Detection of Insider Attacks to the Web Server[*]

Byungha Choi and Kyungsan Cho[†]
Dankook University
Yongin-si, Gyeonggi-do, Korea
notanything@hanmail.net, kscho@dankook.ac.kr

Abstract

In this paper, we propose a detection scheme to protect the Web server by inspecting HTTP outbound traffic from insider attacks which reveal confidential/private information or spread malware codes through Web. Our proposed scheme has a two-step hierarchy with a signature-based detector using Snort, and an anomaly-based detector using HMM. Through the verification analysis under the attacked Web server environment, it has been shown that our proposed scheme improves the detection rate.

Keywords: intrusion detection system, insider attack, outbound traffic, Web server

1 Introduction

Attacks with criminal motives of intentional harm to the victim system evolved from simple spoofing other's password to the complicated Web-based attacks. Because more and more systems are reliant upon the Web server to get and exchange information through the Internet, Web-based attacks have become an important subject in the security field. In addition, defending against Web-based attacks has become increasingly complex and hard, and intruders try to bypass the traditional attack path. Web-based attacks expose the vulnerability of the victim system and spread malwares to other hosts communicating with the victim system[1]. A hierarchical layered schemes with firewall, IDS(Intrusion Detection system), and WAF(Web Application Firewall) are provided to cope with above attacks and they protect the victim system very well. Traditionally they detect external outsider threats by inspecting inward traffic towards the system.

Even though outsider attacks are constantly evolving and increasing, they are well detected and protected with the corresponding technical improvement. However, because an insider directly accesses to the Web server and insider attacks bypass the traditional Web-based intrusion path, insider attacks should be coped with in other ways. In addition, there are detoured attacks aided by the insider. For example, they are infected e-mails sent to insiders and using malicious USB memory or PDA with the aid of insiders. This unusual type of attacks becomes more serious and common threat which uses the legal privileges for malicious purposes. Insider attacks (including detoured attacks aided by the insider) already have overtaken Web-based viruses and worm attacks as the most reported security incident[2]. Up to now, several detection technologies for insider attacks are proposed. It is found that insiders show unusual activities or abnormal behaviors when they access system resources for attacking purpose. Thus, most works are based on identifying abnormal insider's behaviors and finding any significant change in insider's activities. In addition, many insider attacks and detoured attacks to Web servers reveal

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 3, number: 4, pp. 35-45

[*] This research was supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development.

[†] Corresponding author: Dept. of Software Science, Dankook University, Tel: +82-31-8005-3238

confidential/private information or spread malware codes to outside of the victim system, and these harm could be protected by inspecting the outbound traffic.

In this paper, we propose a scheme to detect insider attacks to the Web server by inspecting outbound traffic rather than inbound traffic. We combine two approaches of detection systems to inspect outbound HTTP packets. That is, our proposal is a two-step detection scheme with two popular approaches; signature-based and anomaly-based. Our signature-based detector is implemented with snort and the anomaly-based detector with HMM(Hidden Markov Model). We cannot find any other hybrid system combining two approaches to detect insider attacks. Through the verification, we show that inside attacks are detected by inspecting HTTP outbound traffic from the Web server and our two-step scheme improves detection efficiency compared to each detection approach.

The rest of the paper is organized as follows. In Section 2, we review related works on security vulnerabilities on the Web and solutions to them. In section 3, we propose our two-step detection scheme. In Section 4, we verify our scheme with real datasets collected under the attacked environment. In Section5, a summary is provided.

2 Related Works

In this section, we address related woks on Web-based outsider attack, insider attack, and their countermeasures. In 2010, OWASP announced the updated top 10 most critical Web application security risks to educate about the consequences of the most important Web application security weaknesses and provide basic techniques to protect against these high risk problem areas[3]. The WASC Threat Classification v2.0 shows proper classification of threats into attacks and weakness for static/core view[4]. Both show seriousness of Web-based attacks with focus on application layer of the protocol suite. Application vulnerabilities could provide the means for malicious end users to breach a system's protection mechanisms in order to gain access to confidential and private information or system resources[4].

There are a lot of attack classifications. For example, insider attack and outsider attack are defined by the attack location. To detect and protect from Web-based outsider attacks, a hierarchical Web security system is commonly used. The traditional security system could protect the Web server from external threats by inspecting the inbound traffic through layers of firewall, IDS, and Web application firewall as shown in Figure 1.

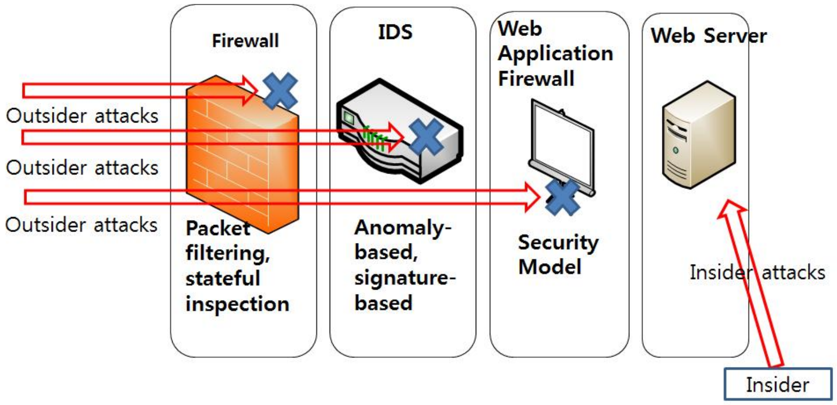


Figure 1: Traditional Security System for a Web server

This approach to defend the Web server allows filtering different attacks in each layer by inspecting inbound traffic. Firewall, which filters the specific network traffic between the network and the Web server, uses packet filtering and stateful inspection to detect the simple intrusions in the first stage of the hierarchical security system. However, the firewall cannot prevent against previously unknown attack types and protect against insider attacks[5]. The second stage, IDS uses signature-based or anomaly-based approaches to protect against attacks that pass the firewall in the first stage. Signature-based IDS finds known pattern of misbehaviors in the message context, and anomaly-based IDS finds any deviation from the normal context pattern. IDS also can be classified according to the location and the purpose as NDIS(Network-based DIS) and HDIS(Host-based DIS). Mostly, NDIS is used in the second stage of the hierarchical security system. Web application firewall filters packets by applying a set of rules to inbound traffic. It uses Positive Security Model and Negative Security Model or both. It filters packets which already pass both firewalls and IDS[6].

Even though most of current intrusion detection systems only use one of the two detection approaches, signature-based detection or anomaly-based detection, both of them have their own limitations. Signature-based IDSs cannot detect any unknown attacks whereas anomaly-based IDSs cannot detect any attacks of untrained type. Thus, the detection rate of the signature-based detection is relative low, but FP rate of anomaly-based IDSs is not negligible. Snort is widely used IDS which allows pattern search for signature-based detection, and new works on using snort rules in IDS have been proposed[7, 8]. Security tools incorporating anomaly-based detection are proposed and HMM, a statistical model of a system as a Markovian process with hidden states, has been shown to provide a high level performance for detecting intrusions[9, 10]. Lately, hybrid IDSs have been proposed by combining the two approaches in one system. For example, a hybrid IDS is obtained by combining packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) or a hybrid intrusion detection system (HIDS) is configured with three sub-modules of misused detection module, anomaly detection module and signature generation module[11, 12]. However, they detect only external outsider attacks. Even though external outsider attacks are constantly evolving and increasing, they are well detected and protected with the corresponding technical improvement.

Insider attacks, unusual type of attacks, become more serious and common threat which uses the insider's legal privileges for malicious purpose. In fact, the traditional security system pays little attention to what is happening inside the system even though they could inspect it. Insider who is defined as an individual with privileged access to an IT system[13], shows unusual activities or behaviors when he accesses system resources for attack purpose. Thus, Information about the user's pattern of behaviors and activities could be inspected for detection purpose. Up to now, there are some verified prevention and detection technology and security program for insider attacks. Most works are based on identifying abnormal insider's behavior and finding any significant change in an insider's activity. For example, Shavlik addressed a detection system through user profiling[14]. However, it may not enough to make a conclusion of a malicious act merely from knowing only user's activity and need further verification[2]. In addition to the insider attack, detoured attacks aided by the insider to the Web server are possible. For example, e-mail attacks with attached malwares and attacks using USB memory/PDA attack with the aid of the insider are also considered as a sort of insider attack. If an e-mail containing a malware is sent to the insider and the insider accepts it, it causes the victim system to be remotely controlled by the outsider. If an outsider connects USB memory or PDA infected malwares to the Web server with the aid of any insider, it causes detoured attack. Insider attacks to the Web server cause information leakage and malware distribution through the outbound traffic and we need other approach to detect them. Insider attack and detoured attack with the aid of insider to Web servers use weakness of OWASP Top 10 or malware codes, thus cause altering HTML documents with tags and JavaScript codes as well as SQL injection in DB. When this falsified Web page is activated by the malware code, it could reveal confidential/ private information and distribute malware codes.

If HTML tags disseminating malware are divided into several parts and stored partially into DBMS and any symptom of the malware appears in the traffic only when activated, it is very difficult for security programs to detect them. However, this kind of attack could be detected by inspecting the outbound traffic.

Traditionally, conventional security solutions monitor network communication without paying much attention to outgoing traffic, due to high processing cost of packet level network traffic analysis[15]. Lately, several works on inspecting outbound traffic for security reasons have been proposed. It is shown that many insider attacks to the Web server show similar patterns in the HTTP outbound traffic[16], and the potential HTTP-based application-level attack exploits the features of Hypertext Markup Language (HTML)[17]. Thus, currently unknown insider attacks also could be detected by inspecting outbound traffic. If any deviation from normal context pattern of tags and JavaScript codes for the specific traffic is found in the outbound HTTP packets, it could be detected as an attack.

Based on above analysis, we propose a two-step detection scheme for insider and detoured attacks by monitoring HTTP outbound traffic. Our proposal has the following properties different from other related works.

1) Operation: Our detection scheme inspects HTTP outbound packets from a Web server rather than inbound packets, and detects information leakage or malware codes generated by insider and detoured attacks.

2) Implementation: We create new 107 detection rules from the analyzed signatures into Snort and configure new HMM models for probability evaluation. Signature-based detection and anomaly-based detection operate in two sequential steps.

3 Proposed Detection Scheme with Two Step Hierarchy

3.1 Overview of the proposal

What our proposed scheme could detect includes insider attacks, detoured attacks and unknown attacks which all show abnormal symptoms when they send HTTP packets through the network as discussed in section 2.

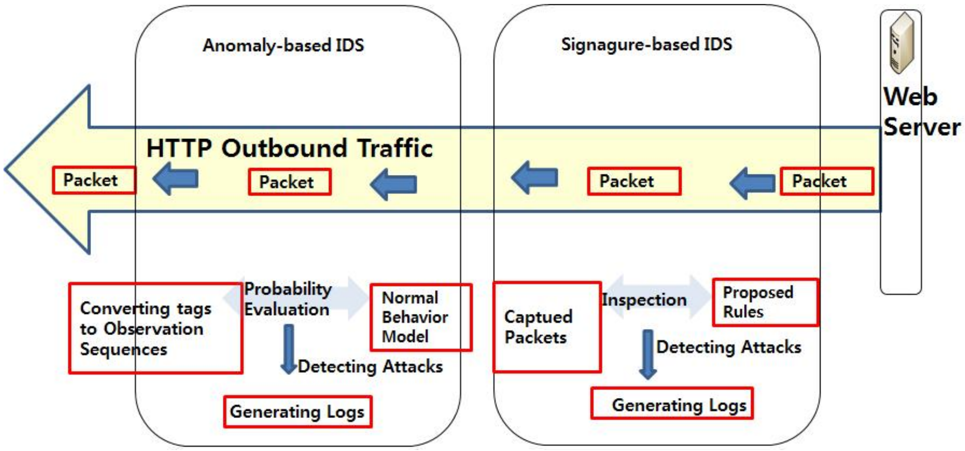


Figure 2: Proposed Two-Step Detection Scheme

Therefore, instead of analyzing user's profiling or user's activities and instead of inspecting inbound traffic towards the system, we propose to inspect and detect abnormal HTTP outbound traffic caused by the insider attack. Our proposed scheme shown in Figure 2 is a two-step detection system composed of a signature–based IDS using Snort and an anomaly-based IDS using HMM.

3.2 Signature-based Detection

The first step of our detection scheme is a signature–based detection implemented using Snort which is an open source network intrusion prevention and detection system combining the benefit of signature, protocol and anomaly-based inspection. A signature is a distinctive mark or characteristics contained in the context of a packet. Our signature-based detector inspects the symptoms of disseminating malware, XSS, URL Spoofing and information leakage from the Web server.

All these symptoms are represented as special forms of tags and JavaScript codes as well as particular context in HTML documents. Thus, above attacks could be detected by finding predefined signatures in the HTML documents transferred from the Web server. Table 1 and Table 2 show typical signatures that appear in HTML tags and JavaScript code spreading malwares. Figure 3 shows the detailed process of creating a rule from a signature in Snort[16]. Snort has a function to detect abnormal context in the outbound packets if proper signatures and rules are provided. We created 107 rules into Snort to detect predefined signatures. Our rules detect the actual vulnerability with signatures extracted in the abnormal HTML documents.

Table 1: Some Signatures that appear in HTML tags

Tag name	Attribute name	Attribute value
	HEIGHT	0
IFRAME	FRAMEBORDER	0
	SRC	(1)
IFRAME	WIDTH	0
	SRC	(1)
LINK	HREF	(1)
SCRIPT	SRC	(1)
(1) It doesn’t start with	”http://Web server IP address	or domain name” or ”./”, ”/”

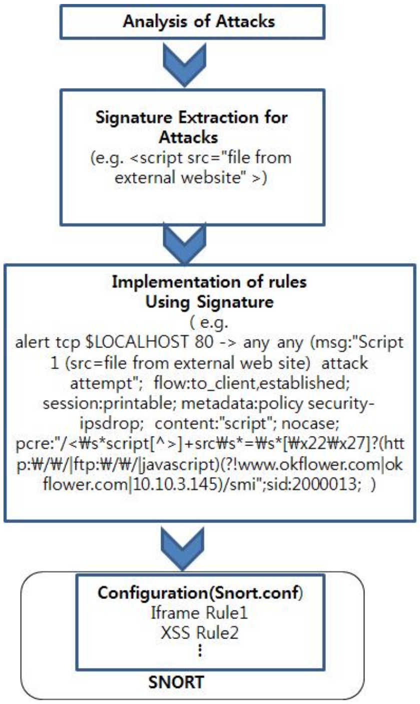
Table 2: Some Signatures that appear in JavaScript

JavaScrpt Code
String.fromCharCode(
decodeURIComponent(
charCodeAt(
document.write(”<OBJ....width=0

3.3 Anomaly-based Detection

The second step of our detection scheme is an anomaly-based detection which detects attacks by using HMM and finding the probability of an observed sequence.

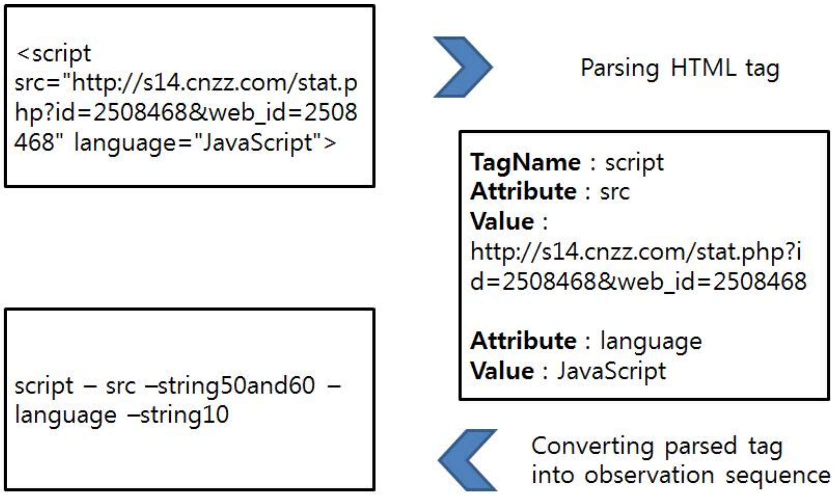
Normal models are created by training with datasets of tags and JavaScript codes found in the normal HTML documents where observation sequences are HTML documents assembled by the packets in the outbound traffic. HMM is a statistical model of a system as a Markovian process with hidden states. An HMM is characterized by the number of states N, the number of distinct observation symbols per state M, the state transition probability distribution A, the observation symbol probability distribution in a state B, and the initial state distribution π. Given appropriate values of N, M, A, B, and π, HMM can generate an observation sequence O. Thus, HMM requires specifications of two model parameters(N, M), observation symbols, and three probability measurements (A, B, π) and the compact notation λ=(π,


Analysis of Attacks
Signature Extraction for Attacks
(e.g. <script src="file from external website" >)
Implementation of rules Using Signature
(e.g.
Configuration(Snort.conf)
Iframe Rule1
XSS Rule2
⋮
SNORT

```
alert tcp $LOCALHOST 80 -> any any (msg:"Script
1 (src=file from external web site)  attack
attempt";  flow:to_client,established;
session:printable; metadata:policy security-
ipsdrop;  content:"script"; nocase;
pcre:"/<₩s*script[^>]+src₩s*=₩s*[₩x22₩x27]?(htt
p:₩/₩/|ftp:₩/₩/|javascript)(?!www.okflower.com|ok
flower.com|10.10.3.145)/smi";sid:2000013;  )
```

Figure 3: Detailed Process of Creating Rules

A, B) is used to indicate HMM model[18]. As an application of HMM to detecting attack, we can find how to compute P(O| λ)under the given model λ=(π, A, B) with observation sequence O and how to adjust the model parameters λ=(π, A, B) to maximize P(O| λ).



```
<script
src="http://s14.cnzz.com/stat.p
hp?id=2508468&web_id=2508
468" language="JavaScript">
```

TagName : script
Attribute : src
Value :
http://s14.cnzz.com/stat.php?i
d=2508468&web_id=2508468

Attribute : language
Value : JavaScript

```
script – src –string50and60 –
language –string10
```

Figure 4: Process of converting into observation sequence

With Baum-Welch algorithm for this problem, we can train with the normal dataset in the same way of finding optimal values for π, A and B to maximize the probability of observation sequence O given λ. Then, use the probability evaluation which finds the probability of the observation sequences(tags or JavaScript codes in outbound traffic) in the normal model to detect attacks. We already addressed an IDS with HMM[19]. Our proposed method using HMM checks whether the tags or JavaScript codes in the

HTTP outbound packet are normal or not. After parsing HTML tags, the parsed tags are converted into observation sequence as shown in Figure 4. Figure 5 shows how to apply HTML tags to HMM model where three hidden states are defined and the probability of observation sequence are initiated.

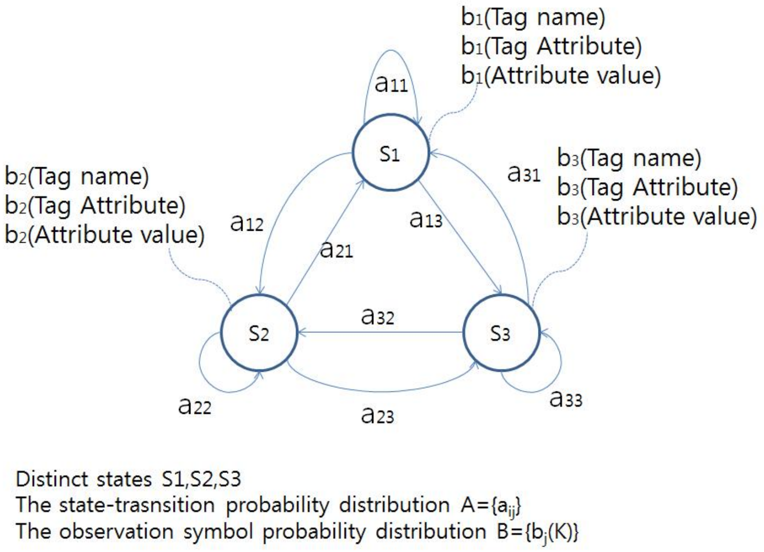


Figure 5: HMM Model for HTML Tags

4 Verification Analysis

4.1 Test Environment

Table 3 shows detailed description of our test environment.

Table 3: Test Environment

Web server	DBMS	MS - SQL 2000
	Server	IIS 5.0
	Web Programming Language	ASP
	Virtual machine	MS Virtual PC
Signature-based IDS	IDS	Snort 2.8.6.1
	Packet capture Library	winpcap 4.0
Anomaly-based IDS	Packet capture Library	Jpcap 0.7
	HTML Parser	Jericho HTML Parser
	JavaScript Parser	Rhino 1.7 R3
	HMM Library	JaHMM
	JDK(language)	Oracle JDK 1.6(Java)

For the verification of our proposed scheme, we configure a Web server environment with the altered HTML documents created under the real attack and by the hacking tool. We configure a virtual network where a Web server is operating under the vulnerable OS-Window 2000.

4.2 Datasets

We use the following datasets for the verification.

- 31 altered HTML documents provided by one of Korean Security Agencies. They are generated by the real inside attacks and detoured attacks. This dataset is used to evaluate detection rate.
- Dataset generated by HDSI which is a SQL injection hacking tool and Dataset generated by Havij which is an automated SQL injection tool. These two data sets are used to show detection efficiency of each detection scheme.

4.3 Verification

To verify the detection rate, we use the first dataset of 31 altered HTML documents.
At first, two detection schemes are tested individually. Evaluated detection rate for each detection scheme is shown in Table 4.

Table 4: Detection results from each detection scheme

	Number of normal documents	Number of detections	Number of misses
signature-based IDS	31 HTML documents	20	11
anomaly-based IDS		29	2

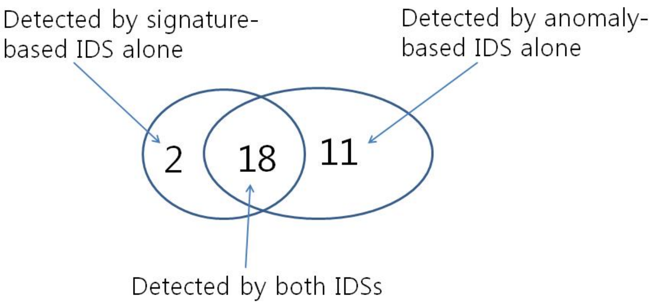

Detected by signature-based IDS alone
Detected by anomaly-based IDS alone
2
18
11
Detected by both IDSs

Figure 6: Venn Diagram of detected attacks

The missed detections in each IDS are different each other, and it causes two IDSs to complement each other as shown in Figure 6. Then, we tested the same dataset in two steps; at first signature-based detection and then anomaly-based detection. Evaluated result shows that all altered documents caused by attacks are detected through two steps as shown in Table 5. Even though each detection method cannot detect all attacks, the combined two-step detection could detect all altered documents.

Table 5: Detection results from two-step detection scheme

Number of altered documents	Number of passed documents after signature-based scheme	Number of passed documents after anomaly-based scheme
31	11(20 are detected)	0 (all are detected)

To verify the detection efficiency of each detection approach, we test two datasets. At first, we detect attacks generated by inputting various SQL queries as Web parameters in HDSI. HDSI tries to attack a DB through Web servers in 5 stages. Each stage generates attacks from Web server to DBMS in order to get detailed data.

Table 6: Detection of attacks generated by HDSI

Stage	Signature-based IDS	Anomaly-based IDS
1st stage	none	19
2nd stage	none	14
3rd stage	none	86
4th stage	none	311
5th stage	30	90

An anomaly-based IDS detects abnormal documents in each stages: 19 anomalies in the 1st stage, 14 anomalies in the 2nd stage, 86 anomalies in the 3rd stage, 311 anomalies in the 4th stage and finally 90 anomalies(3 anomalies per each e-mail, 30 e-mails) in the 5th stage as shown in Table 6. From the analysis result, we can find the efficiency of the anomaly-based IDS in detecting different trained anomalies. On the other hand, the signature–based IDS detects only 30 e-mail attacks in the 5th stage. It is because only error documents are generated in the 1st - 4th stages, thus no signatures are found in them.

Table 7: Detection of attacks generated by Havij

Stage	Signature-based IDS	Anomaly-based IDS
1st stage	9	none
2nd stage	none	none
3rd stage	1	none
4th stage	139	none

Another test is performed using dataset generated by Havij, which is an automated SQL injection tool that helps penetration testers to find and exploit SQL injection vulnerabilities on a Web page through 4 stages. As shown in Table 7, signature-based IDS detects attacks in the 1st, 2nd and 4th stages because they have known distinctive marks in the context. However, anomaly-based IDS cannot detect any attacks because attacks change tag's content strings in the context rather than tag's attributes or JavaScript code. They are not trained in our normal models as an observation symbol of HMM. From table 6 and 7, we find that each approach alone is not sufficient for detecting insider attacks. Analysis of the results in Table 4, 6 and 7 justifies our two-step scheme.

5 Summary

Even though external outsider attacks are constantly evolving and increasing, they are well detected and protected with the corresponding technical improvement. Insider attacks, unusual type of attacks, become more serious and common threat which uses the insider's legal privileges for malicious purpose. Most works on detecting insider attacks are based on identifying abnormal insider's activity and finding any significant change in insider's activities. It is found that many insider attacks and detoured attacks to the Web server disclose confidential/private information or spread malware codes to outside, and these harm could be protected by inspecting HTTP outbound traffic.

In this paper, we propose an improved detection scheme for insider and detoured attacks based on the analysis addressed in section 2. Our proposed scheme has two-step hierarchy to detect abnormal tags and JavaScript codes in HTML documents. The first step is signature-based detection using Snort and the second step is anomaly-based detection using HMM. Through the verification analysis with HTML documents created under the attacked Web server environments, we show that insider attacks can be detected by inspecting HTTP outbound traffic from the Web server, and our proposed two-step scheme improves the detection rate compared to each detection scheme.

As an extension, our proposed scheme could be applicable to the backdoor attack which is a method of bypassing authentication and securing remote access to a Web server. It has been a serious threat over past few years, because outbound traffic generated by backdoor attacks have various types of packets and becomes more dangerous due to the unusual control of a Web server.

References

[1] J. Crist, "Web based attacks," SANS Institute, Tech. Rep. 1, 2010.

[2] M. Salem, S. Hershkop, and S. Stolfo, *A Survey of Insider Attack Detection Research*. Springer US, 2008.

[3] M. B. et al., "OWASP top 10 - the ten most critical web application security risks," OWASP, Tech. Rep. 1, 2010.

[4] The Web Application Security Consortium, "The WASC threat classification v2.0," The Web Application Security Consortium, Tech. Rep. 1, 2010.

[5] S. K. and H. P., "Guidelines on firewalls and firewall policy," Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, Tech. Rep. 1, 2009.

[6] L. D. et al., "Bridging the gap between web application firewalls and web applications," in *Proc. of the 4th Workshop on Formal Methods in Security Engineering (FMSE'06), Virginia, USA*. ACM, November 2006, pp. 67–77.

[7] J. Gómez, C. Gil, N. Padilla, R. Baños, and C. Jiménez, "Design of a snort-based hybrid intrusion detection system," in *Proc. of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living (IWANN'09), Salamanca, Spain, LNCS*, vol. 5518. Springer-Verlag, June 2009, pp. 515–522.

[8] A. Mitra, W. Najjar, and L. Bhuyan, "Compiling pcre to fpga for accelerating snort ids," in *Proc. of the 3rd ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS'07), Florida, USA*. ACM/IEEE, December 2007, pp. 127–136.

[9] S.-B. Cho and S.-J. Han, "Two sophisticated techniques to improve HMM-based intrusion detection systems," in *Proc. of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID'03), Pittsburgh, USA, LNCS*, vol. 2820. Springer-Verlag, September 2003, pp. 207–219.

[10] W. Khreich, E. Granger, R. Sabourin, and A. Miri, "Combining hidden markov models for improved anomaly detection," in *Proc. of the IEEE International Conference on Communications (ICC'09), Dresden, Germany*. IEEE, June 2009, pp. 965–970.

[11] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[12] Y.-X. Ding, M. Xiao, and A.-W. Liu, "Research and implementation on snort-based hybrid intrusion detection system," in *Proc. of the 8th International Conference on Machine Learning and Cybernetics (ICMLC'09), Darwin, Australia*, vol. 3. IEEE, July 2009, pp. 1414–1418.

[13] J. Hunker and C. W. Probst, "Insiders and insider threats - an overview of definitions and mitigation techniques," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 2, no. 1, pp. 4–27, March 2011.

[14] J. Shavlik and M. Shavlik, "Selection, combination and evaluation of effective software sensors for detecting abnormal computer usage," in *Proc. of the premier International Conference on Knowledge Discovery and Data mining (KDD'04), Washington, USA*. ACM, August 2004, pp. 276–285.

[15] M. Skrzewski, *Analyzing Outbound Network Traffic*. Springer Berlin Heidelberg, 2011.

[16] B. Choi, S. kyo Choi, and K. Cho, “An efficient detection scheme of web-based attacks through monitoring http outbound traffics,” *Journal of The Korea Society of Computer and Information*, vol. 16, no. 1, pp. 125–132, 2011.

[17] X. Wang, J. Luo, M. Yang, and Z. Ling, “A potential HTTP-based application-level attack against tor,” *Future Generation Computer Systems*, vol. 27, no. 1, pp. 67–77, 2011.

[18] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” in *Proc. of the IEEE*. IEEE, February 1989, pp. 257–286.

[19] B. Choi, S. kyo Choi, and K. Cho, “Anomaly detection scheme of web-based attacks by applying HMM to HTTP outbound traffic,” *Journal of The Korea Society of Computer and Information*, vol. 17, no. 5, pp. 33–40, 2012.

Byungha Choi received the MS degree from the Dept. of Information and Communication Technology, Dankook University. He is currently a Ph.D. student of Dept. of Computer Science and Engineering at Dankook University. His research interest is Network Security.

Kyungsan Cho received his B.Sc. in Electronics Engineering(Seoul National University, 1979), master degree in Electrical and Electronic Engineering(KAIST, 1981), and his Ph.D. degree in Electrical and Computer Engineering(the University of Texas at Austin, 1988). During 1988-1990, he served as a senior R&D engineer at Samsung Electronics Company . He joined Dankook University in March 1990, where he is currently a professor in the department of software science. He authored several books in Computer Architecture and Computer Networks and published over 40 academic papers. His research interests include mobile networks, network security and traffic analysis.