Basic Forensic Procedures for Cyber Crime Investigation in Smart Grid Networks

Igor Kotsiuba
G.E. Pukhov Institute for Modeling in Energy Engineering, National Academy of Sciences of Ukraine
Kyiv, Ukraine
i.kotsiuba@gmail.com

Inna Skarga-Bandurova
School of Engineering, Computing and Mathematics
Oxford Brookes University
Oxford, United Kingdom
iskarga-bandurova@brookes.ac.uk
ORCID ID: 0000-0003-3458-8730

Alkiviadis Giannakoulias
School of Electrical and Computing Engineering
National Technical University of Athens
Athens, Greece
alkiviadis.giannakoulias@eurodyn.com

Oksana Bulda
Ghent University
Ghent, Belgium
oksana.bulda@ugent.be

***Abstract*—The paper outlines some aspects of developing a cyber-forensic framework for Smart Grid cyber-crime investigations. In this research, we examine a key forensic instrument in reconstructing events, the timeline, followed by correlation of data from different sources. Then, we deal with the tasks of collecting and storing the monitored data. The paper also covers some aspects of the legal ramifications from collecting this data and touches on the preconditions that must be met to enable network forensics. Then we present the logging architecture, based on the recommendations of the UK National Cyber Security Center. The final part presents the methodological framework that is the result of applying the OSCAR methodology and relevant open source tools in order to ensure that necessary forensic information can be collected, stored and used as legal evidence in court.**

Keywords—forensic, cyber-crime, investigation, incident network, logging architecture, chain of custody, OSCAR

I. INTRODUCTION

Security Information and Event Management (SIEM) systems are an emerging technology for monitoring and controlling real-time critical infrastructure. SIEM [1] is a combination of Security Information Management (SIM) and Security Event Management (SEM) systems designed to analyze the security information provided by monitoring infrastructure in order to identify security breaches [2], [3].

In [4], M. Nicolett et al. provided an overview of the widespread SIEM technologies, focusing on their strengths and weaknesses. R. Leszczyna et al. [5] evaluated three open source SIEM systems for Smart Grid. In particular, the platforms explored are AlienVault OSSIM [6], Cyberoam iView [7], and Prelude SIEM [8]. According to the authors' evaluation criteria, AlienVault OSSIM and Prelude SIEM perform best. However, existing SIEM systems have three significant limitations in the energy sector. To start with, their functionality focuses only on information and communication technologies (ICTs) without the ability to manage other infrastructures, such as industrial systems. Secondly, even if they can work in the industrial sector, they usually use the appropriate correlation rules for several industrial protocols. Finally, the huge amounts of data generated by several power grid components, cannot be effectively processed. According to MC. Di Sanora et al. [9] SIEM systems lack several features for Critical Infrastructure Protection. In particular, three limits have been identified:

- SIEM systems enable to define security policies, but they do not provide the means for resolving a policy conflict between policies that are both activated at the same time but allow different actions. Several mechanisms and conflict resolution strategies have been developed, such as those presented in [17], [18], [19], [20], [21] and [22]
- Monitoring of critical infrastructure is performed by deploying communication networks that enable information to be exchanged between monitoring facilities and management systems. To avoid specific links from external networks to internal networks, security policies create severe restrictions on data flows. For example, the operation of re-recording the firmware of the sensor can only be done from certain hosts on the authorized LAN, where there are privileged accounts and restricted access to domain experts. SIEM systems do not have a methodology for identifying and controlling all possible data transmission paths that exist in controlled infrastructures.
- SIEMs generate alarms when attack signatures are detected. Today, only a few commercial SIEMs provide these requirements with modules that sign alarms using cryptographic algorithms such as RSA or DES. Typically, the signature module is not designed to attack the module that generates the signed records.

It is certain that SIEM systems have great potential to ensure the security of smart grids, but need further refinement to overcome the above limitations. This is possible by attaching new components and methodologies, adding new metrics such as voltage, current, phase, power and frequency, thereby adapting them to current smart grid security tasks. Thus, an important problem of security analysis is the disclosure of uncertainties caused by the variety of goals, properties and features of the investigated Smart Grid infrastructure components. In this contexts, an essential part of analyzing a system's vulnerability is a network modeling. During the simulation process, several hundred thousand iterations must be done to identify strong attack combinations. In early studies, a cascading failure simulator DCSimSep [10] was used to measure system data, proposed and described in [11], [12]. DCSimSep enables to find combinations of the n-k type, $2 \leq k \leq 5$, which cause large blackouts in large-scale power systems. With the help of this simulator, which is used in most well-known works, the simulation should be performed each time with a manual input of the line, the actions of attackers are aimed at compromising or disabling it. This approach, when evaluating large infrastructure projects, takes a lot of time. In addition, Smart Grid system is constantly evolving due to the involvement of private networks. Also, Smart Grid is the

Kotsiuba, I. Skarga-Bandurova, A. Giannakoulias, and O. Bulda, "Basic Forensic Procedures for Cyber Crime Investigation in Smart Grid Networks," in 2019 IEEE International Conference on Big Data (Big Data), doi: 10.1109/BigData47090.2019.9006215

The published version is available at IEEE Xplore Digital Library: https://ieeexplore.ieee.org/document/9006215

ever-evolving system by leveraging private networks. New connections can be added on a daily basis, changing the configuration and data channels.

Furthermore, advanced forensic frameworks that can preserve users' privacy, are developed but the trade-off between forensic effectiveness and user privacy has not yet been addressed. Besides, current forensic tools suffer from various attacks in their log and event files, making the forensic investigation infeasible. As a result, attackers can launch an attack, and then erase their traces by deleting logs. This suggests, that an advanced Forensic Readiness Framework (FRF) should be designed aiming to collect necessary information from the smart grid systems while being able to protect personal data, and thus, ensuring the fundamental user rights during the preparation of the forensic procedures that will secure a detailed and complete report of the launched attack to the court.

A key aspect here relates to the collection and protection of network traffic data that may subsequently be useful for evidential purposes, which raises two main forms of legal compliance issues. The first is the need for the data to meet generally acceptable standards of evidential reliability and cogency (so as to be usable in subsequent criminal investigation and possible judicial proceedings); the second, concerns the need for processing of such data, where it qualifies as personal data, to satisfy the requirements of applicable data protection law.

Finally, in order to ensure that the results of the network forensic task are reproducible and accurate, forensic investigators will perform their activities within a methodological framework. Within this study the OSCAR methodology [13] will be used.

A. Challenges Relating to Network Evidence

Various studies suggest that for a successful network investigation, the network itself must be equipped with the infrastructure to fully support this investigation. Network-based evidence presents the following challenges:

- **Acquisition.** Due to the many possible sources of evidence within a network environment, it can be tricky to locate specific evidence or, even difficult to gain access to it for political or technical reasons.
- **Content.** Due to the limited storage capacity of the network devices, it might be difficult to get evidence with the desired level of granularity. This is because only selected metadata about the transaction or data transfer is kept instead of complete records of the data that traversed the network.
- **Storage.** As already mentioned, network devices have limited storage capacity and usually they do not employ secondary or persistent storage, resulting in high volatile data.
- **Privacy.** Due to the unique nature of network-based acquisition techniques and the General Data Protection Regulation (GDPR) processing of personal data triggers compliance requirements related to privacy.
- **Seizure.** Seizing a network equipment can be disruptive, bringing an entire network segment down.
- **Admissibility.** Network forensics is a new method related to digital investigations, resulting in conflicting legal precedents for admission of various types of network-based digital evidence.

II. THE FORENSICS PROCESS

The forensics process comprises of the following phases, as shown on Fig. 1:

1. **Collection.** This is where all data related to a specific event is identified, labelled, recorded, and collected, while ensuring that its integrity is preserved. In this phase the Chain of Custody process is initiated.
2. **Examination**. In this phase, we use a forensically sound process to collect data while forensic tools and techniques appropriate to the types of data collected are executed to carve out the relevant information from the collected data while protecting its integrity. During this phase, the results of the investigation process are recorded and noted in the Chain of Custody, including the disposition of any collected evidence used in the examination and how it was used.
3. **Analysis**. In this phase, analysis of the examination results is performed, using justifiable methods and techniques aiming to derive useful information that addresses the questions posed in the particular investigation. The Chain of Custody reporting 'may' be involved in this step.
4. **Reporting**. This final phase includes the documentation of the examination and analysis. It includes details on the use of the various tools, a description of the analysis of various data sources, issues and vulnerabilities identified and recommendations for improvements to policies, guidelines, procedures, tools and other aspects of the forensic process.

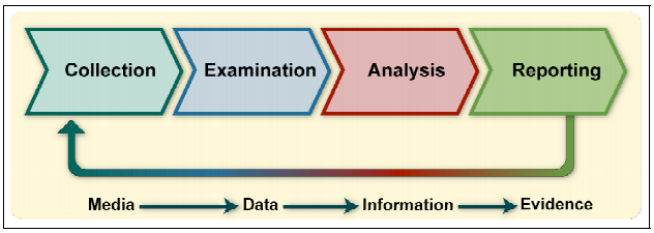

Figure 1 – Forensics process (Source: K. Kent et al., NIST SP 8086, [14])

III. LOGGING AND MONITORING

Within this paper, logging is defined as the passive mechanism used by networking components, where they issue notifications (log messages) periodically or when certain conditions are met [15]. Monitoring is defined as the active probing in order to gather information about the state of systems and networks.

Logs are the most widely-used source data for network and endpoint investigations. They contain application or platform-specific information characterizing handled or observed by the log creator. Logs are widely available and process often in place to analyse them. However, as logs contain varying levels of detail in several formats, parsing is required in order to add context. Moreover, if not already aggregated, finding a specific log entry can involve significant time and effort before analysis can start.

Logging by itself is not suitable for detecting attacks. Together with monitoring, it allows to generate alerts and send notifications in a case of a known pattern being detected. Logging includes DHCP server logs about the assigned IP addresses, webserver logs storing network information about the visitors of the webserver's pages, network devices logs storing network traffic metadata such as connections between IP address, connection date and time, used ports and protocols. However, the majority of logs contain IP address that is considered personal information and as such, they should be protected and should not be kept longer than strictly necessary.

During investigation, event logs provide valuable information, such as system access (in particular, server logins and logouts), startup and shutdown times, errors and problems, information related to network functions, such as DHCP lease histories or network statistics. Moreover, event logs include records of network activity, such as remote login histories.

A. Timeline

During investigations timestamps help us to answer time-related questions and reconstruct the occurrence of events. To be successful, we need to know where to look for timestamps, how to correctly interpret them, how to convert the many different formats available and how to correlate them, since they come from various sources, different operating systems and multiple time zones.

The most common timestamp sources are the timestamps carried by every file on almost every file system, usually being referred to as MAC times (Modification or last written time, Access time, or Change time of a certain file). There are lots of different other sources for timestamps such as meta-data embedded within files (e.g. last printed), server log files, Windows Event Logs, LastWrite timestamps of MS Windows Registry keys, meta-data from the file system itself, web browsing and e-mail artefacts, database timestamps, network captures, etc.

During investigations comprehensive timelines consists of millions and millions of timestamps built from a large number of sources. Fortunately, there is a very powerful, yet free and open-source tool, named Plaso that can combine multiple timestamp sources into a single timeline. So, with respect to network forensics, we can add logs from many different network sources (firewall logs, proxy logs, web server logs, mail server logs, packet captures, etc.).

B. Aggregation, Correlation and Normalisation

Event logs are some of the most valuable sources of evidence for forensic investigators, particularly, when they are stored on a secure central server and can be correlated with multiple log sources. Application servers, firewalls, access control systems, network devices, and many other types of equipment generate event logs and are often capable of exporting them to a remote log server for aggregation.

Analysis of these logs means that events will be aggregated to reduce the total number of events, and group "similar" events together before examination. However, for analysis to be achievable we have to make sure, that the records are comparable and can be correlated to the information they contain, such as timestamps, network addresses, process identifiers, user identifiers, vulnerability identifiers (CVE), etc.

According to [16]: "In event aggregation, similar entries are consolidated into a single entry containing a count of the number of occurrences of the event. For example, a thousand entries that each record part of a scan could be aggregated into a single entry that indicates how many hosts were scanned. Aggregation is often performed as logs are originally generated (the generator counts similar related events and periodically writes a log entry containing the count), and it can also be performed as part of log reduction or event correlation processes.

Event correlation is finding relationships between two or more log entries. The most common form of event correlation is rule-based correlation, which matches multiple log entries from a single source or multiple sources based on logged values, such as timestamps, IP addresses, and event types. Event correlation can also be performed in other ways, such as using statistical methods or visualization tools. If correlation is performed through automated methods, generally the result of successful correlation is a new log entry that brings together the pieces of information into a single place.

In log normalization, each log data field is converted to a particular data representation and categorized consistently. One of the most common uses of normalization is storing dates and times in a single format. Normalizing the data makes analysis and reporting much easier when multiple log formats are in use. However, normalization can be very resource-intensive, especially for complex log entries.

When investigating log aggregation tools, **ELK** short for Elasticsearch, Logstash, and Kibana, is the most popular open source log aggregation tool on the market. It is used by Netflix, Facebook, Microsoft, LinkedIn, and Cisco. Elasticsearch is essentially a NoSQL, Lucene search engine implementation. Logstash is a log pipeline system that can ingest data, transform it, and load it into a store like Elasticsearch. It can also dynamically unify data from different sources and normalize them. Kibana is a visualization layer on top of Elasticsearch, but it lacks an alerting function. However, it offers a geolocation capability, which supports plotting IP addresses on a map. **Graylog** is another option for log aggregation. It uses Elasticsearch, MongoDB, and the Graylog Server under the hood. This makes it as complex to run as the ELK stack. Contrary to ELK, it ships with alerting built into the open source version. Another open-source tool is **Squert** that provides a web interface to view and analyse data coming from Squil databases. **Syslog-ng** is another open source log aggregator that was developed as a way to process syslog (an established client-server protocol for system logging) data files in real time. In essence, network devices sent syslog messages to a syslog server that in turn sends alerts to administrators. As syslog-ng has disk-based buffering, we do not need external buffering solutions to enhance scalability and reliability, making our logging infrastructure easier to create and maintain. Disk-based buffering became part of syslog-ng Open Source Edition (syslog-ng OSE) in version 3.8.1. **Splunk** is a commercial competent, enterprise-grade log management and analysis platform trusted by the world's leading organizations. It correlates log files and store the results in a searchable repository allowing us to search through the information to extract what we need.

To allow correlation of events based on their IP addresses, we should perform **address normalization**, since IP-addresses are only assigned temporarily making correlation of events by IP-address impossible. To overcome this problem investigators should get the logs from DHCP servers and NAT gateways.

Regarding name resolution, although it offers an advantage when messages are read by humans, as it is easier to understand a message like "connection from mail.x.com to mail.y.com port smtp" than "connection from 10.0.1.1 to 10.2.3.4 port 25", it presents a significant problem: translation of numbers to names isn't unambiguous neither globally consistent! Furthermore, as this translation is based on local files for resolution or port numbers or use hard-coded names (/etc/hosts, /etc/services) they can be manipulated by attackers. Moreover, lookups to network services, such as DNS or LDAP, could also fail for various reasons, or be itself susceptible to attacks manipulating the name resolution, like DNS cache poisoning.

For all the above reasons during investigations we should use numbers instead of symbolic names in messages. First, using numbers has also the side effect of being more performant and closes opportunities for denial-of-service attacks as adversaries can inject traffic that would lead to slow name-lookups. Second, it puts less trust in the operating systems involved, which is good from a forensic point of view.

As already presented, during investigations the timeline will assist us in displaying and analyzing the occurrence of events. Timeline generation is based on the timestamps extracted from the data, such as: system logs, file metadata or packet dumps. However, as these are usually generated with the time value and format of the generated system, this leads to two problems: a) synchronization to a central time source (NTP) and b) multiple formats, including daylight savings and multiple time zones, especially in cases where investigations span large areas within the EU. Moreover, the EU that has 4 time zones and their naming is not uniform, with Ireland and the UK using different naming.

Hence, during investigations, we should:

(a) synchronize clocks on all systems using NTP or a similar system,
(b) standardize time formats as much as possible,
(c) include complete, high-precision timestamps (full four-digit year) with time zone information, in the form of an offset, not the name of the time zone and
(d) normalize timestamps to UTC as early as possible in the log chain.

C. Collection and Storage

Using multiple and distributed security probes in the network, will allow us to acquire security event data (log files, flow-data, full content data and statistical data) that will eventually be stored centrally in a secured and dedicated evidence server, for long-term storage, namely the **Forensics Repository** (FR). This central location can be a whole distributed environment, given the large amounts of data to process and may require multiple external storage devices, to handle the large storage requirements. These mounted drives will be protected using file system encryption, via dm-crypt and LUKS.

Once the security event data have been collected, they must be protected from: a) tampering and b) unauthorised access, not only at file system level, but also content level by controlling which event records individual users are allowed to access.

For the on-disk storage **flat files,** including an indexed directory structure (YYYY/MM/DD/HH/file), will be used. This is the less complex storage format and allows us to store syslog, nfdump, pcap, and other data sources. However, file integrity is not guaranteed and thus, has to be supported by external tools like OpenSSL. Additionally, rotation and retention of ever-growing log files can be a challenge on its own, as we have to make sure that no log lines are dropped or written to the wrong file when switching the logfile.

Regarding transferring of log and NetFlow data to the central forensics repository, they should be protected against attackers on the network by incorporating TLS certificates. To be able to deal with network outages and the missing syslog protocol transport security, we should make use of the Reliable Event Logging Protocol (RELP) as specified by Gerhards in [32], which is designed to ensure reliable transfer of rsyslog messages at a higher layer and has provisions for protection from message loss and can also be used over TLS.

Additionally, the logging architecture should provide support for reliability. This can be addressed with offshoots of the syslog daemon including syslog-ng the "next generation" syslog daemon and rsyslog the "rocket-fast system for log processing".

Finally, transfer of forensic images or .pcap files to the central forensics repository has to be done through secure file transfer protocols, such as SSH and HTTPS, since they do not have an intrinsic transfer protocol.

D. Legal Basics

During investigations we have to: a) know the laws that apply to the situation, b) act according to these laws and c) abide by the law, especially since matters may be taken to court. The single most significant piece of legislation relating to privacy protection in the EU is the General Data Protection Regulation (GDPR) (EU) 2016/679.

Network forensics will inevitably collect privacy-related data, foremost IP addresses, but packet captures as well as log files may contain all kinds of data, including passwords, usernames, etc. Depending on the legislation which applies, there will be limitations about what data items can be logged/captured at all, whether data needs to be encrypted, the time period data can be stored, etc. Most significantly, the purpose for which data is logged must be accurately defined. Moreover, consent have to be obtained. All these constraints should be identified and their corrective actions defined in a Data Protection Impact Assessment (DPIA).

E. Prerequisites to Enable Network Forensics

1) Monitoring Policy

Bellow we will detail the preconditions that must be met to enable network forensics.

First, a **policy** on what kind of events should be monitored has to be developed. According to [23] the following approaches are available:

- **Blacklist monitoring** – also called misuse detection in [24].
- **Anomaly (whitelist) monitoring**. The approach is the opposite of the above. Known good behaviour on the network is written down and the monitoring looks for deviations from that norm.
- **Policy (specification) monitoring**. The goal of policy monitoring is to compare events discovered on the network to ensure that they are approved and acceptable. [23]
- **Hybrid**. For example blacklist monitoring on servers reachable from the internet, policy monitoring on high-security parts of the network and anomaly detection in the rest of the network.

There are a wide variety of approaches for selecting the policies to monitor. Once policies are selected, we must determine the environment in which these policies are to be applied. For this we will need an accurate network map, highlighting the underlying functions, applications, and users. Structured, documented network knowledge is fundamental. By deploying tools for documenting and understanding the network environment, we can begin prioritizing security alerts based on how they affect the network.

2) Monitoring Targets

For security to be productive we should target specific systems. By selecting monitoring targets, we can focus and prioritize to the most critical systems, making the most of the security monitoring equipment and resources. We should identify good monitoring targets, if we want to improve our chances of finding the more serious security threats. There are multiple approaches on how to select the critical systems, presented in. [23]

Event collection is always necessary to support investigations and incident response, even if we don't intend to take action. However, for targeted monitoring we should avoid collecting events that we cannot mitigate since they are a distraction.

With policy based security monitoring all unrelated events are filtered out. Effectively this means that when responding to an incident, we will be focusing on specific systems affected during a specific period. However, "secondary" events that are normally filtered out from the monitoring system could help us illuminate a problem, aid in mitigation, trace activity, and attribute actions. Moreover, considering that an intruder will tamper the logs of the compromised systems, it becomes apparent that we should safely collect and store all events (monitoring and secondary events) and log messages in a centralized archival storage for as long as possible.

Armed with the selected policies and the documented network topology (map), we should then conduct a structured assessment of the systems that comprise our network.

The result of this structured assessment, will be a list of systems for which we can target security monitoring.

3) Additional Data Sources

Additional data sources besides data from capturing probes, flow-tools, intrusion detection systems and the like will be needed in a forensic investigation. These include:

a. IP-address information. DNS and DHCP logs should be used to build a timeline of a suspect's activities.
b. WHOIS information, kept in the WHOIS system. However, we should note that due to advice given by ICANN on the impact of GDPR on WHOIS data some major issues have surfaced, as discussed in [25], [26] and [27].
c. Hostnames, which while is bad practice to log them, we might need to deal with DNS names especially in case of dynamics DNS updates. In that case we would need to access the DNS logs

F. Logging Architecture for Incident Investigation

To enable an effective logging capability on smart-grid infrastructures we adopted the UK National Cyber Security Center four step program presented in [15]:

Step 1: Initially, we should generate list of logs that could be used to determine whether infrastructure has been compromised and to what extent.

Step 2: Then, we need to decide how to retain logs. Two primary decisions will shape the approach we will take to logging:

- Once the log sources are identified we need to see how these logs should be properly collected, stored and secured.
- Armed with this information, the next consideration is architectural.

Once we have settled these two matters, we will have outlined the work needed to prime our logging system.

There are essentially three (3) types of log architectures:

(1) *Local architecture.* The default configuration for most operating systems, applications, physical devices, and network equipment. Here logs are collected on individual local hard drives. However, local log aggregation presents issues during investigations since:

- Log collection from multiple systems requires lots of effort and modifies the local system under investigation;
- In successful attacks the attackers usually manipulate the compromised host in order to stop logging any useful information or even log false information.
- Time skew on disparate local systems is often significant and can make it very difficult to correlate logs and create valid timelines.
- Log formats vary between systems.
- Due to the limited storage capacity, limited number of logs can be stored.

(2) *Remote Decentralized architecture.* Common in environments where there is decentralized management of IT resources. Here different types of logs are stored on different remote storage servers. This architecture presents the following advantages:

- Requires far less effort in collecting logs from endpoint devices.

- Increases the forensic value of the collected logs, since they are less likely to be affected by a local system compromise.
- Time skew can be partially mitigated by stamping incoming logs as they arrive, although time skew between servers may still be an issue.

However, there are some drawbacks, when logs are sent across the network:

- Collecting logs from different log servers may still require substantial effort and coordination between teams.
- In case of a network outage, logs may be dropped and lost forever, thus affecting the reliability of the logging infrastructure. This is possible when an attacker executes a denial-of-service (DoS) attack or initiate a network outage thus preventing critical information from being logged. This can be mitigated through the use of protocols that provide support for reliability such as TCP or RELP.
- Since logs are usually forwarded through plain text TCP connection, logs can be intercepted, read and modified by an attacker, thus affecting the **security** of the logging infrastructure. This can be mitigated through the use of encryption protocols such as TLS. However, in decentralized environments this can be cumbersome for network administrators.

(3) *Remote Centralized architecture.* Typically, the most desirable architecture for the purposes of network forensics due to the following reasons:

- In the event of an endpoint device compromise stored logs are not subject to modification or deletion.
- Time skew can be addressed by stamping incoming logs as they arrive, although it does not solve the issue of network transit time.
- Easy access to log data.
- Reliability and security of logs in transit can be centrally addressed, through the use of protocols that provide support for reliability such as TCP or RELP and encryption protocols such as TLS.
- Support for centralized log aggregation and analysis.

This architecture presents the same drawbacks as the decentralized one namely reliability, time skew, confidentiality, integrity and security.

To manage the problem of time skew between servers, clocks on all systems should be synchronized using NTP or a similar system. To address the problem of timestamp format, we should follow the solutions described before, namely include complete, high-precision timestamps (full four-digit year) with time zone information, in the form of an offset, not the name of the time zone. To maintain the confidentiality of logs in transit, TLS/SSL should be used. Finally, to maintain the integrity of event logs in transit we should authenticate the server and client event logging systems.

Considering that an intruder will tamper the logs of the compromised systems, it becomes apparent that we should safely collect and store all events and log messages in a remote centralized archival storage for as long as possible.

Step 3: This step depends on the decision we made above. Since we've chosen a remote centralized solution shown in Fig.2, we need to consider how to implement the following components:

- Logging source established in Step 1.
- Log transport is dictated by the logging source and the service that ingests logs.
- Storage. Accepts pushed logs from device sources and loads them into a data repository, namely the Forensics Repository. Here we should maintain the Chain of Custody.
- Regarding last component querying and analytics, it authenticates users and allows searches to be performed on the data set.

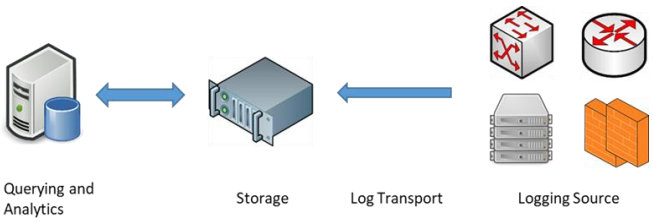

Figure 2 – Centralised Logging Architecture

Step 4: Finally, we should **validate** that the logging capability works as intended. The recommendation is to review and validate the logging strategy every 6-12 months, to capture any relevant changes that have happened in the meantime. This essentially means repeating Step 1 to see if our log sources should have changed. We also need to ensure that existing logs are still being captured as expected. To that extend we should either automate a way of alerting when log messages stop arriving centrally, or initiate a test event that should be captured, and validate that it has been captured by the logging service. And, since we might not need to query large portions of the data set for long periods of time, we should check that any storage mechanism works as intended. Otherwise, we may discover that our logging data is unavailable during an incident.

IV. ANALYSIS

Log files, captures of network traffic and analysis of network related events is at the core of network forensics, whether an incident is still ongoing, or when an incident has closed. If the incident is still in progress, a network tap can be used, collecting live traffic data for later analysis. If the incident has already happened, the evidence log files can be collected for further investigation.

The purpose of the forensic analysis is to answer research questions based on the collected data, like: which systems/people are involved, what happened, where the attack has been carried out and where it has worked out, when the incident/attack occurred, why it happened and how it happened. Other questions include [28] : How long has this activity been going on, is the activity still going on, how many systems were affected, what data was taken, was any sensitive, proprietary, or confidential information taken.

The SANS Network Forensics and Analysis Poster [29], illustrates possible sources of network data:

- **Switches** that through "port mirroring" can send a copy of all packets from a switch port to another switch port.
- **Routers** that can provide NetFlow data for monitoring network traffic.

- **Layer 2-7 devices** that can capture network data, not just devices that provide services but even an endpoint.
- **Network taps** that can duplicate data packet streams and send the data stream(s) to another physical network port or a storage medium.

Network devices store information related to network traffic metadata, including connections between IP address, connection date and time, used ports and protocols. Since the GDPR states that IP addresses (including dynamic IP address) should be considered personal data, they should not be kept longer than strictly necessary.

Monitoring of log files or network traffic on the other hand can generate alerts in case of a known pattern (attack signature) being detected. In case of an incident, (specific) network traffic can be also manually saved in log files for later analysis or review. In general, a network monitoring system monitors the network for problems and an Intrusion Detection System (IDS) monitors a network for threats.

A. Chain of Custody process

When an incident occurs, evidence should be collected and stored securely, while at the same time it should be protected against degradation. When handling evidence, careful registration is very important, as it ensures the integrity and traceability of the evidence from origin to courtroom. Breaches of this integrity affect the legal value of the evidence. Lack of strict control over the personnel responsible for the evidence, at any given point of time, may result in its degradation or compromise.

Integrity ensures that the evidence has not changed from its original state or "reference version". On the other hand, the ability to prove that the evidence has not been corrupted after its creation, ensures its **authenticity**. In other words, authenticity refers to the ability to confirm the integrity of evidence to its original state. Any breach to this integrity will directly affect the legal value of the evidence.

The most common technique used to establish the Chain of Custody (CoC) and demonstrate that the forensic copy has not been altered is that of "hashing". Hashing involves the use of a mathematical algorithm to create a hash value, which is a numeric value of fixed length that uniquely identifies data. Hashing is a one-way function meaning that it's nearly impossible to derive the original text from the hash value. A hash value allows us to compare a forensic copy to the original one, as long as the hash value of the original evidence file has been calculated in advance.

However, there is a small probability that two different forensic copies produce the same hash value. This is called a hash collision attack. MD5 and SHA-1 should not be used for hashing. The hash function should have appropriate strength to make offline brute-force attacks extremely impractical (delay an offline brute-force attack for at least a matter of months). The European Union Agency for Network and Information Security (ENISA) has published a detailed assessment of cryptographic measures. The report is entitled “Algorithms, Key Sizes and Parameters Report - 2014. It supports that, MD-5 should not be used even for legacy applications and SHA-1 is acceptable for legacy systems but should not be used when creating new systems. Instead we should use strong cryptographic algorithms such as SHA-256 or better.

Furthermore, to preserver evidence integrity we should document, preserve and make available for review all activities relating to the seizure, examination, storage, or transfer of digital evidence. Any breach to this integrity will directly affect the legal value of the proof.

Traceability in forensic investigation process includes discovering of complex and huge volume of evidence and connecting meaningful relationships between them. Traceability is not only important to avoid misleading in decision making but also to ensure the valuable information collected is complete and accurate [31], resulting from the combination of information from different sources.

A good monitoring chain can help prove that the evidence in the chain was never left without supervision. Moreover, lack of strict control over who is responsible for the evidence, at any given point of time, may result in its degradation or compromise.

These issues are addressed by the Chain of Custody process that validates the collection, storage, movement and protection of evidence. It provides the forensic link, an audit trail of ‘who did what’ and ‘when it happened’ to a particular piece of evidence. An example of a CoC form is available and can be downloaded from the National Institute of Standards and Technology (NIST) website.

V. SMART GRID NETWORK FORENSICS METHODOLOGY

In order to ensure that the results of the network forensic task are reproducible and accurate, forensic investigators should perform their activities within a methodological framework. Within this paper, the OSCAR methodology is used. This step-by-step process stands for:

- **Obtain information.** During this step we obtain information about the incident itself and the environment, by reviewing the network architecture, and relevant IT policies and procedures. Regarding the **event logs sources** we should be able to know what event logs exist, where they are stored, how we can access them, whom we should contact to get permission to access and collect them and how forensically sound they are. Then we should identify the resources we have available for event log collection, aggregation and analysis including evidence storage space, available time, tools, systems, and staff for collection and analysis. Moreover, we have to consider how the sources of evidence and network itself will be **impacted** by evidence collection, since we might experience network or equipment slowness or outages. Hence, we should be able to know if they can be removed from the network, if they can be powered off, if they can be accessed remotely (active acquisition) and as a last resort if we can access them at specific times or schedule a downtime, to minimise the impact.
- **Strategize.** As this step deals with the planning of the investigation, initially we should **review** all the information obtained, including the time frame for investigation and goals, the potential sources, the available resources, such as hard drives for storing event logs, staff, forensics workstations, and time and how the equipment and networks might be impacted by evidence collection. Once we’ve finished obtaining information we need to **prioritize** sources of evidence, by reviewing the list of possible sources of evidence and identify those that are likely to be of the highest value to the investigation. Next we should consider the

effort needed to obtain them. Since the majority of our evidences are stored in the central logging server (FR), it is fairly straightforward to gather copies of them. Nevertheless central evidence aggregation presents its own unique challenges that might affect our plan and budget. For instance we have to research the underlying log collection architecture to ensure that it is forensically sound and meet our needs for evidence collection. To that extend we should know the transport-layer protocol in use for log transmission, as well as the means to securely forward log files to the FR such as mechanisms for authenticating the logging client and server and encrypting data in transit, to determine the risk of event log loss or modification.

Based on this information, we can prioritize our evidence accordingly.

Succeeding, we should **plan** the acquisition by discussing with our primary contact in order to determine which organisation personnel (system and/or network administrators) can provide us access to the evidence. Once this is settled we should plan the acquisition method. Physical access to the system rather than remote is preferred, in order to avoid generating network traffic that in turn will modify router statistics, flow record data, and access logs. The time of day may be also important if the investigation must remain secret, or if the central logging server is under heavy load at certain hours.

Regarding collection and analysis of router network flow statistics (NetFlow, IPFIX, sFlow), it can be implemented using different software for probes, collectors and analysers. Probes usually come implemented in the operating system of the routers. If this is the case they should configured to export network flows to the NfDump collector running on the forensics repository (FR). However, to support scenarios where the routing equipment does not support data flow (NetFlow) exporting, we should install the softflowd software probe, on the FR, that should also be configured to receive all network traffic through the switch “port mirroring”.

Regarding transferring of logs and network flows to the FR, they should be protected against attackers on the network, by encrypting syslog traffic with TLS. The following procedure should be followed:

a. Set up a certificate authority (CA). Here we should note that the CA must not be connected to the rest of the smart-grid network, since if compromised, the overall system security is breached. The resulting ca-key.pem file should be safeguarded, as if some third party obtains it, our security will be broken.

b. Generate certificates for each of the machines and the FR. Generation of both the private and public keys, should be done on the CA (which is NOT a server!) and then copy them over to the respective machines, by distributing a copy of ca.pem, cert.pem and key.pem.

c. Set up the FR, by copying over ca.pem, fr-key.pem and fr-cert.pem to the central server. Ensure that no user except root can access them (even read permissions are really bad). Then we should configure the server so that it accepts messages from all machines in the identified subdomain.

d. Set up the client(s), in order to communicate only with the FR. This is an important step, as it can prevent man-in-the-middle attacks. To do so, we should generate and copy over the ca.pem, pilot-machine-key.pem and pilot-machine -cert.pem to the client(s). Like before we should ensure that no user except root can access them (even read permissions are really bad). Then we should configure the client so that it checks the server identity and sends messages only to the FR.

Finally, once our plan is complete we should communicate the final plan to everyone involved.

- **Collect evidence**. Regarding collection of evidence from the central logging server, there are various methods. If the server comes with a log analysis tool, such as Splunk or ELK, we can use the dedicated web-interface to access the evidence. If this is not the case, we can access the central log server using services such as SSH, RDP, or direct console connection depending on the specific configuration.

  If however, the FR is geographically farther away than we could access otherwise, we should utilise a manual remote connection through a VPN service. One drawback is that we will modify the system under examination simply by accessing it remotely. We will create network activity through the process of manual remote examination, which can also contribute to network congestion, especially if we need to transfer large amounts of data across the network.

  Additionally if the IT staff are unaware of the investigation or uncooperative we should acquire evidence **passively.** Nevertheless, this method is only effective in environments where we can access the network segments over which the event log data is transmitted, and when the log data is not encrypted in transit.

  Nonetheless, it is advisable to either take a forensic image or simply copy the evidences of interest of the central logging server’s hard drive(s), by **physically** connecting to it, as it does not modify the environment under investigation and in general minimizes the network footprint of the investigation.

  Imaging the logging server’s media by making a bit-for-bit copy of all sectors on the media is a well-established process that is commonly performed on the hard drive level, hence often referred to as forensic imaging. The creation of a true forensic hard drive image can be very resource-intensive and is a highly detailed process. However, it allows a forensic copy of the server’s drive to be preserved and presented later. It can also allow for a very detailed analysis of the logging server configuration.

  The forensic image should be performed by an objective third party, to avoid accusations of evidence tampering or spoliation and increase our chances of

obtaining admissible evidence as a result of our discovery efforts.

In order to copy the evidences we simply use a physical port (i.e., eSATA or USB). This has the strong advantage of having a relatively low impact on system resources (i.e., copying files takes far less time, storage space, and I/O than making a forensic image) and the system does not need to be taken offline or powered down in order to copy files.

When we use a physical docking station to connect the logging server's media (evidence hard disk) to our computer, the operating system will leave traces on the evidence disk and our proof material will be altered. What we need is a hardware write-block device (write blocker) to make the forensic image or to simply copy the evidences of interest. With the write blocker between the evidence disk and the capture device there will be no traces on the evidence disk.

Finally in this step we initiate the **Chain of Custody** process, by documenting everything we do such as systems accessed and all actions taken during evidence collection, including time, source of the evidence, acquisition method and the involved investigator(s).

Regarding the forensic imaging process, we should write down the characteristics of the disc (size, label, serial numbers and stickers). Then we should create and note the hash value of the forensic disk image and duplicate the forensic disk image to a working/investigate image. The industry standard for imaging currently recommends the use of the MD5 algorithm for hashing. The reason why we need two copies of that if we ever damage our working copy, we can make a new copy from the forensic image again, without having to touch the original evidence disk after taking the forensic image. Even if we simply copy the evidences of interest, we should also capture cryptographic checksums of the source and destination files to ensure that we have made an accurate duplicate.

- **Analyse**. During this step we recover evidence material from the forensic working image using a number of different methodologies and tools.

According to Brian D. Carrier, "*After the obvious evidence has been found, then more exhaustive searches are conducted to start filling in the holes*." This suggests that the selected method for analysis depends on the case and what leads are already present. As a result, several iterations of examination and analysis are foreseen to support a theory. During the analysis process the following should be considered as essential:

  - Correlation of data from different sources.
  - Timeline of activities, allowing us to understand who did what, when, and how is the basis for any theory of the case.
  - Isolation of Events of Interest.
  - Corroboration of events though multiple sources due to the low fidelity of data and the problem of "false positives."
  - Recovery of additional evidence, since all previous steps could increase the evidence collected and analyzed.
  - Interpretation (educated assessments) of the meaning of the collected evidence, aiming to help investigators identify additional sources of evidence, and construct a theory of the events that were likely to have occurred. Attention must be paid in differentiating between the interpretations of the evidence from the fact, since interpretations are hypothesis that may be proved or disproved.

- **Report**. This is the most important aspect of the investigation as it conveys the results of the investigations to the client(s). The report must be understandable by non-technical persons like managers, judges, etc. It must be factual and defensible in detail We should ensure that the forensic report incorporates evidence from the event logs, such as:

  - Graphical representations from the event log analysis tools, including charts and graphs.
  - Detailed information regarding the event log source and the process followed for collecting them.
  - Information regarding the methodology and the analysis tools used, since widely known and tested tools, are more likely to be accepted in a courtroom.

  Finally we should preserve and reference the original sources of evidence so that we can support the reported findings.

  Additionally if we must modify the system configuration, we should record the changes and collect screen captures or other documentation whenever possible. Moreover, when connecting to a networked device via console interface, we should keep a record of all commands issued and their output. This can be done using client-side tools; for example, the "screen" command on Linux includes an option (-L) for keeping a full log of the session, while providing excellent documentation of the forensic investigative techniques

  It should be mentioned also that all claims must be supported by evidences.

CONCLUSIONS

Network forensic investigations pose a myriad of challenges. To meet these challenges, investigators must carefully assess each investigation and develop a realistic strategy that takes into account both the investigative goals and the available resources. [30]

As Sun Tsu wrote 2,500 years ago: "A victorious army first wins and then seeks battle; a defeated army first battles and then seeks victory." Strategize first; then collect your evidence and conduct your analysis. By considering the challenges unique to your investigation up front, you will meet your investigative goals most efficiently and effectively. [30]

Using the OSCAR based methodological framework and relevant open source tools, necessary forensic information can be collected, stored and used as legal evidence in court, while its integrity and authenticity cannot be judged. To support this a number of preconditions must be met to

enable network forensics, including an effective logging architecture based on the UK National Cyber Security Center four step program. With retrospective analysis, it enables IT specialists and information security officers to investigate network incidents in detail, addressing infrastructure vulnerabilities even before devastating and critical consequences occur as well as gathering an exhaustive evidence base to protect the legitimate interests of the smart-grid stakeholders from internal and external threats.

ACKNOWLEDGMENT

This research work is based upon the concept of the SPEAR project that has received funding from the European Union Horizon 2020 Research and Innovation programme under the Grant Agreement No. 787011 (SPEAR).

REFERENCES

[1] Carr D. Security Information and Event Management. // Baseline, 2005, no. 47, p. 83.

[2] Aguirre I. and Alonso S. Improving the automation of security information management: A collaborative approach. // IEEE Security Privacy, vol. 10, no. 1, pp. 55–59, Jan 2012.

[3] Cerullo G., Formicola V., Iamiglio P., Sgaglione L. Critical infrastructure protection: having SIEM technology cope with network heterogeneity. CoRR, vol. abs/1404.7563, 2014. [Електронний ресурс] // URL: http://arxiv.org/abs/1404.7563. (дата звернення: 01.11.2018).

[4] Nicolett M., Kavanagh M., Magic Quadrant For SIEM, Gartner Technical Report, 2011, Last Access: 29/01/2016. [Електронний ресурс] // URL: http://goo.gl/2upsI4 (дата звернення: 21.11.2018).

[5] Leszczyna R., Wrãsbel M.R. Evaluation of open source siem for situation awareness platform in the smart grid environment. // 2015 IEEE World Conference on Factory Communication Systems (WFCS), May 2015, pp. 1-4.

[6] AlienVault [Електронний ресурс] // URL: https://www.alienvault.com/blogs/industry-insights/of-dragons-elephants-aliens-a-decade-of-ossim (дата звернення: 05.12.2018).

[7] Cyberoam iView [Електронний ресурс] // URL: https://www.cyberoam.com/cyberoamiview.html

[8] Prelude Universal Open-Source SIEM project [Електронний ресурс] // URL: https://www.prelude-siem.org/

[9] Di Sarnoa C., Garofalo A., Matteucci I., Vallini M. Ensuring Cyber-Security in Hydroelectric Dam through a novel SIEM system. [Електронний ресурс] // URL: https://pdfs.semanticscholar.org/3578/a35321b07328b75ca7a136b9ea261939137e.pdf (дата звернення: 05.12.2018).

[10] Phines P. DCSimSep [Електронний ресурс] // Github. URL: https://github.com/phines/dcsimsep (дата звернення: 9.11.2018).

[11] Hines P., Cotilla-Sanchez E., Blumsack S. Do topological models provide good information about vulnerability in electric power networks? // Chaos: An interdisciplinary journal of non-linear science, 2010. vol. 20, no. 3

[12] Eppstein M., Hines P. A "Random Chemistry" algorithm for identifying collections of multiple contingencies that initiate cascading failure // IEEE Transactions on Power Systems, 2012, vol. 27, no. 3, pp. 1698-1705.

[13] Network Forensics Investigative Methodology (OSCAR) http://comp.org.uk/network-forensics-investigative-methodology-oscar.html

[14] Kent, K., Chevalier, S., & Grance, T., Guide to Integrating Forensic Techniques into Incident Response, NIST SP 800-86, 2006

[15] Introduction to logging for security purposes, Laying the groundwork for incident readiness, UK National Cyber Security Center, July 2018

[16] NIST Special Publication 800-92: Guide to Computer Security Log Management – Recommendations of the National Institute of Standards and Technology, Kent, K. and Souppaya, M., September 2007

[17] I. Matteucci, P. Mori, M. Petrocchi, Prioritized execution of privacy policies, in: Data Privacy Management and Autonomous Spontaneous Security, 7th International Workshop, DPM 2012, and 5th International Workshop, SETOP 2012, Pisa, Italy, September 13-14, 2012. Revised Selected Papers, 2012, pp. 133–145. doi:10.1007/978-3-642-35890-6_10.

[18] F. Cuppens, N. Cuppens-Boulahia, M. B. Ghorbel, High level conflict management strategies in advanced access control models, Electr. Notes Theor. Comput. Sci. 186 (2007) 3–26.

[19] E. C. Lupu, M. Sloman, Conflicts in policy-based distributed systems management, IEEE Trans. Softw. Eng. 25 (6) (1999) 852–869.

[20] E. Syukur, Methods for policy conflict detection and resolution in pervasive computing environments, in: WWW05, ACM, 2005,.

[21] A. Masoumzadeh, M. Amini, R. Jalili, Conflict detection and resolution in context-aware authorization, in: Security in Networks and Distributed Systems, IEEE, 2007, pp. 505–511.

[22] N. Dunlop, J. Indulska, K. Raymond, Methods for conflict resolution in policy-based management systems, in: Enterprise Distributed Object Computing, IEEE, 2003, pp. 98–109.

[23] Security Monitoring – Proven Methods for Incident Detection on Enterprise Networks, Fry, C., Nystrom, M., O'Reilly, 2009, ISBN: 978-0-596-51816-5

[24] Network Intrusion Detection and Prevention – Concepts and Techniques, Ghorbani, A. et al., Springer, 2010, ISBN 978-0-387-88770-8

[25] ICANN GDPR and Data Protection/Privacy Update, sep. 2018

[26] TR-53 - Statement about WHOIS and GDPR, Computer Incident Response Centre Luxemburg, 2018

[27] The GDPR and WHOIS privacy, Michael Hausding, SWITCH-CERT, 2018

[28] Joshua Goldfarb, Use Cases In the Enterprise, FireEye, July 2014

[29] SANS Digital Forensics & Incident Response (DFIR), Network Forensics and Analysis Poster

[30] Network Forensics, Tracking Hackers through Cyberspace, by Sherri Davidoff and Jonathan Ham, 2012, ISBN-13: 978-0-13-256471-7

[31] Selamat et al., A Forensic Traceability Index in Digital Forensic Investigation, Journal of Information Security, 2013

[32] Gerhards, R., RELP – The Reliable Event Logging Protocol, April 2014, http://www.rsyslog.com/doc/relp.html