Future Generation Computer Systems 61 (2016) 26–36

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Unifying intrusion detection and forensic analysis via provenance awareness

Yulai Xie *, Dan Feng, Zhipeng Tan, Junzhe Zhou

School of Computer, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, PR China

HIGHLIGHTS

- We design and implement a provenance-aware intrusion detection and analysis system.
- PIDAS integrates both online intrusion detection with offline forensic analysis.
- PIDAS has high detection rate with low false alarm rate.
- PIDAS can explicitly mark out system vulnerabilities or intrusion sources.

ARTICLE INFO

Article history:
Received 12 November 2015
Received in revised form
5 January 2016
Accepted 17 February 2016
Available online 2 March 2016

Keywords:
Provenance
Intrusion detection
Forensic analysis
False alarm

ABSTRACT

The existing host-based intrusion detection methods are mainly based on recording and analyzing the system calls of the invasion processes (such as exploring the sequences of system calls and their occurring probabilities). However, these methods are not efficient enough on the detection precision as they do not reveal the inherent intrusion events in detail (e.g., where are the system vulnerabilities and what causes the invasion are both not mentioned). On the other hand, though the log-based forensic analysis can enhance the understanding of how these invasion processes break into the system and what files are affected by them, it is a very cumbersome process to manually acquire information from logs which consist of the users' normal behavior and intruders' illegal behavior together.

This paper proposes to use provenance, the history or lineage of an object that explicitly represents the dependency relationship between the damaged files and the intrusion processes, rather than the underlying system calls, to detect and analyze intrusions. Provenance more accurately reveals and records the data and control flow between files and processes, reducing the potential false alarm caused by system call sequences. Moreover, the warning report during intrusion can explicitly output system vulnerabilities and intrusion sources, and provide detection points for further provenance graph based forensic analysis. Experimental results show that this framework can identify the intrusion with high detection rate, lower false alarm rate, and smaller detection time overhead compared to traditional system call based method. In addition, it can analyze the system vulnerabilities and attack sources quickly and accurately.

1. Introduction

Cryptography and access control have long been used to enforce computer security, yet attackers can still exploit a wide variety of system vulnerabilities (e.g., unpatched operating systems, programming bugs, firewall misconfigurations, weak passwords, etc.) to compromise computer systems, leading to leakage or corruption of sensitive data. For instance, in April 2010, the account information of more than six million Internet users of China Software Developer Network was leaked [1]; in April 2014, about half a million of web servers in the wild are vulnerable due to the Heartbleed security bug found in OpenSSL [2].

This work explores how to defend against host-based intrusions. Existing approaches can be classified as *online* and *offline* approaches. Online approaches, which we term as *intrusion detection* approaches (e.g., [3–5]), often examine system call sequences of a running process and infer its system status in real time; in contrast, offline approaches, which we term *forensic analysis* approaches (e.g., [6,7]), conduct forensic analysis on system logs so as to trace back the root cause of why and how an intrusion has happened. Due to performance constraints, existing intrusion detection approaches do not keep detailed logs to support forensic analysis. On the other hand, existing forensic analysis approaches

* Corresponding author.
E-mail address: ylxie@hust.edu.cn (Y. Xie).

http://dx.doi.org/10.1016/j.future.2016.02.005
0167-739X/

often take substantial human efforts to mine the logs, thereby prohibiting real-time detection. Although some studies [8,6] have attempted to reduce the size of logs that need to be queried by carefully differentiating the types of intrusions and prioritizing the important components for inspection, digging out the useful information from the system logs remains relatively slow when compared to real-time intrusion detection.

A key motivation of this work is to design a unified approach that seamlessly integrate both online intrusion detection and offline forensic analysis, so as to effectively defend against host-based instructions. Such a unified design is necessary due to the coherence of the two components, that is, we can leverage the results of intrusion detection to facilitate forensic analysis to accurately and efficiently trace back the root cause of an intrusion. Since the effectiveness of the unified design heavily depends on how we interpret intrusions, we pose the following question: *how can we explicitly and efficiently capture and represent intrusion information, so as to simultaneously support both online intrusion detection and offline forensic analysis?*

In this paper, we propose to leverage *provenance*, which represents the history of a system object (e.g., file, process, socket), to capture the data flows and dependency relationships among objects in a *structured* manner [9]. Provenance has been widely used in different communities such as scientific computing, databases [10], and storage systems [9]. Its typical applications include experimental documentation, debugging, search [11], and security [12]. By carefully collecting and storing provenance information during intrusion detection, we preserve the performance of online intrusion detection, while keeping sufficient evidence for accurate offline forensic analysis. Provenance has been well recognized as a useful element for defending against intrusions [9]. Nevertheless, to the best of knowledge, there is no existing intrusion defense system that systematically realizes the benefits of provenance.

To this end, we design and implement *PIDAS*, a Provenance-aware Intrusion Detection and Analysis System that integrates both online intrusion detection and offline forensic analysis. PIDAS functions in three steps: collecting provenance, detecting intrusions, and analyzing system vulnerabilities and intrusion sources. The provenance collecting step uses PASS [9] to collect and store the provenance of objects (e.g., files, processes, and sockets) in key–value databases. The detecting step extracts dependency relationships from these provenance information, and builds a normal database. Then it judges whether an invasion has happened by comparing the weight factor of a certain length of path that comprises a series of dependency relationships to a predefined threshold. The weight factor represents the ratio of dependency relationships that are judged as abnormal in the path. A big weight factor indicates a large number of abnormal dependency relationships, and thus the program that generates the path will more likely be a real intrusion. The warning report in the detection step explicitly reveals the system vulnerabilities or intrusion sources. Thus it can help reduce the false alarm rate, and can also provide important clues for further forensic analysis. Finally, the analysis step constructs provenance graphs that clearly mark out the entire attack path. The administrator can analyze each event on the invasion path, thus conveniently finding all the files affected by the invasion process and easily analyzing system vulnerabilities.

We compare PIDAS with system call based method in terms of the false alarm rate, detection rate, and detection time on a series of typical security-critical applications. We also investigate how warning reports provided by PIDAS can help reduce false alarm rate and facilitate forensic analysis.

The contributions of this paper are:

1. A provenance-based intrusion detection algorithm that has high detection rate with lower false alarm rate and smaller detection time overhead compared to system call based intrusion detection approach.
2. The warning report during intrusion detection can explicitly mark out system vulnerabilities or intrusion sources, provides detection points for provenance graph-based forensic analysis, and can further reduce false alarm rate.
3. The design and implementation of PIDAS that builds upon an existing provenance tracking framework.
4. The evaluation of PIDAS on extensive and representative security-critical applications.

The rest of the paper is organized as follows. We describe the design goal and threat model in Section 2 and elaborate the design and implementation of PIDAS in Section 3. In Section 4, we evaluate the performance of PIDAS. In Section 5, we summarize the related work. In Section 6, we conclude the paper.

2. Setting

PIDAS aims to detect intrusions and analyze system vulnerabilities and intrusion attack sources. In this section, we first give an overview of provenance, then we describe the PIDAS design goals and threat model.

2.1. Provenance

Provenance represents the history or lineage of an object. In the system area, it means all the processes and inputs that affect the data. Provenance is commonly in the form of directed acyclic graph, where the nodes are digital objects such as files and processes, and the edges are the dependency relationships between files and processes. For instance, a process issuing a read system call on a file indicates that this process depends on the data information from the file, and this implies an edge from a process pointing to a file. A number of provenance systems (e.g., PASS [9], SPADE [13], Story Book [14], ES3 [15], TREC [16]) have been built by the provenance community. For instance, the provenance-aware storage system (PASS) [9] intercepts the system calls events that represent the dependency relationships between files and processes, converts them to causality-based provenance graphs, then stores the provenance graphs in BerkeleyDB. PASS can use the local, network attached [17] or cloud [18] as the storage backend. It provides the storage, query, and management of provenance.

PIDAS mainly concentrates on the provenance (or history) of three kinds of objects: files, processes, and sockets. The provenance of these objects includes their own attributes such as name, id, version number, port number, etc., and the dependency relationships between them, for instance, the causality relationship exists between a virus process and the file it infects. In this model, a series of consecutive dependency relationships comprise an intrusion path, and different intrusion paths can have common edges. For example, Fig. 1 shows a provenance graph that describes how remote attack exploits the vsftpd daemon and tampers the files.

2.2. Threat model

The primary threat that PIDAS guards against is the intrusions induced by exploiting the application or process vulnerabilities. For instance, a remote attacker can exploit the vulnerabilities in the local server process (e.g., vsftpd, samba, distccd, etc.) and take full control of the system. The intruder then can read or tamper the data in the file system, and download any worm or trojan programs into the system. PIDAS can track this whole process by intercepting