Preemptive Intrusion Detection

Phuong Cao, Key-whan Chung,
Zbigniew Kalbarczyk, Ravishankar Iyer
Coordinated Science Laboratory
University of Illinois at Urbana Champaign
{pcao3,kchung10,kalbarcz,iyer}@illinois.edu

Adam J. Slagell
National Center for Supercomputing Applications
University of Illinois at Urbana Champaign
slagell@illinois.edu

ABSTRACT

This paper presents a system named SPOT to achieve high accuracy and preemptive detection of attacks. We use security logs of real-incidents that occurred over a six-year period at National Center for Supercomputing Applications (NCSA) to evaluate SPOT. Our data consists of attacks that led directly to the target system being compromised, i.e., not detected in advance, either by the security analysts or by intrusion detection systems. Our approach can detect 75 percent of attacks as early as minutes to tens of hours before attack payloads are executed.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: Security and protection

General Terms

cyber-system, security incident, graphical model

Keywords

credential stealing attack, factor graph, tagging, timeliness

1. INTRODUCTION

Cyber-systems, such as super-computers or data centers, host mission-critical services and valuable data, making them an enchanting target of attacks. Credentials to those systems may be tied to previously leaked credentials: millions of them can be bought from black-markets for a low cost [1]. However, attacks targeting these systems are often discovered when they are in the final stage, resulting in suspension of critical system services or confidential data leak [4].

We focus on detection of attacks that take advantage of stolen credentials ahead of time, i.e., before the system is misused, while minimizing the false positives is a difficult problem. Major challenges are: (i) an early detection means only a partial knowledge of the attack (e.g., system-level events such as logging in from a remote host) is available, (ii) the attackers enter the target system as legitimate users with known credentials – not leaving many traces, (iii) semantics of event logs may be difficult to correlate with the attacker's actions. Examining an event in isolation is not sufficient to detect such attacks. An event represents a fact, but its semantics can be interpreted differently depending on the context. For example, remote login may indicate that the user is traveling or the user account is being compromised. Restarting the SSH daemon could indicate a maintenance activity or an integrity violation of the daemon [4].

This paper presents SPOT – an approach to identify attacks before attack payloads are executed. Events leading to attacks are associated with user states and attack states to understand *user intentions* and *attack semantics*. We assume the events are not contaminated. Theory of *factor graph* (FG) is used to develop a probabilistic model that captures transitions and relations between the events and the states. In a FG, a variable node can be an event (e.g., a security alert) or a state (e.g., the user state is suspicious). The variable nodes are connected by factor functions describing relationships between the nodes. Using the defined factor graph, SPOT *tags*, i.e., determines, the most probable state of the user (e.g., a user is compromised) based on the events observed in real-time. It results in identifying attackers early, i.e., before the system is misused.

2. APPROACH

System model. Consider a *target system* of functional *objects* (components) and *monitors* (deployed to probe objects). A user u interacts with objects in the system. Monitors probe the interactions and emit an event sequence E. An *event* e is a tuple of the user u, the object o, and the event $\mathfrak{e}$ which belong to a finite set of events $\mathcal{E}$. An alert is a critical event that may violates security property of the target system. Additional evidence regarding the user behavior and system performance can be obtained from the user profile u and measurement metrics $M = \{m^i\}$. For example, a metric can be the number of login attempts.

Problem definition. Given *observed evidence* $X = \{$ event sequence E, user profile u, metrics $M\}$, SPOT infers *security state* $Y = \{S\}$ of the target system. Each event e is tagged with state variables: user state s_u and attack state s_a. A *user* can be in one of the three states: $s_u \in \{benign, suspicious, malicious\}$. At first, when no event is observed, a user is *benign*. Its state changes according to observed events. When the user transitions from the *benign* to the *suspicious* state (i.e., the user performs some

HotSoS '14 Raleigh, NC USA

http://dx.doi.org/10.1145/2600176.2600197.

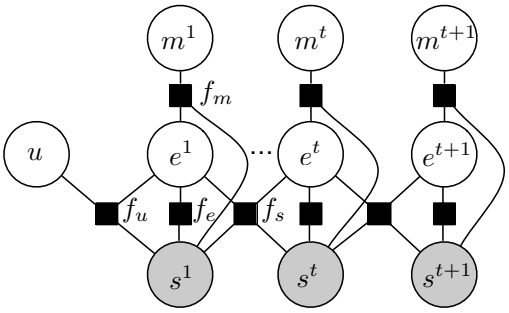


Figure 1: Factor graph model of SPOT. Variable nodes are: user u, events E, metrics M, and states $S = \{s^1, ..., s^t\}, s^i = \{s^i_u, s^i_a\}$ where s^i_u is user state and s^i_a is attack state. Four types of factors connecting variable nodes shown in this example are: user profile factor f_u, metric factor f_m, event factor f_e, and state transition factor f_s.

abnormal activities such as logging in remotely), additional monitors can be enabled to closer scrutinize the user. If further alerts are observed or the user violates a security policy of the target system, the user can be *malicious*, i.e., the user is an attacker. Initially five attack states are considered: {*no attack, gathering information, executing attack payload, establishing backdoor, and cleaning traces*}. An attack state represents attack semantics.

Estimating user state and attack state. Factor graph has been successfully applied in computer vision and robotics [3]. It can represent both Bayesian network and Markov models. We use factor graph to capture relations between observed evidence and the security states. Fig.1 illustrates a factor graph model. Observable variable nodes are user profile u, events $E = \{e^i\}$, and metrics $M = \{m^i\}$. Hidden variable nodes are security states $S = \{s^i\}, s^i = \{s^i_u, s^i_a\}$ where s^i_u is user state and s^i_a is attack state. Factor nodes are: user profile factor f_u, metric factor f_m, event factor f_e, and state transition factor f_s. Formally, relationships between the variable nodes can be expressed as a set of factors F. Each factor f is defined on a variable configuration $x = \{u, E, M\}, y = \{s\}$ and returns a real number in the range $[0, 1]$ as follows: $f : (x, y) \rightarrow \mathbb{R}$. A returned value 1 means a likely variable configuration and 0 means an unlikely variable configuration. For example, a user profile factor f_u can return the value 1 to capture a variable configuration: if the user profile u shows the user has been compromised before and an event e shows the user logs in remotely, then the state of the user s_u is *suspicious*. A factor can capture a rule (return 0 or 1) or a probabilistic knowledge (return a value based on a probability distribution).

The security state S of the target system is determined from the outputs of factors. It is estimated by maximizing the conditional probability over possible combinations of security states $Y = \{S\}$ given the observable evidences $X = \{u, E, M\}$. The conditional probability can be factorized as follows (Hammersley and Clifford theorem [2]):

$P(Y|X) = \arg\max_Y \frac{1}{Z} \prod_{f \in F, x \in X, y \in Y} f(x, y)$

Approximation techniques such as Loopy belief propagation can be used to estimate the security state S [3].

3. EVALUATION

NCSA provides security logs and ground truth of real-world incidents collected over a six-year period (2008-2013). We analyzed 24 credential stealing incidents. During those incidents, total of 5027 users logged in to the target system, and 32 of the users were compromised (i.e., their credentials were stolen and used to gain entry to the target system). *Timeliness* and *accuracy* are evaluated for detection capability (Fig. 2). Let t_0 is the time of the first observed event, t_m is the time when the user is identified as *malicious*, and t_n is the time of the last observed event. Specifically, *detection timeliness* $(t_m - t_0)$ characterizes the responsiveness of SPOT to an attack. *Preemption timeliness* $(t_n - t_m)$ represents the time buffer that security analysts or intrusion response systems have to react to the attack.

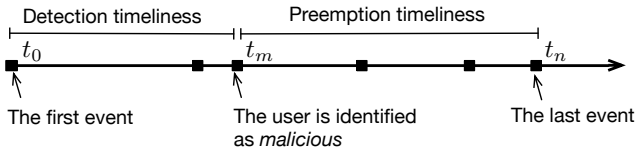
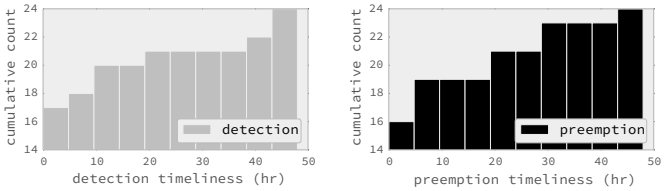


Figure 2: An attack timeline. Each square dot is an event.



(a) Detection timeliness (b) Preemption timeliness

Figure 3: Empirical cumulative number of compromised users identified as malicious, in a function of timeliness

SPOT detects 24 (out of 32) compromised users (75%), with a low false detection rate 1.64%, i.e., a benign user is classified as a malicious user. Fig. 3 plots detection and preemption timeliness for the 24 identified compromised users. Fig. 3a shows empirical cumulative count of users and the corresponding detection timeliness. For example, 17 users (out of 24) are identified as compromised within 5 hours since the first observed event. Similarly, in Fig. 3b, there are 16 users who are identified within 5 hours before the attack payloads are executed. Note that most attacks happen in a 24 hour period. In an extreme case, an attack is identified 48 hours before the attack payloads execute.

4. CONCLUSION

We presented a theoretical framework based on factor graph to understand user intentions and attack semantics from security logs (events). Experimental results on six-year data of real-world incidents are: 75% of attack can be detected early, with a low false positives (1.64%).

5. REFERENCES

[1] Bilton, N. Adobe breach inadvertently tied to other accounts, Nov. 2013.

[2] Hammersley, J. M., and Clifford, P. Markov fields on finite graphs and lattices.

[3] Pearl, J. *Reverend Bayes on inference engines: A distributed hierarchical approach.* UCLA, 1982.

[4] Sharma, A., Kalbarczyk, Z., Barlow, J., and Iyer, R. Analysis of security data from a large computing organization. In *Dependable Systems & Networks (DSN), 2011* (2011).