Behavior Rule based Intrusion Detection

Masayoshi Mizutani
Keio University, Graduate School of Media and Governance
5322 Endo Fujisawa-shi
Kanagawa, Japan
mizutani@sfc.wide.ad.jp

Keiji Takeda
Keio University, Faculty of Environment and Information Studies
5322 Endo Fujisawa-shi
Kanagawa, Japan
keiji@sfc.keio.ac.jp

Jun Murai
Keio University, Faculty of Environment and Information Studies
5322 Endo Fujisawa-shi
Kanagawa, Japan
jun@wide.ad.jp

Categories and Subject Descriptors

H.4.0 [**Information Systems Applications**]: General

General Terms

Security

Keywords

Internet, Information Security, Intrusion Detection, Malware, Botnet

1. INTRODUCTION

Network-based intrusion detection system (IDS) has been changing in accordance with a development of network and appearances of new threats. Traditional IDSes (e.g. Snort[1]) have historically focused on finding patterns appearing in network traffic of attack attempt. However, today huge attempt traffic are observed in each network because of attack automation and rampant malicious software (malware) including computer viruses, worms and bot. Since attack attempts are not always successful, security officers can't know results of attack attempts by traditional IDSes. Security officers have to investigate damages by attack attempts with many resources and it has became difficult to select a high risk attack attempt from huge observed attempts. Therefore, the approach finding traffic patterns of attack attempts had become ineffective.

We propose *behavior rule based intrusion detection* method to analyze correlation of communication behaviors by rules which can be described. An idea of the method is similar to scenario-based intrusion detection (e.g. NetSTAT[2]) and outbound intrusion detection[5]. The method considers an overall picture of communications and can detect incidents with high accuracy. In our approach, we use *auxiliary variables* in each behavior rules to figure out various correlations between events and can describe communication behaviors of various software and attack scenarios. Applications of the method achieve detection of encrypted network traffic of P2P software, unknown malware activities and malware infection over web browsers. Our method is expected to help security officers to know status and incidents in their network.

CoNEXT Student Workshop'09, December 1, 2009, Rome, Italy.
ACM 978-1-60558-751-6/09/12.

2. BEHAVIOR RULE BASED INTRUSION DETECTION

2.1 Concept

In this paper, we defined *behavior* is communication procedures of software. Since all software runs and communicates in accordance with predefined programs, there is a communication pattern which consists of sent and received character strings, destinations, communication protocols, sending and receiving intervals and so on. Although traditional IDSes have mainly focused on packet data including packet headers/payload data pattern, more flexible and high accurate detection can be achieved by the use of other information. For example, BLINC[4] can identify application types by only transport layer communication patterns, excluding payload data pattern. On the other hands, a bot detection method proposed by Binkley and Singh[3] focused on initiation patterns of multiple TCP sessions. These researches indicate that a powerful intrusion detection method that has more flexibility and accuracy can be achieved by a detection method of combination with communication patterns and payload data pattern.

The behavior rule based intrusion detection which uses correlations of packet/payload data patterns and communication patterns. Scenario-based intrusion detection method has similar features based on state transition machine, However scenarios of compromise consist of not only sequential events but also random order events and certain scenarios have to be described complicated correlations between communications. Therefore we focused on 2 types of correlations:

- **asynchronous event sequence**: Handling various event appearance patterns including random order event sequence and repeating of event. When multiple processes or multiple threads are running in most operating systems, a sequence of communications are asynchronous and unsteady. Thus an order of certain events can't be estimated and robustness and flexibility to handle event sequences is necessary to detect a communication behavior of software. For example, asynchronous event sequence can deal a scenario that includes events ($E_{1,2,3}$): $(E_1 \wedge E_2) \rightarrow E_3$ that means observing E_3 after both of E_1 and E_2. We have to prepare 2 event sequence: $E_1 \rightarrow E_2 \rightarrow E_3$ and $E_2 \rightarrow E_1 \rightarrow E_3$ to describe by generic state transition machines. The combination of event sequence is increased along complicacy of correlations.

- **data pattern comparison**: Handling comparison between packet headers/payload data on multiple communications. When 2 or more communications have correlation, same identities (IP addresses, port numbers, domain names, URLs and so on) or string data (sending/receiving commands, queries and so on) are appeared on all communications. Thus correlations can be verified by comparing some packet headers/payload data.

2.2 Architecture

The behavior rule based intrusion detection use auxiliary variables for describing correlations between events in each communication. R is a behavior rule and has N_s *session rules* (s_n) and N_v *variables* (v_n). Sessions mean TCP sessions, a pair of UDP source and destination port number and ICMP request and response and session rules contain events (E_x).

$$R = \{\forall s_{n_s}, \forall v_{n_v} | 0 \leq n_v < N_v, 0 \leq n_s < N_s\} \quad (1)$$

$$s_n = \{\exists E_x | 0 \leq x < N_e\} \quad (2)$$

Session rules are similar to signatures of traditional IDSes in pattern matching of packet headers and payload data and variables work as semaphores in the architecture. When detecting traffic corresponded with some signature (occurring a event), variables are updated to predefined values according to the rules and other session rules can refer the variables. Variables can be substituted integer values, string data and binary data as the situation demands and when they are substituted integer value they can also be incremented and decremented. In order to check asynchronous event sequence, when detecting a event a variable is substituted a certain value and the variable works as a flag. On the other hand, a variable works as a counter by incrementing values and works as a storage by substituting string data and binary data (e.g. IP addresses). Figure 1: The figure shows a example of handling events of random order.

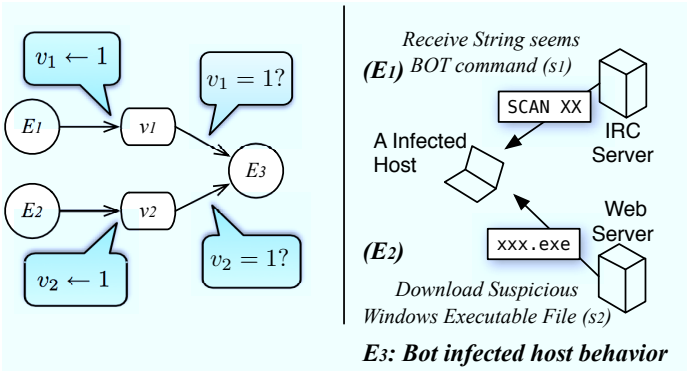

Figure 1: The figure shows a example of correlation among random order communications. The left of figure shows relations of events and variables and the right of figure shows actual security events. In this case, event 1 ($E_1 \in s_1$) is defined receiving bot command and event 2 ($E_2 \in s_2$) is defined downloading a suspicious windows executable file. When both of E_1 and E_2 are observed on the same host, the behavior is highly likely that of bot. If $E_{1,2}$ are observed, each variables value $v_{1,2}$ are updated to 1. (Initial values of them are 0) Since an occurrence condition of E_3 is that both of values $v_{1,2}$ are 1, whichever event is observed on ahead, bot behavior can be detected.

3. IMPLEMENTATION AND EVALUATION

We designed and implemented an intrusion detection system, ROOK[7], which is based on the behavior rule based intrusion detection method. ROOK decodes packets from live network traffic on the fly and analyze correlations of communications based on rule files written in XML. ROOK is implemented in C with libpcap (ver. 0.9.5), libxml2(ver. 2.6.30) and libpcre (ver. 7.4) on Linux 2.6.18 and MacOSX 10.5. The ROOK implementation philosophy is high expand-ability and flexibility. Therefore, components and modules of ROOK have high independency.

We evaluate the accuracy of ROOK using a traffic data set of real malware activities that we prepared in a test environment[6]. We collected 299 traffic data of malware activities and study whether ROOK can detect the data set as malware activities. The one traffic data was generated from one malware on one virtual machine. ROOK applied 3 rules that are described common malware activities: *spread scan with a command message*, *download new malware*, *sending malware to other hosts*. In the evaluation, ROOK detected 98.33% malware activities and false positive ratio of ROOK is 1.67%. The rules for the evaluation are not tuned for these malware traffic. On the other hand, although we also check these data set with snort installed 3,387 rules for malware, it is able to detect 73.91% traffic data and fail to detect 26.09% malware traffic data. The result indicate that our approach is efficient to detect unknown malware activities.

4. REFERENCES

[1] *SNORT-LIGHTWEIGHT INTRUSION DETECTION FOR NETWORKS*, 1999.

[2] Giovanni Vigna,Richard A. Kemmerer. NetSTAT: A Network-based Intrusion Detection Approach. *ACSAC*, 1998.

[3] S. S. James R. Binkley. An Algorithm for Anomaly-based Botnet Detection. *SRUTI '06*, pages 43–48.

[4] Karagiannis, Thomas and Papagiannaki, Konstantina and Faloutsos, Michalis. Blinc: multilevel traffic classification in the dark. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 229–240, New York, NY, USA, 2005. ACM Press.

[5] S. Mandujano. Outbound intrusion detection.

[6] Masayoshi Mizutani, Akira Kanai, Keiji Takeda, Jun Murai. A Malware Detection Method based on Communication Commonality - Implementation and Evaluation. *IPSJ Journal*, 50(9):1234–1243, Sep 2009.

[7] M. Mizutani, S. Shirahata, M. Minami, and J. Murai. Rook: Multi-session based network security event detector. In *SAINT*, pages 48–54, 2008.