ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/269303114

Towards a Framework to Detect Multi-stage Advanced Persistent Threats Attacks

Conference Paper · April 2014

DOI: 10.1109/SOSE.2014.53

CITATIONS
56

READS
1,864

3 authors:

Parth Bhatt
CPQD
5 PUBLICATIONS **62** CITATIONS
SEE PROFILE

Edgar Toshiro Yano
Instituto Tecnologico de Aeronautica
34 PUBLICATIONS **152** CITATIONS
SEE PROFILE

Per M. Gustavsson
George Mason University
78 PUBLICATIONS **409** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Cyber Security Awareness View project

Project Education, Training and Simulation - methods and technology View project

All content following this page was uploaded by Per M. Gustavsson on 08 June 2015.

The user has requested enhancement of the downloaded file.

Towards a Framework to Detect Multi-Stage Advanced Persistent Threats Attacks

Parth Bhatt[1], Edgar Toshiro Yano[2]
Dept. of Electronics and Computer Engineering
Instituto Tecnológico de Aeronáutica
São José dos Campos,SP,Brasil
parthbhatt09@gmail.com[1],
yano@ita.br[2]

Dr. Per M. Gustavsson[3]
[3]Combitech Sweden / Swedish National Defence College /
George Mason University, USA
per.m.gustavsson@combitech.se[3]

Abstract— **Detecting and defending against Multi-Stage Advanced Persistent Threats (APT) Attacks is a challenge for mechanisms that are static in its nature and are based on blacklisting and malware signature techniques. Blacklists and malware signatures are designed to detect known attacks. But multi-stage attacks are dynamic, conducted in parallel and use several attack paths and can be conducted in multi-year campaigns, in order to reach the desired effect. In this paper the design principles of a framework are presented that model Multi-Stage Attacks in a way that both describes the attack methods as well as the anticipated effects of attacks. The foundation to model behaviors is by the combination of the Intrusion Kill-Chain attack model and defense patterns (i.e. a hypothesis based approach of known patterns). The implementation of the framework is made by using Apache Hadoop with a logic layer that supports the evaluation of a hypothesis.**

Keywords—APT; Multi-stage Attack; Hadoop; Intrusion Kill Chain;

I. INTRODUCTION

Currently, cyber systems are being attacked by complex, persistent and stealthy attacks, also referred as APT (Advanced Persistent Threats) [1][2]. An APT usually has multiple stages [13]. At each stage the attacker gets more privileges, information and resources to penetrate deeper within the organization. Persistency means that the attacker will persist patiently for a long time in their attempts to reach the desired goal. The attacker does not give up easily and deviate from their targets. Thus, he has a well defined goal and he will persist until his goal is achieved. The attackers are supposed to be supported by organizations or nations with capabilities and resources to support their aims.

To deal with this new scenario it is necessary that the defense adopts a proactive behavior. That is, an attack must be perceived and treated before it causes significant impacts to the business. The current approach to security, with static models of risk, compliance to standards and regulations and incident handling after impact, is no longer acceptable. Security must be dynamic, with risks assessed continuously and proactively with treatment actions being performed before significant impacts are realized.

A current weakness to deal with this new scenario is the difficulty in constructing, operating and maintaining an appropriate defense system. Even larger organizations with sophisticated defenses are targets of attacks. So, there is a demand for frameworks to support the implementation of effective solutions in order that even organizations with fewer resources and knowledge can reasonably handle complex and persistent attacks. In this paper we present a research framework to handle complex attacks. The central basis of the framework consists of an Intrusion Management System and a multi-stage attack model. The multi-stage attack model is used to identify prevention and detection controls that provide logs used by the Intrusion Management System, and it is also used as a guide to logs correlation activities.

In the next section we present characteristics of APTs and difficulties of current approaches to treat them. In section III we present the framework with the underlying models and architectural principles. In Section IV we present a process with correlation patterns to detect an APT. In section V we present related work. And finally in section VI, we present conclusions and suggestions for future works.

II. APT – ADVANCED PERSISTENT THREATS

A. APT Attack scenario

A complex attack can overwhelm the defenses of a system through a well-planned operation to explore existing weaknesses. The attacker first identifies potential targets in the organization. The selected targets are ceasing to be services or applications, as these are generally better protected and monitored. A common target is a user within the organization with a closer access to assets desired by the attacker. He or she may be the target of a focused phishing attack, or receives a gadget at a conference or exhibition, or is convinced to bring a malicious device inside the organization. Once inside the supposedly secure network, the malware establishes a stealthy communication channel with the attacker, and exploring other weaknesses, the attack advances over other users and resources to achieve its final goal.

B. *Explored Weaknesses*

This scenario is possible because even well designed defenses have blind spots. An anti-virus is unable to detect a malware not registered in the database of signatures, an IDS (Intrusion Detection System) is only effective if an attack triggers a registered detection rule, and frequently it generates many false positives and negatives and so it is often overlooked by security administrators (if the organization has a fully qualified one). Alerts from different security sensors are hardly correlated. Already fixed vulnerabilities remain present for a long time. Vulnerable applications settings are used by many users. Even users with access to sensitive assets do not have adequate training and awareness. These different vulnerabilities could be discovered by an adversary through a combination of social engineering and network reconnaissance attacks.

III. PROPOSED FRAMEWORK

A research framework is being developed to support the detection and analysis of multi-stage cyber-attacks. The framework has the following main components:

- A Multi-stage Attack Model.
- A Layered Security Architecture.
- A Security Event Collection and Analysis System

A. *Multi-stage Attack Model*

The treatment of a cyber attack requires the use of an appropriate attack model. Using an attack model it is possible to recognize the current state of an attack and its possible future states. An attack model is as a model of hypothesis which will be used to infer possible actions of attackers. We adopted the Intrusion Kill Chain (IKC) [3] model as the central basis of our attack model. IKC is a model of seven phases that an attacker inescapably follows to plan and carry out an intrusion. The IKC phases are as follows:

- *Information Gathering* – Selection of targets, collecting information about the target, technologies the target uses, potential vulnerabilities, etc.
- *Weaponization* – developing malicious code to explore identified vulnerabilities, coupling the developed code with unsuspected deliverable payloads like pdfs, docs, and ppts.
- *Delivery*- Transferring the weaponized payload to the target environment.
- *Exploitation* - Use of vulnerability of a target system to execute a malicious code.
- *Installation* - Remote Access Trojan’s (RAT) are generally installed which allows adversary to maintain its persistence in the targeted environment.
- *Command and control (C2)* - Adversary requires a communication channel to control its malware and continue their actions. Therefore, it needs to be connected to a C2 server.
- *Actions* – it is the last phase of the kill chain in which adversary achieves its objectives by performing actions like data exfiltration. Defenders can be confident that adversary achieves this phase after passing through previous phases.

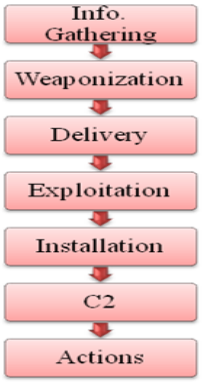

Figure 1: Intrusion Kill Chain (IKC)

To defeat more sophisticated defense systems, attackers may require the execution of one or more IKCs to circumvent different defensive controls. So, an adequate representation of a complex attack is a multi-stage model, with each stage represented by an IKC divided in its seven phases.

B. *Layered Security Architecture*

The detection of a complex attack in its earlier stages is possible if we increase the difficulties for the attacker to access the valuable assets. The attacker will need to invest more resources and time to reach the targets. The likelihood, that one or more sensors are activated and the attack is detected, increases with the number of interactions of the attacker with the targeted system. A pattern to facilitate detection of a complex attack is to protect assets by using a layered model. Most valuable assets should be in the inner layers. The logic is to force the attacker to execute an attack with multiple stages. For each layer, at least once, the seven phases of an IKC will need to be executed. So, there will be at least seven opportunities for detecting an attack on a layer.

To be effective, the layered model should attend the following requirements:

- The access to a layer will only be possible through processes and applications of the immediately outermost layer. The attacker will have first to get an access to the outermost layer.

- To circumvent the controls to get an access to a layer, the attacker will have to execute a kill chain from the outermost layer.
- The probability of finding common vulnerabilities in controls, that are used to defend the different layers, must be very low. The idea is to minimize the reuse of knowledge about vulnerabilities of a layer to attack another layer. The defense can hinder the attack, forcing the adversaries to collect more information and to develop new weapons to bypass each different layer.

C. A Security Event Collection and Analysis System

An effective detection is possible only with appropriate sensors that detect different facets of an attack. One possible approach is to provide each layer with sensors to detect different phases of an IKC. The sensors are triggered by rules established in accordance with patterns of a malicious behavior. Each layer must have its own set of sensors configured to detect an IKC inside that layer. Alerts and logs collected by the sensors should be stored and correlated to identify stages and phases of attacks in progress.

The process of collecting and correlation requires an infrastructure that can become difficult to properly operate and maintain. A small network (about 100 hundred hosts) can generate around 100 GB of daily logs and alarms [4,10]. Considering that an APT attack can last months or even years, a large organization may require a significant investment to establish a system for collecting and analyzing logs.

In order to attend this need, a model of collecting data based on Big Data technology was designed. This model was implemented using Hadoop. Apache Hadoop [5] is an open source framework that allows distributed processing of large collection of data using cluster of computers each having local computation and storage. Hadoop provides high availability, fault tolerance and faster processing speeds of large (structured, semi-structured or un-structured) data sets even with cheap commodity hardware.

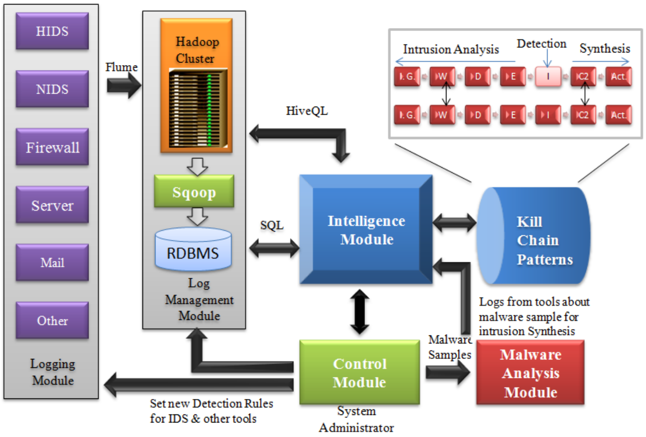

Figure2. Overview of Complete framework

Our framework using Hadoop is divided into 5 modules namely, Logging Module, Log Management Module, Malware Analysis Module, Intelligence Module and Control Module.

Logging Module

This module of consists of sensors from the security architecture. It typically consists of HIDS (Host intrusion detection system) and NIDS (Network intrusion detection system), Firewall logs, Web Server logs, Mail Server logs, etc. The rules and configuration for log generation can be set by the administrator using the Control Module. This Module executes a normalization task [6] to enable uniformity in the analysis process.

Log Management Module

All the logs generated in the Logging Module are moved to this module, stored and pre-processed in the Hadoop Distributed File System (HDFS) [7]. The logs are accessed using Hive queries and for point queries on a small amount of logs a MySQL data base is used.

Intelligence Module

Intelligence module contains the algorithms for log correlation and is responsible for automatic IKC search based on potential malicious events detected. Trigger events are the events on which the Intelligence Module that can initiate an IKC reconstruction. Trigger events can be rule based or a system administrator input. Generally, a trigger is a NIDS or HIDS high risk alert. A multi-stage attack may persist for a long time period. In order to enable this type of analysis, the intelligence module has a campaign analysis component. With the campaign analysis previous attacks data are collected and correlated in order to identify a potential multi-stage attack.

The Intelligence Module activities are explained in section IV.

Malware Analysis Module

Malware analysis module consist of a malware analysis virtualized Lab Environment with detection tools. Explaining malware analysis in detail is out of scope of this paper. The primary approaches for malware analysis are Code Analysis and Behavioral Analysis. There are several tools that help to perform such analysis of executables. The malware analysis module provides a more detailed understanding of the possible actions and effects of a malware.

Control Module

Using the control module, the administrator governs the framework. The administrator can set new rules for the logging module, manage the cluster of the log management module, or test hypothesis with the intelligence module.

IV. Intelligence Module Log Analysis Process

The defense of a layer must be able to detect an IKC identifying one of its seven stages. For each layer, it is elaborated a defense plan, as illustrated in Table I. A defense plan identifies for each phase of an IKC the attack mechanisms that can be used at each stage, and the controls to prevent and detect the attacks. Each row of the table can be understood as a defense line which can prevent or detect a phase of an IKC. The attack mechanisms were extracted from CAPEC list maintained by Mitre [8].

TABLE I. Defense Plan

Phase, Defense Line	***CAPEC Attack Mechanisms***	***Prevention***	***Detection***
Info. Gathering	Social Engineering	Security Awareness and Training	User monitoring
	Network Recognition	IPS[a], Firewall	NIDS[b], NABD[c]
	Data leakage	IPS, Firewall, Proxy	NIDS, NABD
	Fingerprinting	IPS, Firewall, Information Obfucation	NIDS, NABD
	Footprinting	IPS, Firewall	HIDS[d],NIDS
Weaponization		Patching, Auditing, Vulnerability Scanning	
Delivery	Spear Phishing	Content Filtering, Identity Verification, Blacklisting	Source Correlation
	Action Spoofing	Content Filtering	Proxy
	Injection	IPS, Input Filtering	NIDS, HIDS
	Supply-Chain Attack	Security Life Cycle	NIDS, HIDS
	Hacking Hardware Devices	Configuration Control	HIDS
Exploit	Data Structure Attacks	Patching	HIDS
Execution	Privilege Escalation	Password Control, Firewall	HIDS
C2		Firewall, Proxy, Encryption Use Control, blacklisting	HIDS, NIDS, Content Analysis
Actions	Exploitation of Privilege/Trust	Password Control, Firewall	HIDS
	Resource Manipulation	Firewall, Proxy, Encryption Use Control	HIDS, NIDS
	Resource Depletion	IPS	HIDS, NIDS

IPS – Intrusion Prevention System

NIDS – Network Intrusion Detection System

NABD- Network Anomaly Behavior Detection

HIDS-Host Intrusion Detection System

It is worth noting the absence of attack mechanisms for the phases of Weaponization and C2. At the stage of Weaponization the attacker does not interact directly with the system to be attacked. In C2 phase, the attacker has obtained sufficient privileges to establish a communication channel using authorized means. He will try to maintain a usage profile that will not provoke attention from defensive controls.

The main input to the intelligence module is the collected logs from the different prevention and detection controls. Each log is normalized [6] to provide attributes that identify the control, date and time, type of attack, source, destination, and payload attributes. Each collected log is like a part of a puzzle. The complete puzzle is an IKC of a multi-stage attack. The process to analyze the logs is composed of the following activities:

- *Identify the Defense Line.*
- *Identify the phase of an IKC.*
- *Rebuild an IKC.*
- *Identify a multi-stage attack.*

The process starts with the trigger of an alarm activated by the different controls. Alarms are logs classified as critical and requiring immediate security management attention.

A. Identify the Defense Line

The same type of control can be used in different defense lines. The most likely defense line can be identified by the rule that triggered the alarm with correlation with other logs from controls of the defense line. The main attribute for correlation is the time-stamp. Logs of different controls are verified by time proximity. The other attributes (type of attack, source, target, and payload) can be used to correlate with other IKCs already identified. The defenders have the advantage of being able to simultaneously view events in different hosts, networks and controls. For example, a log in the same kind of control in other parts of a network may mean a coordinated attack in progress.

B. Identify the Phase of an IKC

The identification of the defense line may not be accurate which leads to one or more possible phases. For example, Privilege Escalation type of attack may be an IKC at the Actions or Installation Phase. The different alternatives must be marked for review by the IKC rebuild activity.

C. Rebuild an IKC

An identified phase leads to a process of searching for the earlier phases of an IKC. The main element of correlation is still time. However, it should take into account that an APT attack can take a long time. The time interval between two phases can be long. For example, between C2 and Actions phases an attacker may choose to keep the malware dormant until he or she feels confident to start the Actions phase. This can take months or even years. However, between the Exploit and Installation phases time is usually short because the attacker in general, due to peculiarities of the exploited vulnerability, has a limited window of time to finish the

Installation phase operations. The reconstruction of an IKC is an activity that leads to reduction of false positives. The discovering of earlier phases increases the probability that there is an ongoing IKC.

D. Identify a multi-stage attack

Each IKC is a stage of a complex attack. An IKC in a layer may have started from an IKC executed in the outer layer. But it can also be initiated by an IKC in the same layer. For example, an attacker got first an access with restricted privileges and then ran a second IKC to increase his privileges. The linkage between two IKCs may be detected correlating attributes from earlier phases of an IKC with Actions phase of older IKCs.

SEMI-AUTOMATIC PROCESS

The correlation process requires the command of an experienced analyst. The beginning of each activity is performed in accordance with guidelines from an expert. Upon receiving an alert, the system identifies possible defense lines involved. The expert selects the most promising IKC phase and calls for the reconstruction of the IKC.

The automatic event correlation to detect APTs is a research issue. Some promising approaches are the using of probabilistic techniques such as Hidden Markov Models (HMM) [9]. Our framework can be used as a research platform for the development of algorithms for detecting complex multi-stage attacks.

V. IMPLEMENTATION

The framework described in the previous section was implemented for basic testing purposes. A Hadoop Cluster with 5 nodes was implemented using commodity hardware obtained from a previous project, each machine powered by Intel ® Core™ 2 Duo CPU E4500 @ 2.20 Ghz × 2 with 2GB of RAM, 80GB Hard Disk, 32bit machines forming a homogeneous cluster. One machine was set as master node and other four as slave nodes. The master node was configured with Apache Sqoop [14] and MySQL. A Fast Ethernet Switch was used for the networking within the cluster nodes.

Furthermore, simple implementation of Intelligence module was realized using a intrusion analysis algorithm in java program that could access data present on the HDFS in Hive external tables using Hive Thrift service, Sqoop using SqoopOptions class and MySQL. The inputs to the algorithm are alerts generated by intrusion detection systems present in the logging module of the framework. The Flume was configured to transfer data synchronously from different components of the logging module into HDFS. Malware analysis module was left for future implementation.

VI. EXPERIMENTS AND RESULTS

A. Experimental Intrusion Scenario

A scenario of a university network getting attacked by APT is considered. We consider that the network is equipped with a complete framework as described in the Section III. As it is a layered architecture, the attackers are able to reach only the outermost layer in their first attempt. In the outermost layer, the easiest way is to get into university professors mailboxes. The attacker targets some of the professors and performs an initial reconnaissance about their interests. He gets a conference of potential interest and weaponizes its pdf flyer. Next, he crafts a Targeted malicious email (TME) with a malicious pdf flyer in the attachment and finally sends this email to the target completing the delivery phase of an IKC.

Upon reception of the email, the professor downloads the pdf flyer to get more details. As soon as the malicious pdf flyer is opened, the malicious code gets executed, and within fraction of seconds the original pdf flyer is displayed to the professor. This complete process appears to be normal, but the execution of the malicious code was due to exploitation of one of the vulnerabilities of the pdf reader application which leads to the installation of a malicious code. During the installation process, the HIDS generates an alert about a file modification of the windows file "explorer.exe".

B. Experiment

Logs were simulated according to the experimental intrusion scenario. Log entries were created for components of logging module such as OSSEC [15] logs and Mail logs.

The alert from OSSEC syscheck informing the file modification of "explorer.exe" becomes the input for the intelligence module and it starts the IKC reconstruction. The first step was to associate this alert to one of the phases of the IKC. It was identified as the installation phase. Thus, the previous phases of this IKC are needed to be discovered. The intelligence module performs the intrusion reconstruction. Some of the phases of any IKC, such as installation and exploitation are generally in a very close temporal proximity, thus logs based on such temporal proximity are moved from Hive external tables into MySQL using Sqoop and where many point queries can be performed in realtime to search information to complete the kill chain reconstruction. Phases such as weaponization depend on the malware analysis module that was not implemented yet.

The total number of log records fed into Hadoop HDFS were 69,969,233 and 5 Hadoop nodes were used. The process time to get the information based on the IDS alert was 7 minutes and 38 seconds. Using 5 node Hadoop cluster, we were able to process huge amount of semi-structured logs.

The IKC reconstruction process results is given in table II:

TABLE II: KILL CHAIN FOR OUTERMOST LAYER

Info. Gathering	Mailing List , Xyz conference Website, University Website
Weaponization	Malware Analysis Lab
Delivery	adm@xyzconference.com ip : 161.xyz.pq.35 Sub: Xyz conference 2013 xyz2013.pdf
Exploitation	0-day PDF
Installation	Windows file modification detected " explorer.exe"
C2	Left for intrusion Synthesis phase
Actions	-

VII. Related Works

Due to increase in number of sophisticated threats and great increase in volume of data traffic, the landscape of analysing log data has drastically changed, as now working with log data has entered in the category of Big Data problem [10].

J. Howes, J. Solderitsch, I. Chen and J. Craighead [11] proposed an analytical security model considering the security analytics using Big Data. Their architecture is directed towards dealing with operational concerns in security organizations that aim to use existing security tools with Big Data analytics. Since their work is aimed towards operational side of security analytics therefore, it does not demonstrate any methodology of practical analysis of security threats as compared to our framework.

J. Therdphapiyanak and K. Piromsopa [12] used Hadoop map reduce model to analyze high volume of log files from server and distributed intrusion detection system and they proved that their frameworks performance was better than a standalone intrusion detection system. They were able to extract important information from the large security logs using their analysis and scalability of Hadoop, but their work was limited to use of K-means clustering algorithm from Mahout[5] for detection of the deviated behavior Clusters from normal behavior Clusters. Using the proven capabilities of Hadoop for log analysis as in [12], our proposed framework is directed towards practical analysis of dealing with Targeted threats.

VIII. Conclusions

APT attacks are a major challenge for current cyber defenses. We present the design of a framework for detecting APTs. The conception used well known defense patterns in order to increase the difficulty in performing multi-stage attacks and thereby increasing the likelihood of early detection of such attacks. The use of the IKC attack model allows a better tuning of the configuration of security controls and it can be used as a hypothesis model to improve the correlation of logs and thereby facilitate the identification of ongoing attacks.

A prototype of the proposed framework was developed and it was successfully tested with simulated attack data. Currently, we are going to test it with data of a real installation, and we are also making efforts to improve the human interface to facilitate experimentation with different correlation algorithms.

Acknowledgments

The authors would like to thank the following organizations for their support: University of Skövde (HiS),Sweden, Swedish National Defence College (SNDC),Sweden, Combitech AB, Sweden, and National Council for Scientific and Technological Development (CNPq), Brazil.

References

[1] Li F, Atlasis A,“ A Detailed Analysis of an Advanced Persistent Threat Malware”, 2011, SANS Institute InfoSec Reading Room

[2] Sood A.K., Enbody R.J. “ Targeted cyber attacks: A Superset of advanced persistent threats” Security & Privacy, IEEE Volume 11 , Issue 1 2013

[3] Hutchins Eric M., Cloppert Michael J., Amin Rohan M,“Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains” ICIW2011

[4] Brad Hale, “Estimating Log Generation for Security Information Event and Log Management” ,http://content.solarwinds.com/creative/pdf/Whitepapers/estimating_log_generation_white_paper.pdf[accessed 1 Jan 2014]

[5] Tom White “Hadoop: The Definitive Guide”, 2009, 978-0-596-52197-4

[6] Kruegel, Christopher, Valeur, Fredrik and Vigna, Giovanni. *Intrusion Detection and Correlation - Challenges and Solutions*. Vol. 14. : Springer, 2005.

[7] HDFS Architecture Guide http://hadoop.apache.org/docs/stable1/hdfs_design.html [accessed 1 Jan 2014]

[8] CAPEC – Common Attack Pattern Enumeration and Classification online Mechanism of Attack at http://capec.mitre.org/data/definitions/1000.html [accessed 1 Jan 2014]

[9] Dr. Dirk Ourston, Ms. Sara Matzner, Mr. William Stump, and Dr. Bryan Hopkins,Applied Research Laboratories University of Texas at Austin Applications of Hidden Markov Models to Detecting Multi-stage Network Attacks, Proceedings of the 36th Hawaii International Conference on System Sciences, 2003.

[10] MacDonald, Neil, 2012, Information Security is Becoming a Big Data Analytic Problem, Gartner, (23 March 2012), DOI= http://www.gartner.com/id=1960615

[11] J. Howes, J. Solderitsch, I. Chen & J. Craighead, “Enabling trustworthy spaces via orchestrated analytical security”, ACM, CSIIRW 2013, Article No. 13

[12] J. Therdphapiyanak, K. Piromsopa, “Applying Hadoop for log analysis toward distributed IDS” ACM ICUIMC 2013, Article No. 3

[13] Vries, J.D. and Hoogstraaten H. and Berg, J.V.D. and Daskapan S, “Systems for Detecting Advanced Persistent Threats CyberSecurity” 54-61, IEEE Computer Society 2012

[14] Apache Sqoop http://sqoop.apache.org/ [accessed 1 Jan 2014]

[15] Hay Daniel Cid, R. B. A. OSSEC Host Based Intrusion Detection Guide. [S.l.]: Syngress Publishing, Inc., 2008.

View publication stats