Actionable threat intelligence for digital forensics readiness

Nikolaos Serketzis
School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece

Vasilios Katos
Department of Computing, Bournemouth University, Poole, UK

Christos Ilioudis
Department of Information Technology, Technological Educational Institute of Thessaloniki, Thessaloniki, Greece

Dimitrios Baltatzis
International Hellenic University, Thermi, Greece, and

George J. Pangalos
School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece

Received 15 September 2018
Revised 9 November 2018
Accepted 20 November 2018

Abstract

Purpose – The purpose of this paper is to formulate a novel model for enhancing the effectiveness of existing digital forensic readiness (DFR) schemes by leveraging the capabilities of cyber threat information sharing.

Design/methodology/approach – This paper uses a quantitative methodology to identify the most popular cyber threat intelligence (CTI) elements and introduces a lightweight approach to correlate those with potential forensic value, resulting in the quick and accurate triaging and identification of patterns of malicious activities.

Findings – While threat intelligence exchange steadily becomes a common practice for the prevention or detection of security incidents, the proposed approach highlights its usefulness for the digital forensics (DF) domain.

Originality/value – The proposed model can help organizations to improve their DFR posture, and thus minimize the time and cost of cybercrime incidents.

Keywords Information security, Cybersecurity, Cyber threat intelligence, Digital forensic readiness, Digital forensics, Indicators of compromise

Paper type Research paper

1. Introduction

Assisted by the proliferation of sophisticated, free or low-cost malware toolkits, attackers can easily produce various samples of badware while staying undetected even by the most updated anti-malware software solutions. It is estimated that approximately 600 million malware samples appeared by the end of 2016, while the average lifespan of malware file hashes is less than an hour (European Union Agency For Network And Information Security (ENISA), 2017).

Information & Computer Security
Vol. 27 No. 2, 2019
pp. 273-291
© Emerald Publishing Limited
2056-4961
DOI 10.1108/ICS-09-2018-0110

Such threats require efficient digital forensics mechanisms. Digital forensics (DF) has been maturing over the last four decades, but developments within the information technology (IT) industry and the disruptive nature of IT innovation challenge its effectiveness (Garfinkel, 2010).

The need for time-efficient and cost-effective DF were the reasons the concept of digital forensic readiness (DFR) emerged (Tan, 2001). However, research in DFR is yet at an abstract level, with most of the academic publications containing generic steps for DFR modeling while lacking contextualization.

The contribution of this paper is the development of a holistic DFR methodology which combines elements from the fields of information security, DF and cyber threat intelligence (CTI) to facilitate the DFR process. This approach aims to efficiently analyze, correlate and discover potential threats by cross-matching diverse large-volume e-evidences, collected from the local operational environment achieving to reduce the analysis time and cost, required to detect and respond to security incidents.

This paper is structured as follows. A brief review of the current threat landscape and the existing defense approaches are reported in the introduction. Section 2 presents the background work in the fields of DF, DFR and CTI. Section 3 discusses the architecture of the proposed model, while Section 4 presents a prototype of the model implemented. Section 5 outlines the results of the initial experiments, and Section 6 outlines the benefits of this research.

2. Background theory and related work

2.1 Digital forensics and digital forensic readiness

According to Grobler and Louwrens (2007), Kebande and Venter (2015), Palmer (2001), Pangalos and Katos (2010) and Rowlingson (2004), DF is a methodology that utilizes scientifically proven methods toward the preservation, identification, acquisition, analysis, interpretation and documentation of digital data and/or root cause analysis and presentation of digital sources for reconstructing suspicious events. The above definition indicates that DF mainly concerns about forensic procedures, rules of evidence and legal approaches, but lacks consideration for big volume and quality of collected data.

DFR is a term that firstly introduced by Tan (2001) and aims to maximize the associated mechanisms' capability of collecting DF data while minimizing the cost of forensic investigation during incident response (Rowlingson, 2004). Achieving forensic readiness presumes the coordination of multiple factors like legislation, people, processes, policies, governance and technology (Grobler *et al.*, 2010). Proactive readiness, in turn, must take into account several factors like data collection, analysis and storage methodologies limitations. The next section discusses existing methodologies of DFR systems.

2.2 Digital forensic readiness systems

The need for immediate response to incidents has driven the proposal of various DFR systems. Focusing on WLAN communications, Ngobeni *et al.* (2012) described a model that proactively collects, analyzes and stores network traffic that is exchanged between wireless devices and access points, and is thus present and easily retrievable.

Spyridopoulos and Katos (2011) introduced the concept of forcing cloud node failures to cause the transfer of digital evidence between jurisdictions aiming to solve the challenges of evidence collection efficiently. However, such a strategy would potentially cause a significant up-rise of high-volume data transfers.

Kebande *et al.* (2016) presented an approach for achieving forensics readiness in the cloud. Their work uses non-malicious bots that inject cloud machines and transfer potential evidence to a centralized data store where pre-incident detection and analysis takes place.

Inspired by well-established intrusion detection systems (IDSs) and security event management solutions (SEMs), the DFR management system (Reddy and Venter, 2013) proposed aims to coordinate the mass amount of resources an organization possesses, and thus improve its readiness state. The system collects events from various resources, compares them against a predefined set of definitions and alerts users for taking further actions.

Following a different approach, Shields *et al.* (2011) introduced a system that is designed to answer queries about user actions. In essence, they achieved user tracking through the analysis of a calculated set of terms produced by the files each user accessed (Shields *et al.*, 2011).

2.3 Cyber threat intelligence

Threat intelligence is analyzed information about the intent, opportunity and capability of malicious actors, concerning potential or current attacks that threaten an organization. Also known as CTI, it refers to what cyber threat information becomes after collected, evaluated in the context of its source and reliability, organized, analyzed through rigorous and structured tradecraft techniques and refined by security analysts. The primary purpose of threat intelligence is helping organizations understand the risks of the most common and severe external threats, such as advanced persistent threats (APTs) and zero-day exploits. Threat intelligence includes in-depth information about specific threats to help organizations protect themselves from the types of attacks that could do them the most damage (Wigmore, 2015).

Conventional information security defense techniques for detecting and mitigating security incidents raise questions about their effectiveness, as they are time-critical. According to Verizon (2018), 87 per cent of the compromises occurred within minutes are often unable to identify incidents that rely on zero-day exploits. On the other hand, recent developments in the domain of Big Data and CTI seem to better serve the cyber-war battle, and thus seem to be very promising approaches (Virvilis *et al.*, 2014).

Several frameworks have been developed that aim to describe and share threat intelligence in a structured manner. STIX (MITRE, 2017), OpenIoC (Mandiant Corporation, 2013) and IODEF (Danyliw *et al.*, 2007) are among them, with the former being considered the de facto threat intelligence representative (Sauerwein *et al.*, 2017). Using markup languages like XML and JSON, these frameworks can communicate threat information between systems without user intervention, thus narrowing the time needed to respond to security incidents.

2.4 Threat intelligence platforms

The need for effectively managing threat information has emerged the development of threat intelligence platforms (TIPs). TIPs are repositories of threats that enable organizations to collect, analyze, integrate external intelligence and even develop their own. The utmost goal of any TIP is to share intelligence that will be embedded into organizational workflows and would serve decision makers (ENISA, 2017).

TIPs are divided into three categories, namely, open source, commercial and community-based. Open-source TIPs like CRITs, CIF, GOSINT, MANTIS and MISP are freely available tools that collect, correlate and exchange information about threats. Commercial TIPs like ThreatStream, EclecticIQ, Soltra Edge and ThreatConnect are paid services that aim to provide registered organizations intelligence about cyber threats, and thus assist them to react to incidents. Community-based TIPs like Alienvault OTX, Facebook Threat Exchange and IBM X-Force Exchange are online platforms that allow subscribers to mutually exchange intelligence about cyber threats (ENISA, 2017).

While CTI solutions are considered very promising approaches for building a strong information security posture (Pomenon Institute, 2015), to the knowledge of the authors, they are rarely used to improve the DFR levels of organizations. The next section presents a novel approach for filling this gap.

3. The threat intelligence informed DFR model

In their previous work (Serketzis *et al.*, 2017), the authors proposed a generic framework for enhancing DFR using threat intelligence. This framework involves the following main steps:

- Use of threat intelligence sharing techniques to collect, store, analyze and correlate IOCs, from various sources, in the form of graphs.
- Collect, correlate, preprocess and store logs from various security devices of an organization.
- Correlation of the two sets, IOCs and logs, and identification of relationships between them, with the aim to create an effective subset of event logs (triage) that an investigator can primarily consult, in the event of an incident.

This paper extends the above framework by introducing a methodology that uses the indicators of compromise (IoCs) selected from diverse sources, formulated and normalized for compatibility reasons and finally, cross-matched against locally collected logs, aiming to discover potential threats. The above loose methodology steps are framed into three newly created modules, namely, "*IoC Collection Module*", "*Audit Log Processing Module*" and "*Threat Identification Module*" that operate independently (Figure 1) while coordinated to present an effective DFR model. The proposed methodology offers automated methods for discovering threats resulting in a time-efficient DFR model.

3.1 IoC type selection methodology

Threat intelligence is often presented in the form of IoCs, and different TIPs incorporate various IoC structures, including characteristics such as IP addresses, URLs and hash values. For example, STIX v2 utilizes 19 objects (MITRE, 2017), each containing several properties, while OpenIoC v1.1 contains 599 IoC properties (Mandiant Corporation, 2013). Although these called IoC components, objects or properties attempt to cover all systems' characteristics, they are rarely all used in practice (Tounsi and Rais, 2018). One of the advantages of this methodology is to discover and propose the most commonly used ones with the highest forensic value, thus improving the efficiency of the forensic discovery process.

The methodology reviewed the IoC characteristics provided by the most well-known TIPs, referred in Table I. This review revealed eight common IoC characteristics. In particular, Table I indicates that IP addresses and hash values are the most commonly exchanged IoCs. Feeds containing URLs and fully qualified domain names (FQDNs) also appear frequently among the data sources. Some sources share emails, filenames, mutexes and registry values. However, the amount of those IoCs is limited. It is worth noting that data sources label IoCs differently. For example, labels "Domain", "Host" and "FQDN" usually represent the same IoCs.

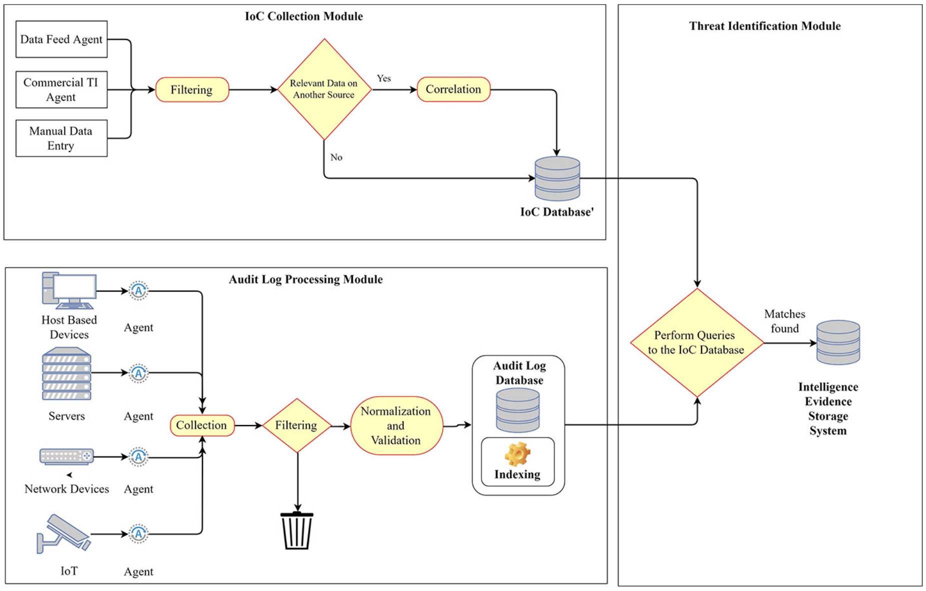


Figure 1.
The threat intelligence informed digital forensic readiness process model

Moreover, a statistical analysis of IoCs within the two most renowned open-access TIPs, Alienvault (Figure 2) and ThreatConnect (Figure 3), confirmed the above-mentioned findings.

3.2 IoC collection module

3.2.1 IoC selection. The purpose and methodology for generating IoCs vary. Some IoCs are produced manually, like the ones security analysts produce after examining malware samples, while others can be downloaded from IoC sources. Although there are many IoC characteristics like the ones presented in Section 3.1, only a few are frequently used in practice (Tounsi and Rais, 2018).

One of the main criteria for selecting IoC sources is the type of threats an organization faces and expects to defend. Apart from that, the nature of organizations, geographic areas they operate and legislative regime they depend on are some of the additional factors that play a significant role in the IoC source selection process. Another criterion is the level of exposure of their assets to those threats. So, it becomes clear that selecting IoC sources is a complex procedure, which must comply with an organization's information security policy.

3.2.2 IoC collection. Ideally, IoCs are obtained directly from sharing platforms using technologies such as STIX and TAXII. However, not all IoC repositories support automated approaches; thus, a hybrid IoC collection process would be feasible. This newly developed hybrid mechanism utilizes custom agents that connect to TIPs and obtain their content through application programming interfaces (APIs). In special cases, it is also possible for end-users to add or elaborate IoCs manually to further optimize the collected information.

3.2.3 IoC filtering. IoC data sources share vast amounts of IoCs, but some of them can challenge the effectiveness of the model. For instance, non-filtering private IP addresses

Table I. Review of characteristics of threat intelligence platforms

No.	Source	Website	Hash	IP address	URL	FQDN	Filename	Email	Mutex	Registry
1	Abuse.ch	https://abuse.ch		×	×	×				
2	AlienVault OTX	https://otx.alienvault.com	×	×	×	×		×	×	
3	Bambenek Source	http://osint.bambenekconsulting.com/feeds		×		×				
4	BotScout	https://botscout.com		×				×		
5	Botvrij	https://botvrij.eu	×	×	×	×				
6	CI Army List	http://cinsscore.com/list/ci-badguys.txt		×						
7	Cryptam	https://www.cryptam.com	×							
8	Cybercrime Tracker	http://cybercrime-tracker.net		×	×					
9	Firehol IP Lists	http://iplists.firehol.org		×						
10	GreenSnow	https://greensnow.co		×						
11	Hybrid Analysis	https://www.hybrid-analysis.com	×	×	×	×	×			
12	Malshare	https://malshare.com	×							
13	MalwareConfig	https://malwareconfig.com	×	×		×			×	
14	MalwareDomainList	www.malwaredomainlist.com		×		×				
15	MalwareDomains	www.malwaredomains.com				×				
16	OpenPhish	https://openphish.com			×					
17	PhishTank	https://www.phishtank.com			×					
18	Sarvam	http://sarvam.ece.ucsb.edu	×							
19	ThreatCrowd	https://www.threatcrowd.org	×	×	×	×				
20	ThreatMiner	https://www.threatminer.org	×	×	×	×	×	×	×	×
21	VirusShare	https://virusshare.com	×							
22	Virussign	www.virusign.com	×							
23	Vxvault	http://vxvault.net	×	×	×					
24	VirScan	www.virscan.org	×				×			
Totals			*13*	*15*	*10*	*10*	*3*	*3*	*3*	*1*

(IANA, 2017) may produce correlation errors. In particular, the experiments revealed that different malware families shared common IoCs, like established network connections to loopback addresses, and thus inaccurately seem to correlate. Moreover, experiments indicated that DNS services like “dyndns.org”, “no-ip.com” and so forth, might generate false positives with regards to malware family correlations. Thus, these IoCs need to be carefully handled during the filtering process. Also, IoC filtering seems to improve the efficiency of the model by excluding IoCs (domains, hostnames, URLs, IP addresses) that relate to the Alexa’s top one million domains list. As such, it becomes clear that filtering for imported IoCs is crucial and is supported by the proposed model.

3.2.4 IoC correlation. IoC correlation refers to the information contained in the adjacency matrix of a connected graph between the different entities, namely, IoCs, TIP sources and malware objects. Every IoC that has passed the filtering phase is imported to the *Local IoC Database*. In parallel, a pre-configured set of TIP sources is queried to reveal any relationships an IoC may have. These relationships among IoCs are explained in the next section. For instance, the authors consider the fan-in of an IoC from TIP sources to be representative of the assurance, quality and reliability of the IoC, whereas the existence of links between IoCs will contribute to the forensic discovery process, as this provides the potential to reveal intrinsic relationships between security incidents, threat actors and families of malware. As an example, Figure 4 shows a minimal graph with two TIPs sharing the same IoC. However, as noted later, both in the *Local IoC Database* and the implementation sections, graphs describing real cases are far more complex, illustrating the importance of both IoC correlation and filtering.

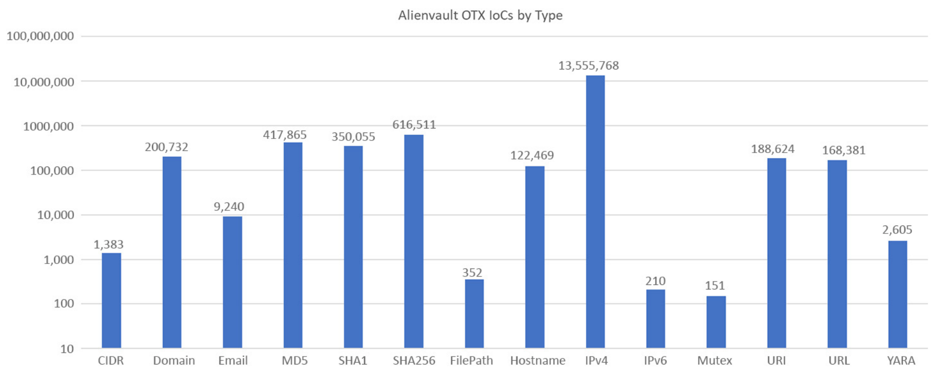


Figure 2. Number and types of IoCs hosted in Alienvault TIP

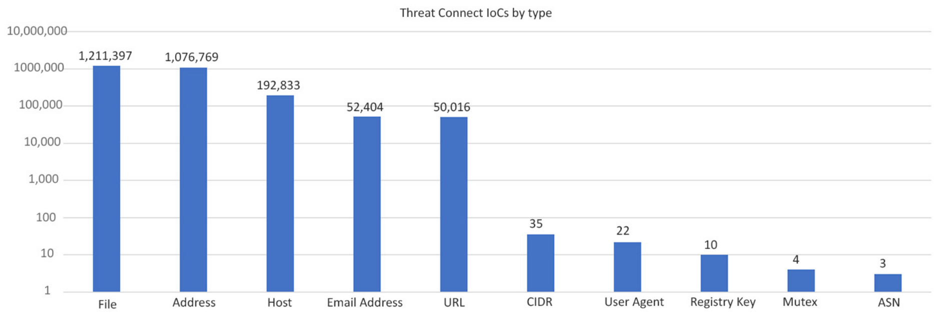


Figure 3. Number and types of IoCs hosted at Threatconnect TIP

3.2.5 IoC database. Although the methodology presented in Section 3.1 identifies the most common types of IoCs, scalability and correlation needs are the primary factors that shape the IoC storage architecture. Relational databases require the definition of an initial schema, which is more restrictive and gives less flexibility when it comes to schema modifications. In turn, graph databases are designed for better performance with highly interdependent data. Such database models uses nodes as data repositories and edges as the inline relationships among them. The ability to pre-materialize those relationships improves the performance of complex queries. Moreover, graph databases are by nature schema-less which overcomes the scaling issues that relational databases face, and are thus considered a viable solution for storing IoCs (Mahmood *et al.*, 2014; Moniruzzaman and Hossain, 2013; Nayak *et al.*, 2013; Neo4j, 2017).

The need for a robust, extensible, interconnected IoC repository platform and the fact that an initial schema is not a preliminary requirement make graph databases an ideal structure for the proposed model. Influenced by the STIX framework, the architecture of the proposed IoC database uses nodes as objects of various types IoCs and edges to describe their relationships. Both objects and their relationships can contain a collection of properties. Also, while it is possible for graph databases to have bi-directional relationships, they are not efficient (Neo4j, 2017), and not a requirement. Therefore, this model only supports directed edges. The proposed IoC database schema is presented in Figure 5 along with and described in more detail in the following subsections.

3.2.5.1 File object. File hashes as IoCs are prevalent in malware analysis for labeling and indexing malware samples. Calculated hash values are properties of the *file object* for the unique identification of malware related files. While there are many algorithms for calculating hash values, MD5, SHA1 and SHA256 are the ones TIPs most commonly use, and thus become properties of the *file object*. The complete *file object* description of properties is shown in Figure 5.

Typical static malware analysis can reveal that *file objects* may relate with IP addresses, URLs, FQDNs and emails (Modi and Doupé, 2017); however, in-depth analysis can unveil more relationships like mutexes and files that are created, opened, deleted or otherwise handled (Figure 5). Moreover, *file objects* can point to *malware* counterparts, providing analysts with additional information about malicious files themselves.

3.2.5.2 IP address object. As mentioned earlier, an *IP address object* is a prominent type of IoC. IP addresses are essential elements of the layer three network connectivity analysis, and their presence in cyber investigations can potentially reveal valuable information about the tools, tactics and procedures (TTP) malware attackers use. For instance, most sophisticated attacks use command and control (C2) servers for administering malware instances. Identifying IP addresses of C2 servers are significant leads allowing cyber analysts to trace attackers. Figure 5 depicts the structure of the *IP address* object and the way it relates with other objects.

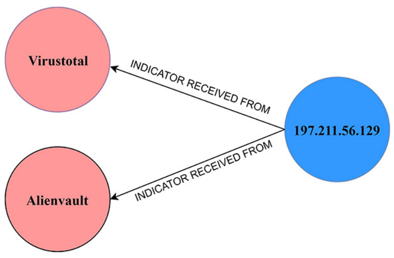


Figure 4.
Relationships between IP address and DataSource objects

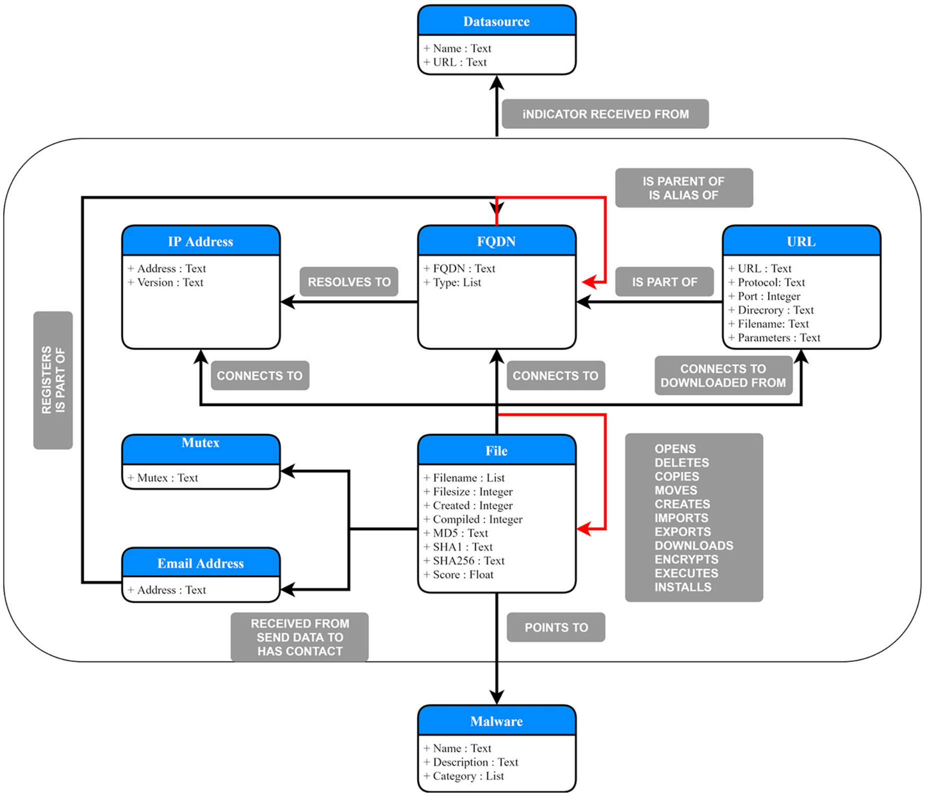


Figure 5.
The local IoC database schema

3.2.5.3 FQDN object. An *FQDN* is the complete domain name for a specific computer, or host, on the internet. It consists of two parts, the "hostname" and the "domain name" (Indiana University, 2017). As shown in Section 3.1, both FQDNs and their associated domain names are frequently used as IoCs, as they describe internet nodes, so both FQDN and domain IoCs can be hosted under the same object as they are similar.

FQDN and *IP address objects* can establish direct relationships that are marked with the label "RESOLVES TO". As shown in the diagram (Figure 5), relationships can also exist between two *FQDN* objects, in cases of child–parent (FQDN–domain) nodes or domain aliases.

3.2.5.4 URL object. In contrast to *FQDN* objects that can abstractly describe the contents of a network source, universal resource locators (URLs) directly point to specific web addresses. Such information is sometimes valuable for the investigation process because it provides the forensic examiner with a holistic view of an attacker's originating attack path. Under certain circumstances, it can also provide the opportunity to fully replicate an attack up to the application layer in a safe environment, thus supporting a better understanding on how it operates and to examine the extent of the attack at the specific environment (i.e. malware forensics).

Text containing the full URL path is the main attribute of the *URL objects* of this model. For efficiency reasons, URL is further decomposed into five attributes (protocol, port, directory, filename, parameters) that become properties of the *URL object* itself. Both the properties and relationships of this object are presented in Figure 5.

3.2.5.5 Email address object. One of the primary targets attackers may have is the establishment of secure channels giving them to control over the infected systems and exfiltrate data. For this reason, one of the steps that precede an attack is the acquisition of one or more remote systems that will be used as a stepping stone to break into the targeted systems. Stepping stone attacks are commonly used as attribution evasion techniques and pose significant challenges to the forensic analyst.

At the same time, hardcoded IP addresses are not preferred by attackers who often tend to use FQDNs that are more dynamic and easily configurable during the preparation of an attack (e.g. during the weaponization phase).

Using FQDN usually presumes a prior registration of a domain name, which in turn requires a valid email address. While registering a free email address is a relatively straightforward process, such information is often crucial for investigations, as it can be used to reveal malicious actors. Also, email addresses can be indicators of ownership of FQDNs (e.g. domain administrators of FQDN "example.com" can only create emails like info@example.com). Figure 5 shows how the *email object* fits within the *local IoC database.*

3.2.5.6 Mutex object. Mutex (mutual exclusive) is a method commonly used to avoid concurrent access to a specific resource. Malware developers usually use mutexes to deter variants of a particular malware being installed on the same system which will be detrimental to the malware itself and system and will potentially allow its detection. At this point, malware instances try to identify whether a mutex exists on the compromised system when they are first executed. The presence of that mutex on the system reveals that the malware is already installed (Blasco, 2009).

Forensic analysts also apply the same techniques to discover malicious software running on a system. In this case, a list of previously known mutexes is used in the form of IoCs. It thus becomes clear that mutexes are essential elements for investigations, and thus, the proposed model includes them as additional objects.

Mutexes do not have a common format. Rather, they are text values threat actors define to evaluate the existence of multiple malware instances. Thus, the "Mutex" key is the only property the *mutex object* contains. Mutexes appear when files execute, and consequently relate to them. However, the relationships between these two objects have already been described in the *File object* section.

3.2.5.7 DataSource object. The *DataSource object* is a supplementary structure in the IoC database. This object does not store IoCs; rather, it keeps information about the origin of every IoC stored in the IoC database. Every object described above relates to the *Datasource object*, and this is why it is presented in Figure 5 as an external entity to the IoC structure (objects within the rounded retackle).

3.2.5.8 Malware object. Similar to *DataSource*, the *malware object* is the element that describes incidents that possibly challenged or bypassed an organization's security measures. The properties it contains can provide abstract information about malware instances, including malware name, description and the category it belongs. The relationships of the *malware object* have already been described in the *File object* section.

3.3 The audit log processing module

The diverse logging formats and the ever-increasing volume of audit data modern information systems produce pose significant difficulties in their efficient management (Chuvakin *et al.*, 2012; Mahmood and Afzal, 2013).

The *audit log processing module* collects and stores data from various sources in a centralized manner and transforms them accordingly, to easily be queried against the IoC database. Detailed analysis of the processes of the *audit log processing module* follows.

3.3.1 Audit log collection. The audit log collection process contains a set of pre-configured agents that gather audit logs produced from security systems and devices. These agents can operate in active and passive mode depending on the limitations of the auditing source. For instance, in cases where collection agents cannot be installed on devices due to computational limitations, listeners deployed on more powerful nodes for event data capturing would be a more feasible approach (passive mode).

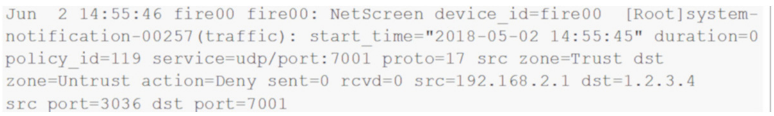
The proposed model does not aim to substitute security information and event management (SIEM) systems that operate in real time or near real time (Murphy, 2017) but intends to augment their forensic capabilities.

3.3.2 Audit log filtering. Audit logs often carry additional and substantial – in volume – information that is not necessarily useful for a security analysis process. For instance, debug-level logging produces detailed information on a system's operations, which is sometimes required for troubleshooting, but does not add value to a forensic investigation. The *log filtering* process enables the collection of suitable data while disregarding the non-relevant characteristics. Considering that there may be incompatibilities among audit logs from different devices, the *log filtering* process utilizes a set of predefined patterns and filters out irrelevant information.

3.3.3 Audit log normalization and validation. Normalization is the process of adapting the raw data collected from various sources into a standard format (Chuvakin *et al.*, 2012). To this point, the data that pass the *log filtering* process are parsed and divided into components. For comparison reasons, in the proposed model, the main log characteristics are identical to the IoC object's properties.

Figure 7 presents an example normalization process for an instance of an audit log produced from a firewall (Figure 6). Note that the IP addresses and timestamps that are among the elements of interest are described in a JSON representation in Figure 7. Also, notice that the normalized JSON structure does not contain the private IP address "192.168.2.1" because the log filtering process filtered it.

Homogeneity of data is essential for efficient analysis. Thus, the log data to be compared against the collected IoCs must be accurate. *Data validation* processes ensure the accuracy of data and guarantee that similar elements have the same data format (i.e. same time format).



```
Jun  2 14:55:46 fire00 fire00: NetScreen device_id=fire00  [Root]system-
notification-00257(traffic): start_time="2018-05-02 14:55:45" duration=0
policy_id=119 service=udp/port:7001 proto=17 src zone=Trust dst
zone=Untrust action=Deny sent=0 rcvd=0 src=192.168.2.1 dst=1.2.3.4
src port=3036 dst port=7001
```

Figure 6.
Sample raw audit log data produced by Netscreen Firewall

```
{
   "ID":"000000001",
   "SourceID":"fire00",
   "IoC":"IPv4",
   "IoC_Type":"1.2.3.4",
   "TimeStamp":"2018-05-02T14:55:46+00:00"
   "Rawdata":"……….."
}
```

Figure 7.

3.3.4 Audit log database and indexing. Unsurprisingly, modern systems produce a wealth of logs – in terms of – variety and volume. As such, log analysis is increasingly becoming time-consuming (Quick and Choo, 2016), requiring a more methodological and systematic approach in developing log analysis processes. This involves approaches and functions such as indexing to improve data manipulation speed (Kim *et al.*, 2011). A particular indexing method which has been shown to be suitable for offering effective analysis of audit log data is that of inverse indexing.

With regards to the underlying database storage paradigm, the proposed methodology favors NoSQL databases as the driving storage technology because one of the main benefits they offer is their schema-less nature, which allows the inclusion of previously unseen data fields into the database schema at execution time (Moniruzzaman and Hossain, 2013).

From a practical implementation perspective in this paper, Elasticsearch was selected as the best candidate tool as it provides an efficient indexing service allowing the processing of complex search queries (Kalyani and Mehta, 2017).

3.4 Threat identification module and intelligence evidence storage system

Last but not least, the *threat identification module* (TIM) analyzes the entries originating from both the *local IoC* and *audit log databases* and checks whether there are matches between them. The existence of a match indicates the presence of a relevant threat. In this case, the TIM keeps a backup of the relevant audit log records in a separate database view called the *intelligence evidence storage system* (IESS). This particular database can be viewed as the result of the triage activity allowing the security and forensics analysis processes to obtain an initial and prioritized overview of the incidents that had the potential to reach the attack surface of the underlying systems successfully. This is particularly useful as it also considers "near misses", which are attack events and incident that have taken place but were not fully successful, and therefore did not raise any incident detection alerts.

The TIM takes advantage of the relationships kept in the IoC database and facilitates the comparison process between the IoC and *audit log database*, succeeding to uncover hidden patterns of threats. Figure 8 presents the algorithm the TIM uses.

4. Implementation

The proposed framework was tested for its applicability in an operational environment with a prototype topology that attempted to simulate a simple office environment (SOHO). The environment contained three Windows 7 virtual machines (VMs), an OSSEC 2.9.3 (OSSEC, 2018) virtual appliance which was configured to act both as gateway and IDS and a virtual appliance running the Ubuntu 16.04 OS hosting the *threat intelligence informed DFR process model*. The OSSEC virtual appliance produced NetFlow, and IDS alerts of the network traffic passed through its interfaces. In addition, OSSEC host agents were installed on the Windows machines for collecting and exporting host audit logs. Both host and network audit data forwarded to the *threat intelligence informed DFR process model* for further handling. More information about the implementation is given below.

4.1 IoC collection module

The prototype agent for the collection of IoCs from various sources was developed with Python programming language. This agent first obtained a list of hash values from a set of pre-configured data sources (Alienvault's OTX and MalwareConfig) and verified their format. Then, it searched several TIPs like Virustotal for establishing

```
$\mathcal{T} \longleftarrow IoC\ Type$
 $\mathcal{D}_1 \longleftarrow Local\ IoC\ Database$
 $\mathcal{D}_2 \longleftarrow Audit\ Log\ Database$
 $O \longleftarrow Intelligent\ Evidence\ Storage\ System$

 $For\ i\ in\ \mathcal{D}_2\ where\ t_1, t_1 \in T$
    $For\ j\ in\ \mathcal{D}_1\ where\ t_1, t_1 \in T$
       $if\ i == j$
          $O[x] = j, x \in \mathbb{N}$
          $y = x$
          $x = x + 1$
          $For\ k\ in\ \mathcal{D}_1\ where\ t_2,\ t_1 \neq t_2$
             $For\ l\ in\ \mathcal{D}_2\ where\ t_2,\ t_1 \neq t_2$
                $if\ k == l$
                   $O[x] = k$
                   $correlate(O[x], O[y])$
                   $x = x + 1$
                $End\ if$
             $Next\ l$
          $Next\ k$
       $End\ if$
    $Next\ j$
 $Next\ i$
```

Figure 8. The algorithm of the threat Identification module

relationships with relevant IoCs. This process produced a set of inter-related IoC objects, which were stored in a Neo4j graph database that acts as the *local IoC database*. Figure 9 depicts a simple instance of the IoC database. This picture shows the way *file object* connects to other IoCs like IP addresses, FQDNs, mutexes, malware instances, etc.

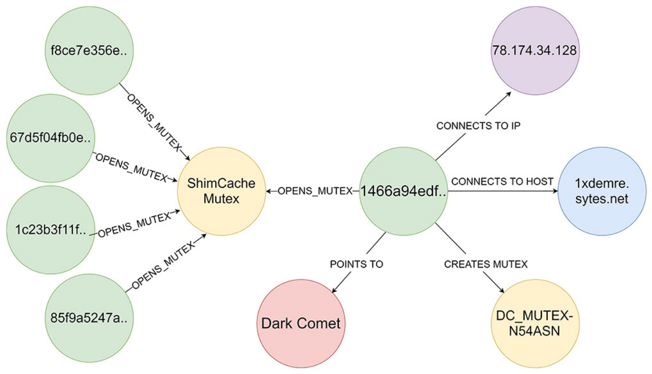


Figure 9. Relationships among IoCs in the local IoC database

4.2 Audit log processing module

The *audit log processing module* entails several open-source tools. Audit log collection performed at the network level with the Bro Security Network Monitor Tool. At the host level, the OSSEC Host Intrusion Detection deployed agents on the Windows VMs and captured the changes made at the file system level.

Collected data have been forwarded to the Logstash application by Elastic, which in turn parsed, filtered and normalized every record it received, and subsequently transformed raw entries into predefined fields. These fields were similar to those of the *local IoC database*. For integrity reasons, Logstash kept raw data at their original state.

At the storage level, Elasticsearch was selected as the most appropriate solution because of its interoperability with the Logstash application and its efficient inverse indexing service, which is powered by the Lucene search engine. Every record pushed into Elasticsearch is automatically tokenized and indexed, and is thus capable of instant retrieving.

4.3 Threat identification module

The TIM was implemented with Python programming language. Its effectiveness relied on the rapidness of accessing and comparing inverse indexed tokens against the local stored IoCs. To avoid duplication, IESS was implemented within the *audit log database* by introducing a set of special fields that acted as binary flags. The result of the TIP algorithm specifies the *audit log database* entries where these flags are set, by updating the status of the relevant NoSQL documents. Finally, a Web-based frontend interface was developed to visualize and represent as graphs the contents of the IESS.

5. Results and evaluation

In addition to the implementation phase discussed earlier aiming to verify the applicability of the proposed framework, the evaluation part aims to measure its effectiveness with appropriate data. To this extent, the authors chose to simulate the operation of an IDS with captured data that are part of the CTU-13 data set from the Stratosphere Lab (Stratosphere Lab, 2015). It is worth highlighting that a recent research performed by Małowidzki *et al.* reported that the CTU-13 is the most valuable data set to date (Małowidzki *et al.*, 2017).

So, the first experiment is based on CTU13 – Mixed 1 data set that was produced from an attack scenario simulating the operation of the Bubble Dock malware. This malware variation is known of generating various intrusive third-party ads while tracking user activity. The Mixed-1 data set contains 52,374 rows with bidirectional network flows of both normal and malicious network traffic. Table II depicts the structure of the data set.

This data set contains 14 fields; however, considering that the proposed methodology uses text comparison techniques to identify similarities between common types of IoCs stored in the "*Local IoC*" and "*Audit Log*" databases, it becomes clear that IP addresses are the only elements of this particular data set that the TIM can use. The contents of the above-mentioned data set have been imported to the Elasticsearch indexing service, which have been acting as the *audit log database*, thus producing a set of 2,659 unique IP addresses.

On the other side, the *local IoC database* has been populated with 1,500 random MD5 hash values following the procedure described in Section 4.1, which in turn resulted in the establishment of a graph containing 35,000 IoCs in approximate, of which 3,649 were IP addresses.

The contents of both databases compared through the TIM indicating that 13 IP addresses were in common, and it thus updated the contents of the IESS accordingly. By querying the *local IoC database*, the TIM found that 19 malware samples were related with

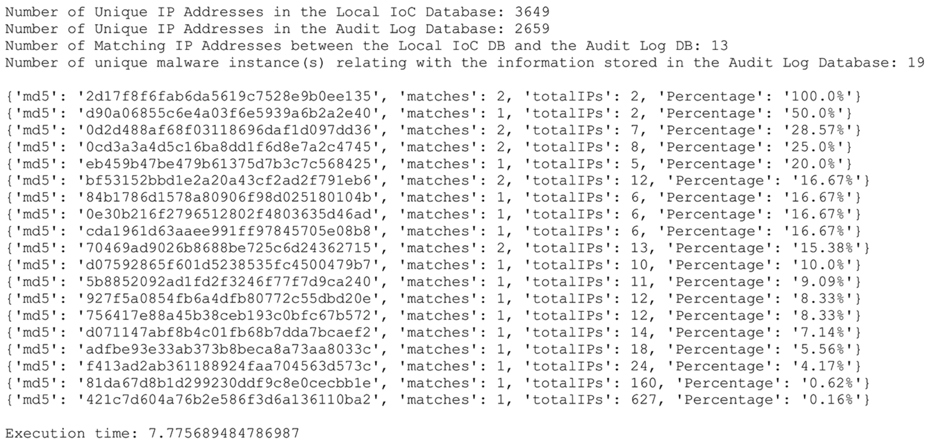
Field	Description
StartTime	Record start time
Dur	Duration of a flow
Proto	Transaction protocol
SrcAddr	Source IP address
Sport	Source port
Dir	Direction of transaction
DstAddr	Destination IP address
Dport	Destination port
State	Transaction state
STos	Source TOS byte value
dTos	Destination TOS byte value
TotPkts	Total transaction packet count
TotBytes	Total transaction bytes
SrcBytes	Bytes sent from source to destination
SrcUdata	Data sent from source to destination
dstUdata	Data sent from destination to host

Table II. Structure of the dataset used for the experiment

these IP addresses (Figure 10). Despite the increased number of these reported malware samples, the scoring algorithm of the TIM designated the file having an MD5 value of "2d17f8f6fab6da5619c7528e9b0ee135" as the one that relates most with the data set used in the experiment. The scoring algorithm used in this experiment calculated the percentage of the related IP addresses of a *file object* that are present in the *audit log database*. The evaluation of the scenario confirmed the accuracy of the results of the experiment.

The whole process of cross-checking 3,649 IP entries of the *local IoC database* against 52,374 records contained in the *audit log database*, obtaining related objects, running the scoring algorithm and presenting the results lasted approximately 8 s (Figure 10), which is considered an acceptable timeframe.

Further analysis, which performed by applying more complex queries against the *local IoC database*, uncovered additional patterns of possible malicious activity but produced more information that needs to be evaluated by forensic analysts. Figure 11 presents additional relationships among the malware instances mentioned above (purple nodes),



```
Number of Unique IP Addresses in the Local IoC Database: 3649
Number of Unique IP Addresses in the Audit Log Database: 2659
Number of Matching IP Addresses between the Local IoC DB and the Audit Log DB: 13
Number of unique malware instance(s) relating with the information stored in the Audit Log Database: 19

{'md5': '2d17f8f6fab6da5619c7528e9b0ee135', 'matches': 2, 'totalIPs': 2, 'Percentage': '100.0%'}
{'md5': 'd90a06855c6e4a03f6e5939a6b2a2e40', 'matches': 1, 'totalIPs': 2, 'Percentage': '50.0%'}
{'md5': '0d2d488af68f03118696daf1d097dd36', 'matches': 2, 'totalIPs': 7, 'Percentage': '28.57%'}
{'md5': '0cd3a3a4d5c16ba8dd1f6d8e7a2c4745', 'matches': 2, 'totalIPs': 8, 'Percentage': '25.0%'}
{'md5': 'eb459b47be479b61375d7b3c7c568425', 'matches': 1, 'totalIPs': 5, 'Percentage': '20.0%'}
{'md5': 'bf53152bbd1e2a20a43cf2ad2f791eb6', 'matches': 2, 'totalIPs': 12, 'Percentage': '16.67%'}
{'md5': '84b1786d1578a80906f98d025180104b', 'matches': 1, 'totalIPs': 6, 'Percentage': '16.67%'}
{'md5': '0e30b216f2796512802f4803635d46ad', 'matches': 1, 'totalIPs': 6, 'Percentage': '16.67%'}
{'md5': 'cda1961d63aaee991ff97845705e08b8', 'matches': 1, 'totalIPs': 6, 'Percentage': '16.67%'}
{'md5': '70469ad9026b8688be725c6d24362715', 'matches': 2, 'totalIPs': 13, 'Percentage': '15.38%'}
{'md5': 'd07592865f601d5238535fc4500479b7', 'matches': 1, 'totalIPs': 10, 'Percentage': '10.0%'}
{'md5': '5b8852092ad1fd2f3246f77f7d9ca240', 'matches': 1, 'totalIPs': 11, 'Percentage': '9.09%'}
{'md5': '927f5a0854fb6a4dfb80772c55dbd20e', 'matches': 1, 'totalIPs': 12, 'Percentage': '8.33%'}
{'md5': '756417e88a45b38ceb193c0bfc67b572', 'matches': 1, 'totalIPs': 12, 'Percentage': '8.33%'}
{'md5': 'd071147abf8b4c01fb68b7dda7bcaef2', 'matches': 1, 'totalIPs': 14, 'Percentage': '7.14%'}
{'md5': 'adfbe93e33ab373b8beca8a73aa8033c', 'matches': 1, 'totalIPs': 18, 'Percentage': '5.56%'}
{'md5': 'f413ad2ab361188924faa704563d573c', 'matches': 1, 'totalIPs': 24, 'Percentage': '4.17%'}
{'md5': '81da67d8b1d299230ddf9c8e0cecbb1e', 'matches': 1, 'totalIPs': 160, 'Percentage': '0.62%'}
{'md5': '421c7d604a76b2e586f3d6a136110ba2', 'matches': 1, 'totalIPs': 627, 'Percentage': '0.16%'}

Execution time: 7.775689484786987
```

Figure 10. Results of the threat identification module on CTU-13 – Mixed1 dataset

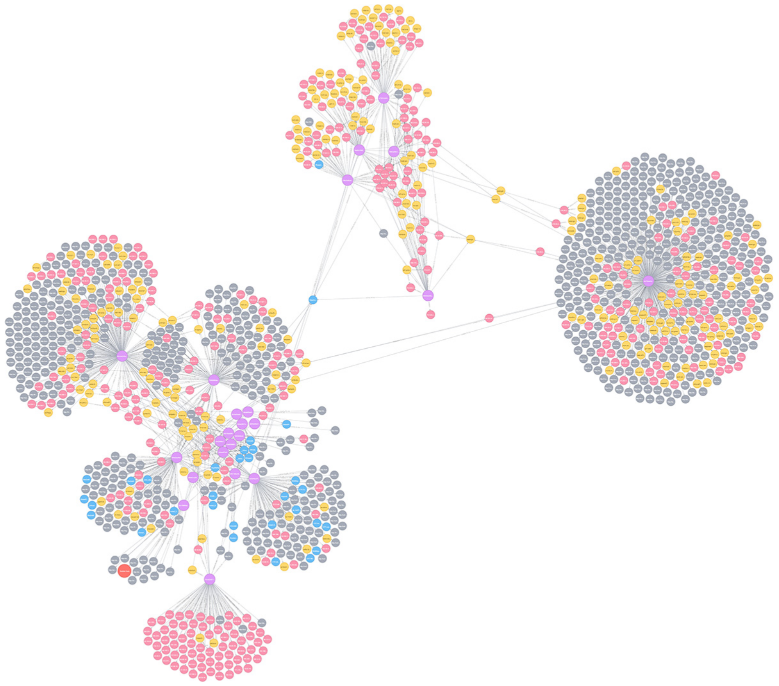


Figure 11.
Graph produced after more in-depth analysis

meaning that these interconnections can be based on other objects like URLs (gray nodes), FQDNs (yellow nodes) and mutexes (blue nodes). This picture also indicates the increased volume of data that has been collected and elaborated during the pilot implementation, given that MD5 hash values was the only source of information.

The second experiment utilized the twenty-fifth data set of the CTU scenarios. This data set contained 241,243 rows of firewall event logs collected over a period of four months that relate to Zbot Trojan horse family. Zbot malware samples are known for their attempts to steal confidential information from the compromised computers (Symantec, 2010).

Both this data set and the contents of the *local IoC database* fed the *TIM*, which managed to cross-check 4,157 records of the *local IoC database* against 241,243 event logs and provide relevant results in approximately 30 s. Moreover, the scoring algorithm reported that four malware samples were the most probable instances that produced the traffic the data set contains. One of those instances matched the one described in the scenario, meaning that the proposed framework has accurately reasoned how the incident emerged. It is also important to underline that the rest of the samples that have been assigned high degrees of confidence belong to the same malware family.

The results of the experiments mentioned above are very promising, as they indicate that the proposed framework can substantially narrow the information digital forensic analysts need to examine for uncovering patterns of malicious activity. Additionally, the transformation of data to intelligence can provide digital forensics specialists with important directions about existing forensic artifacts. For instance, the contents of the IESS can be exported in a list of IP addresses, URLs, FQDNs, filenames and file hashes, which in turn can be imported into forensic investigation tools for accurately and efficiently identifying malicious activities.

6. Conclusion

There are two key factors to evaluate for implementing a successful DFR procedure; the time needed to identify a potential incident and the extreme volume of data a security analyst must examine. In our days, sophisticated malware and APTs are designed to stay uncovered as long as possible for their exploiters to achieve their targets. The timely uncovering of such a threat is crucial, as the shortest it stays uncovered, the less damage the target system suffers. Combined with the huge amount of data collected from security devices, and logs generated from local information systems, an analyst needs to work with, compose a demanding environment with high risks. The methodology presented here succeeds to overcome these barriers by automating the malware uncovering procedure, and thus reducing the time needed to take actions against a potential threat while handling all locally stored evidence and taking advantage of all TI accumulating in across various TIPs.

References

Blasco, J. (2009), "Malware: Exploring mutex objects", *Alienvault*, available at: www.alienvault.com/blogs/labs-research/malware-exploring-mutex-objects (accessed 11 May 2018).

Chuvakin, A., Schmidt, K. and Phillips, C. (2012), *Logging and Log Management*, 1st ed., Syngress.

Danyliw, L. Meijer, J. and Demchenko, Y. (2007), "RFC 5070. The incident object description exchange format", available at: https://tools.ietf.org/html/rfc5070

ENISA (2017), "Exploring the opportunities and limitations of current threat intelligence platforms", December, pp. 1-42.

European Union Agency For Network And Information Security (ENISA) (2017), *ENISA Threat Landscape Report 2016*, Athens, available at: https://doi.org/10.2824/92184.

Garfinkel, S.L. (2010), "Digital forensics research: the next 10 years", *Digital Investigation*, Vol. 7, pp. S64-S73.

Grobler, C.P. and Louwrens, C.P. (2007), "Digital forensic readiness as a component of information security best practice", *International Information Security Conference*, Vol. 232, pp. 13-24.

Grobler, C.P., Louwrens, C.P. and Von Solms, S.H. (2010), "A framework to guide the implementation of proactive digital forensics in organizations", *ARES 2010 – 5th International Conference on Availability, Reliability, and Security*, pp. 677-682.

IANA (2017), "IANA IPv4 Special-Purpose address registry", available at: www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml (accessed 7 May 2018).

Indiana University (2017), "About fully qualified domain names (FQDNs)", available at: https://kb.iu.edu/d/aiuv (accessed 8 May 2018).

Kalyani, D. and Mehta, D.D. (2017), "Paper on searching and indexing using elasticsearch", *International Journal of Engineering and Computer Science*, Vol. 6 No. 6, pp. 21824-21829.

Kebande, V. and Venter, H.S. (2015), "A functional architecture for cloud forensic readiness large-scale potential digital evidence analysis", *Proceedings of the 14th European Conference on Cyber Warfare and Security 2015: ECCWS 2015*, pp. 373-382.

Kebande, V., Ntsamo, H.S. and Venter, H.S.S. (2016), "Towards a prototype for achieving digital forensic readiness in the cloud using a distributed NMB solution", *European Conference on Cyber Warfare and Security*, pp. 369-379.

Kim, J., Abbasi, H., Chacon, L., Docan, C., Klasky, S., Liu, Q., Podhorszki, N., *et al.* (2011), "Parallel in situ indexing for data-intensive computing", *2011 IEEE Symposium on Large Data Analysis and Visualization, IEEE*, pp. 65-72.

Mahmood, K., Risch, T. and Zhu, M. (2014), "Utilizing a NoSQL data store for scalable log analysis", *Proceedings of the 19th International Database Engineering &#38; Applications Symposium*, pp. 49-55.

Mahmood, T. and Afzal, U. (2013), "Security analytics: big data analytics for cybersecurity: a review of trends, techniques and tools", *2013 2nd National Conference on Information Assurance (NCIA), IEEE*, pp. 129-134.

Małowidzki, M., Berezi, P. and Mazur, M. (2017), "Network intrusion detection: half a kingdom for a good dataset", *ECCWS 2017 16th European Conference on Cyber Warfare and Security*, pp. 1-6.

Mandiant Corporation (2013), "OpenIOC", available at: www.openioc.org

MITRE (2017), "STIX: a structured language for cyber threat intelligence", available at: https://oasis-open.github.io/cti-documentation/ (accessed 5 March 2017).

Modi, A. and Doupé, A. (2017), "Automated Confidence Score Measurement of Threat Indicators".

Moniruzzaman, A.B.M. and Hossain, S.A. (2013), "NoSQL database: new era of databases for big data analytics-classification, characteristics and comparison", *ArXiv Preprint ArXiv:1307.0191*, Vol. 6 No. 4, pp. 1-14.

Murphy, J. (2017), *Security Information and Event Management (SIEM) Implementation*, CreateSpace Independent Publishing Platform.

Nayak, A., Poriya, A. and Poojary, D. (2013), "Type of NoSQL databases and its comparison with relational databases", *International Journal of Applied Information Systems*, Vol. 5 No. 4, pp. 16-19.

Neo4j (2017), "From relational to Neo4j", available at: https://neo4j.com/developer/graph-db-vs-rdbms/#_from_relational_to_graph_databases

Ngobeni, S., Venter, H. and Burke, I. (2012), "The modelling of a digital forensic readiness approach for wireless local area networks", *Journal of Universal Computer Science*, Vol. 18 No. 12, pp. 1721-1740.

OSSEC (2018), "Open source HIDS SECurity", available at: www.ossec.net/ (accessed 10 May 2018).

Palmer, G. (2001), "DTR - T001-01 technical report. A road map for digital forensic research", Utica, New York, NY, available at: www.dfrws.org/2001/dfrws-rm-final.pdf

Pangalos, G. and Katos, V. (2010), "Information assurance and forensic readiness", *Next Generation Society. Technological and Legal*, pp. 181-188.

Pomenon Institute (2015), "The importance of cyber threat intelligence to a strong security posture", available at: www.webroot.com/shared/pdf/CyberThreatIntelligenceReport2015.pdf%5Cnhttps://www.webroot.com/shared/pdf/CyberThreatIntelligenceReport2015.pdf

Quick, D. and Choo, K.R. (2016), "Big forensic data management in heterogeneous distributed systems: quick analysis of multimedia forensic data", *Software: Practice and Experience*, Vol. 39 No. 7, pp. 701-736.

Reddy, K. and Venter, H.S. (2013), "The architecture of a digital forensic readiness management system", *Computers and Security*, Elsevier Ltd, Vol. 32, pp. 73-89.

Rowlingson, R. (2004), "A ten step process for forensic readiness", *International Journal of Digital Evidence*, Vol. 2 No. 3, pp. 1-28.

Sauerwein, C., Sillaber, C., Mussmann, A. and Breu, R. (2017), "Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives", pp. 837-851.

Serketzis, N., Katos, V., Ilioudis, C., Baltatzis, D. and Pangalos, G.J. (2017), "A Socio-Technical perspective on threat intelligence informed digital forensic readiness", *International Journal of Systems and Society*, Vol. 4 No. 2, pp. 57-68.

Shields, C., Frieder, O. and Maloof, M. (2011), "A system for the proactive, continuous, and efficient collection of digital forensic evidence", *Digital Investigation*, Vol. 8, pp. 3-13. Elsevier Ltd.

Spyridopoulos, T. and Katos, V. (2011), "Requirements for a forensically ready cloud storage service", *International Journal of Digital Crime and Forensics*, Vol. 3 No. 3, pp. 19-36.

Stratosphere Lab (2015), "Stratosphere datasets", available at: www.stratosphereips.org/ (accessed 10 August 2018).

Symantec (2010), "Trojan.Zbot", available at: www.symantec.com/security-center/writeup/2010-011016-3514-99 (accessed 20 October 2018).

Tan, J. (2001), "Forensic Readiness", available at: https://isis.poly.edu/kulesh/forensics/forensic_readiness.pdf, Cambridge, MA 02139 USA.

Tounsi, W. and Rais, H. (2018), "A survey on technical threat intelligence in the age of sophisticated cyber attacks", *Computers and Security, Elsevier Ltd*, Vol. 72 September pp. 212-233.

Verizon (2018), "2018 Data breach investigations report", available at: www.verizonenterprise.com/resources/reports/rp_DBIR_2018_Report_execsummary_en_xg.pdf

Virvilis, N., Serrano, O. and Dandurand, L. (2014), "Big data analytics for sophisticated attack detection", *ISACA Journal*, Vol. 3.

Wigmore, I. (2015), "What is threat intelligence", available at: https://whatis.techtarget.com/definition/threat-intelligence-cyber-threat-intelligence (accessed 7 July 2018).

Corresponding author
Nikolaos Serketzis can be contacted at: nserketzis@auth.gr