KIDS – Keyed Intrusion Detection System

Sasa Mrdovic, Branislava Drazenovic

University of Sarajevo
Faculty of Electrical Engineering
Zmaja od Bosne bb
71000 Sarajevo
Bosnia and Herzegovina
{sasa.mrdovic,branislava.drazenovic}@etf.unsa.ba

Abstract. Since most current network attacks happen at the application layer, analysis of packet payload is necessary for their detection. Unfortunately malicious packets may be crafted to mimic normal payload, and so avoid detection if the anomaly detection method is known. This paper proposes keyed packet payload anomaly detection NIDS. Model of normal payload is key dependent. Key is different for each implementation of the method and is kept secret. Therefore model of normal payload is secret although detection method is public. This prevents mimicry attacks. Payload is partitioned into words. Words are defined by delimiters. Set of delimiters plays a role of a key. Proposed design is implemented and tested. Testing with HTTP traffic confirmed the same detection capabilities for different keys.

Keywords: Network intrusion detection, anomaly detection, word model, Kerckhoffs' principle, keyed IDS

1 Introduction

Methods of intrusion detection made a huge advance from the time of their onset in 1980s. But, during this period attacks evolved from Morris to Conflicker worm. For each method of attack, the method of defense must be changed by modifying the detection algorithm. Once the detection algorithm is known, a new attack is created, and this race may continue forever. This paper is a try to hinder the creation of mimicry attacks based on the information of the detection method.

Network intrusion detection system (NIDS), based on anomaly detection is subject of this paper. Its role in the defense system is to detect network attacks that do not have signature yet. An anomaly based NIDS is based on a model of the normal behavior of the protected network segment. The deviation of the incoming traffic from model of normal behavior is considered as an attack. Since NIDS analyze network packets, the model is built from packet elements. Currently, most frequent attacks are at the application level [1]. To detect application level attack, the network packet payload analysis is obligatory.

Regardless of data used to model normal system behavior, anomaly based IDSs are susceptible to mimicry attacks. Simply said, a model of normal behavior is a set of allowed parameter values spans. An atypical behavior is detected when a value of one or more parameters is outside of its normal span. When parameters and their normal values are known, it is possible to create malicious attack packets having the same model as normal payload. Mimicry attack packets look as normal ones, and that prevents their identifications as unusual.

This paper offers a new approach to detection that could prevent mimicry attacks. It combines and further develops two previous works by the same authors [2,3]. The main idea is to use basic cryptography principle: the system security is not in secrecy of its design but in secrecy of a key. The intrusion detection is based on the analyses of the full packet payload and should be able to detect attacks on application level. Model of normal traffic and detection of deviation from normal is based on division of payload into "words". Words are consecutive payload bytes separated by a set of delimiters, selected byte values. Each implementation of method uses its own set of delimiters, a key, which results in its own model of the normal network traffic. Attackers cannot generate mimicry attacks if they do not know the key. A network intrusion detection method employing this principle is developed and tested.

The paper is organized as follows. Related work is addressed in section 2. Section 3 explains a new implementation of the intrusion detection method and how a key can be used. Results of testing are presented in section 4. Conclusion and discussion as well as directions for future research work are in section 5.

2 Related Work

Papers published in recent years deal with various aspects of payload analysis. Papers [4,5,6] advocate a partial payload analysis. Modern tools for packet manipulation could create attack inserted in payload in diverse forms. The form could be fine-tuned to become unrecognizable as an attack by above detection methods.

The consideration of a single application protocol makes detection easier. Number of papers analyze HTTP requests only [7,8,9,10]. This approach showed good results.

Another research direction seeks exploit code in the packet payload [11,12,13,14]. All these papers make assumptions on how attack code might look and based on those hypotheses analyze packet payload. With current rate of creation of new attacks and mutations of existing ones it is questionable if such approach can keep pace. Paper [15] states that it does not pay to try to model all versions of polymorphic sequences of payload bytes. It suggests that signature based detection is limited and that detection of anomaly from normal is more promising.

The technique proposed in this paper is close to methods that use payload division or byte grouping. PAYL, approach advocated in [16] and [17] uses single byte frequencies to make the model. Reported results are excellent, but since a single byte model is too simple, it is easy to create attack packets that fit the model of normal.

In Anagram detector [18] a model of normal traffic is constructed using fixed length payload byte sequences, named n-grams. It uses simple formula that is fast to

calculate and gives excellent detection results. On the other hand, it is extremely sensitive to attacks in training data. A single attack in these data will make all of its n-grams part of normal model and make that and similar attacks absolutely undetectable.

Division of payload into byte sequences that do not have fixed length is another approach. This approach needs delimiters, byte values that divide payload into parts. Important result of [19] is generation of delimiter sets for a variety of protocols. Proposed delimiters for HTTP are used in [20]. Authors compared results for words, payload byte sequences separated by delimiters, model with n-grams models, and showed that they are comparable in accuracy of detection, but have a much smaller computational load.

First real mimicry attacks were described relatively recently [21,22]. Ideas on how to avoid detection by anomaly based NIDS are even newer. Based on suggestions published in [23], papers [24,25] show how to create attacks that current NIDS cannot detect. Paper on Anagram [18] is the only one that partly addresses mimicry issue.

Detection based on multi-byte payload analysis that prevents mimicry attacks is an approach that will be presented next.

3 Proposed Detection Method

Proposed detection method continues work of other researchers mentioned above. The idea is to combine payload partitioning into words from [20] with an improvement of simple anomaly score calculation in [18]. This should provide a simple, easily storable model, and a fast detection. A new part of the model data on transition from one word to another. The most important extension of the model is use of key.

3.1 Model building principles

The model of the normal packet payload depends on the way of partitioning payload into words. Term “word” has the same meaning as in written humane language: sequence of symbols between two delimiters. In human written language delimiters are spaces and punctuation marks. In the payload symbols that separate words are some predetermined bytes. The selection of delimiters is not unique or obvious as in written human language, and might depend on the application level protocol used.

Since each set of delimiters produces a unique set of normal payload words, and accordingly a unique model it may be used as a key. Each implementation should use an individual set of delimiters. The important question is how a particular set of delimiters affects the detection capability of an IDS. This problem will be addressed in the next section by investigating relationship between keys detection capabilities.

3.1.1 Learning and detection

The model of normal behavior is built during the training phase. Normal, attack free, payloads is partitioned into words. Words in normal traffic with their frequencies constitute the first part of the model. A malicious payload will have word frequency distribution significantly different from the normal payload.

The other part of the model is word transition frequency distribution. Probability of appearance of a word in a language usually depends on the previous word. Assumption is that payload words should behave in the same way. A malicious payload should show a word order different than in a normal payload. During the training phase appearances of any pair of words are counted and stored.

In the detection phase a new packet payload that was not used in learning, is analyzed and assigned two scores, named word score Sw and transition score St.

The word score is calculated using the following formula:

$$S_w = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n(w_i)} \quad (1)$$

In this formula k is the number of words in a payload and $n(w_i)$ is the number of appearances of the word w_i in the learned model. For the words that were not seen in the normal traffic thus having $n(w_i) = 0$, the corresponding term in the sum is set to two instead of infinity. This value is twice the value for the word that was seen only once during training, that has $n(w_i) = 1$. In this way words that did not appear in training payloads will have the highest contribution to anomaly score, but that contribution will not be such to make contribution from rare words completely insignificant.

The formula used to calculate the transition score follows:

$$S_t = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n(t_i)} \quad (2)$$

In this formula m is the number of transitions in a payload and $n(t_i)$ is the number of appearances of the transition t_i in learned model. Value of m is one less than k in (1). Similarly to the formula for the word score, transitions not seen during training will have $n(t_i) = 0$. Again, term $1/n(t_i)$ is set to two instead of infinity. The same reasoning can be applied to formula (2) as for formula (1).

For a faster calculation of scores by avoiding division, of the inverse values of $n(w_i)$ and of $n(t_i)$ are calculated and stored before the detection phase.

Scores based on formulas (1) and (2) are used to obtain total score S. To keep the number of false positive detections low, both word score and transition score must be high to have a high total score. For this reason the total score is found as their product.

$$S = S_w * S_t \quad (3)$$

4 Testing

Although proposed detection method should work with any protocol, HTTP is selected for testing, since it is probably the most widely used application protocol nowadays. Standard TCP HTTP port 80 is generally open on most firewalls. Increasing number of Web applications increases number of vulnerabilities. According to SANS institute statistics Web application vulnerabilities represent almost half of all vulnerabilities discovered in 2007 [26]. The open port and application vulnerabilities attract attackers. Web based attacks make majority of all attacks [27]. HTTP protection seems to be the most needed and any improvements would be welcome.

The major issue in IDS testing is evaluation dataset. A very active recent thread on Security Focus IDS related mailing list showed how a good evaluation dataset is badly needed. The current problems in HTTP testing methods are pointed out in [28]. The best known, and once most widely used data sets for IDS testing, were created by DARPA/MIT Lincoln Laboratories in 1998 [29] and 1999 [30]. There are two main reasons why DARPA data sets are not adequate for current HTTP IDS. Both traffic and attacks in those data sets are obsolete. There are only four web attacks in the data sets. Recent HTTP anomaly detection papers mostly use their own data sets [7,8,9,10,17,18]. Those that do use DARPA data sets, also use their own [16,20].

In this paper EE department of the Sarajevo University traffic was used for testing. Real traffic with outside world was recorded for 12 days in November 2007. Traffic was recorded on the inner side of department external router / firewall. Traffic was first cleaned from attacks using fully tuned Snort, signature based NIDS, with latest signatures combined with manual inspection. The cleaned traffic therefore may have only few attacks. It is a question if it is possible at all to get real traffic that is guaranteed to be attack free [31]. For detection purposes recorded traffic was intentionally combined with attacks generated using Metasploit framework. Attack details are provided later.

4.1 Initial Set of Delimiters

The testing had two parts. The aim of the first part was to assess the ability of proposed score to detect anomalous payload using the delimiter set generating maximal number of meaningful words in HTTP and HTML protocols. The influence of key size and content on the learning process, the normal behavior model and detection was the task in the second part.

Initial key of 20 delimiters taken from [20] were:

CR LF TAB SPACE , . : / \ & ? = ( ) [ ] " ; < >

A set of delimiter symbols may result in any length of a words and any total number of words. In order to keep the number of words down, and thus get a smaller model, allowed word length was limited to the range from 3 to 16. Words shorter than 3 bytes are ignored, since they hardly could be an important part of an attack. Also, a word always ends after 16 consecutive bytes, and a new word begins.

4.1.1 Learning and detection

The recorded traffic, cleaned from attacks, was used to build a model of normal payload. All incoming traffic packet payloads were scanned and partitioned into words, Words, word pairs, and their respective frequencies were stored in a hash table. Number of learned words leveled after the 96 hours of learning. At this point it was concluded that there would be no significant increase in number of words in by continuing the training. Testing should confirm if the conclusion was correct. Total number of learned normal words was about 33 000.

Proposed model consists of normal word frequencies and word transition frequencies. A 33 000 x 33 000 matrix would be needed to store all the word pairs and their frequencies. The distribution of word frequencies showed that a small number of words appeared very frequently, while a huge number appeared only several times. This word distribution was used to create a reduced size transition matrix, instead of full size but sparse matrix. First, the set of words that appear more than ten times in traffic was found. Then, a set of pairs of these words was selected to enter reduced size matrix. The resulting matrix size was 80 times smaller than the full matrix. All the other word pairs were considered as rare, and they were assigned a high partial anomaly score. A question remained would a similar word distribution hold for a different set of delimiters.

To test the ability to detect attacks, packet payload was scored using formula (3). The test checked how detection method scores real attacks. For this purpose 17 HTTP attacks were created using Metasploit. Attacks with related vulnerabilities, CVE references and used attack payloads are given in Table 1. It should be pointed out that the attacks that were used mostly exploit vulnerabilities in Web servers. Attacks that target Web applications, like SQL injection, XSS or similar should be further tested.

Table 1. Attacks with related vulnerabilities and used payloads

No.	Vulnerability	CVE	Payload
1	Apache Chunked-Encoding	2002-0392	adduser
2	Apache Chunked-Encoding	2002-0392	meterpreter-reverse_tcp
3	Apache Chunked-Encoding	2002-0392	shell-reverse_http
4	Apache mod_jk overflow	2007-0774	adduser
5	Apache mod_jk overflow	2007-0774	shell-reverse_tcp
6	Apache mod_rewrite	2006-3747	shell-bind_tcp
7	Apache mod_rewrite	2006-3747	shell-reverse_tcp
8	Apache mod_rewrite	2006-3747	vncinject-reverse_http
9	Apache mod_rewrite	2006-3747	vncinject-reverse_tcp
10	IIS 5.0 IDQ Path Overflow	2001-0500	shell-reverse_http
11	IIS 5.0 IDQ Path Overflow	2001-0500	shell-reverse_tcp
12	IIS ISAPI w3who.dll	2004-1134	exec
13	IIS ISAPI w3who.dll	2004-1134	shell-reverse_tcp
14	Oracle 9i XDB HTTP PASS	2003-0727	shell-reverse_tcp
15	Xitami If_Mod_Since	2007-5067	shell-reverse_tcp
16	HP OpenView Network Node Manager CGI Buffer Overflow	2007-6204	shell-reverse_tcp
17	BSD Mercantec SoftCart CGI Overflow	2004-2221	shell-reverse_tcp

Six days of traffic that were not used during learning were combined with attacks from Table 1. Results of the test are presented on ROC curve in Fig. 1. Threshold for an anomalous score was varied in the range from 0.2 to 2.0. False positive rate scale goes from 0 to 0.005 to provide enough detail in the part of the picture where ROC changes. Results are equal or better then the results reported by other researchers. Real comparison is difficult due to reasons explained in [28].

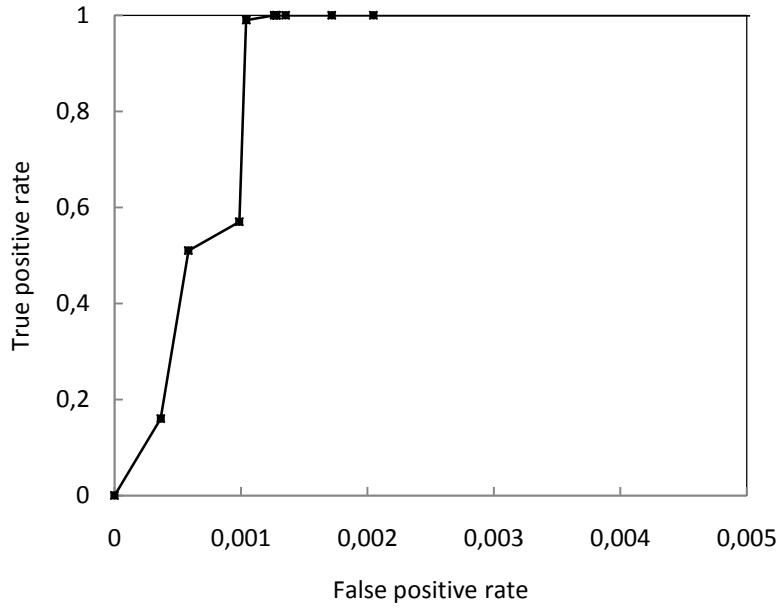

Fig. 1. ROC curve for initial set of delimiters

4.2 Arbitrary Sets of Delimiters

Since a set of delimiters is a key, each implementation should use an individual set of delimiters. A set of delimiters not optimized for the highest percentage of meaningful words, might negatively affect the detection capability. In addition, number of words might increase substantially. Also, word distribution might change, preventing usage of reduced matrix for transition storage. To test above issues various 120 keys were created. Key sizes were 15, 20, 25, 30, and 30 different sets were made for each size. Delimiters were chosen using function “rand” to generate a number between 0 and 255. For each of these keys, the initial set tests were repeated.

4.2.1 Results

For every set of delimiters, number of learned words leveled after 96 hours of learning, as was the case with initial set of delimiters. For those sets, total number of learned words was between 40000 and 50000, for 20% to 50% increase. The increase was expected, but it is not huge. It did not have any important adverse effect on model building or storage. Word distribution has not changed for any of the sets. This is important since the distribution was basis for the reduction of the model size.

Curves in Fig 2 present min, average and max true positive rate for fixed false positive rate points among all 30 delimiter sets for each of four set sizes. Although results vary, there are some keys that are as effective as the initial one. Before deploying a key for practical operation, experiments could be performed to find a good one. Effort to break the method even for small set of good keys is still high enough to prevent key guessing and practical attacks.

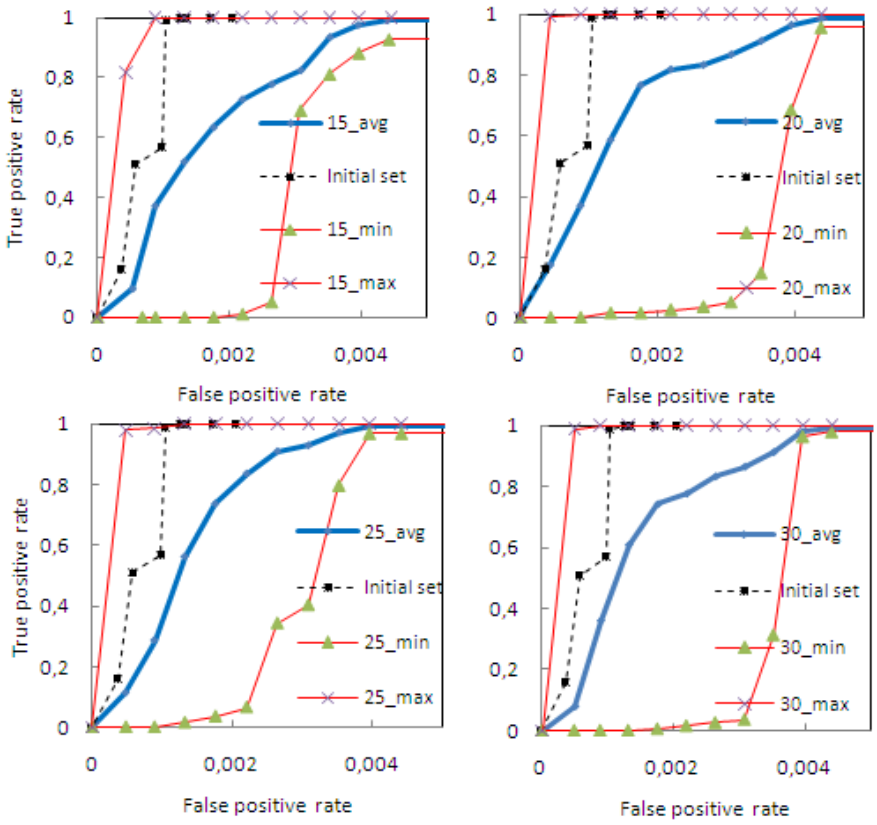

Fig. 2. Average, min and max ROC curves for random sets of 15,20, 25 and 30 delimiters

Since random delimiters can be used, new models of normal payload can be constantly built. While using one model to detect attacks, normal traffic is tokenized

to words with a new set of delimiters. After system has trained enough a new model should be tested. If it is good, it can be used for detection right away or as needed. This enables easy change of keys in regular intervals or on request. Also, the model is now dynamic and can always represent current normal payloads.

5 Conclusion

Keyed intrusion detection system is an implementation of the open design principle. Detection method is public but each implementation uses different secret key. Model of normal behavior is key dependent. Creation of attacks that fit the model is difficult, if not impossible, without the knowledge of the key.

Detection method analyses network packet payloads. Payload analysis enables detection of application level attacks. Payloads are tokenized to words. Model of normal payload is built on frequency distribution of words and transitions between them. Word boundaries are defined by set of delimiters, selected byte values. Set of delimiters plays a role of the key. Different sets of delimiters result in different words and different model.

Method is not protocol dependent. It was tested with HTTP traffic, but should work with all text based application level protocols. This should be tested and it is planned for future work. Future work will further check set of delimiters selection. It has a role of a key and there might be some weak keys.

Keyed IDS might be built with basic detection method different from the one proposed in this paper. It is a novel idea and there might be some better implementations.

References

1. M. Rash, A.D. Orebaugh, G. Clark, B. Pinkard, and J. Babbin, Intrusion Prevention and Active Response: Deploying Network and Host IPS, Syngress, 2005.
2. S. Mrdovic and B. Perunicic, "Kerckhoffs' Principle for Intrusion Detection‖," Telecommunications Network Strategy and Planning Symposium, 2008. Networks 2008. The 13th International, 2008, pp. 1–14.
3. S. Mrdovic and B. Perunicic, "NIDS Based on Payload Word Frequencies and Anomaly of Transitions," Digital Information Management, 2008. ICDIM 2008. Third International Conference on, 2008, pp. 334–339.
4. R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: a new approach for detecting network intrusions," Washington, DC, USA: ACM, 2002, pp. 265-274.
5. M.V. Mahoney, "Network traffic anomaly detection based on packet bytes," Melbourne, Florida: ACM, 2003, pp. 346-350.
6. S. Zanero and S.M. Savaresi, "Unsupervised learning techniques for an intrusion detection system," Nicosia, Cyprus: ACM, 2004, pp. 412-419.
7. C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," Washington D.C., USA: ACM, 2003, pp. 251-261.

8. C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," Computer Networks, vol. 48, Aug. 2005, pp. 717-738.
9. W. Robertson, G. Vigna, C. Kruegel, and R.A. Kemmerer, "Using Generalization and Characterization Techniques in the Anomaly-based Detection of Web Attacks," 2006.
10. K. Ingham, A. Somayaji, J. Burge, and S. Forrest, "Learning DFA representations of HTTP for protecting web applications," Computer Networks, vol. 51, Apr. 2007, pp. 1239-1255.
11. P. Akritidis, E.P. Markatos, M. Polychronakis, and K. Anagnostakis, "Stride: Polymorphic sled detection through instruction sequence analysis," 2005.
12. C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna, "Polymorphic Worm Detection Using Structural Information of Executables," 2005.
13. X. Wang, C. Pan, P. Liu, and S. Zhu, "SigFree: a signature-free buffer overflow attack blocker," Vancouver, B.C., Canada: USENIX Association, 2006, p. 16.
14. M. Polychronakis, K.G. Anagnostakis, and E.P. Markatos, "Emulation-based Detection of Non-self-contained Polymorphic Shellcode."
15. Y. Song, M.E. Locasto, A. Stavrou, A.D. Keromytis, and S.J. Stolfo, "On the infeasibility of modeling polymorphic shellcode," Alexandria, Virginia, USA: ACM, 2007, pp. 541-551.
16. K. Wang and S.J. Stolfo, "Anomalous Payload-Based Network Intrusion Detection," 2004.
17. K. Wang, G. Cretu, and S.J. Stolfo, "Anomalous Payload-Based Worm Detection and Signature Generation," 2005.
18. K. Wang, J. Parekh, and S. Stolfo, "Anagram: A Content Anomaly Detector Resistant to Mimicry Attack," 2006, pp. 226-248.
19. R. Vargiya and P. Chan, "Boundary Detection in Tokenizing Network Application Payload for Anomaly Detection," 2003.
20. K. Rieck and P. Laskov, "Language models for detection of unknown attacks in network traffic," Journal in Computer Virology, vol. 2, 2007, pp. 243-256.
21. D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," Washington, DC, USA: ACM, 2002, pp. 255-264.
22. K. Tan, K. Killourhy, and R. Maxion, "Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits," Recent Advances in Intrusion Detection, 2002, pp. 54-73.
23. O. Kolesnikov and W. Lee, Advanced Polymorphic Worms: Evading IDS by Blending in with Normal Traffic, College of Computing, Georgia Tech, 2005.
24. P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee, "Polymorphic blending attacks," Vancouver, B.C., Canada: USENIX Association, 2006, p. 17.
25. P. Fogla and W. Lee, "Evading network anomaly detection systems: formal reasoning and practical techniques," Alexandria, Virginia, USA: ACM, 2006, pp. 59-68.
26. SANS Institute, "SANS Top-20 2007 Security Risks (2007 Annual Update)."
27. Internet Security Threat Report, Symantec Corporation, 2008.
28. K. Ingham and H. Inoue, "Comparing Anomaly Detection Techniques for HTTP," Recent Advances in Intrusion Detection, 2007, pp. 42-62.
29. R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," 2000, pp. 12-26 vol.2.
30. L. Richard, J.W. Haines, D.J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," Computer Networks, vol. 34, Oct. 2000, pp. 579-595.
31. C. Gates and C. Taylor, "Challenging the anomaly detection paradigm: a provocative discussion," Germany: ACM, 2006, pp. 21-29.