NTNU

Norwegian University of
Science and Technology

Multisensor Fusion for Intrusion Detection and Situational Awareness

Christoffer V Hallstensen

Master in Information Security
Submission date: June 2017
Supervisor: Katrin Franke, IIK

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Multisensor Fusion for Intrusion Detection and Situational Awareness

Christoffer V. Hallstensen

01-06-2017

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Norwegian University of Science and Technology, 2017

Supervisor : Prof. Katrin Franke, PhD

Preface

This work is a Master's thesis at the Department of Information Security and Communication Technology at NTNU. It was carried out during the spring semester of 2017. The basis for this research originally stemmed from my passion for network security monitoring, intrusion detection, and situational awareness in large scale network systems, and for open source security technology. Moreover, my experience with incident response, intrusion detection, system administration and networks, provides me with the knowledge about real complications regarding intrusion detection and digital forensics. The work has been done as a part of solving the problem of network security monitoring in the largest university in Norway.

Gjøvik, 01-06-2017

Christoffer V. Hallstensen

Acknowledgment

First and foremost, I would like to thank Anastasiia Moldavska for all her support and help through this thesis work. Secondly, I wish to express my gratitude to Kiran B. Raja for reading the work and providing a good feedback. I would also like to thank my supervisor, Prof. Katrin Franke, for her moral support and belief in this work. I extend my gratitude to my colleagues at NTNU IT, and especially the NTNU Digital Security section, for their support and patience during the writing of this thesis. Lastly, I would like to thank fellow students at NTNU Digital Forensics group and staff of Institute for Information Security and Communication Technology for good discussions, challenging ideas, and valuable comments.

C.V.H.

Abstract

Cybercrime damage costs the world several trillion dollars annually. And although technical solutions to protect organizations from hackers are being continuously developed, criminals learn fast to circumvent them. The question is, therefore, how to create leverage to protect an organization by improving intrusion detection and situational awareness? This thesis seeks to contribute to the prior art in intrusion detection and situational awareness by using a multisensor data fusion. The model for multisensor data fusion system incorporates human cognition reasoning into a hybrid multisensor fusion, i.e. vertical fusion, horizontal fusion within a network segment, and horizontal fusion between the network segments. The proposed model is able to reduce false positive alarms for intrusion detection, improve the detection of unknown threats, and provide coverage for the whole cyber kill-chain.

Contents

Preface . . . . . **i**
Acknowledgment . . . . . **ii**
Abstract . . . . . **iii**
Contents . . . . . **iv**
List of Figures . . . . . **vii**
List of Tables . . . . . **viii**
List of Abbreviations . . . . . **ix**
1 Introduction **1**
1.1 Keywords 1
1.2 Topics covered 1
1.3 Problem description 1
1.4 Justification, Motivation, and Benefits 2
1.5 Research Questions 2
1.6 Contribution 3
1.7 Outline 3
2 Theoretical Background **5**
2.1 Digital Forensics 5
2.1.1 Introduction to digital forensics 5
2.1.2 Reactive Digital Forensics 11
2.1.3 Proactive Digital Forensics 13
2.1.4 Active Digital Forensics 15
2.1.5 Challenges in Digital Forensics 17
2.2 Network Security Monitoring 18
2.2.1 Introduction to Network Security Monitoring 18
2.2.2 Collection phase 19
2.2.3 Detection phase 25
2.2.4 Analysis phase 28
2.2.5 Challenges in Network Security Monitoring 33
2.3 Cyber Threat Intelligence 35
2.3.1 Introduction to Cyber Threat Intelligence 35
2.3.2 Cyber Situational Awareness 39
2.3.3 Strategic Cyber Threat Intelligence 43
2.3.4 Tactical Cyber Threat Intelligence 43
2.3.5 Operational Cyber Threat Intelligence 44
2.3.6 Challenges in Cyber Threat Intelligence and Situational Awareness 44

2.4 Big Data Principles and Technology . . . 46
2.4.1 Data, Information, and Data Structure . . . 48
2.4.2 Big Data Architecture Designs . . . 49
2.4.3 Big Data Transfer . . . 51
2.4.4 Big Data Storage . . . 53
2.4.5 Big Data Processing . . . 55
2.4.6 Challenges in Big Data . . . 61
2.5 Multisensor Data Fusion . . . 62
2.5.1 Introduction to Multisensor Data Fusion . . . 62
2.5.2 Intelligence Cycle . . . 65
2.5.3 The Boyd Control Loop model (OODA Loop) . . . 67
2.5.4 JDL Data Fusion Model . . . 68
2.5.5 Visual Data Fusion Model . . . 70
2.5.6 Waterfall Data Fusion Model . . . 71
2.5.7 Omnibus Data Fusion Model . . . 72
2.5.8 The Dasarathy model . . . 73
2.5.9 Multisensor Data Fusion Systems Design . . . 75
2.5.10 Challenges in multisensor data fusion . . . 75
3 Related Work . . . **77**
3.1 Digital forenics . . . 77
3.2 Cyber Threat Intelligence and big data . . . 77
3.3 Big Data and Network Security Monitoring . . . 79
3.4 Multisensor data fusion for intrusion detection . . . 80
4 Methodology . . . **82**
4.1 Literature Study . . . 83
4.1.1 Sources . . . 83
4.1.2 Search terms . . . 83
4.1.3 Method discussion . . . 84
4.2 Questionnaire . . . 84
4.2.1 Expert survey . . . 84
4.2.2 Method discussion . . . 84
5 Relationships between domains . . . **85**
6 Proposed model for a MSDF system for ID and SA . . . **88**
6.1 Requirements . . . 88
6.2 Proposed model . . . 90
6.2.1 Vertical and Horizontal Fusion . . . 92
6.2.2 Device / Sensor (S1,S2,S3,...,Sn) . . . 92
6.2.3 Data Refinement (L0) . . . 94
6.2.4 Object Refinement (L1) . . . 94
6.2.5 Databases . . . 95
6.2.6 Intrusion Analysis Engine (L2) . . . 95
6.2.7 Target Tracking Engine (L2) . . . 96

6.2.8 Situation Assessment Engine (L2) . . . 96
6.2.9 Threat Assessment Engine (L3) . . . 96
6.2.10 Data Mining and Learning . . . 96
6.2.11 Process Refinement (L4) . . . 96
6.2.12 Cognitive Refinement (L5) . . . 96
6.3 Applicability of model . . . 97
6.4 Model assessment . . . 97
6.5 Limitations . . . 98
6.6 Implementation considerations . . . 98
7 Discussion and Implications . . . 100
7.1 Theoretical implications . . . 100
8 Conclusions . . . 102
9 Further work . . . 103
Bibliography . . . 104
Appendix . . . 113
A Questionnaire . . . 114

List of Figures

1 Forensic Science [1] . . . 6
2 Hypothesis-Driven Crimescene reconstruction [2] . . . 10
3 Digital Forensic Investigation Process [2] . . . 12
4 Network Security Monitoring Cycle, based on [3] . . . 19
5 What can each NSM tool detect? (Figure 7-6 [4]) . . . 29
6 Relationship of Data, Information and Intelligence [5] . . . 35
7 The Intelligence Pyramid [6] . . . 43
8 Lamda Architecture, based on [7] . . . 49
9 Kappa Architecture, based on [7] . . . 50
10 Flume Dataflow Model[8] . . . 52
11 MapReduce, Figure 3-1 in [9] . . . 56
12 The Spark Stack[10] . . . 58
13 Spark Streaming[11] . . . 59
14 The Intelligence Process [5] . . . 65
15 Boyd's control loop (OODA Loop) . . . 67
16 Revised JDL Data Fusion Model [12] . . . 68
17 Visual Fusion Model [12] . . . 70
18 Waterfall Fusion Model [12] . . . 71
19 Omnibus Fusion Model [12] . . . 72
20 Apache Metron Logical Architecture [13] . . . 80
21 Research methodology . . . 83
22 Relationship between domains . . . 86
23 Proposed model for Multisensor data fusion for ID and SA . . . 91
24 Sensor perspectives . . . 93
25 Distributed sensor network . . . 94

List of Tables

1 Order of Volatility [14] 9
2 Latency in Big Data [9] 51

List of Abbreviations

API Application Programming Interface

C2 Command and Control

CF Computer Forensics

CNA Computer Network Attack

CND Computer Network Defense

CoC Chain of Custody

CTI Cyber Threat Intelligence

CyberSA Cyber Situational Awareness

DF Digital Forensics

FPC Full Packet Capture

FPCD Full Packet Capture Data

HDFS Hadoop File-System

HSQL Apache Hive SQL

HTTP Hyper-Text Transfer Protocol

ID Intrusion Detection

IoC Indicator of compromise

IP Internet Protocol address

NF Network Forensics

NSM Network Security Monitoring

OOV Order of Volatility

PSD Packet String Data

RDD Resilient Distributed Dataset

SA Situational Awareness

SIEM Security Information and Event Management

SQL Structured Query Language

SSL Secure Socket Layer

Syslog System Log

TCP Transport Control Protocol

UDP User Datagram Protocol

YAF Yet Another Flow Meter

1 Introduction

This chapter presents the idea behind the project, providing an introduction to the topic to be researched. The problem is described and the research questions are developed to address the given problem.

1.1 Keywords

Forensics, Forensic Readiness, Intrusion Detection,Situational Awareness, Sensor Fusion, Network Security Monitoring, Cyber Threat Intelligence, Big Data, Incident Response.

1.2 Topics covered

Digital Forensics, Cyber Threat Intelligence, Situational Awareness, Network Security Monitoring, Big Data, Multisensor data fusion.

1.3 Problem description

Nowadays, criminals and hackers are using more sophisticated means for hiding their activities in corporate networks. This development is making both detection and forensic investigations a more complicated task. In addition to the increase in more sophisticated attacks, the sheer volumes of data stored in corporate network and traversing the gigabit network connections are ever increasing. The analysis of the growing amount of data becomes impossible for humans to analyze manually and near-impossible to detect and protect against in real-time. Suitable tools are being developed for data collection and intrusion detection (ID), supporting the principle of forensic readiness and enabling the incident response process, and many of them are already reasonably good. But with the ever increasing use of encryption and obfuscation, a single tool alone is not capable to detect, collect, and triage for responding to an incident and performing the forensic investigation. Most of the tools today work fairly well within their respective domains. But with the increasing threats corporate networks are facing today, a single tool will not cover the basis for detection and supporting investigations in an ever increasing stream of potential intrusions. Corporate networks need to utilize multiple tools to cover each domain, to look at the network from different angles, to correlate events. But as security technology usage increases, so does the complexity. It is easy to loose the overall situational awareness (SA), miss successful attacks and adapt accordingly to the current threats. Every network is different, and there is no one-fits-all solution to securing it and apply forensic readiness.

1.4 Justification, Motivation, and Benefits

As the Norwegian society and companies become more and more dependent on digital infrastructure, the need to protect this digital infrastructure become more critical, as we are far beyond the point of being dependent upon computers. Security reports show that threats against digital infrastructure and information systems are increasing [15, 16, 17]. The increase in bandwidth, computing power, storage sizes, and mobile computing, bring you own device and cloud services increase the complexity of detecting and responding to threats in current networks. Reports also show that threat actors evolve their tactics, techniques and procedures and become more sophisticated [18, 19, 20]. The usage of anti-forensic techniques, in combination with the mentioned factors, increases the difficulty of detecting intrusions and perform digital forensics. Intrusion detection becomes more resource intensive than ever.

Symantec reports that threat actors now are hiding in plain sight by using network administrative tools or benign software that are already available in the targeted network, making them harder to detect as less zero-days and malware are being used after footprint inside the network is set up [18]. Mnemonic is describing in their threat report that industrialization of cyberattacks has evolved to the point that if you remove the malicious intent, it is difficult to differentiate an attack group from a normal organization [19]. Cisco ASR reports that most organizations use more than five different security vendors. 28% of vendors list adapting advanced security solutions as their top constraint because of product compatibility. Organizations only manage to investigate 56% of security alerts per any given day, and from those, only 28% are True Positive [20].

Many domain-specific tools and methods today serve different types of intrusion detection and digital forensics. All tools have advantages and disadvantages depending on what kind of intrusion that are to be detected. Often, threat actors use multi-stage attack tactics that can not be fully unveiled by network-based intrusion detection or netflow alone. To fully understand the attack and to gain situational awareness require to look at the same attack from different angles.

Fusion of data from multiple sources and sensors can provide collaboration, better threats detection, threat intelligence, and situation assessment.

1.5 Research Questions

To address the current problem with intrusion detection and situational awareness, the goal of this work is to develop a new approach to intrusion detection and situational awareness to better serve the purpose of incident response and digital forensics. In order to achieve the goal, the following three research questions are developed.

In the context of modern networks, there are many potential sources of data, but because of modern networks complexity, i.e. velocity, volume, veracity, and variety of data, a careful selection of what to collect, process, and store, is needed

due to resource constraints. This problem poses the background for the first research question:

Q1: *What data and why should be collected for event analysis?*

a. What defines an effective event analysis?
b. What type of data and data sources are required for effective event analysis?
c. What are the appropriate tools for data collection?

After data is collected, it needs to be processed, structured, correlated, given a context, and made available for use. This leads to the second research question:

Q2: *What approaches can support data processing for intrusion detection and situational awareness?*

a. In order to enable proactive, active, and reactive digital forensics.
b. In order to support the principle of forensic soundness.
c. In order to ensure flexibility and scalability of data processing.

There already exist several domain specific tools for network security monitoring. These often perform their task well in operation environments within the constraints of their design. But they cannot provide a complete picture which brings us to the third research question:

Q3: *How can data from different sensors be combined?*

a. In order to enhance intrusion detection.
b. In order to enhance situational awareness.

1.6 Contribution

The planned contributions of this research project are:

(I) To produce knowledge about application of multisensor data fusion, e.g., how to combine data from different types of sensors (perspectives), to increase reliability and confidence of intrusion detection and situational awareness in computer networks.
(II) To provide new knowledge about how the domains of digital forensics, network security monitoring, cyber threat intelligence, multisensor data fusion and big data technology enables and improve each other.
(III) To provide the model for a MSDF system for ID and SA.

1.7 Outline

This thesis is structured as follows. Chapter 2 presents theoretical background based on literature-studies, explaining the concepts behind this work in depth. Chapter 3 presents related work. Chapter 4 describes the methodology used to achieve the goal of this research. In chapter 5 the new approach to intrusion detection and situational awareness using multisensor data fusion is presented.

Chapter 6 presents a model for MSDF system for intrusion detection and situational awareness to serve the purpose of incident response and digital forensics. Discussion and implications of the model are presented in section 7 that is followed by the concluding remarks.

2 Theoretical Background

In this chapter the background theory and concepts are presented. Literature review was performed to study each concept in depth. The chapter is divided into six sections that cover the main concepts behind this thesis, 2.1 Digital Forensics, 2.2 Network Security Monitoring, 2.3 Cyber Threat Intelligence, 2.4 Big Data Principles and Technology, 2.5 Multisensor Data Fusion. Moreover, tools used by industry in each of the five domains are reviewed in Section 3.

2.1 Digital Forensics

> When you have eliminated the impossible, whatever remains, however improbable, must be the truth.
>
> Sherlock Holmes

2.1.1 Introduction to digital forensics

Forensic Science is the application of science and technology to investigate and establish facts of interest to criminal or civil court of law. Forensic Science was established as a domain within science during the 1800s to early 1900s. By using science in criminal investigations, the effectiveness of law enforcement increased significantly [21]. Forensic science is a multi-disciplinary scientific domain where the sub-domains are anything that can be related to a crime, which may encompass most of both the physical and digital world. Franke and Srihari [22] provide a general list on how Forensic Science is used to:

1. Investigate and to reconstruct a crime scene;
2. Collect, analyze and trace evidence;
3. Identify, classify, quantify and individualize persons, objects and processes;
4. Establish linkages, associations and reconstructions; and
5. Utilize those findings in the prosecution or the defense in a court of law.

As seen in Figure 1, Forensic Science is built upon and evolved around the research on new methodologies and technologies or reapplication of existing ones, in the application to forensic related problems.

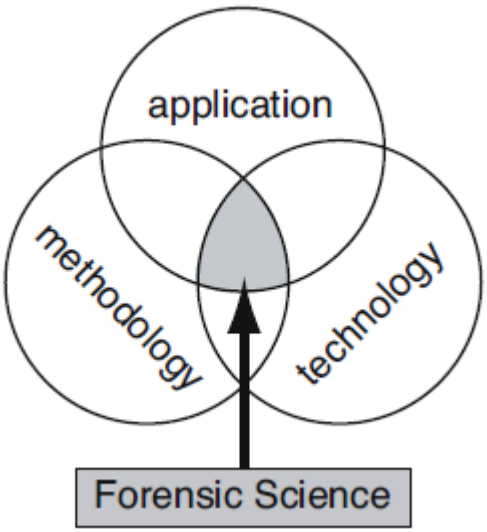


Figure 1: Forensic Science [1]

Computer forensics (CF) was the starting point for what we call digital forensics (DF) today. CF was described by Farmer and Venema [14]. "*Forensic analysis of a computer system is all about capturing the data and then processing the information gathered to prove, or disprove that an event has or has not occurred*" [14].

Network Forensics (NF) is often used interchangeably with DF. For example, Nelson et al. [23] defines NF (Digital Forensics) in the following way. "*Network Forensics is the process of collecting and analyzing raw network data and tracking network traffic systematically to ascertain how an attack was carried out or how an event occurred on a network*". However, in the context of this work, network forensics is seen as a sub-domain of digital forensics.

Digital Forensics is an umbrella term used to describe the application of forensic science to digital devices. Digital Forensics can be defined as "*the process of employing scientific principles and processes to analyze electronically stored information and determine the sequence of events which led to a particular incident*" [24].

Computational Forensics is described by Franke and Srihari [22] as an emerging multi-disciplinary research domain where computers are being used for *hypothesis - driven investigations* of problems in forensic science with the primary goal of knowledge discovery and advancement of the forensic discipline. Computational Forensics involves modeling and computer simulations (synthesis), and/or computer-based analysis and recognition in order to achieve (1) in-depth understanding of a forensic discipline, (2) evaluation of a particular scientific

method, and (3) systematic approach to forensic sciences, by applying computer science, applied mathematics and statistics.

The main terminology and principles of Digital Forensics are presented below.

Digital Evidence is any digital data that contains reliable information that supports or refutes a hypothesis about an incident [2, 21].

Comprehensive Digital Evidence is evidence that will have evidentiary weight in a court of law. It contains all evidence—relevant, sufficient, and necessary—that with a great level of certainty can determine the root cause of an event and the responsible party that will lead to a successful prosecution of the perpetrator [25].

Evidence acquisition The more accurate and complete the extracted data is, the better and more comprehensive the analysis can be, and more accurate results can be reported to support conclusions drawn from an investigation [14].

Forensic copy is a bit-by-bit copy of the original evidence to ensure that the original evidence is not altered in any way during an investigation. A forensic copy ensures that the evidence integrity is intact and ensures Forensic Soundness as someone else can start from scratch with the original evidence. This also ensures that if new techniques are being developed, they can be tested without compromising evidence integrity.

Digital Fingerprints is a way to ensure evidence integrity and Chain of Custody. Digital Fingerprinting is most often a one way cryptographic hash algorithm like MD5, SHA1 or SHA2 that are used, for instance, to make sure that a forensic copy is the same as the original evidence, or to ensure and verify that a procedure or evidence extraction method yield the same result using the same method [2].

Anti-Forensics (Counter Forensics) is methods used by an perpetrator that subverts the probability of a successful Forensic Investigation that results in a collection of comprehensive digital evidence. Anti-Forensics methods are designed for preventing evidence collection, increase the resources and time needed for an investigation, deceive the investigator by leaving misleading evidence, and prevent detection of the event or crime altogether [26, 27]. Methods used to achieve this are Encryption, Steganography, Obfuscation, Proxies, Memory only execution, Secure Deletion, and Data Tampering [28].

Evidence dynamics is any influence that changes, relocates, obscures, or obliterates evidence regardless of intent.

Multi-Tool Verification In digital forensics, the need for multi-tool verification when extracting digital evidence is needed as different tools might yield different results. Some tools are good for specific problems while other might fail or provide false positive or false negative result on the same dataset. The purpose of Multi-Tool verification is to discover human or software errors, and ensure that repeatability of the results is possible. The tools being used and put up against each other should ideally be developed by different people or companies.

Chain of custody Chain of Custody is the complete documentation of evidence acquisition, control, analysis, and disposition of it in both digital and physical form. Chain of custody is often performed in the form of timestamps and cryptographic hash values, checklists, notes, photos and reports. The Chain of Custody documentation should at least make sure that the following is documented [29, 21]: the time of evidence acquisition; location of evidence collection; the reason for collecting the evidence; the person handling the evidence; method of collection, examination, and analysis; and processes and procedures performed on the evidence [29, 21].

Evidence integrity Evidence Integrity refers to the preservation of the evidence in its original form. This is a requirement for both the original evidence and forensic copies of the original evidence.

Forensic Soundness Forensic Soundness refers to the fact that the method or tool adhere to digital forensics principles and processes after best practice and legal requirements. A typical interpretation is: source data is not altered in any way; every bit is copied, no data is added to the image [2, 21]. The two basic principles needed for Forensic Soundness is *Chain of Custody* and *Evidence Integrity*. This means that acquisition and processing of Digital Evidence need to be done and documented in a manner where somebody else can follow the steps documented in the forensic report, working their way from a forensic copy and reproduce the results as the former investigation. But when dealing with digital evidence, this is not always possible as it is sometimes not possible to perform acquisition or analysis without altering and potentially compromise evidence integrity. In these case, Forensic Soundness is defined by using procedures and methods that are peer-reviewed and deemed best practice by forensic experts.

The Order Of Volatility The heisenberg principle of digital data gathering and system analysis says that "*it's not simply difficult to gather all the information on a computer, it is essentially impossible*" [14]. In a digital forensic investigation, the system under investigation may change as a result of the evidence collection itself or evidence might be corrupted or even destroyed

before the investigator get the chance to acquire it. An investigator must therefore carefully plan a forensic acquisition based on incident hypothesis, taking into account which forensic artifacts are important to prove this hypothesis. This is where the principle of Order of Volatility comes in. In a computer system, changes may occur every millisecond and the analysis of one part of the system will affect other parts of the system. During a computer forensic investigation, the goal is to secure a copy of the whole system state for analysis, in practice this is not possible in most cases, therefore prioritization should me made according to where evidence is located (Table 1) [14].

Type of data	**Life Span**
Registers, Peripheral Memory, caches etc.	Nanoseconds
Main Memory	Ten Nanosecounds
Network State	Milliseconds
Running Processes	Seconds
On Disk	Minutes
External backup media, USB Pen drives etc.	Years
DVD-ROM, Printouts etc.	Decades

Table 1: Order of Volatility [14]

Crime Scene Reconstruction

Crime Scene Reconstruction is a method to determine the most probable hypothesis or sequence of events by applying scientific methods to interpret events that surround the commission of a crime. Hypotheses can be tested using statistical or logical reasoning. The process of hypothesis-driven crime scene reconstruction is presented in Figure 2

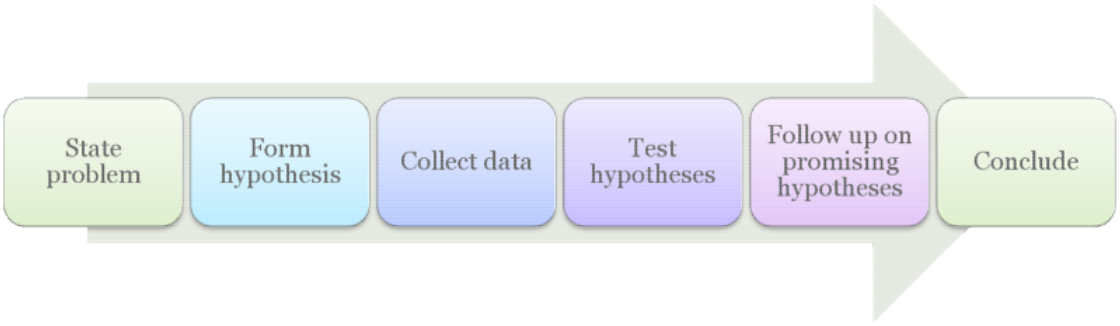


Figure 2: Hypothesis-Driven Crimescene reconstruction [2]

In the following subsections the three categories of forensic investigation are presented in chronological order of digital forensics evolution, (1) reactive digital forensics (2.1.2), proactive digital forensics (2.1.3) and active digital forensics (2.1.4).

2.1.2 Reactive Digital Forensics

Reactive is defined in the Oxford Dictionary [30] as "*acting in response to a situation rather than creating or controlling it*". Gobler et al. [25] identify Reactive Digital Forenics as:

> "*Analythical and investigative techniques used for the preservation, identification, extraction, documentation, analysis and interpretation of digital media, which is digitally stored or encoded for evidentiary, and/ or root-cause analysis and the presentation of digital evidence derived from digital sources for the purpose of facilitation or furthering the reconstruction of incidents.*" [25]

Reactive Digital Forensics, or post-mortem forensics is focused on the traditional computer and digital forensics methods. An organization can never really prevent all incidents from happening, and the nature of some incidents requires that an investigation is launched. The purposes of reactive digital forensics are to determine the root-cause of the incident, to link a perpetrator to it, to minimize the impact of it, and to successfully investigate it [25].

A forensic investigation is a systematic process of identifying whether a fact is true or false. The purpose of a criminal investigation or intrusion forensics is to identify the key elements in the given case. Investigators are encouraged to set clear objectives for the investigation. A common methodology used for this is to follow the 5WH formula. The 5WH formula consists of who, where, what, when, why, and how questions: (1) who is relevant to the case, are there witnesses, victims and suspects? (2) Where did it happened, and is there other locations that are relevant to the given case? (3) What happened, i.e. a fact-based description of the event? (4) When did it happen, i.e. the time of the event and other relevant events? (5) Why did it happen, i.e. the motive behind the crime and why the target was at the time at the location? (6) How was the offence committed? [21, 31].

To ensure a systematical approach, the digital forensics investigation process is commonly divided into five phases with separate goals (Figure 3) [21]. Each phase aims at answering the questions presented in the 5WH formula while ensuring both evidence integrity and chain of custody.

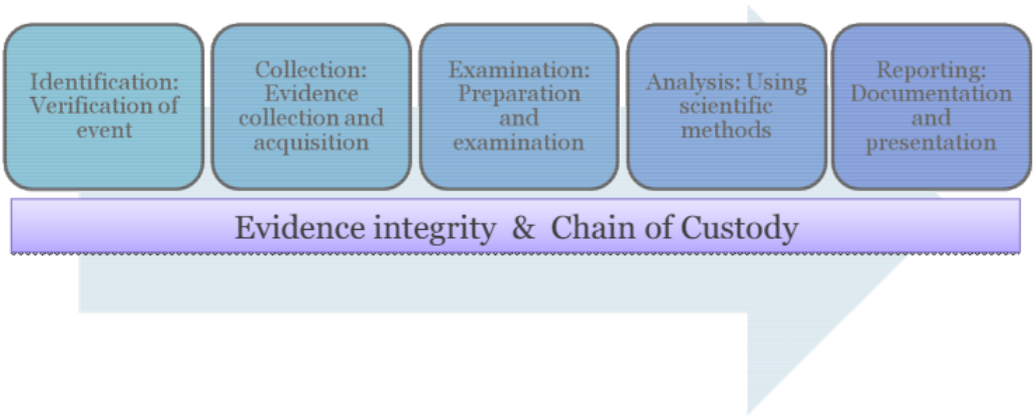


Figure 3: Digital Forensic Investigation Process [2]

Identification Phase The goal of the identification task is to detect, recognize, and determine the incident or crime to investigate.

Collection Phase The goal of the second phase is to collect data from digital devices and to make a digital copy using forensically sound methods and techniques.

Examination Phase The third phase focus on the preparation and extraction of the relevant information to retrieve potential digital evidence from collected data while protecting integrity.

Analysis Phase The fourth phase includes the processing of information that addresses the objective of the investigation with the purpose of determining the facts about an event, the significance of the evidence, and responsible person(s).

Reporting Phase Reporting phase includes sharing analysis results in the form of reports generated during the analysis phase with the interested parties, supported by actions taken and the evidence found.

2.1.3 Proactive Digital Forensics

Proactive is defined in the Oxford Dictionary [30] as "*creating or controlling a situation rather than just responding to it*". Proactive digital forensics is the preparation of an organization to ensure a successful and cost effective active or reactive digital forensic investigation with minimal disruption of business while ensuring acquisition of evidence in a forensic sound matter. The goal of proactive forensics is to make the organization digital forensics ready. This is done in the form of enhancing IT and information security governance programs and strategy to support digital forensics [25]. Proactive digital forensics is defined as:

> "*the proactive restructuring and defining of processes, procedures and technologies to create, collect, preserve and manage comprehensive digital evidence to facilitate a successful, cost effective investigation, with minimal disruption of business activities whilst demonstrating good corporate governance*" [25].

To become digital forensic ready (Forensic Readiness), an organization must maximize the usefulness of data acquired for evidence before an active or reactive digital forensic investigation while reducing the cost of performing forensic investigations [32]. Digital forensic readiness is the ability of an organization to maximize its potential to use comprehensive digital evidence while minimizing the cost of an investigation [33]. The elements for enabling forensic readiness in an organization is to have policies, routines, and processes to intrusion detection, forensics sound evidence acquisition in generall, and more specific to define how logging is being done, which logs to collect, and how to handle evidence [32]. To implement proactive digital forensics and forensic readiness, Rowlingson [33] proposes a ten step process of key activities:

1. Define the business scenarios that require digital evidence.
2. Identify available sources and different types of potential evidence.
3. Determine the evidence collection requirement.
4. Establish a capability for securely gathering legally admissible evidence to meet the requirement.
5. Establish a policy for secure storage and handling of potential evidence.
6. Ensure that monitoring is targeted to detect and deter major incidents.
7. Specify circumstances when escalation to a full formal investigation (which may use the digital evidence) should be launched.
8. Train staff in incident awareness, so that all involved actors understand their role in the digital evidence process and the legal sensitivities of evidence.
9. Document an evidence-based case describing the incident and its impact.
10. Ensure legal review to facilitate action in response to the incident.

When these ten steps are implemented, the organization should be able to

follow five phases of proactive digital forensics which starts after alert is received [34]:

Alert According to the organizations information security policy and local law, the incident response team of the organization should have a system in place to show alarms in a cataloged manner (NSM Detection 2.2.3).

Identification (Phase 1) Data should be identified in the order of volatility and priority related to the specific requirements of the organization. Cyber threat intelligence is a good source to base a collection strategy on (Covered by NSM Collection 2.2.2).

Collection (Phase 2) Collection of live data should be automated. Targeted automated evidence collection can be performed by the collection of live data on the trigger of an event that will start live data collection when certain criteria are met from different types of incident alert (Covered by NSM Collection 2.2.2).

Preservation (Phase 3) Preserving and ensuring evidence integrity by automating preservation of evidence related to the trigger (alarm) with cryptographic hashing methods.

Analysis (Phase 4) Automated live analysis of the collected evidence utilizing data mining, machine learning, or other computational forensic techniques to utilize automated hypothesis driven investigation.

Documentation (Phase 5) Automatic generation of documentation for the human analyst.

2.1.4 Active Digital Forensics

Active is defined in the Oxford Dictionary [30] as "*participating or engaged in a particular sphere or activity*". Applied to the domain of digital forensics, this means engaging actively in a collection and preservation of volatile digital evidence in live production environments (Also called live forensics). Active digital forensics is defined by Grobler et al. [35] as:

> "*the ability of an organization to gather (identify, collect, and preserve) comprehensive digital evidence in a live environment to facilitate a successful investigation*" *[35]*.

Active digital forensics is the capability of an organization to easily collect and preserve digital evidence in a live environment (production environments) so that an effective and meaningful investigation can take place in the event of an incident. The difference between digital forensics and incident response is often fuzzy, but they can be distinguished by their goals. Digital forensics focuses on the port-mortem analysis while incident response focus on handling incidents in live systems [36]. While incident response is concerned with remediating from the incident and to bring systems back to a normal state, active digital forensics is concerned with acquireing volatile evidence to support a later reactive digital forensics investigation. Active digital forensics utilizes live and remote forensics tools, techniques, and method to acquiring volatile digital evidence in a forensically sound matter. Active forensics often utilizes network forensics to identify and acquire the evidence while the incident is ongoing, the reason for this is that it enables acquiring of volatile evidence that is not possible to gather by having known proactive measures (see 2.1.3), e.g., logs, or reactive digital forensic because the evidence will simply be gone. By building active digital forensics capability an organization can (1) reduce the effect, cost, and impact of an ongoing incident, and (2) collect relevant digital evidence on live systems using proven and trusted tools and methods, which preserve evidence integrity and ensure forensic soundness. Active digital forensics provides a meaningful starting point for a successful reactive forensic investigation after the incident is over [35]. Grobler et al. [35] proposed four phases of the active digital forensic investigation process:

Incident response and confirmation What distinguishes active digital forensics from traditional incident response is that the investigator must comply with the steps in the reactive digital forensic investigation process and identify what volatile evidence must be acquired for a successful reactive investigation [35].

Active digital forensic investigation During this phase the incident responder performs evidence acquisition targeting volatile evidence using forensically

sound methods and tools. This ensures that evidence integrity and chain of custody are being preserved. Due to the nature of volatile evidence, evidence acquisition during this phase must be automated. The investigator must continuously acquire evidence, i.e. assess if all the pieces of the puzzle identified have been collected successfully. In other words, this phase overlaps with the next phase of Incident Reconstruction [35].

Limited Incident Reconstruction During this phase all the collected data are being put together to reconstruct the evidence. The goal is to make sure that the evidence is as complete as possible and all the pieces of the puzzle are in place. If there are any holes in the puzzle, the investigator should go back to the second phase and make an attempt to collect missing parts. If all possible evidence is collected, it needs to be documented in a forensically sound matter before moving to the next phase [35].

Incident closure This is the last phase which is closing the active digital forensic investigation before an reactive forensic investigation can start. An active forensic investigation is finished or completed when all possible evidence has been collected and documented and the incident declared over. The reactive forensic investigation can now take place by incorporating all the volatile evidence acquired during the incident [35].

2.1.5 Challenges in Digital Forensics

Big data and computational forensics As the amount of data being processed, transferred, and stored on digital devices is increasing rapidly, there is a significant challenge to collect this evidence, perform automated evidence analysis on it, and perform event reconstruction and timelining. The challenge of Digital Forensics investigations today is that there might be tiny pieces of evidence hidden in large complex and mostly chaotic environments. Using Big Data technologies in the domain of Digital Forensics is a current challenge and an active research topic to address forensic investigations on ever increasing large data sets. Machine learning and computational forensics are popular topics used to reduce the time complexity of evidence analysis by using pattern recognition to find links which a human analyst cannot and by providing visualization to help the human investigator to focus on the right parts of the data set [21].

Embedded systems and Bring your own device The internet of things is here; in the near future, digital forensics of fridges and coffee machines might enter the arena of digital forensics investigation. There is a significant challenge in performing digital investigations of mobile and embedded devices since they are often proprietary and closed, and both the software and hardware are device specific, making forensic acquisition hard. In addition, the data on the devices themselves is often in a binary format, that require reverse engineering, or data is even encrypted as a result of the Snowden leaks [1]. A good example of this is the FBI's investigation of the San Bernardino bombings [2] [21].

Cloud and Internet Forensics As cloud technologies and services become more widely adapted, new evidence acquisition techniques, legal frameworks, and methods are required. Regardless of private or public cloud services, the cloud domain provides significant challenges when it comes to digital forensics and incident response [21].

Anti-Forenics Hackers and criminals advance their technologies to remain hidden on compromised computer systems or to hide in plain sight on the network. There is an increase in the use of methods of obfuscation and encryption to subvert any forensic investigation. Often, the malware, which was placed in the system, leaves minimal traces because it runs in the memory only and leaves a scarce footprint in the filesystem, making a reactive forensic investigation unsuccessful.

[1] http://www.nytimes.com/2014/09/27/technology/iphone-locks-out-the-nsa-signaling-a-post-snowde
html

[2] http://www.reuters.com/article/us-california-shooting-san-bernardino-idUSKCN0VR2I1

2.2 Network Security Monitoring

> The rising of birds shows an ambush.
>
> Sun Tzu, "The Art of War"

Due to the scope of the thesis, this section presents only detection and response, while analysis is discussed in the proposed approach that utilize multisensor data fusion for detection and respond.

2.2.1 Introduction to Network Security Monitoring

Network Security Monitoring (NSM) is a key piece of Computer Network Defense (CND) which is a sub category of Computer Network Operations and is the opposite of Computer Network Attack (CNA). NSM can be viewed as a three step loop consisting of collection, detection, and analysis of network security data to address the four key elements in CND which are:

Protect To protect the network by focusing on securing systems and to prevent exploitation and intrusion from occurring by hardening network and computer systems, vulnerability scanning and vulnerability management, risk assessments and risk management [3].

Detect To detect threats towards the network by focusing on detecting intrusions that are currently active or intrusions that were successful in the past, by monitoring systems, sensing attacks and issue alarms and warnings [3].

Respond To respond to threats in the network by focusing on responding to intrusions, isolating compromised assets, performing host and network forensics, malware analysis and reporting [3].

Sustain To sustain the operational capabilities of CND by focusing on managing people, processes, and technologies in the forms of capability development, systems implementation, staffing, policies development, and routines writing [3].

NSM focus on the collection of data that describes the network environment to the greatest extent possible, providing incident responders, security professionals, and forensic analysts with the background data for responding, understanding, recovering, and protecting assets of an organization from security breaches. By collecting relevant information to the extent of technology and policy, the likelihood of intrusion detection rises significantly as well as an understanding of intrusion by the analyst [37]. Network Security Monitoring is all about Indicators (2.3.1) and Warnings. The use of strategic (2.3.3) monitoring of threats

against the network environment, NSM aims to, based on indicators and warnings, assist in *detection* and *validation* of intrusions [37]. Network security monitoring is built up by three key elements, arranged in a cyclic process (Figure 4)—collection, detection, and analysis. After analysis is completed, the need to collect new events might be identified and the cycle starts over by defining a new collection strategy to support detection and analysis. Each of the elements will be described in the next three subsections.

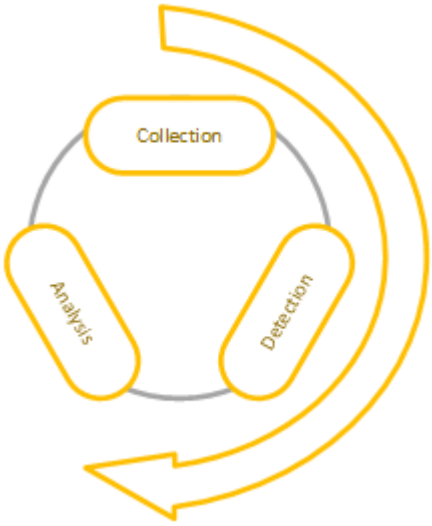


Figure 4: Network Security Monitoring Cycle, based on [3]

2.2.2 Collection phase

Network Security Monitoring begins with the hardest, labour-intensive, and important step,— collection of data from different sensors. Collection is being done by a combination of hardware and software that are used to generate, transfer, process, and store data for NSM detection and analysis. Collection is the most crucial part of the NSM cycle, i.e. it is extremely important to do it right, because the quality of collection will greatly affect the organization's performance, ability, and chance of success in intrusion detection and analysis [3].

NSM data can be collected in various ways and from many locations in the ICT infrastructure. The most commonly collected NSM data is full packet capture data, packet string data, session data, statistical data, log data, alert data, and meta data [3]. These types are described in detail in the following sub-sections.

Full Packet Capture Data

Full Packet Capture is a method to collect every network packet between two points in the network. Full Packet Capture Data can require quite a lot of storage and computing power to process due to its completeness, but even though the cost of Full Packet Capture Data is high, the complete detailed view into the

network communication is of a very high value for providing analytical context. Full Packet Capture Data is especially valuable in network forensics investigations and can be compared to a record from a surveillance camera and serve as evidence that a crime or an event did or did not occur. It is also very useful for analyzing false positive alarms as the network traffic can be replayed through detection tools as a whole, finding and verifying exactly what triggered an alarm. Ultimately, if an attacker accessed a system over the network, there will be evidence of it in the Full Packet Capture Data, even though no intrusion detection systems are triggered or logs on the compromised system were deleted [3].

Planning Full Packet Capture when deploying sensors is important for several reasons. By collecting FPCD, one can generate/extract almost all other NSM Data. FPCD can be made a primary data type and the root for all network-based detection and analysis. Since the level of detail provides high analytical value, Full Packet Capture Data can be replayed through detection tools such that new detection logic can be written and tested in a safe environment. Moreover, Full Packet Capture Data can be replayed through detection tools to verify and get an exact analysis of why the detection logic triggered [3].

FPCD collection will require a lot of storage compared to other types of collection. An organization should therefore define the minimum acceptable amount of Full Packet Capture Data needed for delivering an Network Security Monitoring Service and the operational ideal which is a reasonable collection goal. In general, there are two retention policies for storage consideration, and the choice depends on the organization and budget [3]:

- *Time-Based Retention Policy* should be chosen when the collection is required to be stored for at least a some period of time. This can be 24 hours, a couple of days, or some weeks at most. This approach is normal in industries with compliance requirements.
- *Size-Based Retention Policy* should be chosen when the collection is based on how much storage space that is available or wanted. For example, if 5 TB of packet capture data is saved in a rolling window, then a 5TB window is always available to look at. This approach is more common among companies that do not have compliance requirements but have limited resources for storage.

Packet String Data

Packet String Data (or Transaction Data) is extracted from Full Packet Capture Data and is generally defined by how it is used. Packet String Data is basically a selection of human-readable data of analytical value extracted from Full Packet Capture Data or Network Packets on the wire. The role of Packet String Data is to be an intermediate data form intersecting Full Packet Capture Data and Session Data. Session Data lacks the granularity to ascertain detailed information about what is going on or what has occurred in the network at a certain time

interval, while Full Packet Capture Data is just too expensive to store over a longer period of time. Having a limited store of Full Packet Capture Data and only Session data makes a reactive forensic analysis of the state of the network in a past time interval less effective or even impossible. To enable a reactive forensic investigation, Packet String Data can be extracted and stored longer than Full Packet Capture Data; they provide fairly detailed context and complement the Session Data. Storage of only Session Data without Full Packet Capturing Data can limit an analyst who will not be able (1) to identify in retrospect and verify a unique HTTP user agent associated with an attacker (2) to identify a SSL Certificate encrypting communication of a newly discovered C2 malware, (3) to determine the extent of how many users clicked on a link in a phishing e-mail, (4) to determine if a certain file was downloaded, (5) to determine if a specific virtual website or URL was accessed [3, 38].

There are two approaches to Packet String Data collection, to extract it from Full Packet Capture Data or to derive it directly from the wire of a monitoring port on a NSM sensor. Regardless of approach, it is important that all NSM data is collected from the same source to limit correlation errors or multiple processing of the same data. Before collecting Packet String Data, a few issues must be taken into account. First, the extent of the collection must be considered, i.e. what Packet String Data is relevant for security, incident response, and reactive forensic investigations. The goal should be to collect as much essential plain-text application protocol data as possible, while storing encrypted protocol data-fields for as long as possible. Second, the time period, Packet String Data has to be stored, should be chosen cautiously, i.e. it should be somewhere in between storage time for Full Packet Capture Data (ranges from hours to days) and Session Data (from months to a couple of years). Third, it should be taken into account that Packet String Data collection varies in a storage need during the day, and spikes in collection may appear [3].

Session Data

Session Data, also called a Flow, provides information about the communication between two network devices. Flow data is one of the most flexible and useful forms of Network Security Monitoring data since it can be used to prove that communication found place, to identify how much data was sent, and to identify which of OSI Layer 3 and Layer 4 addresses and port were used. Session Data is simply a summary of the connection, and therefore does not provide the same level of detail as FPCD does. Nevertheless, session data uses less storage space and can be stored for longer as a record of all network transactions that is very valuable for network forensics [3]. By collecting Session Data like NetFlow or Flow, the events seen from other perspectives of the network can be glued together to provide the thorough picture.

Flow records usually include, and are defined by five attributes: the Protocol,

Source IP address, Source Port, Destination IP address, and Destination Port. In addition,time-stamps for communication start and termination will be available along with logs for protocol flags that have been set and for number of packets and bytes sent in the flow. A flow record is only terminated when the flow hits one of three states. The first one is *Natural Timeout*, when the communications has ended naturally caused by the protocol in use. In connection-oriented protocols like TCP, this will be done by a RST or FIN sequence. The second is *Idle Timeout*, which normally happens when no new packets in the flow have been received in the last 30 seconds (time can be configured) after the last packet was sent. If communication continues after Idle Timeout has been reached, a new flow record will be created. The third is *Active Timeout*, which happens when a flow is terminated after being active for 30 minutes (can be configured) and a new flow will be created [3].

The collection of Session Data requires a Flow generator and a flow collector. There are only two approaches to generate Session Data [3]. The first one is *Hardware Generation*, that can be used by enabling Flow generation on a router or Layer 3 network device and point it to the IP address of the collector which receives and stores the flows on the behalf of a network device. The benefit of this approach is that the network router has the possibility to see all flows that run through it that provides a good view in the the network traffic. The disadvantage of using the network router for Session Data generation is the extra CPU needed to track flows. If the router already has heavy traffic load, enabling Session Data generation may seriously degrade a network performance. Another disadvantage is router is required on every place in the network where Session Data collection is wanted. The second approach is *Software Generation*, which is the most common practice in Network Security Monitoring because a software for generating Session Data comes with advantages over hardware generation. The first advantage is the flexibility, i.e. the software can be installed on a sensor and placed in strategic places in the network. The second advantage is the ability of the sensor to passively monitor the network with no degradation of network performance and no risk of outage since the traffic is not flowing through the sensor, it only gets a copy of it. Common software for Session Data generation is Fprobe, Yet Another Flowmeter YAF, and Suricata [3].

Statistical Data

Statistical Data is derived from the collection, organization, analysis, interpretation, and presentation of existing data [39]. In Network Security Monitoring, statistical data can take several different forms and tell different stories about the Network Security Monitoring data that has been collected, derived, and produced. Statistical Data can play a vital role in detection and analysis when large amount of data is collected and stored. Statistical Data is usually presented to an analyst in form of a dashboard, i.e. a looking glass into the current state of

the network. It provides situational awareness and displays statistical anomalies which should be further investigated. Statistical Data can help to identify positive or negative relationships between two data entities over time [3].

Log Data

Log Data is the intrinsic meaning that a log message has. A log message is what a computer system, device, software, or application generates in response to some sort of stimuli. A log message is basically built by three sections, a timestamp, a Source of the message, and a Log Data [40]. In general, Log Messages can be classified into five categories based on the importance of the message [40]:

- *Informational* log messages are designed to let administrators and users know that some benign event has occurred in the system. But even though events of this kind are considered benign, it is possible to detect anomalies by adding a context to such events.
- *Debug* log messages are designed to provide software developers and system administrators with information about the internal states of a piece of software or hardware so that problems can be identified and troubleshooted [40].
- *Warning* log messages are designed to notify system administrators about problems in the system which are not severe enough to affect system operation [40].
- *Error* log messages inform system administrators that something is wrong somewhere in the system and it is negatively affecting the operation [40].
- *Alert* log messages are notifying the administrator that something interesting (often related to security-related log messages) has happened in the system [40].

Log data can include Web server logs, Application logs, router firewall logs, or system logs. Examples of log sources are SYSLOG Deamon and Microsoft EventLog. The analytical value of logs depends on where the log events are collected from and what kind of information they contain. The storage space and computational requirements can vary a lot depending on the content of the log source [3].

Logs messages can contain a log of information about various domains withing a computer system or network. It can inform an administrator about performance issues, security issues, logic problems in a system or application, etc. The following list covers the whole spectrum of security, operational, and debugging log data [40]:

- *Change logging* is a record of system changes, component changes, updates, and account changes. These logs are usually split into add, delete, update, or modify operations on system objects. Changes can be important and of relevance for both operations and security, often overlapping.

- *Authentication and Authorization logging* is a record of decisions regarding authentication and authorization of subjects on objects in the system. The most common form of these logs are successful and failed logins, and the use of privileges. These logs are mostly for security purpose, but might be used operationally to track usage of systems or services.
- *Data System Access logging* are messages related to Authentication logs, but has focus on logging access to application components and data, like a database or web application.
- Alert logging are generated from traditional intrusion detection tools about other devices and activities that violate the security policy, like firewalls or anti-malware.
- Performance logging is a broad category of messages related to a system and application performance, including thresholds, memory, and computing capability or other finite resource utilization. These messages are mostly operational but can be used for security as well.
- Availability logging are log messages that tell about the operational state of the system, such as reboots, shutdowns, the availability of backups, service availability, and Disk RAID status. These kind of logs are rarely used for security and forensics.
- Miscellaneous errors and failure logging are all other types of system or application errors that do not threat the availability or stability of the system, but operate as a warning for the system administrator.
- Miscellaneous Debug logging: Debug logging is a tool for developers and should not be enabled on production environments.

In Network Security Monitoring and forensics, good log data is essential. Different systems, applications, and services log differently and the quality of log data is important in order to get the benefits from logging without consuming unreasonably storage and processing to manage and use the logs. In general, when designing a strategy for what to log or do not log, it is essential that the collected log messages answer the five questions—What happened? When did it happen? Where did it happen? Who/What was involved? Where he, she, or it came from? [40]

Another important decision in a log collection strategy is whether a central log collection is needed. Since Log Data is being generated on the different devices in the network, to retrieve them and perform analysis for NSM from each of these devices is a time consuming job. Log messages need to be collected in a central location, often a server called a Loghost. Chuvakin et al. [40] points to three advantages of collecting Log messages on a central server: (1) it is one centralized place to store log messages from multiple locations, (2) it is one place to store backup copies of the logs, and (3) it is one place where analysis can be performed on the log data.

Some of the most common protocols for sending and receiving Log Data are

[40]:

- *Syslog*[3] is the most common protocol for storing and transmitting log data nowadays. All *nix flavours and most networking equipment support this today. Syslog is a UDP-based (TCP Implementations do also exist) client/server protocol.
- *SNMP*[4],Simple Network Management Protocol, was originally designed for the use in administrating networking devices. However, over the years it has been adopted by many non-networked systems as a mechanism for sending and receiving log message and status data.
- *Windows Event Log* is a Microsoft's proprietary logging format. It is possible to forward events to a central server by using Windows Event Forwarding or using several third party tools to convert the local Windows Event Log to syslog, like NXlog[5].
- *Database* is also commonly used as a structured way to store and retrieve log messages, especially in applications.

Alert Data

Alert Data is produced by a tool that discovers an anomaly within any of the data it is configured to analyze; the notification or log entry generated is called an Alert. This data usually contains a description of the alert combined with pointers to the data that triggered an alert. Alert data is usually very small and can be retained for a very long time, as it points to other data. Alert data is usually the trigger for analysis of other Network Security Monitoring data [3].

Metadata

Metadata is simply data about data that helps to bring a context and meaning for human analysts to the collected data. Metadata is generated by using tools to understand better the data that has been collected. Examples of Metadata are the WHOIS record for an IP-address and Cyber Threat Intelligence collected from either third parties or generated in house [38].

2.2.3 Detection phase

Detection is a part of the NSM Cycle where collected data is being examined and alerted upon when suspicious or unexpected data is being discovered. Intrusion detection is the process of monitoring the event that occurred in a computer system or a network for indicators of security breaches [41]. The Detection phase of Network Security Monitoring is all about knowing one's detection capabilities, understanding threats and the adversary Tactics, Techniques, and Procedures (TTP's 2.3.4), to apply this knowledge to detection mechanisms, which goes beyond the traditional Intrusion Detection System. NSM Detection is typically done

[3] https://tools.ietf.org/rfc/rfc5424.txt
[4] https://tools.ietf.org/rfc/rfc1157.txt
[5] http://nxlog-ce.sourceforge.net/

through some form of rule-, anomali-, or statistical-based detection which results in generation of alert data (2.2.2). Traditional intrusion detection systems are modeled after four stages—data source, which can be a packet stream or log; data pre-processing, where data are being normalized; detection algorithm, where indicators of compromise are being searched for; and an alert filter, to provide alerts of positive hits. NSM Detection goes beyond traditional intrusion detection systems because it evolves around cognitive detection as well. Traditional intrusion systems do not work on a cognitive level, that is one of their biggest weaknesses [3].

Detection methodology is often classified into two main categories, misuse-based detection and anomaly-based detection. However, there is also a third method described in literature, specification-based detection. It is also common to utilize more than one detection methodology in the same intrusion detection technology; this is called Hybrid detection.

Signature-Based Detection is the oldest form of intrusion detection and is one of the most common ones. A signature is a pattern or a string that corresponds to a known attack or threat. Signature-Based detection is the process to compare patterns against captured events to recognize possible intrusions. Due to the use of the knowledge accumulated by specific attacks and system vulnerabilities, signature-based detection is also known as Knowledge-based detection or misuse-based detection. The advantages of signature-based detection are that it is the simplest and most effective method to detect known attacks and it provides detailed contextual analysis. The disadvantages of using signature-based detection are that it is ineffective against unknown attacks, evasion attacks (deception), and is easy to trick with variants of known attacks (denial). Signature-based attacks also have little understanding of states and protocols. It takes a lot of work to keep the signatures relevant and up to data, and it is time consuming to maintain the knowledge (Situational Awareness) [42].

Anomaly-Based Detection is detection of things that deviate from its normal state. Anomaly-based detection is when the network or a system is monitored and profiled over a period of time so that the normal state of the system or network is learned. These profiles can be either static or dynamic and can be developed for many types of data, like network flows or user log-in attempts. After the profile has been generated, the detection algorithm compares the events seen in the normal state, defined in the profile, with an event (or series of events) which are outside of a given threshold, then an alarm event is being produced. Sometimes, anomaly-based detection is called a behavior-based detection. The advantages of anomaly-based detection are (1) the ability to detect unknown attacks, (2) being less dependent on knowing the technology behind the events, and (3) ability to

facilitate the detection of privileges escalation attempts by, for example, identifying users logging into odd hours from uncommon places. On the other hand, the accuracy of the profile is often degraded if the state of the network changes constantly. Moreover, the detection is unavailable during profile rebuilds, and it is difficult to provide alert data within acceptable time [42].

Specification-Based Detection is when the intrusion detection system knows and can trace the protocol states. Specification-Based detection is also known as Stateful Protocol Analysis detection. Specification-Based detection may look similar to Anomaly-Based detection as it also compares the events against profiles. However, Specification-Based detection uses technical implementation specifications from vendors or protocol standards to define the normality compared to Anomaly-based detection that looks at the events from a network or host. The advantages of a specification-based detection is that it knows how the protocol state-machines works, and can detect deviation from this. This means detection of unexpected series of commands which have useful application in, for example, SCADA systems. The disadvantages of Specification-based detection are the resource consumption needed to analyze all commands/instructions run on the target system or network. Specification-based detection is also ineffective against attacks that look like benign protocol behavior and it might not be a platform independent [42].

Hybrid Detection is when multiple detection methodologies are used to complement each other. This approach is the most common today [42].

The Holy Grail for IDS vendors is 100 accurate intrusion detection. In other words, every alert corresponds to an actual intrusion by a malicious party. Unfortunately, this might never happen. One of the reasons is that IDS products lack a context. Context is the ability to understand the nature of an event with respect to all other aspects of an organization's environment [37].

The detection methodologies described above are related to the detection algorithm of the intrusion detection system, they are generalized and might be used in most perspectives in a ICT infrastructure. In the next sub-section, a brief explanation of where the detection can be deployed is presented.

Scope of Detection

Intrusion Detection technology is often categorized based on the scope or perspective from where it detects intrusions. Three well established categories can be outlined:

NIDS Network-based Intrusion Detection systems work by monitoring the raw network traffic on the wire and searching for patterns of intrusion in the packets that fly by.

HIDS Host-based Intrusion Detection systems monitor events on the local computer installed on for sings of intrusion. HIDS is often combined with Target-Based Intrusion detection for, for example, checking integrity of the files in the file-system, or specific files and folder like C:
Windows or /bin.

AppIDS Application-based Intrusion Detection systems are wired into an application. They monitor the internal states and events of the application for signs of intrusion.

2.2.4 Analysis phase

The final stage of the NSM Cycle is Analysis. This is where a human is being involved. A human analyst interprets the information from the detection stage to make a decision whether the warning is a real intrusion or a false positive alarm. This step often involves gathering information and investigative data from other sources, researching Open Source Intelligence related to the generated alert, and looking into the detection logic that produced the alert. Analysis is often the most time-consuming step in the cycle and might trigger the following tasks: network packet analysis, network forensics, host forensics, and malware analysis [3].

Figure 5 shows NSM Tools and their application for detecting anomalies, infected hosts and C2 traffic. Green means that the tool is detecting the threat, while grey means that it does not. As seen on the figure, some tools have better detection for some problem then others.

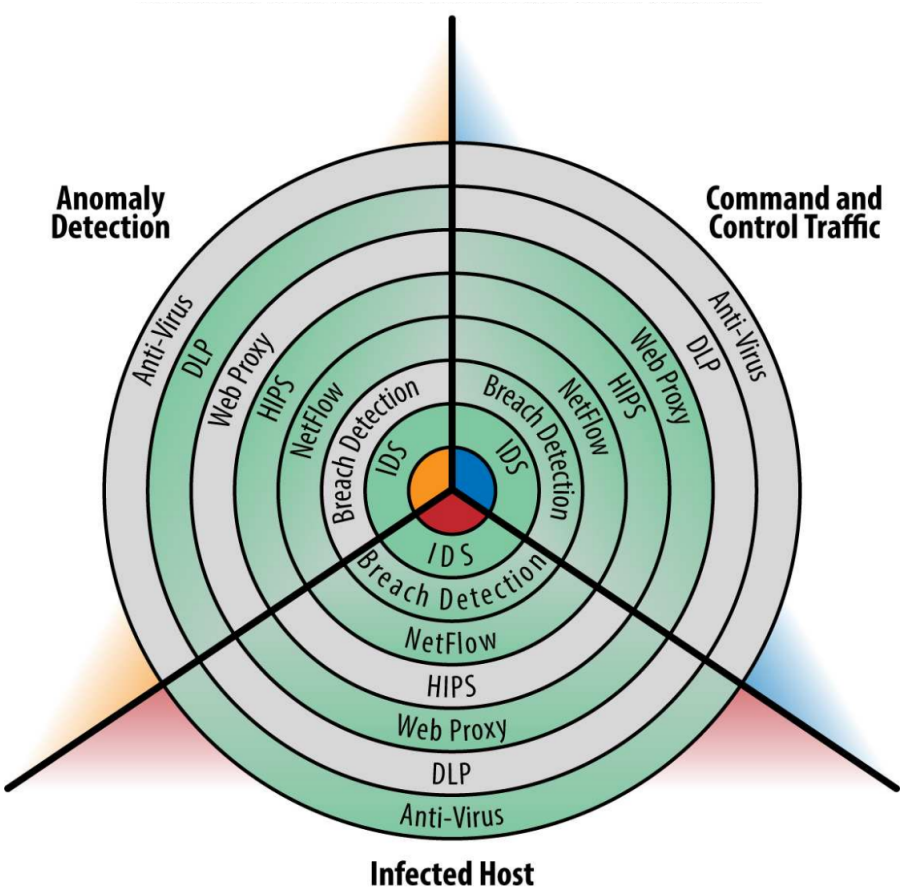


Figure 5: What can each NSM tool detect? (Figure 7-6 [4])

The cyber kill-chain (2.3.1) is commonly used for modeling NSM analysis today as the model provides detection perspectives is derived from intelligence-driven CND. Malicious activity rarely provides a single set of behaviour to detect, and malicious activity often stands out. Malicious activity is often a set of moving activities as threat actors find new innovative means to reach their malicious goals and even though the complexities of attacks might be high due to the complex nature of computer networks, analysis and detection should find as simple means as possible to build detection logic upon. Each of the stages of the attack can be used for detection of unusual or malicious activity. Attacks are rarely so complex that trying to find IoC's in the whole chain is the best approach or a good detection and analysis strategy. Instead, the most reliable of the easiest links in the chains should be the focus, e.g., (1) if a malware author include sophisticated anti-forensics techniques like polymorphism, encryption and packaging of the malware itself, analyzing C2 traffic would be simpler and still detect the compromised asset; (2) if C2 traffic is hard to detect and hiding in plain

sight, then detect on other type of indicators like DNS and IP addresses; (3) if more than one weak link in the kill-chain can be analyzed and detected, confidence level can be increased by combining these by looking if they occur at the same time; and (4) if they can not be tight together, detect on both separately with less confidence [4]. When using the cyber kill-chain as a model for analysis, Reconnaissance activity can be detected by monitoring and analyzing unusual connections and probes, to and from the internal network. Contact from external clients to web applications can also provide early warnings. If clients contact resources of known phishing campaigns, warning of coming infections could be seen analysis. Weaponization can first be analyzed after first delivery and is most often based on malware analysis. The CTI IoC's extracted can be used to find valuable atomic and computed indicators in order to detect elements in other parts of the chain. Delivery can be detected and analyzed by NIDS, HIDS, EPP or EDR. Exploit attempts can be detected by EPP, HIPS and Installation of backdoor's can often be observes either by EPP and HIDS, or by mining system logs by the means of behaviour analysis. Command and control can be picked up by NIDS, firewalls and session data analysis while Action on objectives (I.e. exfiltration) can be observed either by session data, DLP tools or similar.

FPCD Analysis

Because of the high detailed view of packet capture data, in a perfect work one would have FPC everywhere. In smaller environments this capability feasible. FPCD provide comprehensive and volatile digital evidence for investigations in the detailed level where analysis would be able to reproduce network based attacks by replaying the network communication for analysis and IDS rule development. Packet captures provide great analytical value in this cases when it is desirable to understand exactly what happened. Even though it is hard to engineer a good FPC solution, but filtering, storage, indexing and recall possibilities must be considered. For FPC to work and provide value to NSM Analysis, filters for removing unreadable and unusable data must be in place. FPC is the right choice if the target of the collection and analysis is to dive into packet payload data, SSL, IPSec and other encrypted data make FPCD useless if the cryptographic keys are not avaiable and can be discarded. If the goal of analysis is to just extract packet headers, packet string analysis is a better suited choice. If network metadata is the goal, then session data capture is a better option. Broadcast, Multicast and other chatty protocols can also be discared as they provide little analytical value [4].

Packet string analysis

Packet string analysis is a cheaper alternative to FPC where a subset of features is extracted from the packets. Common packet analysis task would be to look at HTTP headers, TLS headers or DNS traffic but it is not limited to just these. From HTTP headers, HTTP host, user-agent, file-type etc. is interesting. I.e. if

a waterhole website is hosted in a web-hotel it is like other websites on the same server as virtual hosts. The IP address of all websites on the server will be the same, but there is only one website that is malicious. To identify which one, HTTP Host can be pulled from the header in order to extract the hostname of the malicious website [4]. Other use cases is performin passive DNS analysis, where the DNS queries and answers are being logged and can be used for statistical analysis. It is also common to use DNS as C2 protocol, sending commands over TXT records.

Session analysis

Unlike other NSM Data, Session data is completely content free metadata about communications on the network. Session analysis is used in almost all investigation to create timelines, identify movement of a threat, provide context to encrypted traffic or to map out communication behaviour or a client in a given timeframe. Session data anlysis works best for analyzing and detecting threats when the content of the traffic is irrelevant for detection. Simple netflow provide information about communication that are taking place now, or happened in the past. Netflow analysis is great for detecting if any client in the network communicated with something malicious in the pas based upon IP address or network address. So even though intrusion detection systems did not pick it up, it can serve as a flight recorder for going back [4].

Statistical analysis

Statistical analysis is a tool and method generating CTI of NSM data in ways that is less obvious than searching for and matching atomic IoC's Statistical analysis can support in finding outliers and patterns in collected data, providing a generalized view of what is happening on the network . Statistic can be used on all levels of data, and can be isolated to single hosts in the network for greater resolution, or work on all the data collected. Statistical analysis is an important tool for situation assessment and in order to uncover the extent of an attack. A use case i.e can be looking at session data (netflow) to detect UDP amplification attacks by performing statistical analysis to detect a sudden positive deviation in total packet count, disproportionate number of UDP server response bytes compared to what the client sends, a positive deviation of the total amount of UDP traffic on the network, and any of the mentioned in correlation with vulnerable UDP protocols like DNS, chargen and SNMP [4]. Statistical data can be the aggregated information of all sessions to provide top talkers on the network, or more interesting, the connections that are happening regular but not so often.

Log analysis

Log aggregation is useful for other purposes than just centralized management of logs. Centralizing log management provide a single place to search logs that span multiple hosts and endpoints in the network. Because of this, mining logs

to provide much richer context for security analysis is possible. I.e. by looking at a single failed login on one host meaning a user forgot his password, but seeing the same user and client with one failed login on all the servers in the network provide whole different story that should be investigated. This kind of anomaly can be picket up quite easy when all the logs are at the same place [43].

Alert analysis

NSM Analysis can be more effective if access to good and actionable CTI is available. Good CTI can increase accuracy of NSM Detection significantly. But Operational CTI which is highly specific and often based on atomic indicators can be malicious this week, but benign next week after. The threat landscape is constantly shifting meaning that CTI must be constantly update in order to provide value to NSM Analysis in order to remain effective [4]

Metadata analysis

SIEM's rely primarily on reputation metadata (Operational CTI) for alerting and prioritization. Detecting based on this metadata is often prone to false positives if not maintained over times as reputation data changes over time. In commercial SIEMs this often happens in a black box environment providing little to nothing about how the analysis is being done. Without intimate knowledge about how metadata is being produces in NSM analysis, an analyst can not estimate the fidelity of the metadata used for detection [4].

Log entries by them self have very little meaning, it is the context that is build up by aggregating and correlating log entries that provide a context and meaning to what the logs represent. Context from NSM data is derived by organizing and sorting raw data into metadata, or groups of metadata, and then apply analysis and context (knowledge) to establish their meaning. Metadata is a collection of features that describes behaviour of an incident. When it is high volume of NSM data, reducing to, and searching with metadata yields understandable and digestible amounts of information. To be able to sort out and cut through the most valuable information efficiently provide more efficient NSM analysis [4].

Human analyst

Even though, in a perfect world where detection and response is fully automated, at some point a human must be involved in the loop to take action. Human response can be validating the alarm, working the incident case, contacting a user or remediate the compromised system from the bad state. It is unlikely that one could ever fully remove the human analyst from the loop of an security incident, because the adversary on the other end will be a human. To reach high NSM analysis efficiency, continuous development, threat hunting, investigating and tweaking detection logic and sensors to detect malicious behavior is a constant evolving task. No computer or software can replace the analytical cognitive capacity of a context-informed human analyst in any foreseeable future. The

reason for this is that compared to computers flowchart-style logic and decision making, human analyst can reason and construct conditional indicators, analyze and establish motivation and other human idiosyncrasies [4].

2.2.5 Challenges in Network Security Monitoring

Challenges in Network Security Monitoring

Network Security Monitoring is a relative new discipline under the umbrella of Information and ICT Security. NSM is a paradigm shift when Threat-Centric Security is replacing Risk-Centric Security practices. NSM is currently a relatively immature scientific discipline with a need for more research since there is a wide gap between what is written about it and what is practically implemented in real infrastructure [3]. The main challenges in NSM are:

Defined common language There is the lack of common language and methods used by practitioners and researchers. There is no sufficient amount of open research within the discipline itself, but there is a good degree of publications regarding sub-disciplines to NSM like Intrusion Detection [3].

Mobile Devices Mobile devices that never cross a network segment, which is being monitored by the sensors, will therefore create a blind-spot [38]

Cloud Cloud services can be out of the network security monitoring perspective, making parts of the organization's assets unprotected.

Cost of implementation Network security monitoring aims to collect as much data as policy and technology allow in order to describe the network environment in the greatest detail possible. This is data intensive due to a need in a lot of storage and computational resources for detection and analysis. One of the biggest reasons for low adaptation nowadays is that NSM requires a Big Data and is expensive to implement [38]

Skilled analysts Like with computer and information security in general, network security monitoring is lacking skilled specialists [3]. Since NSM is a multi- disciplinary domain, it requires a broad set of deep skill-sets and understanding. Moreover, NSM is an emerging discipline, so it also thrives on the ingenuity and creativity of the people working with it, whether it is a Data Scientist, Security analyst, or someone else.

Obfuscated and encrypted traffic Network security monitoring faces the same issues as many other network security related technologies, i.e. obfuscated Layer 3 information using virtual private networks or encrypted communication are degrading analysis capability and eating up storage and processing power.

Network architecture Network architecture and ICT infrastructure are not designed or suited for NSM.

Privacy issues Network security monitoring can be very intrusive in terms of the vast amount of individual identifiable information being collected. Because of this, collection of traffic and events needed for an effective NSM solution is not sufficient, that can, in turn, degrade detection and analysis of performance [38].

Challenges in Intrusion Detection

Intrusion detection is a much older discipline than NSM and several methods and approaches were published over the years. The holy grail of intrusion detection research is to detect 100% of all threats and that 100% of all alarms are true positive. This goal has yet to be achieved. The following measures of intrusion detection systems are presented in literature [44, 45]:

Accuracy The degree level that an intrusion detection system produce True Positive alarms.

Completeness Incompleteness occurs when an intrusion detection system fails to detect an attack (False Negative).

Performance A rate at which an IDS can process the input data and produce alarms. If the IDS performs poorly, real-time detection is impossible.

Fault Tolerance The IDSs itself must be resistant towards the attacks and not let itself be subverted by an attacker. There can be performance related attacks, e.g., a DDoS, or algorithmic based attacks where an attacker manages to crash the intrusion detection engine itself.

Timeliness The intrusion detection system must provide an analyst with its decision (alarm) within a timely manner so that the correct measures can be taken to thwart the attack before extensive damage has been done.

2.3 Cyber Threat Intelligence

> If you know the enemy and know yourself, you need not fear the result of a hundred battles
>
> Sun Tzu, "The Art of War"

In this section the concept of threat intelligence is described, i.e. definition of Cyber Threat Intelligence, sources of Cyber Threat Intelligence, and how it is performed.

2.3.1 Introduction to Cyber Threat Intelligence

In traditional military sense of the word, US Department of Defence (DoD) defines Intelligence as

> "*the product resulting from the collection, processing, integration, evaluation, analysis, and interpretation of available information concerning foreign nations, hostile or potentially hostile forces or elements, or areas of actual or potential operations*" [46].

To understand what this means, Figure 6 shows how operational environment is observed, collected, and turned into data, which, when processed and added enriched with a context, are transformed into information. Information is then turned into intelligence when knowledge is added to the context (See Intelligence Cycle 2.5.2).

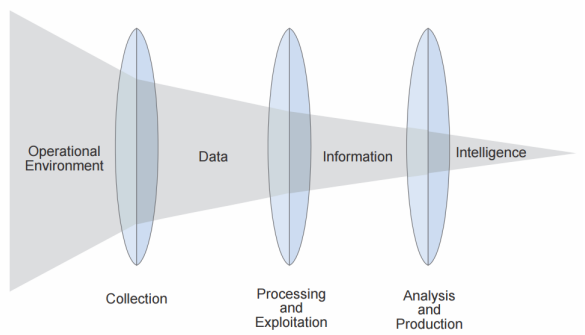


Figure 6: Relationship of Data, Information and Intelligence [5]

The same principle applies to cyber threat intelligence, which is also intelligence in the DoD definition; cyber threat intelligence is used by the military as well as private industry. The key difference is the context and knowledge of the analyst producing the intelligence, and the goal to produce the intelligence. The DoD's definition will suffice for Cyber Threat Intelligence as well since CTI

is just another form of intelligence in military sense, closely related to signals intelligence and electronic warfare. But what defines "cyber threat intelligence"? This is not completely agreed upon within the academic community, but many practitioners from industry and scholars refer to Rob McMillan [47] in Gartner's definition:

> "*Evidence-based knowledge, including context, mechanisms, indicators, implications, and actionable advice, about an existing or emerging menace or hazard of assets that can be used to inform decisions regarding the subject's response to that menace or hazard*" *[47]*.

McMillan's definition of threat intelligence is well applicable to the cyber domain. As with all intelligence, context and knowledge are important, without it, it is just data and information. Cyber threat intelligence has the goal to be actionable, i.e. security teams may use it to actively protect their constituency. For cyber threat intelligence to be actionable, it must be accurate, timely, and relevant to the current goals [48].

Friedman et al. state that cyber threat intelligence is "knowledge about adversaries and their motivations, intentions, and methods that is collected, analyzed, and disseminated in ways that help security and business staff at all levels protect the critical assets of the enterprise" [48]. In other words, cyber threat intelligence is the evidence-based knowledge resulting from the collection, processing, integration, evaluation, analysis, and interpretation of available information concerning a threat that provides motivation, capabilities, and methods to produce actionable advice.

By using threat intelligence as a part of the security operations, and combining both internal and external intelligence, a security team can detect more attacks earlier in the cyber-kill chain. This improves the effectiveness and security team's ability to fulfill its mission of protecting the organization assets. By utilizing threat intelligence, an effective security team may help providing relevant information to the right people at the right time, giving them the time and chance to take appropriate actions to stop an attack and remediate back to normal operations in a timely manner [6]. Cyber threat intelligence can provide great value to an organization in the forms of:

Breach identifications, by being able to search for indicators of compromise and discover breaches that have happened in the past. It is implemented by using threat intelligence indicators to for proactive protective monitoring of network and systems to trigger responses [49].

Breach prevention, by identifying new threats which provide operational space for proactive measures by understanding threat actors targets, tactics, procedures, and capabilities, and then address these in implementation of defensive measures [49].

Fraud and theft minimization, by identifying causes, actors, and threats as well as potential targets for implementing preventive measures to minimize potential damage [49].

Asset protection and risk minimization, by identifying assets in need of changing protection mechanisms to address new risks from threat actors. This by getting insight into detection and protection strategies [49].

User protection and risk minimization, by identifying users that need their protection profile changed in order to address emerging threats [49].

Cyber Kill-Chain

The cyber kill-chain was proposed by Hutchins et al. [50] in 2011 in order to address, at the time, the emerging threat of advanced persistent threats (APT). The kill-chain is a generalized seven step process for penetrating a computer network:

1. **Reconnaissance** is the phase when target is researched, i.e. identification and selection of targets. This can be done by crawling web-sites, mapping out employees and persons of interest, or investigate technologies utilized by the target network.
2. **Weaponization** is the phase of coupling a remote access trojan or dropper into a deliverable payload like a PDF or Microsoft Word document.
3. **Delivery** is the phase of delivering the weaponized document to the target network. This is often done by e-mails and websites, or by planting USB thumbdrives on the physical premises.
4. **Exploitation** is the phase where someone in the target opens and executes the payload from the delivery phase so that the attacker can exploit a vulnerability in either installed software or the operating system.
5. **Installation** is the phase where the attacker installs his tools on the victim's computer. This can be a Remote Access Trojan or another type of backdoor to maintain persistent on the targeted network/computer.
6. **Command and Control** (C2) is the phase where the tools installed by the attacker in the former phases call home (beacon) to provide the attacker with "hands on keyboard" access to the compromised system.
7. **Action on objectives** is the final phases where the attacker can take actions to achieve the objective, e.g., data exfiltration, sabotage, use of the compromised system as a proxy towards another target.

Indicators

The fundamental element of Cyber Threat Intelligence is Indicators. An indicator is a piece of information that objectively describes an intrusion or an event. Indicators can be subdivided into three categories [50]:

Atomic indicators , are those indicators that can not be broken down any further to smaller parts and still keep their meaning and context. Examples of atomic indicators are DNS, Hostnames, IP-addresses, emails, and vulnerability identifiers.

Computed indicators , are indicators derived from the data in an intrusion. These can be computed cryptographic hash values or regular expressions.

Behavioral indicators , are the collections of both atomic and computed indicators that is often a subject to qualification by quantity and combinatorial logic. An examples of behavioral indicators is when threat actors send a phishing email *address* with a weaponized document with *hash* that when executed sends *traffic* to command and control *IP-address*.

Cyber Threat Intelligence Collection

To give some examples of what information can be turned into cyber intelligence, we provide a list of eight common technologies that can be turned into valuable cyber intelligence sources for an organization:

- DNS and Passive DNS can be used to track IP address to hostname mappings. This is a valuable intelligence source because it has the potential to assist in detection of unknown command and control traffic in the Kill Chain Model. The reason for this is that malware and campaign often use a deterministic domain name generator to power a fast-flux network where the atomic indicators like IP-addresses and DNS names change rapidly. Passive DNS can help to spot this by providing information about how long a domain has been alive. This provides an analyst with a list of domains, which had a short time to live, for further investigations.
- Intrusion Detection Systems can be a valuable source of CTI, because when a threat is discovered, rules can be written based on atomic indicators to detect other hosts that also have been compromised. IDS systems come in several types ranging from host-based, network-based to application based, each with its strengths and weaknesses. But eventually they all have the same capabilities depending on the perspective.
- Honeypot is a collection method for studying the attacker after he or she has compromised the system. This can be used to study their techniques and methods so that IDS rules may be written and goals of the attacker be determined.
- System logs are a goldmine for Cyber Threat Intelligence. By having the capability to centrally collect, store, and analyze system logs for the whole network, patterns can be observed and mined. For instance, if an attacker is doing reconnaissance by trying some stolen credentials on several servers in the network, this pattern can easily be picked up over the network even if he only tries a couple of times on each server. This is possible due to the

distributed view on the network, i.e. it may be highly suspicious that the user has 2-3 failed login attempts on a range of servers. Adding the context as what those servers do, one can possible deduce a motive as well.

- Malware sandboxes can be extremely useful for analyzing the attack chain and yield a range of valuable intelligence about the the weaponized file, what packer, crypter, etc. are being used, compile time, timezone of the attacker, and which DNS/IP addresses the malware is using to phone home,etc.
- Netflow can keep a track record of the metadata of all communication that is on the network, e.g. source, destination, TCP/UDP ports, packets sent, amount of data sent, etc. Netflow can be an extremely valuable source of intelligence and it ca help to collect forensic evidence from command and control traffic as well as actions. An example is the transfer of gigabytes of traffic over an uncommon purpose built protocol like DNS TXT records or over ICMP ECHO requests.
- Endpoint Protection products can provide information of malicious activity on the endpoint. This information is valuable because techniques used by attackers today are hidden, obfuscated, or encrypted, traversing the network.
- External intelligence maybe the most common source of cyber threat intelligence which comes from third party security vendors nowadays. Security vendors provide with the whole range of threat intelligence from strategic reports about the threat landscape to indicators of compromise at the operational level. Other external intelligence sources are Common Vulnerability and Exposure, Zero-Day advisories, Media and blogs, etc.

2.3.2 Cyber Situational Awareness

Situational Awareness is a cognitive state of human beings that defines and shapes their perception and understanding of a given situation. This applies to traditional intelligence work as well as to cyber situational awareness (CyberSA). Situational awareness in dynamic environments, like the cyber domain, is defined by Endsley [51, 52] as:

> "*is the perception of the elements of the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future*" [51].

Endsley [51] differentiates between situation awareness as "*a state of knowledge*" that exist within the cognitive space and situation assessment as the "*process of achieving, acquiring, or maintaining situational awareness*". Situational assessment is therefore the tool to gain and maintain situational awareness in a dynamic environment. It is important to understand that situational assessment and awareness are two different things, i.e. situational assessment can be automated with computers, but knowledge and context of the event can only be

understood by human analysts which can see patterns that computers cannot. For example, if computer gets a task to identify all objects in a room that can be used to drink water from, it will probably never finish. But for a human, this task is trivial. Situational awareness can be viewed as three separate evolutionary states—situation recognition, situation comprehension, and situation projection [52].

When applying situational awareness to the cyber domain, at least seven aspects are defined in the literature:

1. **Situation perception** or to be aware of the current situation includes both situation recognition and identification. Situation identification can include identification and classification of the observed attacks, and determination of the source (who, what) and the targeted system. Situation perceptions is beyond intrusion detection and works in the cognitive domain, i.e. meaning in the knowledge and understanding of security analysts. Compared to an intrusion detection system which can neither identify nor recognize an attack, but simply identify an event that might be part of an attack, the human analyst adds port perception and knowledge to the data [53].
2. **Impact assessment** which means to be aware of the impact of the current attacks. Impact assessment can be split into two sub-processes, damage assessment of the current state of the attack and impact assessment to uncover future damage if the attack sustains or increases over time. Both damage assessment and impact assessment include and utilize vulnerability and threat assessments [53].
3. **Situation tracking** is the process of monitoring the situation in order to see and understand how it evolves [53].
4. **Adversary behavior** is an awareness of how adversary behavior evolves over the time by monitoring attack trends and motives to define tactics, techniques, and procedures utilized by the adversaries [53].
5. **Understand the how and why** means to the cause of the current situation by performing root-cause analysis and digital forensics to understand how a system was compromised [53].
6. **Situation recognition** means to be aware of the soundness of the collected cyber threat intelligence and which knowledge-based intelligence decision that derived from these information activa. The metrics for this aspect is the truthfulness, completeness, and freshness of the cyber threat intelligence [53].
7. **Situation projection** is done by assessing plausible futures of the current situation that requires an understanding of the threat, the network, its vulnerabilities as well as the adversary motive, opportunity and capability [53].

Analyst Cognitive Bias

Cognitive Bias is an human error of processing information that leads to incorrect conclusions, distortion of information, or illogical determination of what the information means. To suffer from cognitive bias is a human quality, and all humans have it because it would be impossible to get through life without any preconceived notion about how particular events in life will play out, or living without any ability to estimate and make predictions about the future.

The problem with Cognitive Bias regarding intelligence analysts is the situation when the preconceived notions remain static, even while the surrounding events are changing, making those notions invalid. Liska [6] lists five biases that are common for intelligence analysts:

- The paradox of expertise often impacts the most experienced analysts who are experts in a particular field or area. This cognitive bias can be experienced by an analyst that has been studying and working in the same area over many years. Thus, analyst can dismiss situational changes because they do not fit into the established patterns that have been observed over a long period of time. This leads to dismissing the event as not important or relevant because it is believed to be an error or a mistake [6].
- Confirmation Bias is a when an analyst looks more at, and value the indicators that support his/her hypothesis while dismissing or neglecting the importance and value of indicators that are contradicting his/her believes and hypothesis. In other words, it means what weight and value an analyst assigns to an indicator based on his/her believes [6].
- Coherence Bias is when the analyst assumes that the group or an individual being studied have the same motivation and goals as the analyst himself. Thus, the analyst assigns the same values as he/she has to the subject, making himself unable to be objective, This results in overlooking vital information that in turn leads to the wrong conclusions of the finished intelligence [6].
- Hindsight Bias often involves memory distortion, a phenomenon where memories are being altered to fit a new narrative. It can be expressed as "I know it all along" and "How could anyone miss this". Hindsight bias can be very damaging since it does not provide methodological analysis of past events in order to create new knowledge and learn from past mistakes [6].
- Anchoring Bias is when an analyst relies too much on one aspect of the collected data and weights a single indicator as more valuable then all the other indicators. This can often happen to the indicator an analyst get hold first during an investigation. This bias is often experienced by young or inexperienced analysts, but it is not limited to them [6].

Denial and Deception

Biased analysts is not the only threat to good and actionable cyber threat intelligence. The adversaries themselves may try to influence the process by confusing and misleading the analyst to the wrong track in order to remain hidden and carry on with their objectives. Analysts must keep in mind that the attacker may as well use denial and deception. In intelligence, one rarely has the full and complete picture of a given situation and event, and analysts need to put the pieces together from often large and complex puzzles.

Denial can be used by an adversary as mechanism that seeks to prevent or degrade the ability to collect information about their campaign. To prevent this, there is generally a need to understand the attackers capabilities to be able to subvert them. Knowing the capabilities of the adversary has no value to the target if the target does not have the capabilities to counter this [6].

Deception happens when the adversary instead of trying to deny the target collection of indicators, manipulates the collection systems to provide the analyst with false or misleading information, for instance with time skew. Deception involves manipulation of collection systems either directly or indirectly with two goals in mind: (1) to spoil the collection systems with tainted information, and (2) to sway the analyst to producing finished intelligence based on skewed information. Deception can also come in the forms of diluting, where the attacker produces overwhelming amount of alarms as a decoy to stay incognito with the real attack happening in the background. This can be done by means of a Distributed Denial of Service attack [6].

Denial and Deception attacks can be quite costly for an organization to handle, and if well designed, very hard to detect. An analyst can only look into a finite amount of alarms and indicators per time unit. By distorting or depriving the analysts from access to information within his time window of view, the attacker can make the analyst miss important indicators in favor of the adversary. To work against such attacks, the analyst must be able to do knowledge accounting so that based on what he/she does know knowledge gaps can be identified in the given data. This can help the analyst to compensate for the lack of knowledge but increase the uncertainty and decrease the confidence of the produced threat intelligence. The analyst can also start investigating why the gaps are present and address the reasons for the gaps in the data set.

The following sub-sections present briefly different levels of cyber threat intelligence according to the intelligence pyramid (Figure 7),

Figure 7: The Intelligence Pyramid [6]

2.3.3 Strategic Cyber Threat Intelligence

Strategic Intelligence is meant for the top management in an organization. The role of intelligence at this level is to primarily help senior management to understand the broader picture of threats and potential threats against their organization. Intelligence at this level should try to answer the following questions: Who is attacking the organization? Why are they targeting the organization? Where do they attack it? To answer these questions can be an incredibly easy task or an extremely hard one, but the point of providing answers to these questions is to make senior management able to allocate the appropriate resources to strengthen their defense in the right places. Strategic intelligence in based on educated guesses and estimations regarding future behavior or expected capabilities of threat actors. Analysts working with strategic intelligence are not only in need of deep a subject-matter expertise, they must also have the willingness and capability to understand and adapt to the changes in the threat environment [6].

2.3.4 Tactical Cyber Threat Intelligence

Tactical Intelligence is primarily meant for managers and intrusion analysts in the organizations security team. The role of intelligence at this level is to provide the security team manager with information on how the threat actor is working by looking at their "signature"—a set of the threat actors tactics, techniques and procedures. This type of intelligence aims to answer the questions such as "What are the threat actor doing?" and "When is he doing it?" Tactical intelligence assesses the current immediate capabilities of the threat agent like weaknesses, strengths, and the intentions of threat agents. The goal is to asses and allocate the appropriate resources in the most effective manner at the appropriate time [6].

The purpose of producing tactical intelligence is to provide the security team with information on how the threat actor is working by assessing the current and immediate capabilities observed. Tactical Intelligence is also concerned about

learning the weaknesses, strengths, and intentions of the specific threat actor so that this can be addressed and applied in defenses to mitigate or handle attacks in a timely manner. The production of tactical intelligence is done by creating a "signature" for the specific threat actor, and identify this signature in the collected data sets. A signature in this context means a set of tactics, techniques, and procedures (TTP), that the specific threat actor is using that make them stick out from others. The main goal of intelligence at this level is to asses and allocate the appropriate resources in the most effective way against a threat actor at the appropriate time [6].

Tactical cyber intelligence has a bit shorter time to live than strategic intelligence. As strategic intelligence answers "who" and "which motives", which may not change, tactical intelligence describes "what" and "when". Threat actors can change their methods, tactics, and tools after what is available and still go after the same goal. Therefore, tactical intelligence has weeks to months of time to live before some new technique or procedure is available in the black markets of cyberspace.

2.3.5 Operational Cyber Threat Intelligence

Operational Intelligence is a real-time intelligence. The goal of operational intelligence is to answer the question "How is the threat actor getting into the system?" [6]. Operational intelligence is often derived from technical collections based on tactical intelligence to support the ongoing operation. Operational cyber intelligence is immediate and has a short time to live because the indicators used at this level of intelligence can be changed by the threat actor quite rapidly. For example, IP-addresses of command and control infrastructure, binaries, DNS names or email addresses. Since operational intelligence is vetted and used instantly, an IP-address that is malicious can be benign after a few hours. This means that the time to live of intelligence at this level is from less than a day up to one week. Because of the short time to live, this type of intelligence is prone to have a lot of false-positives alarms.

2.3.6 Challenges in Cyber Threat Intelligence and Situational Awareness

The current challenges in cyber threat intelligence and situational awareness are [53]:

Evaluation of the effectiveness of CTI and SA Since effectiveness evaluation depends on cognitive processes that support the decision making process, measurement of the effectiveness of decision based on cyber threat intelligence is still an open research problem [52].

Fusion Computer network defense is becomes increasingly difficult, i.e. many network protocols are insecure, software still contains a lot of bugs and vulnerabilities, and security mechanisms are often complex and error prone. There are large volumes of relevant security intelligence waiting to be used,

but the current security tools lack the ability to provide the context necessary to implement an effective computer network defense. There is a need in a technology to connect the dots and show the patterns of attacks and corresponding paths of network vulnerability. Traditional security tools generally only point solutions that provide only a small part of the complete picture and give few clues about the TTP of the adversary to uncover a complex multistop attack.

Security tools lack context needed for CND defending complex network with many security vulnerabilities requires a context to be more efficient.

2.4 Big Data Principles and Technology

> In the midst of chaos, there is also opportunity.
>
> Sun Tzu, "The Art of War"

In this section we are looking into Big Data theory, its concepts, principles, and technology. The main focus regarding technology is the Open Source ecosystem evolving around Apache Hadoop as it has become the defacto standard for cheap large scale data processing. The list of tools that are mentioned in this section is far from exhaustive, but rather a selection of the most mature and relevant projects to this thesis and those that are described in the top literature on the subject. The Apache Hadoop ecosystem is already being used for big data security and forensics in projects like Cisco OpenStack and the new Apache incubator project Metron. In addition to the top Apache big data project, we are looking at ElasticSearch as an indexer outside of the Apache Foundation because of its widely adoption as an indexing engine for machine data and logs and as competitor to the commercial Splunk.

The term Big Data and associated technology were pioneered by many of the big internet companies like Google, Amazon, Facebook and Linked'In. The reason for this was the extreme data growth during the last decade. The role of Big Data technologies is to take over in applications where traditional database technologies, like Relational Database Management Systems (RDBMS), are no longer able to scale [54].

Traditional RDBMS systems and Incremental Architectures are *Schema-on-Write* systems. This means that the system imposes the schema on the data when you write it to storage, limiting what the system can store by a given model. In Big Data systems, on the other hand, impose *Schema-on-Read* makes them more flexible for storing raw data; it also allows the application that reads and processes the data decide on the schema at run time.

When we are talking about Big Data, we essentially describe systems, data sets, and applications that have one or more of the properties described in the Four V's:

Volume The first property of Big Data is volume of data to be transferred, stored, and processed. Volume is the property of Big Data that most people associate with the term [55].

Velocity The second property is velocity, i.e. how fast data flows into the system in terms of amount and its continuity. Even small messages can produce quite a lot of computation if the frequency rate of the incoming data is high [55].

Variety The third property is variety, i.e. diversity of data types received by Big Data systems from several sources and processed in order to extract meaning from the data and put it together to tell a story from it [55].

Veracity The fourth property is veracity, i.e. the accuracy of the data coming into the system and data leaving it. Depending on the application of the big data system, the requirement of accuracy may vary. In some systems the accuracy must be absolute, in other systems close enough might be accepted. When deciding on building a big data system, the trade-off between accuracy and computing power must be made since more accuracy requires more computation [55].

One of the challenges related to the application of Big Data systems and distributed computing is that fulfillment of desired properties, i.e. complexity and scalability, can be a difficult task. Big Data systems should not only perform good and be resource efficient, they also need to be simple enough to understand and manage. Marz and Warren [54] describe the following desired properties of Big Data systems:

- **Robustness and Fault tolerance** There is a challenge to make distributed systems robust and fault tolerant. Systems must be able to handle random fall out of nodes, keep the consistency of distributed databases, handle duplication of data, and manage concurrent processing and human failures. The mentioned problems make it hard to check, understand or even make an educated guess about what the system is doing at the current time. Therefore, to make the system understandable, it is crucial to address and avoid these complexities when designing a Big Data system. By building immutability and re-computation into the Big Data design, the system could easily be reset and start over when a fatal error occurs [54].
- **Low latency reads and updates** Most Big Data applications require low-latency reads in a range of a few milliseconds to a few hundred milliseconds to satisfy their use-cases while updating latency may vary a lot, ranging from real-time to some hours. When designing Big Data systems both read and update time must be taken into consideration, but not at the cost of fault tolerance [54].
- **Scalability** A Big Data system needs to be scalable, i.e. resources can easily be added on demand when data or load is increasing [54]. Scaling in Big Data systems is commonly done horizontally by adding more machines to the system, thus increasing both storage and computational power since the system gets additional node to store and process data on. On the other hand, Vertical Scaling is used to add more resources, e.g., CPU, Memory and Disk, to the machines in the system itself. However, Vertical Scaling requires to take down the nodes for service temporary that may degrade the system performance during the take-down.

- **Generalization** A Big Data system must be generalized in order to run a wide range of applications [54].
- **Extensibility**: One would not want to re-invent the wheel every time a change is needed, or that a new feature in implemented. Definition of an extensible system is a system where one can add new features with minimal development cost. Adding a new feature often triggers the need to convert old data in to a new format to put the new feature to use, a part of making a system extensible is to make large-scale data migrations easy.[54]
- **Ad Hoc queries** It is extremely important to be able to do Ad Hoc queries on the whole data set or parts of it. Large data sets are full of yet to be discovered and unanticipated information. To be able to mine the data set arbitrary may provide opportunities to discover new applications for the data and provide the possibility for business or process optimization. One cannot discover new use cases for data if it cannot be queried [54].
- **Minimal maintenance** The components of a Big Data system should be chosen to provide as low maintenance as possible. Maintenance in this context means anticipating when scale up, keeping processes up and running, and debugging when things go wrong in production. Keeping the developers and system administrators occupied by keeping the system up and running instead of extending features for new business cases is an example of poor maintenance planning. An important part of developing a low maintenance system is to choose components with as low implementation complexity as possible. One wants to rely on components that have simple mechanisms and underlying technology. The more complex a system is, the more room for fatal errors exists [54].
- **Debuggability** It is crucial for Big Data systems to provide information necessary for debugging on failure, specially when the the aim is the possibility to trace each value in the systemthe reason why a given value was set [54].

2.4.1 Data, Information, and Data Structure

Data comes in three degrees of structure [56]:

- **Structured data** is often data that originates from databases or spreadsheets. Data that is structured conforms to a relational database model where data can be placed in the various fields with an assigned type. The model also has some restrictions on data types, what data can go into each field, and constraints between the various fields to enforce consistency [56].
- **Semi-Structured data** is data that falls between the categories of structured and unstructured data. It can be considered as loosely structured data, i.e. there is a structure, but this structure is not imposed by an underlying data model. When using semi-structured data, tags are used to identify certain elements within the data. But the data itself is elastic where

complete semantics are hard to extract without further processing [56].

- **Unstructured data** or information refers to data that does not have a pre-defined data model or are not organized in any pre-defined structural manner. An example of this is raw data (untagged) representing network dumps, documents, steaming sensor data, etc. [56].

2.4.2 Big Data Architecture Designs

Big Data architectures are build by combining different technologies to handle various functions for transferring, processing, and storing the data. In this section two main architecture designs that are used with the common toolbox of the Apache Hadoop ecosystem are introduced, the Lambda and Kappa architectures. Both Architectures can be implemented by mixing technologies such as Apache Kafka, Apache HBase, Apache Hadoop (HDFS, MapReduce), Apache Spark, Apache Drill, Spark Streaming, Apache Storm, and Apache Samza [7].

Lambda Architecture

The main idea behind lambda architecture is to build systems that handle big data as a series of layers where each layer satisfies a subset of the required properties while building upon functionality that is being served by the layers underneath.

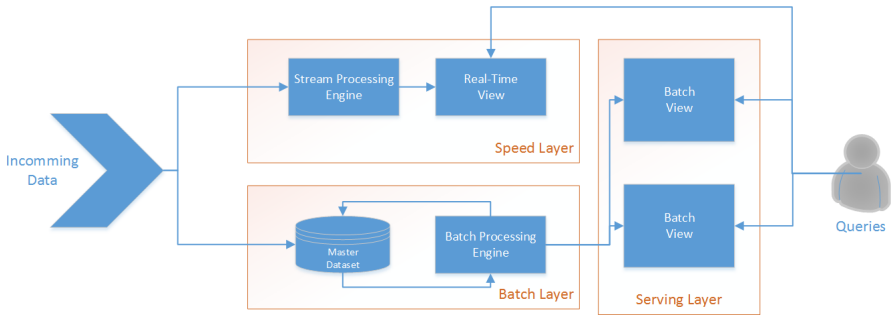


Figure 8: Lamda Architecture, based on [7]

As shown in Figure 8, lamda architecture consists of three layers, batch layer, speed layers, and serving layer. The batch layer has two major tasks. The first task is to manage historical data. The second is to recompute results, similar to re-learning of a machine learning model. The batch layer receives data from the incoming stream and then recomputes result by iterating over the whole data set, combining the fresh data with the historical data. The batch layer computes results with high accuracy but with the cost of longer computation, resulting in a higher latency before the new results are available.

The speed layer is used to provide less accurate but low-latency computational results for the new data. The speed layer implements incremental algorithms that are doing incremental updates on the result of the batch layers computation.

The serving layer is acting like an interface to query the results from both the speed layer and the batch layer.

In more complex use-cases where the output from batch processing and the real-time processing are different, it is not beneficial to merge the batch processing and stream processing. This is similar to the situation with computing / learning a machine learning model that requires quite a lot of time and computational resources so that the best achievable result in real-time processing can be obtained by incrementally updating the computed model from the batch layer. The strength of lamda is the separation of batch processing and stream processing, the weakness is more codebase to develop and maintain [7].

Kappa Architecture

Kappa architecture was proposed in the summer of 2014 by Jay Kreps from Linked'In. It is addressing some of the pitfalls that the lambda architecture has. For example, Lambda architecture can be expensive to maintain and develop since there is a need to maintain a codebase for the batch layer and the speed layer separately. Kappa architecture is not a replacement for lambda architecture since the main idea behind the kappa architecture is to handle both real-time and continuous data reprocessing using a single stream processing engine. In comparison with lamda architecture, kappa architecture is composed of only two layers, speed layer and serving layer (Figure 9).

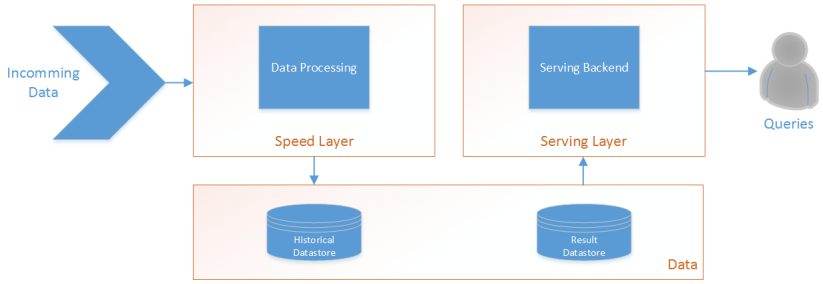


Figure 9: Kappa Architecture, based on [7]

The speed layer runs all stream processing functionality. One process is handling all the real-time stream processing while data reprocessing is only being done when some code of the stream processing engine is being changed. The reprocessing is being done by running another real-time stream processor that replays the whole data set with all previous data.

The serving layer is, like in batch processing, an interface to query the results of the processing stored in some result storage.

When the same data algorithms are being applied to both the real-time and the batch processing of historical data, maintaining the same codebase for both layers is clearly beneficial and cost effective. In this Use-Case, the kappa architec-

ture might be the most beneficial. The strength of kappa architecture is that it is less codebase to develop and maintain; the weakness is that it is not optimizable for complex batch processing jobs on historical data [7, 57].

2.4.3 Big Data Transfer

To be able to store and process data, data must be moved into the system and often between components of the big data system. Common data sources for Hadoop systems are traditional data management systems like relational database systems, log, machine generated data, and other forms of event-data and files of various formats. When designing data transfers for big data systems, there are several considerations must be taken into account: (1) timeliness and accessibility of data ingestion, (2) whether it is incremental updates, (3) data access and processing, i.e. what kind of system the source is and the data structure, (4) how the data is partitioned and split up, (5) in which format should data be stored, (5) whether data transformations are needed [9]. In table 2.4.3, latency classifications for ingestion and processing time in big data systems are shown.

Classification	**Time**
Macro Batch	$< 15min$
Micro Batch	$2min < 15min$
Near-Real-Time Decision Support	$2sec < 2min$
Near-Real-Time Event Processing	$100ms < 2sec$
Real-Time	$< 100ms$

Table 2: Latency in Big Data [9]

In the big data systems, tools for data transfer are usually implemented in the form of a queue. Examples of these are Apache NiFI, Apache Kafka, Apache Flume, Apache Scoop, and Apache Scribe [55].

Apache NiFi

Apache NiFi [58] is a tool to automate the task of building data flow between the systems both within and outside of the Hadoop ecosystem. Apache NiFi was developed by NSA, and was open sourced by the NSA Technology Transfer program in 2011.

Apache Kafka

Apache Kafka [59] is a distributed streaming platform that allows applications to publish or subscribe to topics (channels or message queues) and that allows one to send and receive stream messages in a fault-tolerant way while processing them as they occur on the channel. Kafka is a good choice as an replacement for traditional message brokers when linear scalability and fault-tolerance are needed. Kafka can be used for different purposes in a big data architecture, and is a popular choice to move data between systems or processes data in a stream

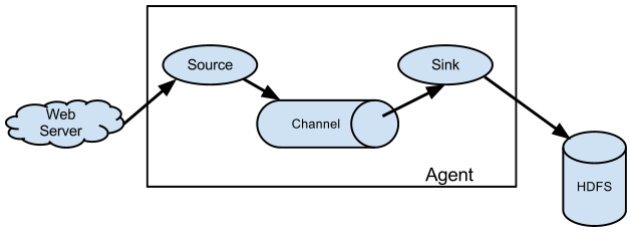


Figure 10: Flume Dataflow Model[8]

processing architecture. Kafka is increasingly popular choice in central log systems because of the high message throughput and abstractions from the source file it provides. When it comes to log files, performance of Apache Kafka can be compared to Apache Flume, but Apache Kafka has a more general approach and can transfer structured data, semi-structured data, or even unstructured data, in comparison to Flume which is bounded by structured and semi structured data. Apache Kafka has also utility in Event Sourcing type of applications where State changes must be logged and acted upon. In addition, Kafka can provide a central component for tracking commit logs in a distributed computing system.

Apache Flume

Apache Flume [8] is a distributed system that is able to efficiently collect, aggregate, and move large amount of log data from different sources to a central place for processing. Although Log data is what Flume was designed for, it is not limited to this use-case alone. Flume is a customizable agent-based tool that is designed for high-volume ingestion of event-based data into Hadoop Distributed Filesystem (2.4.4) [60].

As shown in Figure 10, the Flume agent consumes events from a source (e.g., a farm of webservers). A source in Flume produces events and writes those events to one or more passive channels that temporary store the event until it is extracted from the channel by a sink. A sink is an interface to an external system like HDFS or Hbase. The sink may also forward the event to another Flume source for publication on a new channel. As seen, Flume can build multi-hop flows that enable the system designer to pass the event along to multiple agents before reaching the final destination. Flume also allows the system designer to enable contextual and fail-over routing of the event flows, making the message bus fault-tolerant [8, 60].

Flume has a model that ensures reliability for event messages as each event is a stage in a channel on each agent until delivered to the next hop or destination. The events are only removed from the channel after they have successfully been delivered to the next destination. Treating the events as stages in the channel allows to recover from failures—the system can be configured to store the channel

content persistently on disk or in-memory. In this way, the system designer can choose between stability and recovery or fast processing [8].

Apache Sqoop

Apache Sqoop [61] is a tool for transferring bulk data between Apache Hadoop and structured, semi-structured, and unstructured data sources. Apache Scoop can import data to and export data from structured data sources such as Relational Database Management Systems, and Semi-Structured data sources like Apache Cassandra and Apache HBASE. The main goal of Apache Scoop is to simplify the integration of the Apache Hadoop ecosystems with traditional data storage systems. Apache Sqoop is a mature project and should be used when the application requires to import structured data into hadoop for processing and export the results back in to traditional structured data store when the processing is done [61, 60].

2.4.4 Big Data Storage

It is often desired in a big data system to store massive amount of data over time to perform queries and analysis on the data set to answer specific questions. Examples of big data storage tools are Hadoop Distributed Filesystem, Apache HBASE, Apache Cassandra, Apache Accumulo, and ElasticSearch.

Hadoop Distibuted File-system

Hadoop Distributed File-system is designed for storing extremely large files with the possibility for streaming data access patterns that can run on commodity hardware [62, 60]. Distributed file-systems are needed when the size of files exceeds the physical hardware resources that a single computer can provide, and therefore a cluster computing is needed. One of the biggest challenges with Distributed File-systems is the complex network stack needed as well as fault-tolerance when nodes fail. HDFS works well in use cases where large files in Gigabytes, Terabytes and Petabytes must be written once, but read many times. HDFS is designed to be used with off-the-shelf commodity hardware. HDFS is designed for high throughput and does not work well in use cases where applications need low-latency real-time or near real-time (2.4.3) data access. Metadata about files in HDFS is stored in memory of the namenodes in the cluser, the size of memory dictated how many files that can be stored within HDFS. The Rule of thumb is to caclulate 150 MB of memory per file in the filesystem, which make HDFS less suited for storing small files (less than Gigabytes). In HDFS, only one process is supported to write to a file at a time and must append all new data to the end of the file. It is no support for performing arbitrary file modifications [60].

HDFS has two main components, master and workers:

- Master (NameNode). The master component, also called a namenode is responsible for managing the cluser namespace. The namenode keeps track

of the file-system tree and all the meta-data for all files and directories stored in that three. This information is stored on disk in two files called 'namespace image' and 'edit log'. The namenode keeps track of the whole cluster and has therefore direct impact on the cluster's scalability based on the system resources available to the Namenode. The Namenode knows which worker (datanode) has which block of a specific file [60].

- Workers (Datanodes). Workers is the workhorse of HDFS, they store and retrieve blocks when are told either by a client or a namenode, and periodically report to the master node which filesystem blocks that are allocated. The default block size in HDFS is 128MB compared to a couple of kilobytes in normal filesystems. However, unlike traditional filesystems, if the file does not occupy the whole block, the storage in the host filesystem will be equal to the filesize. The reason for this large blocksize is defined by the desire to reduce the cost of seeks. If the the block is large enough, the time to retrieve it can often surpass the time of actually searching through it [60].

Apache HBASE

Apache HBASE is a distributed column-oriented database that utilizes the Hadoop Distributed Filesystem underneath. Apache HBASE is inspired by Goolge BigTable. The strength and use-case for HBASE is when the application requires a real-time read and write with a random accesses to very large data sets. Compared to traditional relational database systems which are not built to be distributed, the possibility for horizontal scaling is limited because of the complexity of common RDBMS functions like joins, complex queries, triggers, views, and key-constraints. This complexity makes the maintenance extremely expensive in case it can work at all. Because HBASE has much of the same properties as Hadoop, it is possible to horizontally scale it quite well by simply adding more nodes into the cluster [63, 60].

Apache Cassandra

Apache Cassandra [64] is a linear scalable database system that provides high availability and distribution over several data centers. This makes Apache Cassandra suitable in case as a mission critical database system is required.

Apache Accumulo

Apache Accumulo [65] is a sorted, distributed key/value store that provides robust, scalable and retrieval data storage. Accumulo has several novel features such as cell-based access control and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.

ElasticSearch

ElasticSearch [66] is a popular open source search enginge built on top of Apache Lucene [67]. It has gained special popularity regarding the log analysis and security because of the framework Splunk that was developed by the company behind the technology to compete with the industry standard of machine-data indexing.[6]

Apache Lucene is a cross-platform, high-performance, and full featured text search engine library that is suitable for nearly any application that involves full-text search [67]. Apache Lucene is arguable the most advanced search engine library nowadays, both proprietary and open source. Apache Lucene is just a search library and it is not very useful by itself. To use the full potential of Apache Lucene, one needs to build some application around it. This is what Elasticsearch is. Elasticsearch is written in Java, just like Apache Lucene, but Elasticsearch is abstracting away the complexity of the Apache Lucene library behind a simple RESTful application programming interface. This makes Elasticsearh much more than just the indexing engine [68]. Elasticsearch is a framework that makes it possible to:

- Build a distributed semi-structured document store where every file is indexed and searchable.
- Build a distributed search engine with near real-time analytics capability.
- Build a scalable distributed system for indexing and search.

2.4.5 Big Data Processing

In Big Data stacks, Data Processing is the part where different tools are used to perform some kind of processing and transformation, or extract some form of intelligence or information from the data set [55].

Data Processing comes in two levels:

- **High Latency Batch Processing** High Latency Batch Processing is appropriate when the application needs to provide an answer within the order of a few seconds to minutes or hours [56].
- **Low Latency Stream Processing** If a Big Data application needs an immediate answer on live data, a stream processing must be used. A stream processor processes data as it comes in, i.e. a small piece of code performs small operations on the events separately (Apache Storm) or using Micro Batch (Apache Spark Streaming). Stream Processing frameworks are primarily addressing parallelization of computational load over several nodes in a cluster, providing a variety of degree of Fault tolerance. To query the data, a storage back-end is also required [56]. Stream processing is therefore often used in combination with High Latency Batch Processing and a storage back-end, implemented in the form of the Lambda Architecture

[6] https://www.splunk.com

described earlier.

Hadoop MapReduce

The MapReduce programming paradigm was introduced in a paper by Jeffrey Dean and Sanjay Ghemawat from Google, where they describe a programming and implementation model for processing large data sets in the from of a 'Map and Reduce' function on Key-Value pairs [69, 9]. The Map function takes an input pair and produces intermediate Key-Value pairs. Then all intermediate Key-values associated with the same key value are grouped together before it is sent to the Reduce function (Figure 11).

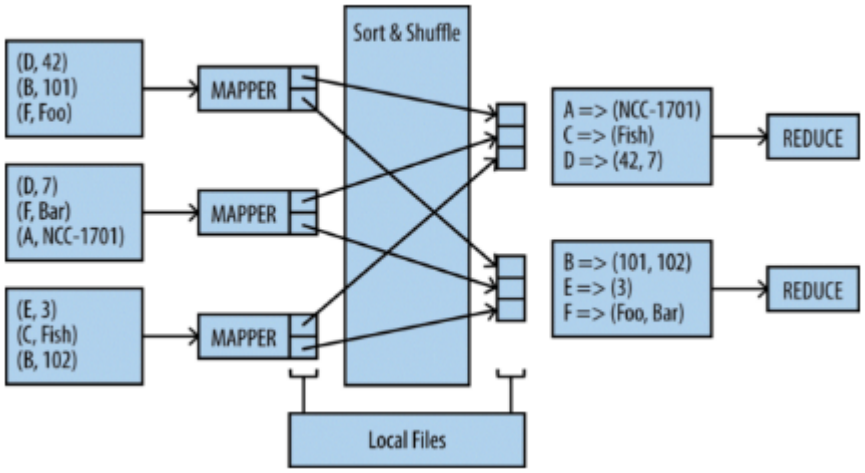


Figure 11: MapReduce, Figure 3-1 in [9]

MapReduce is a very low-level development framework where the developer is responsible for every detail of the processing, which makes it non-trivial to implement. MapReduce follows a very rigid data flow and does not fit well for applications of iterative machine learning or interactive data analysis. Nevertheless, there is a subset of problems where MapReduce is a natural selection for a task. Such tasks are compaction, distributed file-copy or row-level data validation. MapReduce writes data to disk after each job is executed and then re-reads the data from disk again when the next job starts. In other cases, MapReduce, because of being a low-level framework, can take advantage of specified input data for performance optimization in merging very large data sets [9].

Apache Storm

Apache Storm is a distributed realtime computation system that was originally developed by Twitter [56]. This system was initially designed to simplify and make more reliable the processing of endless streams of data. Apache Storm is

a pure-breed stream processing framework.Apache Storm is the realtime processing framework as MapReduce is the batchprocessing framework for Hadoop. Compared to Apache Flink, Storm does not provide any Batch Processing capabilities. Storm is simple and can be used with any programming language. Apache Storm has many use-cases such as realtime analytics, online machine learning, continues computation, distributed RPC, etc. Storm has been benchmarked to processing over 1 000 000 tuples per second per node. It is scalable, fault-tolerant and guarantees that the data will be processed [70].

Storm can be applied to a variety of stream use-cases and can work well with most of the big data technologies in the Hadoop ecosystem. Apache Storm is appropriate when the need is in a scalable distributed stream (realtime) processing engine which has fault tolerance and can guarantee that everything sent to it will be processed at least once. Apache Storm is also an independent programming language, and as long as it can run on a Java Virtual Machine or a *NIX commandline, it can easily be run on a storm platform [55].

Apache Storm uses the following components to build a computation engine [55]:

- Tuple A tuple is an ordered list of values and is the internal data structure used by Apache storm in the topology for passing data between computations.
- Topologies An Apache Storm topology is a graph of nodes where each node represents a single computation task. The topology defines the processing flow in Storm in the form of a DAG. The edges represent data flow between the computations.
- Streams A stream is an unbouded sequence of tuples (tuples are the data format within a Storm topology and it is essential what is being sent between the different computation nodes).
- Spouts A spout is the source of a stream in the Storm topology. Spouts usually read data from an external source, transfers them into tuples, and pass them to the topology.
- Bolts A bolt listens for incoming tuples adjacent to it. It may receive tuples from either spouts or other bolts. When a Bolt receives a tuple, it performs some kind of computation or transformation on the received tuple before sending the result of the computation to its output stream.

Apache Spark

Apache Spark [71] is a fast and general purpose cluster computing platform designed for large-scale data processing that provides high-level APIs, and is optimized for execution graphs [71, 10]. Spark was initially a research project from UC Berkeley AMPlab with the goal to improve the MapReduce framework regarding the rigid data flow enforced, which does not support many Data Science applications or Big Data Problems today. Spark is extending the MapReduce model

to support other types of computations that differ from traditional batch processing like interactive queries and stream processing. By supporting several types of computations in the same engine, Spark makes it less expensive to combine the different approaches (like for instance in lamda architecture) and go from implementing prototypes to scalable production environments. Speed is one of the most valued traits of Spark when comparing to the traditional MapReduce framework. Spark is 100 times faster than MapReduce when is run in memory, and 10 times faster when is run on disk. The speed that Spark provides is one of the main reasons Spark has become quite popular. This is due to the flexibility it provides in regards to interactively exploring very large data sets and experimenting on it before producing production grade systems [10]. The other trait is the easy accessible Big Data computation with simpler APIs in Java, Scala, Python, R and SQL. Apache Spark can run on existing Hadoop clusters and access any Hadoop data source. Apache Spark as shown in Figure 12 is built up by several integrated components, where the main component and the processing engine is the Spark Core.

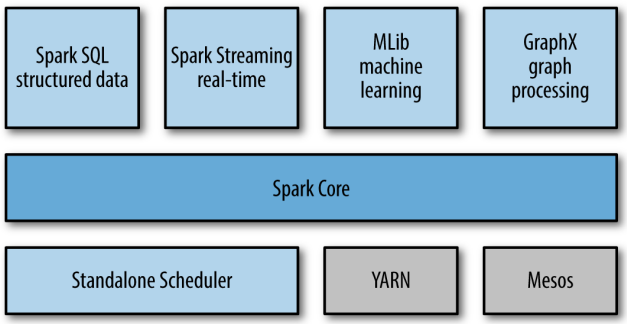


Figure 12: The Spark Stack[10]

Spark Core is responsible for the basic functionality that Apache Spark provides. Spark Core includes components for handling Task Scheduling, Memory Management, fault-recovery. It also interacts with storage back-ends and APIs for defining Resilient Distributed Datasets (RDDs), which is the Spark's main data abstraction. RDDs represent a collection of distributed nodes which allows Spark to do parallel processing on the same data set distributed over the several nodes [10].

Apache Spark provides a package for working with structured data, called SparkSQL. SparkSQL supports traditional SQL languange and the Apache Hive variant HSQL, and queries many sources like JSON, Parquet and Apache Hive Tables. SparkSQL can be combined in an application with logic providing the possibility for complex analytics and Data Warehousing capabilities [10].

Apache Spark Streaming (see Figure 12) is an extension of the core Apache

Spark API that enables a high-throughput fault-tolerant stream processing framework for live data stream processing [72, 10]. Spark Streaming provides an API that is similar to the Spare Core API, that makes porting Spark batch applications to Spark Stream applications easier [10]. However, Spark Streaming is not a true stream processing engine like Apache Storm or Samza, it utilizes Micro-Batch processing [55].

Figure 13: Spark Streaming[11]

Apache Spark comes with a component that handles common Machine Learning functionality and algorithms. MLlib is designed to provide classification, regression, clustering, and collaborate filtering machine learning methods to scale out in a clustered computing environment [10].

Apache Spark is providing a Graph processing library called GraphX. GraphX is built for performing distributed parallel-graph computations and manipulations across a cluster [10].

The advantages of using Apache Spark are [9]:

- **Simplicity** Apache Spark has a significantly cleaner APIs compared to the MapReduce framework, therefore there is no need for high-level abstractions on top of Spark like Hive and Pig for MapReduce.
- **Versatility** as is built from the ground up to be extensible.
- **Reduced Disk I/O** Spark can store its RDD in memory and process them with multiple iterations without storing on the disk. Due to the lack of MapReduce functions, Spark is reading the data when processing starts, and writing the data when there is a need for persistent storage of results.
- **Storage** Apache Spark is highly flexible, so the developer can choose between in memory on a single node, in memory replicated to several nodes or written to disk.
- **Multi Language** Apache Spark has multi-language support and the most popular APIs are those for Python, Java and Scala, which is the language that Spark is written itself.
- **Resource Manager independence** Spark supports both Yarn and Mesos as cluster resource manager, but there exist also a standalone option for developers.

- **Interactive Shell** Spark jobs can be deployed as applications like in MapReduce, but in addition, Apache Spark provides an interactive shell (similar to Python) which allows easy experimentation, debugging, and testing of Spark code.

Apache Samza

Apache Samza is a distributed stream computation framework originally developed by LinkedIn which can be directly compared to Apache Storm. Apache Samza is using Apache Kafka for messaging and Apache Hadoop YARN for providing fault tolerance, processor isolation, security, and resource management [73]. One of the main differences between Spark Streaming, Apache Storm, and Apache Samza, is that the first two can run stand-alone with their own resource managers, while Apache Samza is bounded to run only with Apache YARN as resource manager. Apache Samza is also simpler than Apache Storm, but includes the cost of tunability when it comes to parallelism. In Apache Samza, each processing node is a single entity which is connected with Apache Kafka, compared to Storm where this is handled internally resulting in a much lower latency.

There are several advantage of using Apache Samza over Storm. First, Apache Kafka is used between each processing entity providing a fault tolerant queue between the computations. Second, it also provides the possibility for independent entities to listen on the mediate queues making it easier to grab the result between each computation without disturbing it [55].

Apache Flink

Apache Flink is a unified realtime streaming and batch processing engine that provides data distribution and communication, and fault-tolerant distributed computation over data streams [74]. Apache Flink can be compared to Apache Spark.The main difference is that Apache Spark provides a micro-batch processing engine for streams instead of a realtime stream processor. The implementation of Stream processing engine of Apache Flink is similar to the implementation processing engine of Apache Storm. But Apache Storm is a pure Stream Processing engine, and does not provide any batch processing capability like Apache Flink.

Apache Pig

Apache Pig is a platform, developed at Yahoo in 2007 to abstract the complexity of Apache MapReduce and to analyze large data sets with a high-level language for expressing the analysis programs. It is one of the oldest MapReduce abstraction platforms that are still widely adopted today. The main strength of Apache Pig is that it provides the script language, Pig Latin. Pig Latin ensures easy parallel programming, utilizing the more complex Apache MapReduce underneath. Pig Latin is being compiled to run on an underlying processing engine, most commonly MapReduce. Apache Pig helps the developer with optimization so that he

can focus on the semantics in the analysis instead of program execution performance. Although Apache Pig provides its own language, it is still flexible and allows extensibility, i.e. developers can write their own special-purpose functions to use in their analysis task [75, 9].

There are many reasons to use Apache Pig as an abstraction to MapReduce. The only drawback of using Pig is that developers need to learn a new language and perhaps some new concepts. But the benefits outnumber the drawbacks greatly.

Apache Hive

Apache Hive is a data warehouse platform that provides an abstraction on top of Apache MapReduce, and was originally developed at Facebook. Like Apache Pig, it was one of the first and most adopted MapReduce abstractions that are still widely in use today for easy data analysis. Apache Hive is similar to Apache Pig in many ways, but instead of learning a new language, Apache Hive let developers use the more familiar SQL language to read, write, and manage the data that resides within the Hadoop cluster [76, 9].

Apache Hive is an appropriate choice for all queries that naturally can be expressed as SQL, especially long running queries where fault-tolerance is desirable (2.4). Apache Hive is also the De-Facto standard for handling Meta-Data in the Apache Hadoop ecosystem using the Hive Metastore [9].

2.4.6 Challenges in Big Data

The following Big data Security Challenges can be outlined:

1. Secure computations in distributed programming frameworks.
2. Security best practices for non-relational data stores.
3. Secure data storage and transactions logs.
4. End-point input validation/filtering.
5. Real-time security/compliance monitoring.
6. Scalable and composable privacy-preserving data mining and analytics.
7. Cryptographically enforced access control and secure communication.
8. Granular access control.
9. Granular audits.
10. Data provenance.

2.5 Multisensor Data Fusion

> Startled beasts show that the enemy is closing from several sides.
>
> Sun Tzu, "The Art of War"

2.5.1 Introduction to Multisensor Data Fusion

Data Fusion originates from traditional scientific disciplines like signal processing, statistical estimations, control theory, classical numerical methods, artificial intelligence, and machine learning. Data fusion has historically been used for military applications like automatic target tracking, autonomous vehicle guidance, battlefield surveillance, and automated threat detection. Data fusion has later emerged, been adapted, and applied to civil applications like manufacturing, robotics, video and image processing, medical equipment, and sensor networks [77, 78]. Data Fusion research has come far along in the recent years, but it has not matched the capacity and cognition of the human brain yet [78].

Fusion is defined by Hall and Llinas [77] as "*the integration of information from multiple sources to produce specific and comprehensive unified data about an entity*". Fusion related to data is defined by White [79] as "*a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats as well as their significance*" [79]. Hall and Llinas [77] describe data fusion as the:

> "*process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats as well as their significance*" [77].

The terms *Data Fusion* and *Information Fusion* are usually considered used interchangeably. However, in some cases, Data Fusion is the process of fusing raw data directly from sources while Information Fusion is the process of fusing already processed data (Information). There are several ways to fuse data; according to Castanedo [80], current data fusion techniques can be classified into three non-exclusive categories—data association, state estimation, and decision fusion.

Multisensor data fusion is the term used when data (or information) from multiple sensors or different type of sensors are combined (fused). The Multisensor Data Fusion is widely used in a variety of fields and becomes more relevant and practical in many fields [81, 77, 82, 83, 78]. The purpose of using multisensor data fusion system is to provide accurate situation assessment to ensure

appropriate actions towards a given event. Generally, application of multisensor fusion can be of two types, military and civilian. "*Multisensor data fusion is a technology to enable combining information from several sources in order to form a unified picture*" [78]. The concept of multisensor data fusion is not a new one, even though the methods applicable in real world applications have been discovered recently. Because of increased computational resources and improved algorithms for signal processing, pattern recognition, and machine learning, the application of multisensor fusion has become relevant. Multisensor data fusion system can be compared with the human brain that utilizes the concept of multisensor fusion every day. The human brain is an excellent example of a biological sensor fusion system that is so efficient that we rarely notice it. The sensors—senses like vision, touch, sound, smell, and taste—are being fused immediately and support our decision making [82]. Multisensor data fusion can refer to "*the acquisition, processing and synergistic combination of information gathered by various knowledge sources and sensors to provide a better understanding of the phenomenon under consideration*" [82].

In principle, fusion of data from different types of sensors (multisensor) provides a significant advantage over fusing data from one single source. In addition to the statistical advantage of fusing data from single same-source sensor, the use of multiple sensor-types may increase the accuracy of the decision based on the fused data [77]. Fusing data from different sensors can be used to introduce or enhance intelligence or system control functions. Multisensor fusion systems can be divided into three categories:

Complementary , when the information provided by the input sources represents different parts of the scene and can thus be used to obtain more complete global information. For example, in the case of visual sensor networks, the information on the same target provided by two cameras with different fields of view is considered complementary [84].

Redundant , when two or more input sources provide information about the same target and can thus be fused to increment the confidence. For example, data coming from overlapped areas in visual sensor networks are considered redundant [84].

Cooperative , when the provided information is combined with new information that is typically more complex than the original information. For example, multi-modal (audio and video) data fusion is considered cooperative [84].

The desired outcome of fusing data from multiple sensors is often to obtain lower error probability, increase the reliability of a decision made by the fusion system, and provide accurate situation assessment for decision support. One of the advantages from the use of multisensor instead of a single sensor, which is

the most relevant to this work, is the improved situation assessment capability that multisensor fusion systems provide. Data duplication is considered a good thing in most multisensor fusion systems as it improves the reliability of the system. There are several advantages multisensor data fusion provide over a single sensor system. The first advantage is an improved system reliability and robustness because of an inherit redundancy of data collected from different sensors. This increases the overall system performance and decreases the impact of a single sensor failure. In case of failure, a multisensor fusion system will still be operational, but degraded compared to a single sensor system, which will be in a fail state. The second advantage of multisensor fusion systems over single sensor systems is extended coverage. The perspectives of the system is increased, both spatial and temporal, when multiple sensors are used, i.e. a multisensor system can extend its reach beyond what is possible with a single sensor system. A multisensor system provides an increased confidence in the decision or result compared to a single sensor system because the sensors can confirm or disprove each others observations, so that the confidence in the result can be increased or decreased. A multisensor approach may also decrease response time since several sensors can collect more data than a single sensor in the same timeperiod, thus providing the result faster then the single sensor system. The third advantage of using multisensor fusion is the improved resolution that comes with the interference between events from different sensors [80, 82, 78].

Several models for data fusion have been proposed over the last three decades starting in the 1980's with the Intelligence Cycle, Boyd control loop (OODA Loop) [85] and the JDL Data fusion model [86]. In the 1990's the waterfall [87], Dasarathy [88], Visual-Data fusion [89], omnibus[90], and the Endsley [91, 51] data fusion models were proposed. In the 2000's, the Object-centered information fusion model [92], extended OODA model [93], TRIP model, the Unified data fusion model, the dynamic OODA Loop [94] and the JDL-User data fusion models were proposed [12]. In the next sections, some of these models are briefly described.

2.5.2 Intelligence Cycle

The intelligence cycle was one of the first data fusion models surfacing in the 1980's. It comes from military application for collecting, analyzing, and distributing intelligence. In 2013, the US Department of Defense published the JP2-0 Joint Intelligence [5] that describes the intelligence process (Figure 14). The intelligence cycle consists of five distinct phases. Despite the difference in the terms used, the parallels can be made to most of the other information fusion models. The five phases are, planning and direction, collection, processing and exploitation, analysis and production, and dissemination and integration. The five phases are arranged in a cycle around the mission—goal of collection—where each round enhance the intelligence for the given mission. A short description of each of the five phases has been extracted from [5] and presented below.

Figure 14: The Intelligence Process [5]

Planning and direction, "*planning and direction activities include, but are not limited to: the identification and prioritization of intelligence requirements; the development of concepts of intelligence operations and architectures required to support the commander's mission; tasking subordinate intelligence elements for the collection of information or the production of finished intelligence; submitting requests for additional capabilities to higher headquarters; and submitting requests for collection, exploitation, or all-source production support to external, supporting intelligence entities*" [5].

Collection, "*collection includes those activities related to the acquisition of data required to satisfy the requirements specified in the collection strategy*" [5].

Processing and Exploitation, "*during processing and exploitation, raw collected data is converted into forms that can be readily used by commanders, decision makers at all levels, intelligence analysts and other consumers*" [5].

Analysis and Production, "*during analysis and production, intelligence is produced from the information gathered by the collection capabilities assigned or attached to the joint force and from the refinement and compilation of intelligence received from subordinate units and external organizations. All available processed information is integrated, evaluated, analyzed, and interpreted*" [5].

Dissemination and Integration, "*during dissemination and integration, intelligence is delivered to and used by the consumer*" [5].

2.5.3 The Boyd Control Loop model (OODA Loop)

The Boyd Control Loop [85] shown in Figure 15 is more known as the OODA Loop that stands for Observe, Orient, Decide, and Act. The OODA loop follows the same cyclical format as the intelligence cycle. The OODA Loop is often used to model gaining and maintaining of situational awareness in any dynamic environments.

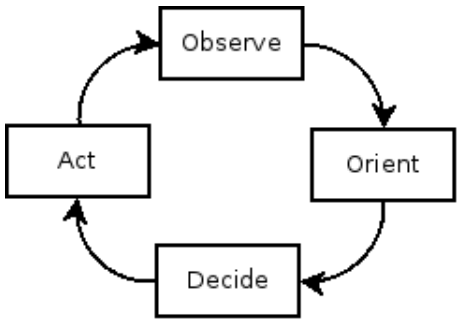


Figure 15: Boyd's control loop (OODA Loop)

Observe Some environments (physical or logical) are being observed, analogical to JDL model level 0.

Orient Analog of JDL Level 1 and 2.

Decide Analog to JDL Level 4.

Act Putting a made decision into actions.

2.5.4 JDL Data Fusion Model

The Joint Directors of Laboratories (JDL) data fusion model (Figure 16) was developed in 1991 [86] with revisions in 2004 [95], 2008 [96], and 2013 [97]. It is a functional model which is designed to be very general and applicable for multiple applications in multiple domains focusing on correlation, filtering, and association. The reason for development of the JDL Data Fusion Model is the historical boundaries regarding data fusion when it came to cross-application lack of common terminology and understanding. Even within the same industry, fundamental terminology and understanding varied depending on application [77]. Because of this, JDL has become the standard model for describing the information fusion process. The model is a two-layered hierarchy which splits processes into several levels of fusion. At each level, algorithms combine data to make interference about the meaning of data by putting it into a context. The multi-level breakdown of the JDL data fusion process is meant to provide and break apart functions rather then to identify a sequential flow [98].

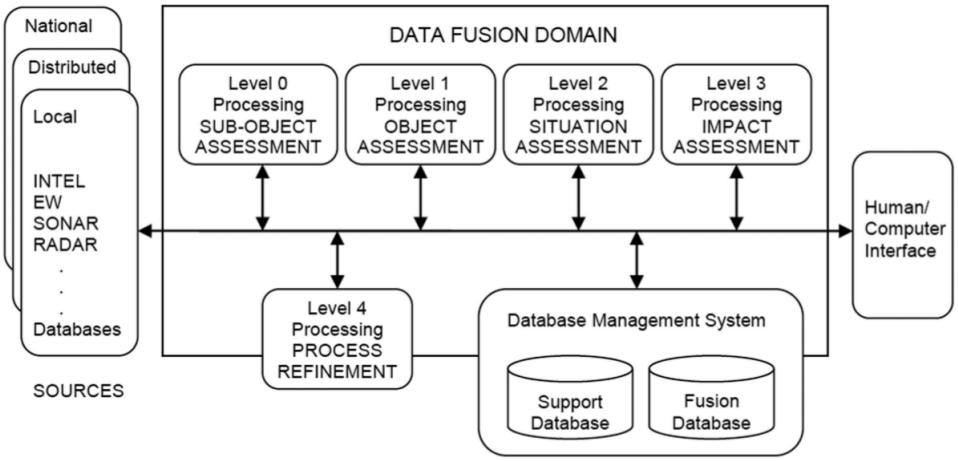


Figure 16: Revised JDL Data Fusion Model [12]

Level 0 - Data refinement Data refinement is the level where preprocessing and normalization of the data before fusion start in order to ease the load, and increase the quality of the fusion process.

Level 1 - Object refinement Object refinement is the level where alignment, association, correlation, and classification are being done to produce refined representation of individual objects. This means (1)to transform the data into consistent set of units, (2) to refine or extend object attributes and features, and (3) to assign data to objects and refine estimates of classifications.

Level 2 - Situation refinement Descriptions of current relationships among objects, events, and the context in their environment, are produces. The goal of the situation refinement is to produce and add meaning to the objects and their relationships resulting in situation assessment.

Level 3 - Threat refinement By projecting the current situation into the future, i.e. drawing interference about threats, vulnerabilities and opportunities, hypotheses about potential outcomes are produced.

Level 4 - Resource refinement Resource refinement is a meta-process to control and measure the fusion process, and is not a part of the fusion itself.

2.5.5 Visual Data Fusion Model

The Visual Data-Fusion (VDF) model (see Figure 17) was proposed by Karakowski [89] as an extension of the JDL data fusion model. Karakowski presented a human participant as an integral part of the model. The human is a central participant in the information fusion who enables cognitive and creative problem solving. The information derived from the fusion process is visualized and shown to the human to help him gain a broader perception of and insight into the data, thus enhancing problem solving. Visualization is a key since humans process images better than data. The advantages of VDF is the maximization of relevant information by means of visualization that enables to create increasingly sophisticated problem queries. Because of the user interaction, tailoring the information fusion capabilities for all type of users is possible. The visual data fusion model is a problem-driven and relates to what the users of the system need—the users perspective [99].

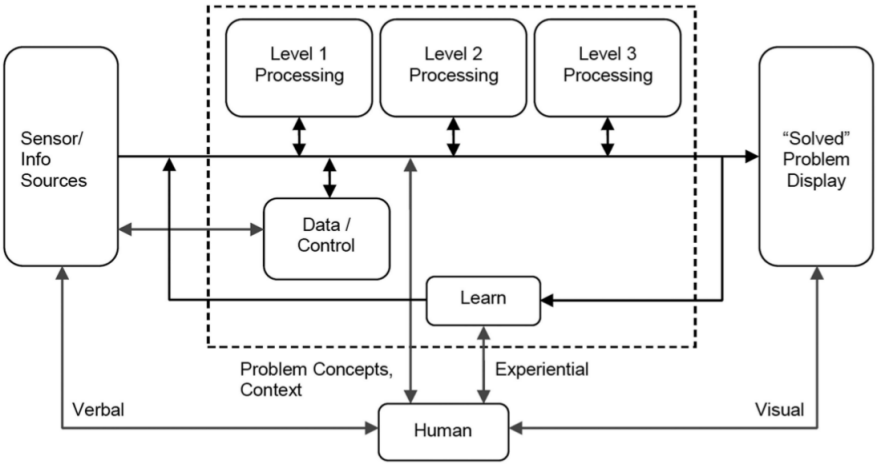


Figure 17: Visual Fusion Model [12]

2.5.6 Waterfall Data Fusion Model

The Waterfall data fusion model was proposed by Markin et al. [87] (Figure 18). The model consists of three levels of data fusion. Level 1 performs the signal processing (sensing) of the raw data, and transforms it into information about the surroundings. Level 2 is extracting features from the raw data and performing pattern matching processing in order to minimize the data content while maximizing the information delivered. Level 3 performs decision making and provides situation assessment by establishing the relationships between the objects and events. The focus of the model is on processing data on the lower levels without any feedback to humans that makes him unable to interact with the process, which is one of the limitations of this model [12, 100].

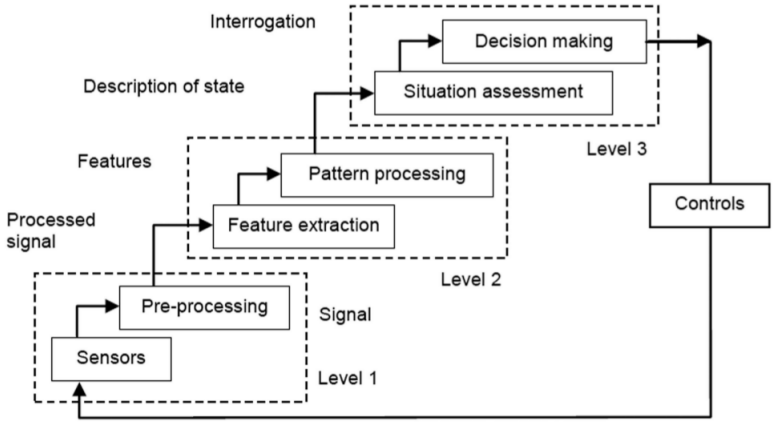


Figure 18: Waterfall Fusion Model [12]

2.5.7 Omnibus Data Fusion Model

The Omnibus model was proposed by Bedworth and O'Brian [90] (Figure 19) as a unified fusion model that "*comprises a flow chart, dual-perspective definition and a structured repository of accumulated expertise*" [90]. The omnibus model embraces the cyclic process of data fusion, and is based on the Boyd control loop and the intelligence cycle combined with the definitions and fidelity that the tasks from Waterfall fusion model provides. The tasks from the waterfall model can be associated and mapped with the levels defined in the JDL and Dasarathy fusion models. Since the omnibus model is cyclic, a feedback loop is explicit. The Omnibus model embraces the concept of loops, and even loops within loops.The model can be used with two purposes: (1) to provide an ordered list of tasks, or (2) to organize functional objectives by means of the ordered structure [90, 12].

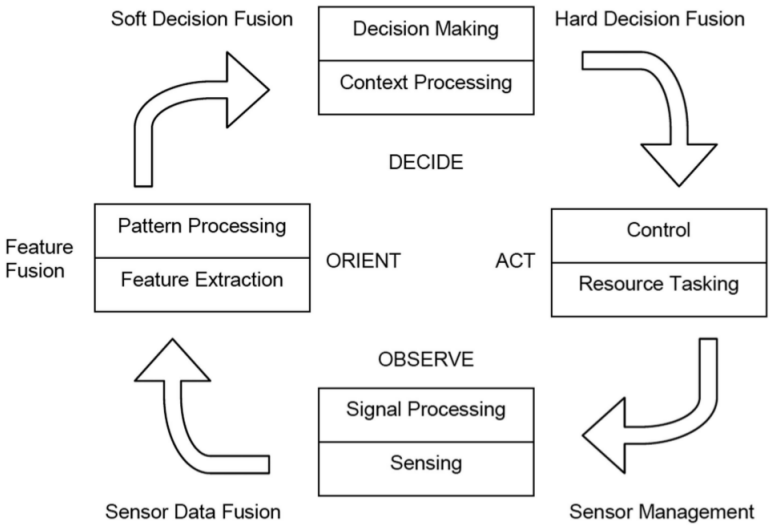


Figure 19: Omnibus Fusion Model [12]

2.5.8 The Dasarathy model

The Dasarathy data fusion model [88] is commonly viewed as a hierarchical model tied to the three general levels of abstraction within data fusion: data (sensor data), features (intermediate-level information), and decisions (symbolic belief values). Dasarathy [88] pointed out that fusion can be done in and across all the three layers. The Dasarathy model was developed to expand the three levels into six categories, or Input-Output modes, that can be combined and mixed in a flexible fusion architectures to build applicable fusion systems and logic [12, 88].

- *Data in - Data out (DAI-DAO)* This is the level of fusion, both the input and output data are fused, and is commonly known as data fusion in literature. This level of fusion is used for traditional signal and image processing domains where there is very little abstraction. Therefore it often takes place in the front of a fusion pipeline. Data fusion at this level requires compatibility of sensors.
- *Data in - Feature out (DAI-FEO)* At this level of fusion, the data input fuses into a feature output. Data from multiple sensor are the input and is being fused to produce some feature(s) (information) that describe the characteristics of the data inputs in an abstracted meaningful manner. An example of this is Flow or Netflow where packets on the wire are being fused into a flow record.
- *Data in - Decision out DAI-DEO* At this level, raw data is the input and a decision is the output. This level is very similar to FEI-DEO and is applicable to many pattern recognition problems.
- *Feature in - Feature out (FEI-FEO)* At this level, both the input and output of the fusion process are features. This level is, therefore, often referred to as feature fusion. It takes place in the middle of a fusion pipeline. Instead of combining data into features, features are combined instead either quantitatively, e.g., in a multidimensional feature space, qualitatively within a heuristic decision logic process, or by a combination of qualitative and quantitative information.
- *Feature in - Decision out (FEI-DEO)* At this fusion level, features are the input, and a decision, e.g., a target class, is the output. This type of fusion is mostly used in pattern recognition systems that involve input from multiple sensors that either need to classify something based on a prior knowledge or utilize training of machine learning algorithms to produce a label (decision) based upon the observed features.
- *Decision in - Decision out (DEI-DEO)* This is the last fusion level in the hierarchy where both input and output are decisions. This type of fusion is appropriate when there a need to fuse decisions of a spectrum of sensors with different configurations and tasks exist. This ensures that decisions

can be made on a high level of abstraction, in contrast to the data and feature level fusion where sensors must be compatible.

2.5.9 Multisensor Data Fusion Systems Design

Multisensor fusion systems are in principle easy to understand, but may be a complex and challenging task to design, implement, and apply to real problems [82]. The purpose of developing a multisensor fusion system is to define where the data flows and where to fuse them [77]. When designing and building a multisensor fusion system for a specific application, several fundamental issues must be addressed. The designer should choose algorithms that are optimal for the application and architecture that takes into account where data flow and where to fuse this data. The individual sensor placement must be planned and taken into consideration in order to get the maximum scope out of data collection and how the data collection environment affect the processing. Moreover, it is important to assess what accuracy can be realistically achieved for the fusion process, how dynamical optimization can be implemented, and under which conditions the fusion improves a system operation. Multisensor data fusion architectures can be divided into three categories: centralized fusion, decentralized fusion, or distributed fusion [77].

There are also practical consideration regarding the design and implementation of multisensor data fusion system. The nature of the sensors available and their resolution, as well as type and accuracy of the data sensors collect, are important issues to address. The following questions should be answered [82]: (1) Are the sensors either geographically or functionally distributed? (2) What are the constraints on the communication between distributed sensors? (3) What is the computational capability of the sensors? (4) Can algorithms be implemented both on the sensor and at the fusion center? (5) What algorithms can support the overall goal of the fusion system? and (6) What kind of system architecture, infrastructure topology, and communication can support this at which fusion level?
.

2.5.10 Challenges in multisensor data fusion

As multisensor fusion is mostly used in physical systems, e.g., robotics, many challenges are related to this, especially when it comes to getting correct readings and binary data fusing. The list of the general challenges regarding data fusion (multisensor and single sensor) are left out of scope. This section includes challenges in the field of data fusion and multisensor (based on [78]) due to the scope of this thesis and leaves out the more general problems.

Conflicting Data Data provided by sensors in a multisensor environment might be conflicting.

Data alignment Data originating from different types of sensors or distributed sensors must be aligned before processing. This means preprocessing and normalization of data that may be challenging.

Data association When performing tracking of multiple targets in an environment, the complexity increases significant comparing to tracking a single target. Two problems rise: (1) 'measurement to track', i.e. the problem of identifying if any events are connected to which target, (2) 'track to track', i.e. the problem of distinguishing and combining the tracks of the target.

Data dimentionality Data originating from sensors might be large and require heavy processing. This challenge also applies to the feature space of the generated data stream since many small features being ingested fast can provide significant processing challenges.

Processing framework The challenge of central versus de-central or distributed processing where all approaches have advantages and disadvantages.

Timing The data might come in with the same timestamps or a spread of timing might occur when the data comes into a central processing framework.

3 Related Work

> We become what we behold. We shape our tools, and thereafter our tools shape us.
>
> Marshall McLuhan

In previous chapter, an introduction to this thesis was given, i.e. the theoretical background of several key disciplines is presented. This chapter provides relevant work related to the topic of the thesis. An overview of state-of-the-art in big data for network security monitoring will be presented, before a state-of-the-art in multisensor data fusion for intrusion detection is presented.

3.1 Digital forenics

As the amount of data for reactive digital forensic investigations increases, work is being done to overcome the Big Data problem of digital forensic investigations. The most notable is the work on Hansken by Netherlands Forensic Insitute [1].

Cisco AMP [101], CarbonBlack [102], and FireEYE [103] are all commercial tools within the enterprise detection and response segment that provide capabilities for limited remote forensics (live forensics). The tools can detect and respond to threats, and collect remote evidence like file hashes of executed files. Cisco Advance Malware Protection (AMP) [101] is a hybrid between traditional endpoint protection platforms and enterprise detection and response, originally developed by sourcefire and bought by Cisco. Cisco AMP comes in three different products—AMP for endpoints, AMP for email, and AMP for networks—which can be fused into the Firepower console. CarbonBlack [102] is a next-generation anti-malware and enterprise detection and response which is highly scalable endpoint solutions. CarbonBlack can aggregate information across endpoints but does not have the network or email view of Cisco. FireEye [103] provides a more unified view to malware detection and covers network, endpoints, email, and content. They focus on the products for enterprise forensic and investigation. All tools incorporate cyber threat intelligence in order to improve their detection and response capabilities.

3.2 Cyber Threat Intelligence and big data

Academic research on cyber threat intelligence related to network security monitoring is scarce at the moment. Academia focuses on threat intelligence sharing,

but not how to apply it. This is why industry must be studied. The tools for cyber threat intelligence used by industry include:

SQRRL [104] is a product based on Apache Accumolu, and is intelligence-, situational awareness-, and analytics-driven with the goal to enable threat hunting within networks. SQRRL is the industry leading threat hunting platform that utilizes link analysis, machine learning, and graphs analysis on a scalabale Hadoop-based platform. SQRRL aims to help analysts to discover threats faster, thus reducing the cost of investigations. SQRRL slogan is "target, hunt and disrupt". Target means the scope of an investigation using indicator- or hypothesis-driven exploratory analysis and automation. Hunt means proactive and iterative search through the network and endpoint data to discover threats. Disrupt means going from hunting to forensic analysis and disrupt the threat from gaining its goals within the network.

Palantir [105] is a big data platform for intelligence gathering and analysis. Palantir emphasizes the use of multiple tools by an organization by collecting data from endpoint protection platforms, logs, and firewalls. However none of these tools are adapted for detection of advanced threats that utilize sophisticated methods for attacks. Palantir integrates structured data with contextual data, and performs large scale data fusion in a single environment where the analyst can interact with it, or use algorithms for detection of patterns related to threats in the data. Palantir enables forensic investigations over multiple dimensions and can support in connecting the dots between seemingly not connected events.

IBM i2 Analyze [106] is another tool for intelligence analytics, not just related to cyber, but also to law enforcement and military intelligence, where it is most known. The IBM i2 platform is a general data fusion stack of tools for intelligence gathering and analysis, and enables cross-domain intelligence analysis. It provides data collection, analytics, interactive visualization, and search. Nowadays, IBM i2 is focused rather more on government, law enforcement, military, and financial intelligence than cyber threat intelligence, even though it can be applied to it.

IBM QRadar [107] is a security intelligence platform, security event, and information management (SIEM) platform from IBM, which claims to reduce noise by their proprietary QRadar Sense Analytics engine. QRadar Sense provides discovery of slow threats, helps to find vulnerabilities, and performs anomaly detection. It can be integrated with third-party and can provide a unified central data fusion system for security analytics. The key applications are fraud detection, cloud security, incident forensics, insider threat monitoring, and risk and vulnerability management.

3.3 Big Data and Network Security Monitoring

There is limited academic research on Network Security Monitoring since it is a young discipline within the field of informatics and cyber security. In contrast, plenty of academic research on intrusion detection systems has been carried out. Related work in the filed of NSM can be found in industry and not academia. Cisco was developing an open source platform for security operations centers, called OpenSOC, for many years. The OpenSOC project was a collaborative development project with the aim to build a large scalable security analytics tool based on the Hadoop ecosystem. The project was providing complex event processing and event enrichment of telemetry data as it was floating through a pipeline [108]. One of the disadvantages of OpenSOC project is that it did not take advantage of full parallelism which Apache Storm provides by treating the enrichment pipeline serially. In addition, it was hard to extent because new Storm topology was needed when adding new sources. OpenSOC had also problems with scaling since increasing of the sources increased the complexity of the system, resulting in redundancy and difficulties of maintaining the code. Moreover, it was lacking a testing of the codebase, thus the quality of large parts of the OpenSOC codebase was not fully validated. Due to the mentioned disadvantages, the OpenSOC project was discontinued in favor of the Apache Metron project. Apache Metron [109, 110] (Figure 20) is being built and developed from the OpenSOC codebase and strives to evolve and advance the state of the art regarding security analytics. Currently, Apache Metron is an incubator project and is being moved into the Apache Software Foundation. It has inspired many of the ideas behind this thesis, especially on the technical and architectural side of the proposed implementation.

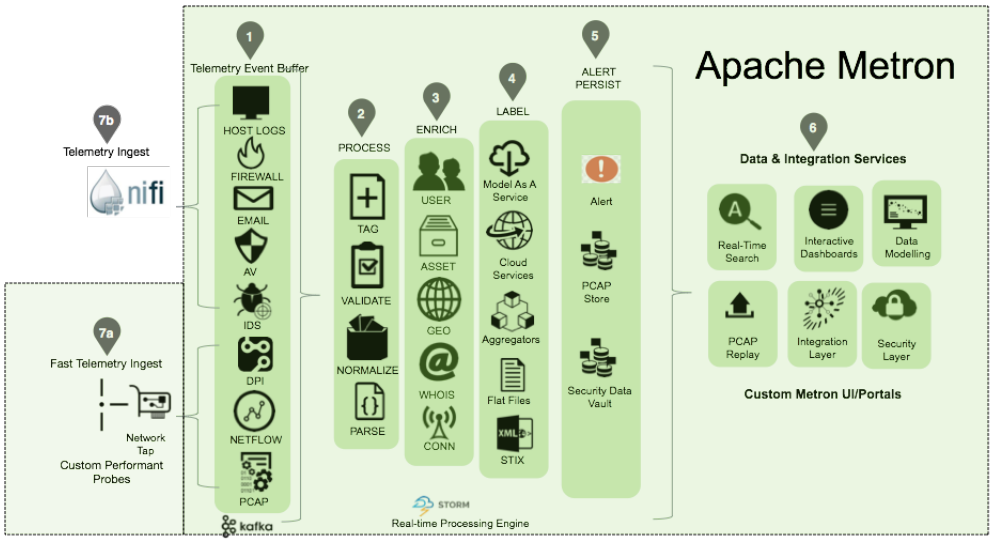


Figure 20: Apache Metron Logical Architecture [13]

3.4 Multisensor data fusion for intrusion detection

Bass [111] describes the concept of distributed sensing in the form of a distributed intrusion detection system that creates inferences about events, activities, and situations. Bass uses sensor data, commands, and a prior knowledge from databases, in his fusion model. The inputs are based upon system log files, snmp traps, packet sniffers, and user queries. The output is the identity and possibly the location of the intruder and the intruder's activity, observed threats, attacks rates, and severity of the cyber attack.

Ballora et al. [98] proposes an adaptation of the JDL Data fusion model for usage in computer network defense. It is said that the military application of multi-sensor fusion applies well to the cyber domain. Ballora et al. describe how humans, as soft sensors, are becoming more important in human-centric information fusion. The use of ad-hoc users as sensors for gathering information about an emerging situation and utilization of human analysts in a joint cognitive fusion system, where the utilization of the human skills related to visual and aural pattern recognition in combination with semantic reasoning, are actively taking part in decision-making and situation assessment, and analyst crowd sourcing of analyzing a situation or a threat. Ballora et al. describe the use of a joint cognitive system that uses reasoning and visual patterns recognition of human analysts in combination with the processing power of computers. The conclusion of their work is that multisensor data fusion in the computer network defence will require multi-disciplinary efforts, (1) within the development of algorithms and technology, (2) within the techniques to improve the analysts ability to understand evolving situation, identify and predict threats, and (3) in

developing collaborative decision making methods. It is believed that success is depended on looking at the process from data to the analyst, and from the analyst down to the data, and that efforts must model both cyber space and the human landscape to address cyber attacks. In [98], Level 0 passes and synchronizes data from various sources and address the problem of assigning unique keys to events. Level 1 combines level 0 data to provide capability to identify individual security events in the observed digital environment. This is where environmental enrichment also happens, e.g., adding system information to the object. Level 2 provides comprehension of a current situation, and Level 3 - projection and prediction of future outcomes of the current state. Level 4 is process refinement providing updates to detection and sensor capabilities while level 5 is described as a cognitive refinement for the analyst to interact with the fusion [98]. The usefulness of fusing intrusion detection data has been documented by Thomas and Balakrishnan [112].

4 Methodology

> The map? I will first make it.
>
> Patrick White

The objectives of most traditional sciences can be defined as to explore, to describe, to explain, and to predict. Such objectives can be achieved by using descriptive, evaluative, exploratory, or predictive research paradigms. However, some studies, often in the field of applied research, might have other objectives, such as prescribing or developing solutions and methods for solving a given problem or developing a new artifact [113]. The result of such studies is the design of artifacts that serve human beings, both tangibles artifacts, e.g., machines, hardware, and intangible, e.g., methods, models, software. This type of studies aims at reducing the gap between the theory and practice, and uses the prescriptive research paradigm. Prescriptive research is defined as a knowledge-using activity corresponding to design science. The goal of this science is to develop knowledge that professionals can use to design solutions for their field problems [114]. Prescriptive research is argued to be an essential part of information systems (IS) discipline, which is usually applied and practical, and the design science activity of building IT artifacts is propound as an important part of prescriptive research in IS discipline [115]. The researcher's experience is often used for the development of the solution and recommendations to the solution.

Since the goal of this work is thus to come up with a model for how to improve intrusion detection and situational awareness to serve the purpose of incident response and digital forensics, a prescriptive research design is chosen. The process of the research is presented in Figure 21. The research process starts with an in-depth investigation of the problem to be solved (Problem Analysis stage). During this stage, the problem is analyzed iteratively using the existing theoretical body of knowledge (i.e. existing theories and research) and the available empirical data (questionnaire, expertise). A thorough understanding of the problem is reached. At the solution design stage, a model of MSDF for improving the intrusion detection and situational awareness to better serve the purpose of incident response and digital forensics is developed. Theoretical and empirical insights from the problem analysis stage forms the basis for the design of the model for MSDF for ID and SA. Solution design is a creative process, and a variety of sources is used to inspire the solution, i.e. existing approaches, researcher's expertise. During the last stage of the research, the proposed model is documented and discussed.

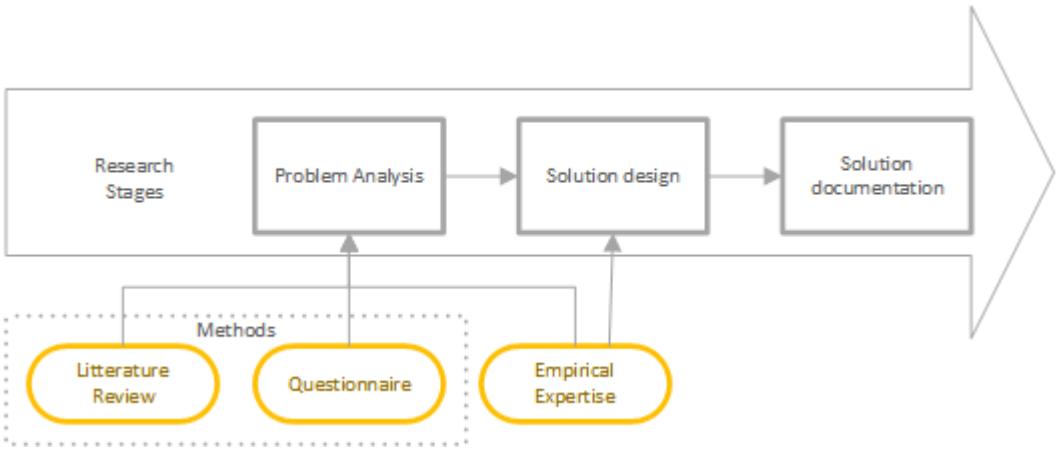


Figure 21: Research methodology

4.1 Literature Study

Because of the multi-dicplinary nature of this research, the need for studying multiple equally important topics was needed. The qualitative method of literature study was chosen to get comprehension of the quite large domains of 1) digital forensics, 2) network security monitoring, 3) cyber threat intelligence, 4) big data principles and technology and 5) multisensor data fusion.

4.1.1 Sources

The main sources used for literature study was Google Scholar, IEEE Explore, ACM, and the institutional archives and databases. The citation lists in the initial works found (especially state of the art reviews) on the topics were also used to backtrack reference work on each of the domains. From these reference works, more recent works and search terms on all of the topics where discovered. The process of discovering new publications continued until the references where starting to go in a loop, returning to the same list of references. In addition to academic resources, security blogs and news sites were used to identify the state of the industry and relevance of the topic of the thesis.

4.1.2 Search terms

The terms used to search for relevant academic resources from Google Scholar, IEEE Explore, ACM, and the institutional archives where:

Multisensor Fusion, *Multi-sensor Fusion*, *Multi-sensor Fusion Intrusion Detection*, *Multisensor Data Fusion*, *Multisensor Data Fusion Models*, *Multisensor Information Fusion*, *Data Fusion*, *Data Fusion techniques*, *Data Fusion Algorithms*, *Data Fusion Mathematics*, *Data Fusion Models*, *Big Data*, *Intrusion Detection*, *Anomaly Detection*, *Data correlation*, *Computer network defense*, *Computer network defence*, *Forensic Readiness*, *Proactive Forensics*, *Active Forensics*, and *Situational awareness*.

4.1.3 Method discussion

The choice of the qualitative method of literature study was essential for comprehending and understanding how each sub-domain works and how each topic of interest is interlinked with topics in another sub-domain. Without utilizing literature study, the foundation for this thesis would not be possible. This method formed the baseline for answering all three research questions, and in combination with questionnaire and empirical expertise enable the problem analysis.

4.2 Questionnaire

To relate the knowledge gained from literature study and to prioritize the focus of this research, the qualitative method of an questionnaire was chosen to complement the literature study and to map the gap between literature and current practice to make the proposed approach for multisensor data fusion applicable to real world scenarios. The result of the questionnaire is included in Appendix A.

4.2.1 Expert survey

The questionnaire was sent to individuals and groups in the Norwegian CERT/CSIRT community to help understand how IT-administrators, security practitioners, and incident responder's work on a daily basis to identify and confirm that problems found in literature related to intrusion detection and forensics are valid. This provides the thesis with external validity.

4.2.2 Method discussion

The questionnaire was important to help identifying tools, technologies, and current practices that are used by practitioners, along with the limitations and challenges that practitioners face on a daily basis. Subjects of the survey were carefully selected from a target group of security practitioners and IT-administrators with security roles within their organization. Ten questionnaires were completed. Although this is not enough for statistical generalization, analytical generalization can be made allowing to link the findings from a theory to the practice. The survey was helpful to clarify problems related to network securing monitoring, tools used, types of data collected, etc. The number of responses is constrained by the unwillingness of many organizations that work with information security to share their problems.

5 Relationships between domains

> We are not creators; only
> combiners of the created.
> Invention isn't about new
> ingredients, but new recipes.
>
> Ryan Lilly

Chapter 2 presented the theoretical background of five different domains: Digital Forensics, Network Security Monitoring, Cyber Threat Intelligence, Big Data Principles and Technology, and Multisensor Data Fusion. This chapter thus describes how these domains are related and how to complement them in order to provide improve intrusion detection and situational awareness. Relationships between domains are shown in a flow chart in Figure 22. The grey and orange boxes represent a support domains and the goal domains respectively. Green arrows carry a meaning 'to improve', while blue - 'to enable' the process it points to. The arrows that are both 'enabling' and 'improving', have the color of their main effect on the domain pointed to. The flow chart is a circular like many of the data fusion models reviewed in 2.5. Five domains are the components of the platform model described in chapter 6. Relationships between domains are described below.

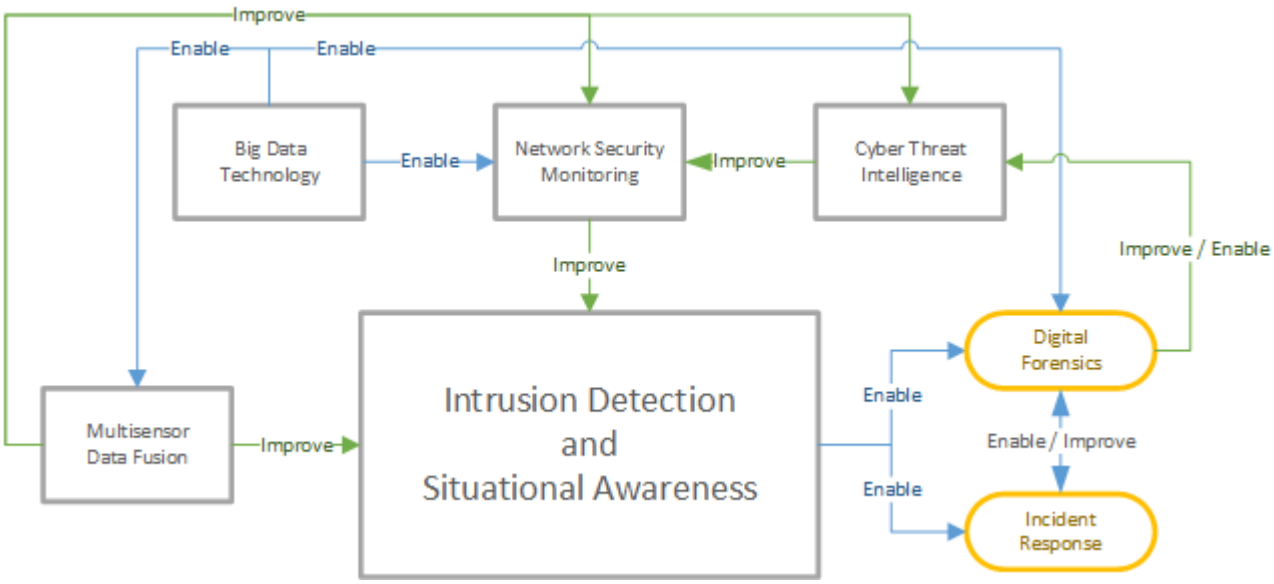


Figure 22: Relationship between domains

In Figure 22, 'Intrusion Detection and Situational Awareness' enables 'Digital Forensics' and 'Incident Response'. 'Incident Response' and 'Digital Forensics' are distinguished for the sake of completeness. Incident response is the first phase of active digital forensics so when active digital forensics is mentioned, incident response is mentioned by transference. Intrusion detection enables incident response, because it is impossible to respond to an incident that neither known nor detected, that in turn enables and improves Digital forensics.

'Digital Forensics' improves and enables 'Cyber Threat Intelligence' by providing indicators of compromise and threat actors tactics, techniques, and procedures (tactical and operational intelligence). Proactive digital forensics also provides strategic cyber threat intelligence as the ten steps process for forensic readiness is being implemented, providing business scope and points to where digital evidence should be collected.

'Cyber Threat Intelligence' improves 'Network Security Monitoring' by utilizing strategic threat intelligence to define NSM collection strategies, and tactical and operational intelligence with computed, behavioral, and atomic indicators of compromise for NSM detection.

'Network Security Monitoring' improves 'Intrusion Detection (NSM Detection) and Situational Awareness' by collecting as detailed data as possible about the current state of the network. This provides possibilities for advanced detection and response that goes beyond regular IDS systems.

The loop is closed, and NSM Detection provides data for NSM Analysis (active

and reactive digital forensics) that feeds 'Cyber Threat Intelligence in a continuous loop where each iteration provides better capabilities for intrusion detection and situation assessment which leads to situational awareness.

'Multisensor Data Fusion' improves 'Network Security Monitoring' by providing the capability to do better analysis. NSM collection provides a variety of data where some types of data have little to no analytical value before it is viewed together with and in relation to other data. By fusing NSM data together, a more detailed picture of a current event can be obtained. This is why 'Multisensor Data Fusion' improves 'Intrusion Detection and Situation Assessment'.

'Multisensor Data Fusion' improves 'Cyber Threat Intelligence' by fusing data together to provide information, fusing information with context to provide knowledge, and with a human in the loop to produce intelligence. The intelligence cycle is based on fusing data, information, and knowledge to obtain intelligence.

'Big Data Technology' enables 'Multisensor Data Fusion' by providing a robust platform to host fusion applications. Big data can scale to handle large volumes of data, in high velocity, variety, and veracity. The MSDF system might need to fuse several types of NSM data—structured, semi-structured, or unstructured. Regardless of this, big data technology can serve this purpose at scale.

'Big Data Technology' enables 'Network Security Monitoring' because one of the challenges that NSM faces, and one of the reasons that it is not been taken more into use, is the amount of data that it might produce. Big data technology is suited to ingest NSM data, store NSM data, and provide tools to process NSM data.

'Big Data Technology' enables 'Digital Forensics' in order to investigate and search for evidence in large amounts of data—structured, semi-structured, and unstructured. Big data technology can support active digital forensics by providing a glasses for looking into a large amount of data in order to find the needle in the haystack. Big data technology can store any data in case it is needed for a reactive investigation.

6 Proposed model for a MSDF system for ID and SA

> The world is full of obvious things which nobody by any chance ever observes.
>
> Sherlock Holmes

6.1 Requirements

Based on the study of each of the five disciplines in the background chapter, requirements for a multisensor data fusion model to address intrusion detection, situational awareness, and forensic investigation, are to be identified. A review of common fusion models and architectures, network security monitoring, digital forensics, and cyber threat intelligence, forms the baseline for a model that aims to reduce false positive alarms for intrusion detection, improve detection of unknown threats, and provide coverage for the whole cyber kill-chain.

The model also seeks to integrate human cognition reasoning capabilities with the heavy work of computational processing. Human analysts have, for now, unmatched reasoning skills that at this point can not be implemented into computer system logic. However, computers have an advantage over humans with fast processing of large amount of data. Thus, by actively combining human reasoning with computer data processing power, a model is developed for (1) guiding both academic research and security operation centers towards operative network defence, (2) developing new methods and technologies for more sophisticated network defense, detection and response, intelligence and situational assessment, and (3) providing meaningful starting point and extended toolsets to ensure successful digital forensic investigations in complex digital environments in order to address the increasing amount of cybercrime. The model seeks to cancel out the individual weaknesses of current technologies, tools, methods, and human reasoning by utilizing their advantages where they are most effective, but without the cost of being blind sighted for the attack vectors of tomorrow. In order to address this, the model must fulfill the following requirements:

Automation and manual analysis The model must support automation of analytical tasks in order to handle large data loads, but not remove the possibility for manual analysis.

Modular and minimalistic The fusion model must be modular and as simple as possible but not simpler. Flexibility and minimalism must be implemented as one does not know today the threats of tomorrow and neither how to detect them. Therefore, one of the core design requirement is to keep the complexity as low as possible. The requirements, thus, are based on the well-known Unix philosophies of making each module perform one thing well, but they should be easily replaceable if new features are needed, or if the module does not serve its purpose anymore. Each module must be designed to expect that their outputs is the input of another module.

Agile, flexible and scalable The model should enable DevOps thinking and rapid change, providing analysts with the room to adapt the tool for analysis, i.e. not adapt analysis for the tools, but still being able to handle big data loads for scalability.

Forensic sound The model must ensure forensic soundness in respect to ensuring evidence integrity and chain of custody by being a source for comprehensive digital evidence.

Support Reactive, Proactive and Active Digital Forensics The model must ensure forensic readiness (proactive) by collecting evidence to ensure the ability for successful reactive investigations. The model must also enable the possibility for acquiring volatile evidence in an active manner.

Cyber Threat Intelligence The fusion model must support the usage of external, and the production and usage of internal cyber threat intelligence.

Enhance situational awareness The platform should be designed for supporting situation assessment in order to help analysts gain and maintain situational awareness.

Human in the loop The model must put the human analyst into the fusion process, but must then also address problems related to human cognitive bias.

Centralized/Distributed fusion hybrid architecture The model must utilize a distributed architecture where individual sensors can operate autonomously with limited processing, but with a central hub for multisensor fusion in order to build situational awareness and intrusion detection both horizontally and vertically in the monitored environment.

An hybrid Complementary, Redundant and Cooperative data fusion model The model must ensure complementary data fusion in order to cover a broader and multi-perspective view on the environment. The model must ensure redundant fusion in order to comply with the forensic principle of multi-tool

verification. The fusion must be cooperative, delivering fused information from different processes, in order to provide more complex intelligence, detection and analysis, and a greater sum of sensing.

In the next section, a model is proposed to address the described requirements.

6.2 Proposed model

The multisensor fusion model (Figure 23) is a hybrid model based on a distributed and central multisensor fusion architecture. The reason for the hybrid approach is to make it modular, robust, and scalable. It is also provides a defense mechanism in order to not have one single point of failure. Even with the fall out of the central fusion architecture, each sensor can operate, do some limited processing, and store the data locally. The model is based on JDL and Visual Data Fusion models, but adapted for the network defense.

Data is collected at the bottom, and are refined into information. After the central fusion component, the data has been transformed into knowledge that a human analyst can utilize in order to refine it into intelligence.

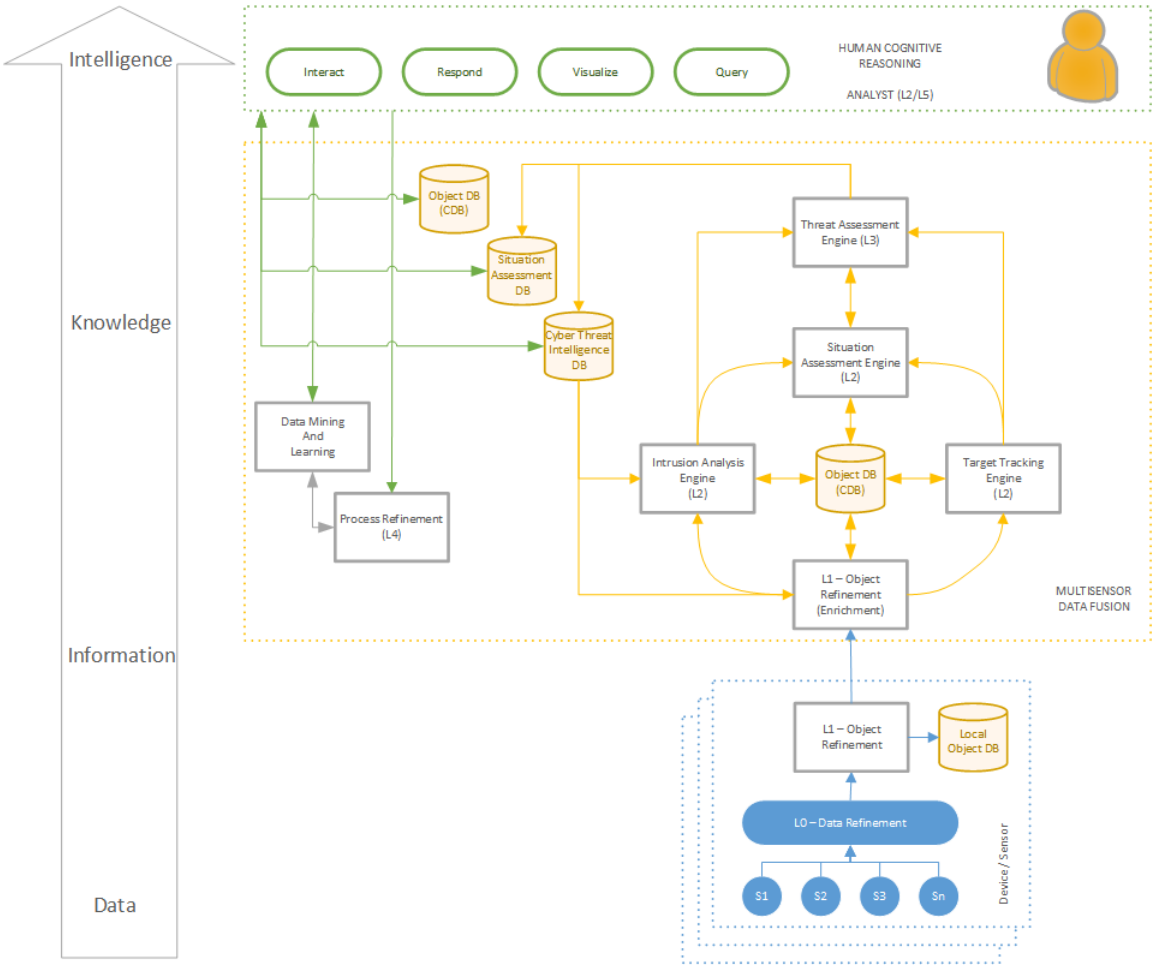


Figure 23: Proposed model for Multisensor data fusion for ID and SA

The model has three main parts: Device (or Sensor), Multisensor Data Fusion, and Human Cognitive Reasoning.

Device can have one or many sensors providing different perspectives within their view. A device can be anything, e.g., a server, client, mobile device, firewall, software, intrusion detection system, etc. Sensors are hooked into their perspective, providing data feed to the the fusion process. The sensors on the device are retaining data locally, both processed and raw, before sending it to the central fusion system.

Multisensor data fusion is the central fusion system that fuses data from multiple devices in order to perform advanced intrusion analysis, target tracking, situation assessment, and threat assessment. The central component

is built upon the big data technology in order to scale and tolerate loss in services.

Human Cognitive Reasoning is the third component of the model, and represents how the human being can interact and utilize reasoning and knowledge in the fusion processes.

6.2.1 Vertical and Horizontal Fusion

Vertical Multisensor data fusion (Figure 24) is the process of complementing evidence and artifacts from the different layers in a computer or device in order to recreate as complete picture as possible over what happened in the system during a reactive forensic investigation. When it comes to the application of intrusion detection, vertical fusion can be used to build detection logic that is contextual, aware, and is based on either anomalies or signatures detect breaches or attempted breaches. To give an example: one can detect in either the network segment or network perspective that the given host has communicated with, either an unknown system or a known bad system. This information alone has little analytical value, or detection value, and is often results in high false positive rates. But if this information is combined with information about a new executive file, that was placed in the user perspective (e.g., in a temp folder), the cryptographic hash (signature) from this file can be collected and fused with the indicators from the network perspective. The fact that the computer communicated with an unknown host or downloaded a unknown file does not necessarily mean that it is compromised. This is where the system perspective come in. If this file gets executed and spawns a new process or installs as a driver in the host perspective, then it is highly likely that this might be a new strain or a rebuilt malware.

Horizontal Multisensor data fusion (Figure 24) is the process of performing decision fusion based on decisions or features recorded from the different hosts on a given network segment, or features collected across multiple network segments (Figure 25 below) to get a clearer picture.

Each component of the model is explained in depth in the following subsections.

6.2.2 Device / Sensor (S1,S2,S3,...,Sn)

This level is marked as blue in the Figure 23. The dotted box itself represent the device. Everything inside of the box happens on the device itself. The blue circles are sensors (S1,S2,S3,...,Sn). Sensors can be places and can monitor any of the perspectives shown in Figure 24.

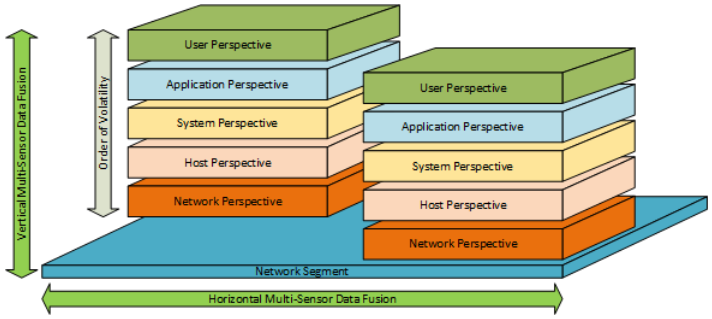


Figure 24: Sensor perspectives

Six perspectives on the device / sensor can be outlined (Figure 24):

User perspective There is a need for sensing in the user perspective as user space is the most common attack vector used by adversaries today. The user perspective in this model is split in two, cognitive and technical. The cognitive part means that the user himself is a soft-sensor who lets a security analyst know about abnormal behavior,. The technical part is indicators of compromise that exist within the users files and folders on his/her digital device.

Application perspective The application perspective can be on client devices; it can be the user applications that are not running as services or a part of the operating system itself. On a web server, the web-application is an example.

System perspective The system perspective is the services and daemons running on top of the operating system, fulfilling some function to the upper layer.

Host perspective The host perspective is all files and software that are the part of the operating system itself such as device drivers, kernel, processes, process management, network management, memory and process states, and other low-level forensic system artifacts.

Network perspective The network perspective is what can be collected from the network itself but limited to the host it belongs to. It means that this perspective it limited to network packets that are being transferred over the local physical network adapter connected to the host. The network perspective is not connected to the network segment perspective since in today's connected and mobile world, one can not be sure that collection on the network segment perspective is always possible because device might be connected to another ISP, Mobile provider, open WLAN in a cafe, etc.

Network segment perspective The network segment perspective is the network broadcast domain for all hosts connected to the given local area network, wireless wide area network (UTMS, LTE, etc.).

6.2.3 Data Refinement (L0)

Data Refinement (JDL Layer 0) is the sensing level of the model and has the capability to identify individual events in the monitored perspective. In the data refinement level, any preprocessing or error correction techniques are being applied, meaning synchronizing data from multiple sources and normalize it in a predefined data format before sending to Object refinement. Data refinement in this case is a Data in - Feature out fusion process.

6.2.4 Object Refinement (L1)

Object refinement (JDL Layer 1) combines the data from data refinement in order to identify individual security events in the monitored perspective. Because of the distributed architecture in the model, this is happening in two places.

The first Object Refinement is happening on the sensor, combining the features from the data refinement into objects containing features that describes the single event. In addition to combining observations from sensors into a single object, this level also adds metadata about the object before it is cached in the local Object DB. Metadata in this case can be information about the sensor, timestamp, classification, event type, id, etc. The features in the object itself is the observations, e.g., an alarm, netflow record, an event log entry, a process entry, a file hash, and any other atomic CTI indicator.

The second Object refinement is happening centrally, and is combining the objects sent from the different sensors into fused objects based upon some criteria, pattern matching, or classification. This can be a fusion based upon IP addresses and ports, timestamp, target system, or other identifying factors, that can tight the objects together to a single event, and then reduce the overall amount of events. At this level, the objects are being enriched with atomic or computed CTI indicators. This can happen by adding metadata helpful to the Intrusion Detection engine or target tracking engine on the next level.

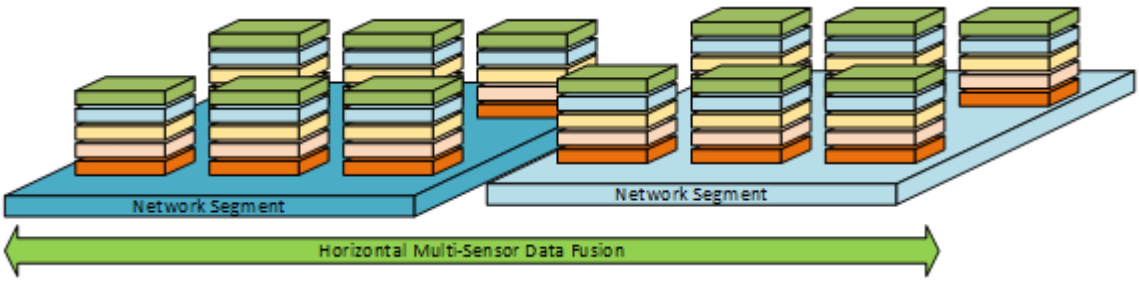


Figure 25: Distributed sensor network

6.2.5 Databases

Databases is any data store in the model, and there are two 'General Purpose Object Databases', and two 'Purpose Specific Databases':

Local Object DB Because of the distributed architecture, the objects are stored in a local object database as a local cache to be able to replay the data transfer in case of failure or corrupted data in the chain. This local object database also serves a forensic value in case of compromised system since the data in it is stored on disk. The local object database can be split into two functions, where one is storing the preprocessed data, and another one preserves the raw data. Enriched is events with added annotation and metadata like source. Forensic copy is a raw copy of the data before preprocessing or with minimal processing that does not affect evidence dynamics and ensures forensic integrity. Forensic integrity must be ensured by cryptographic hash functions, and a copy of the hash functions should be sent centrally.

Object DB (CDB) The object database is the central storage that stores all objects. This database is shared among all the processes in the model and also works as an inter process communication system.

Cyber Threat Intelligence DB Both internal threat intelligence and external threat intelligence are merged into one database for lookups. This databse contains Strategic, Tactical, and Operational cyber threat intelligence in an indexed and searchable manner for easy access and integration into the fusion stream. Indicators of compromise are tight to strategic intelligence; heuristic detection methods are tight to tactical intelligence; and reputation based or signature based detection methodologies are tied to operational intelligence.

Situation Assessment DB Situational assessment database is storing the decisions related to Situation Assessment Engine, and the predictive decisions of the Threat Assessment Level.

6.2.6 Intrusion Analysis Engine (L2)

The Intrusion Analysis engine (JDL level 2) provides deeper contextual understanding about objects from level 1. The intrusion analysis engine combines multiple perspectives in order to achieve a deeper understanding of the current state of the environment. The intrusion analysis engine is modular, and each module is performing only one type of analysis. This is where the detection playbooks are being implemented. The intrusion analysis engine can pull both operational and tactical cyber threat intelligence and is working as an modular and parallelized anomaly detection engine.

6.2.7 Target Tracking Engine (L2)

The target tracking engine (JDL Level 2) performs real-time search and tracking of targets in the environment in order to sort out objects relevant to an ongoing active digital forensic investigation.

6.2.8 Situation Assessment Engine (L2)

The situation assessment engine (JDL Level 2) is performing situational analysis in order to provide impact assessment, and monitoring a current situation in order to track how its evolvement, adversary behavior and projecting potential next evolvements.

6.2.9 Threat Assessment Engine (L3)

The Threat Assessment Engine (JDL Level 3) performs predictive analytics in order to understand the current and future threats to the network. It combines information from all Level 2 fusion engines in order to do so.

6.2.10 Data Mining and Learning

The data mining and learning module is a general knowledge discovery and classifier training module that can extract knowledge over time. This module utilizes machine learning, statistical methods, and other forms of analysis and search, and works as a glasses into all the databases.

6.2.11 Process Refinement (L4)

Process Refinement (JDL Level 4) does not add any fusion capabilities, but is a support process that is integrated into and is controlling all the other processes in the model. Process refinement controls, reconfigures, and tracks changes in the infrastructure in order to make the processes more efficient.

6.2.12 Cognitive Refinement (L5)

The security analyst is the key element in the model as the other parts are just there to serve the analyst's needs for information and detection in order to gain and maintain situational awareness, and the capability to respond to alarms and perform forensic or intrusion investigations.

Query The analyst must be able to perform ad-hoc queries that reach all of the data, information, and knowledge stored inside of the fusion system in order to perform threat hunting, find unknown unknowns, or gather information for a reactive forensic investigation.

Visualize In order to perform good quality analytics to get situational awareness, the analyst must be able to visualize any data, models, and data flow that he/she wish. This is because the human brain is able to perform visual pattern processing more efficient then by looking at text or numbers (Data / Information). By visualizing, the analyst will easier detect patterns in the data.

Respond When an intrusion is detected, or that an analyst is tracking an ongoing event or incident, he/she must be able to respond to the threat in order to do an active forensic investigation or neutralizing the threat.

Interact The analyst must be able to interact with automated models and machine learning in near real-time (online learning), with visual 'point and click' tools. This allows to fuse human reasoning and contextual understanding into, for example, classification of an event or incident.

6.3 Applicability of model

The model (Section 6.2) is applicable in real applications based on technologies that is available today, or that will be available in the near future. As seen above in related work section (Section 3), the commercial security vendors are already building a lot of the functionality required to capture evidence from the different layers in the proposed model (Figure 24). It has also been shown that vendors are working on the problem of large scale security analytics utilizing big data tools. The big data tool set identified in Section 2.4 makes building large scale and flexible fusion systems possible. Many of the open source big data tools are already 'production ready' and there are companies that provide consulting and commercial support for big data platforms, e.g., Hortonworks[1], Cloudera[2] and Elastic[3]. This means that building large-scale storage and computational architectures is easier than before. Projects like Apache Metron proves that the computing architecture is possible and feasible. The challenge lies in the development of the logic behind the data fusion methods. Therefore, the proposed model provides a road-map on generalized processes to focus the development on.

6.4 Model assessment

The model satisfies the requirements defined based on the literature study. The model seeks to integrate human cognition reasoning capabilities with the heavy work of computational processing. It is done by providing the analyst with the steering wheel to the system where he/she can run manual analysis by interacting with it, or deploy automatic analysis to the different engines that is controlled by the process refinement function. Each of the engines is just a platform to run analysis code as a minimalistic module that takes inputs and provides outputs. Engines can be used interchangeably that makes them flexible in terms of adding and removing functionality. Since the fusion system runs on the top of Apache Spark, it is scalable in terms of adding and removing compute power and storage. All logic is running as code and is floating on top of the platform. The fusion system supports the principle of forensic soundness by ensuring evidence integrity

[1] https://hortonworks.com/
[2] https://www.cloudera.com/
[3] https://www.elastic.co/

and chain of custody. The devices always store a forensic copy of what they send to the central fusion platform, but send a copy of the hash in order to ensure evidence integrity. The evidence flow in the model is designed to be tracked from source to the analyst. The model supports proactive forensics by being flexible in order to change collection policies, data types, and data processing, defined by the organization needs. The model supports active digital forensics—it is able to start collecting or rerouting evidence actively if needed—and reactive digital forensics—it is able to store the evidence both centrally and on the devices ensuring that there is always a copy available for an investigation. The model supports both production and utilization of cyber threat intelligence, and can easily integrate this data into the engines. The situation assessment can combine data from the other engines in order to provide the analyst with answers to his/her questions about the current situation, so that situational awareness can be achieved. Due to the flexibility of the model and the platform that lies underneath, the support for a hybrid fusion approach is achieved. The fusion lies in code submitted to the engines, and all data is available to the processes being executed in the engines to perform the desired fusion.

6.5 Limitations

The limitations of the model is based on its design that satisfies the purpose of the development, i.e. to be a decision support tool. The designed distributed sensor fusion will not be able to provide a real-time or near-real-time event processing and point in time detection. Detection will always happen with a certain delay in the range of some seconds to a couple of minutes or hours. This restricts the use of the model as a tool to target, track, and block threats in real-time at network scale like intrusion prevention systems do. The system is designed to be a general fusion system for fusing NSM Data with CTI, and NSM Data with human cognitive reasoning, in order to help the analyst to get the information that he/she needs. Thus, this is a system that constantly requires a human intervention to function effectively.

6.6 Implementation considerations

When designing and building a multisensor fusion system, the nature of the sensors must be taken into account, e.g. what are their resolution, type, and accuracy of the data that they collect. Sensors' functional or geographical spread must also serve as input to the architectural design. For a network security monitoring purpose, and especially with full packet capture, exporting the data to a central point would be expensive. In order to do this, the same amount of collected data must also be exported. This is why a distributed-central model is chosen. Each sensor can perform some processing and fusion locally, and only forward the results to the central multisensor fusion system. A local raw packet dump should be retained, but the query and processing of this must happen on

the device where the data is collected. A view into the raw packet data can be done with tools like Sguil[4]. Sguil facilitates a deep looking glasses into raw NSM data and supports event-driven analysis.

One of the key issues of designing a multisensor data fusion system is to define where the data flows and where to use it. This can be resolved by using Apache NiFi as a flow controller and Apache Kafka as a data bus between different processors.

[4] https://bammv.github.io/sguil/index.html

7 Discussion and Implications

> That's the beauty of argument, if you argue correctly, you're never wrong.
>
> Christopher Buckley

In the previous chapters, theory and related work have been presented, methodology has been explained, and relationships between five domains that create a base for the proposed model are described. Moreover, a model for multisensor data fusion for intrusion detection and situational awareness has been proposed.

The key problem addressed by this thesis is intrusion detection and situational awareness, and utilization of data fusion from multiple sensors to achieve it. One of the main problems with intrusion detection is the often high false positive rates with anomaly detection approaches, and the lack of detecting unknown attacks with signature detection approaches. The drawbacks of intrusion detection systems is that they are not context aware and intelligent, this is where situational awareness comes in. Situational awareness is a cognitive state of an analyst or a group of analysts. By fusing human situational awareness into the intrusion detection process, it is believed that unknown threats can be detected, and the false positive rate decreased. Situational awareness, can be fed into the intrusion detection system by actively writing rules or define indicators of compromise based on either cyber threat intelligence, or digital forensics.

7.1 Theoretical implications

In this thesis, an approach to provide better intrusion detection and situational awareness has been proposed. The thesis covers multiple domains within informatics in order to cover the breadth and complexity of detecting threats in modern, complex, distributed, and high bandwidth networks. The adversaries are constantly evolving their tactics, techniques, and procedures in order to avoid detection long enough to reach their malice goals.

To answer first research question, literature study in combination with questionnaire has been used. The answer is simple, but yet complex. Network Security Monitoring defines several data types (sources) (2.2.2): full packet capture data, packet string data, session data, statistical data, log data, alert data, and meta data. These are the general categories of data available for collection, but

to define what data to collect is a harder answer to give. To define what to collect, a combined approach of proactive digital forensics (2.1.3) and cyber threat intelligence (**??**) is needed. Proactive forensics and cyber threat intelligence take business objectives as basis to determine what to collect. The business case and policies in proactive forensics provide the answer of what evidence is important for the organization to collect in order to protect their information, and by defining this, the probability of a successful investigation rises while the cost of performing investigations gets lower. Cyber threat intelligence defines the threats against the assets that the organization wants to protect, and by continuously consume and produce threat intelligence, the data collection strategies must adapt as the threat landscape adapts.

To answer to second and third research questions, the proposed model has the goal to support data processing for intrusion detection and situational awareness. The appropriate approach should include three types of fusion: horizontal within one network segment, horizontal between several network segments, and vertical between perspectives on the device. Moreover, this three types of fusion should be combined with human cognitive reasoning in order to support principles of forensic soundness, proactive, active and reactive digital forensics, and to ensure flexibility and scalability of data processing. Hybrid data fusion enables efficient combination of data from different sensors that enhance both intrusion detection and situational awareness.

8 Conclusions

In this thesis, the domains of digital forensics, network security monitoring, cyber threat intelligence, big data principles and technology, and multisensor data fusion have been studied to understand how they are related to each other. It has been established that they are closely related and sometimes overlap. Together they can, in theory, both enable and improve each others performance significantly. Based on how these domains relate and fit together, solving and limiting each others weaknesses, a model for a multisensor data fusion system has been proposed. The model incorporates human cognitive reasoning, and proposes a new approach to fuse data from sensors with different perspectives on both devices and in the network, providing the concept of horizontal and vertical fusion.

9 Further work

Based on the literature study, a combination of domains, and the developed model, the three directions of further research are proposed.

Applied cyber threat intelligence

Academia mainly focuses on how to share cyber threat intelligence, and how to fuse threat intelligence with threat intelligence in an effective and secure manner. But there is very little done academically on how to produce, use and integrate cyber threat intelligence, and how to measure the quality of cyber threat intelligence. In this thesis, it is discovered that cyber threat intelligence is an integral part of both network security monitoring and intrusion detection, digital forensics and so on in a way that make it comparable to cyber risk and security governance. There is a research gap in how to produce, apply, and measure quality of actionable cyber threat intelligence.

Applied cognition (human - machine interaction)

There has been some research about how to build better interfaces and information sharing between humans and computer. But there is a gap in research how to make human cognition an integrated part of multisensor data fusion. Application of this kind of research can for instance be for near-real-time decision making where the human analyst can select and classify features on the fly, actively helping the system and algorithms to learn new classes while being used (Online machine learning).

Multisensor fusion for intrusion detection

This thesis was theoretical and form the base theory for sensor fusion for intrusion detection and propose a model that is flexible enough and suited for doing research in sensor fusion on NSM data. But there is still applied research to be done on how different types of NSM Data and sensors can be fused, and the results of doing so.

Bibliography

[1] van der Steen, M. & Blom, M. A roadmap for future forensic research. Technical report, Technical report, Netherlands Forensic InsTtute (NFI), The Hague, The Netherlands, 2007.

[2] Axelsson, S. 2015. Imt4012 digital forensics, lecture 1.

[3] Sanders, C. & Smith, J. 2013. *Applied network security monitoring: collection, detection, and analysis*. Elsevier.

[4] Bollinger, J., Enright, B., & Valites, M. 2015. *Crafting the InfoSec Playbook: Security Monitoring and Incident Response Master Plan*. " O'Reilly Media, Inc.".

[5] Dempsey, M. 2013. Joint intelligence. *Joint Publication*, 2–0.

[6] Liska, A. 2014. *Building an Intelligence-led Security Program*. Syngress.

[7] Forgeat, J. 2015. Data processing architectures - lamda and kappa. https://www.ericsson.com/research-blog/data-knowledge/data-processing-architectures-lambda-and-kappa/. Accessed 25.04.2016.

[8] Apache flume. http://flume.apache.org. Accessed 25.04.2016.

[9] Grover, M., Malaska, T., Seidman, J., & Shapira, G. 2015. *Hadoop Application Architectures*. " O'Reilly Media, Inc.".

[10] Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. 2015. *Learning spark: lightning-fast big data analysis*. " O'Reilly Media, Inc.".

[11] Apache spark streaming programming guide. https://spark.apache.org/docs/latest/streaming-programming-guide.html. Accessed 25.04.2016.

[12] Foo, P. H. & Ng, G. W. 2013. High-level information fusion: An overview. *J. Adv. Inf. Fusion*, 8(1), 33–72.

[13] Apache metron (incubator). https://cwiki.apache.org/confluence/display/METRON/Metron+Architecture. Accessed 25.04.2016.

[14] Farmer, D. & Venema, W. 2005. *Forensic discovery*, volume 6. Addison-Wesley Upper Saddle River.

[15] 2017. Pst trusselvurdering 2017. http://www.pst.no/media/82444/PST_Trusselvurd-2017.pdf. [ONLINE] (NORWEGIAN) Accessed 07.12.2016.

[16] 2017. Nsm risiko 2017 (norwegian). https://nsm.stat.no/globalassets/rapporter/rapport-om-sikkerhetstilstanden/nsm_risiko_2017_lr_0404_enkelts_v3.pdf. [ONLINE] (NORWEGIAN) Accessed 07.12.2016.

[17] 2017. The norwegian intelligence service - focus 2017. https://forsvaret.no/fakta_/ForsvaretDocuments/Fokus2017_2002_ENGELSK_v2.pdf. [ONLINE] Accessed 07.12.2016.

[18] 2017. Symantec internet security threat report 2017. https://www.symantec.com/security-center/threat-report. [ONLINE] Accessed 07.12.2016.

[19] 2017. mnemonic security report 2017. https://www.mnemonic.no/SecurityReport. [ONLINE] Accessed 07.12.2016.

[20] 2017. Cisco - 2017 annual cybersecurity report. https://www.cisco.com/c/en/us/products/security/security-reports.html. [ONLINE] Accessed 07.12.2016.

[21] Årnes, A., Flaglien, A. O., Sunde, I. M., Dilijonaite, A., Hamm, J., Sandvik, J.-P., Bjelland, P. C., Franke, K., & Axelsson, S. 2016. Digital forensics, textbook for imt3551, imt4009, imt4012 and imt4114.

[22] Franke, K. & Srihari, S. N. *Computational Forensics: An Overview*, 1–10. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. URL: http://dx.doi.org/10.1007/978-3-540-85303-9_1, doi:10.1007/978-3-540-85303-9_1.

[23] Nelson, B., Phillips, A., & Steuart, C. 2010. *Guide to Computer Forensics and Investigations*. Course Technology, 4th edition.

[24] Raghavan, S. 2013. Digital forensic research: current state of the art. *CSI Transactions on ICT*, 1(1), 91–114.

[25] Grobler, C., Louwrens, C., & Von Solms, S. H. 2010. A framework to guide the implementation of proactive digital forensics in organisations. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, 677–682. IEEE.

[26] Alharbi, S., Weber-Jahnke, J., & Traore, I. 2011. The proactive and reactive digital forensics investigation process: A systematic literature review. In *Information Security and Assurance*, 87–100. Springer.

[27] Ademu, I. O., Imafidon, C. O., & Preston, D. S. 2011. A new approach of digital forensic model for digital forensic investigation. *Int. J. Adv. Comput. Sci. Appl*, 2(12), 175–178.

[28] Kittelsen, J., Franke, K., & Hämmerli, B. 2010. Digital forensics ontology framework.

[29] Laliberte, S. & Gupta, A. 2004. The role of computer forensics in stopping executive fraud.

[30] Soanes, C. & Hawker, S. 2005. Compact oxford dictionary.

[31] Stelfox, P. 2013. *Criminal investigation: An introduction to principles and practice*. Routledge.

[32] Tan, J. 2001. Forensic readiness. *Cambridge, MA:@ Stake*, 1–23.

[33] Rowlingson, R. 2004. A ten step process for forensic readiness. *International Journal of Digital Evidence*, 2(3), 1–28.

[34] Roger, A. E. & Achille, M. M. 2012. Multi-perspective cybercrime investigation process modeling. *Int J Appl Inf Syst (IJAIS)*, 2, 14–20.

[35] Grobler, C., Louwrens, C., & von Solms, S. H. 2010. A multi-component view of digital forensics. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, 647–652. IEEE.

[36] Cowen, D. 2013. *Computer Forensics InfoSec Pro Guide*. McGraw Hill Professional.

[37] Bejtlich, R. 2004. *The Tao of network security monitoring: beyond intrusion detection*. Pearson Education.

[38] Bejtlich, R. 2013. *The practice of network security monitoring: understanding incident detection and response*. No Starch Press.

[39] Dodge, Y. 2006. *The Oxford dictionary of statistical terms*. Oxford University Press on Demand.

[40] Chuvakin, A., Schmidt, K., & Phillips, C. 2012. *Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management*. Newnes.

[41] Bace, R. G. 2000. *Intrusion Detection*. Macmillan Technical Publishing.

[42] Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., & Tung, K.-Y. 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1), 16–24.

[43] Brotherston, L. & Berlin, A. 2017. *Defensive Security Handbook: Best Practices for Securing Infrastructure*. " O'Reilly Media, Inc.".

[44] Debar, H., Dacier, M., & Wespi, A. 1999. Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8), 805–822.

[45] Porras, P. A. & Valdes, A. 1998. Live traffic analysis of tcp/ip gateways. In *NDSS*.

[46] Dod dictionary of military and associated terms. http://www.dtic.mil/doctrine/new_pubs/dictionary.pdf. [ONLINE] Accessed 22.05.2017.

[47] McMillan, R. Definition: Threat intelligence. https://www.gartner.com/doc/2487216/definition-threat-intelligence. Accessed 22.04.2016.

[48] Friedman, J. & Bouchard, M. 2015. *Definitive guide to Cyber Threat Intelligence*. Cyberedge Press.

[49] NCCgroup. Threat intelligence: Benefits for the enterprise. https://www.nccgroup.trust/globalassets/our-research/uk/whitepapers/2015/11/ncc-group-threat-intelligence-paperpdf. Accessed 19.05.2016.

[50] Hutchins, E. M., Cloppert, M. J., & Amin, R. M. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1, 80.

[51] Endsley, M. R. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1), 32–64.

[52] Tadda, G. P. & Salerno, J. S. 2010. Overview of cyber situation awareness. In *Cyber situational awareness*, 15–35. Springer.

[53] Barford, P., Dacier, M., Dietterich, T. G., Fredrikson, M., Giffin, J., Jajodia, S., Jha, S., Li, J., Liu, P., Ning, P., et al. 2010. Cyber sa: Situational awareness for cyber defense. In *Cyber Situational Awareness*, 3–13. Springer.

[54] Marz, N. & Warren, J. 2015. *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co.

[55] Allen, S. T., Jankowski, M., & Pathirana, P. 2015. *Storm Applied: Strategies for real-time event processing*. Manning Publications Co.

[56] Big data taxonomy. https://cloudsecurityalliance.org/download/big-data-taxonomy/. Accessed 25.04.2016.

[57] Kreps, J. 2014. Questioning the lambda architecture. https://www.oreilly.com/ideas/questioning-the-lambda-architecture. Accessed 25.04.2016.

[58] Apache nifi. https://nifi.apache.org/. Accessed 15.05.2017.

[59] Apache kafka. https://kafka.apache.org. Accessed 25.04.2016.

[60] White, T. 2012. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.".

[61] Apache scoop. http://Scoop.apache.org. Accessed 25.04.2016.

[62] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. 2010. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, 1–10. IEEE.

[63] Apache hbase. http://hbase.apache.org. Accessed 25.04.2016.

[64] Apache cassandra. https://cassandra.apache.org/. Accessed 25.04.2016.

[65] Apache accumulo. https://accumulo.apache.org/. Accessed 25.04.2016.

[66] Elastic search. https://www.elastic.co/products/elasticsearch. Accessed 25.04.2016.

[67] Apache lucene. https://lucene.apache.org. Accessed 25.04.2016.

[68] Gormley, C. & Tong, Z. 2015. *Elasticsearch: The Definitive Guide*. " O'Reilly Media, Inc.".

[69] Dean, J. & Ghemawat, S. 2008. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.

[70] Apache storm. https://storm.apache.org/. Accessed 25.04.2016.

[71] Apache spark. https://spark.apache.org. Accessed 25.04.2016.

[72] Apache spark streaming. https://spark.apache.org/streaming/. Accessed 25.04.2016.

[73] Apache samza. https://samza.apache.org/. Accessed 25.04.2016.

[74] Apache flink. https://flink.apache.org. Accessed 25.04.2016.

[75] Apache pig. https://pig.apache.org. Accessed 25.04.2016.

[76] Apache hive. https://hive.apache.org. Accessed 25.04.2016.

[77] Hall, D. L. & Llinas, J. 1997. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1), 6–23.

[78] Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. 2013. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1), 28–44.

[79] White, F. E. Data fusion lexicon. Technical report, DTIC Document, 1991.

[80] Castanedo, F. 2013. A review of data fusion techniques. *The Scientific World Journal*, 2013.

[81] Waltz, E., Llinas, J., et al. 1990. *Multisensor data fusion*, volume 685. Artech house Boston.

[82] Varshney, P. K. 1997. Multisensor data fusion. *Electronics & Communication Engineering Journal*, 9(6), 245–253.

[83] Hall, D. L. & McMullen, S. A. 2004. *Mathematical techniques in multisensor data fusion*. Artech House.

[84] Durrant-Whyte, H. F. 1988. Sensor models and multisensor integration. *The international journal of robotics research*, 7(6), 97–113.

[85] Boyd, J. R. 1987. *A discourse on winning and losing*. publisher not identified.

[86] Kessler, O. e. a., Askin, K., Beck, N., Lynch, J., White, F., Buede, D., Hall, D., & Llinas, J. 1992. Functional description of the data fusion process. *Office of Naval Technology, Naval Air Development Center, Warminster, PA*, 16.

[87] Markin, M., Harris, C., Bernhardt, M., Austin, J., Bedworth, M., Greenway, P., Johnston, R., Little, A., & Lowe, D. 1997. Technology foresight on data fusion and data processing. *The royal aeronautical society*.

[88] Dasarathy, B. V. 1997. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1), 24–38.

[89] Karakowski, J. 1998. 'towards visual data fusion. *NSSDF International Open Session*.

[90] Bedworth, M. & O'Brien, J. 2000. The omnibus model: a new model of data fusion? *IEEE Aerospace and Electronic Systems Magazine*, 15(4), 30–36.

[91] Endsley, M. R. 1994. Situation awareness in dynamic human decision making: Theory. *Situational awareness in complex systems*, 27–58.

[92] Kokar, M. M., Bedworth, M. D., & Frankel, C. B. 2000. Reference model for data fusion systems. In *AeroSense 2000*, 191–202. International Society for Optics and Photonics.

[93] Shahbazian, E., Blodgett, D. E., & Labbé, P. 2001. The extended ooda model for data fusion systems. In *Proceedings of 4th International Conference on Information Fusion*.

[94] Brehmer, B. 2005. The dynamic ooda loop: Amalgamating boyd's ooda loop and the cybernetic approach to command and control. In *Proceedings of the 10th international command and control research technology symposium*, 365–368.

[95] Steinberg, A. N. & Bowman, C. L. 2004. Rethinking the jdl data fusion levels. *NSSDF JHAPL*, 38, 39.

[96] Steinberg, A. N. & Bowman, C. L. 2008. Revisions to the jdl data fusion model. In *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*, 45–67. CRC Press.

[97] Blasch, E., Steinberg, A., Das, S., Llinas, J., Chong, C., Kessler, O., Waltz, E., & White, F. 2013. Revisiting the jdl model for information exploitation. In *Information Fusion (FUSION), 2013 16th International Conference on*, 129–136. IEEE.

[98] Ballora, M., Giacobe, N. A., McNeese, M., & Hall, D. L. 2012. Information data fusion and computer network defense. *Situational Awareness in Computer Network Defense: Principles, Methods and Applications*, 141–164.

[99] Roy, J. & Wark, S. 2007. *Concepts, models, and tools for information fusion*. Artech House.

[100] Esteban, J., Starr, A., Willetts, R., Hannah, P., & Bryanston-Cross, P. 2005. A review of data fusion models and architectures: towards engineering guidelines. *Neural Computing & Applications*, 14(4), 273–281.

[101] Cisco advanced malware protection. https://www.cisco.com/c/en/us/products/security/advanced-malware-protection/index.html. [ONLINE] Accessed 22.05.2017.

[102] Carbon black. https://www.carbonblack.com/products/cb-response/. [ONLINE] Accessed 22.05.2017.

[103] Fireeye. https://www.fireeye.com/. [ONLINE] Accessed 22.05.2017.

[104] Sqrrl: Target. hunt. disrupt. https://sqrrl.com/. [ONLINE] Accessed 22.05.2017.

[105] Palantir: Cyber security. https://www.palantir.com/solutions/cyber/. [ONLINE] Accessed 22.05.2017.

[106] Ibm i2 analyze. https://www.ibm.com/us-en/marketplace/enterprise-intelligence-analysis. [ONLINE] Accessed 22.05.2017.

[107] Ibm qradar. https://www-03.ibm.com/software/products/no/qradar. [ONLINE] Accessed 22.05.2017.

[108] Cisco opensoc. https://opensoc.github.io/. Accessed 25.04.2016.

[109] Apache metron (incubator). https://metron.incubator.apache.org/. Accessed 25.04.2016.

[110] Sirota, J., Porter, C., & Bittman, M. Apache metron proposal (final). https://wiki.apache.org/incubator/MetronProposal. Accessed 15.08.2016.

[111] Bass, T. 2000. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4), 99–105.

[112] Thomas, C. & Balakrishnan, N. 2012. Usefulness of sensor fusion for security incident analysis. *Situational Awareness in Computer Network Defense: Principles, Methods and Applications: Principles, Methods and Applications*, 165–180.

[113] Lacerda, D. P., Antunes, J. A. V., & Dresch, A. 2015. Design science research: A method for science and technology advancement.

[114] Voordijk, H. 2011. Construction management research at the interface of design and explanatory science. *Engineering, construction and architectural management*, 18(4), 334–342.

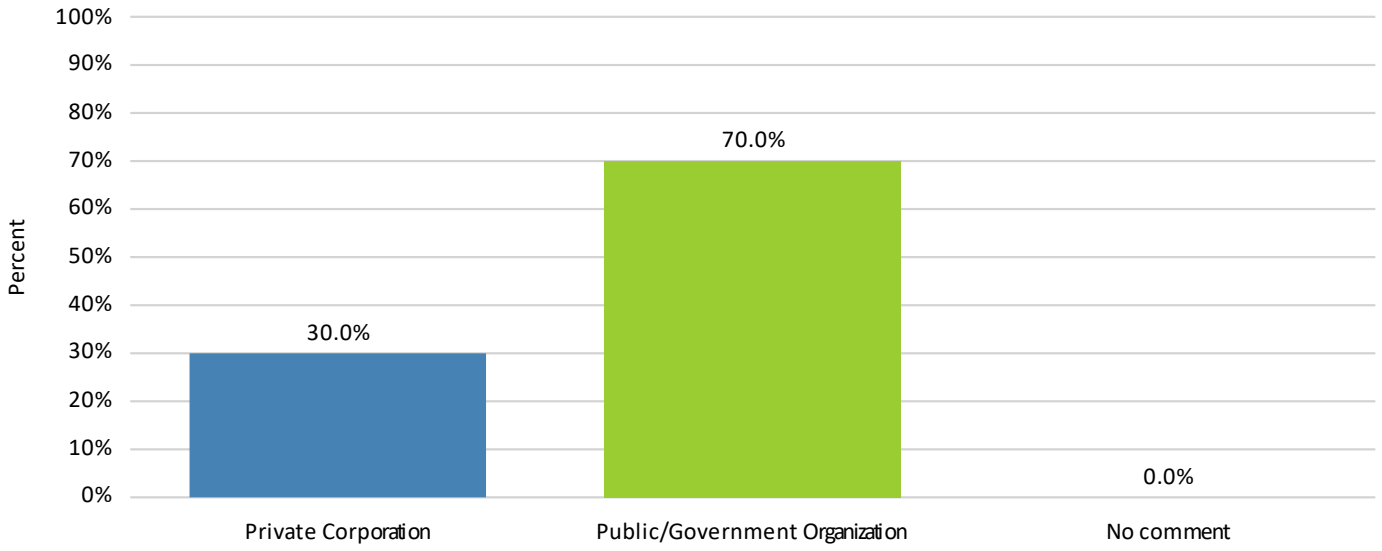
[115] Hevner, A. & Chatterjee, S. 2010. *Design research in information systems: theory and practice*, volume 22. Springer Science & Business Media.

Appendix

A Questionnaire

Multisensor Fusion for Intrusion Detection and Situational Awareness Questionaire

1. Is your organization a:



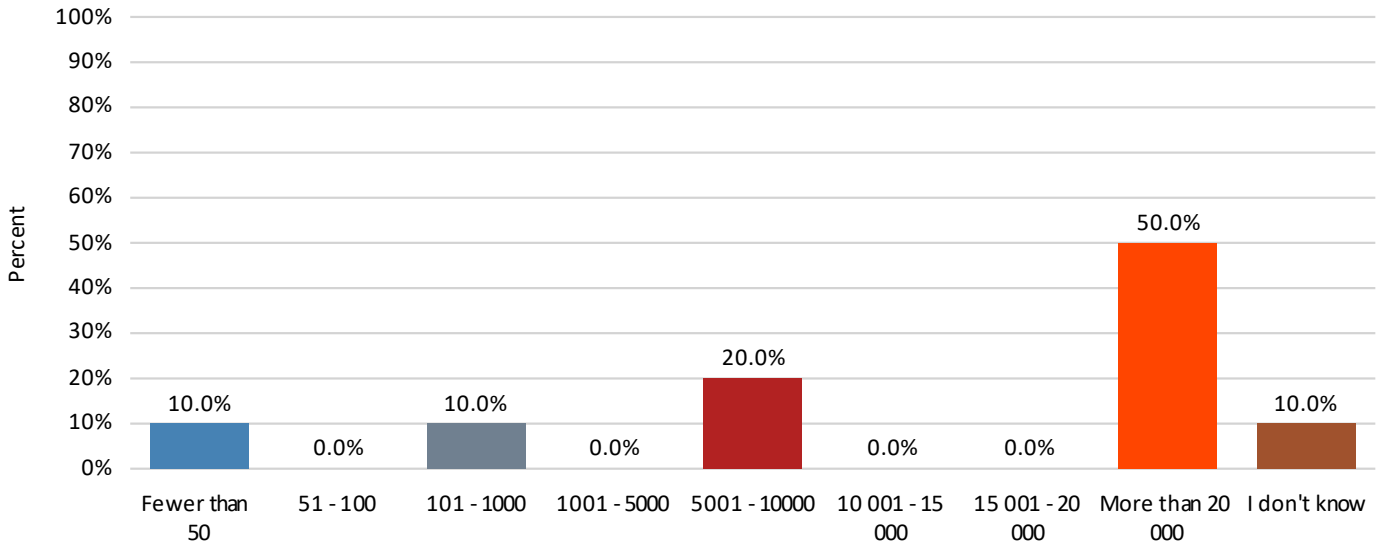
Name	Percent
Private Corporation	30.0%
Public/Government Organization	70.0%
No comment	0.0%
N	10

2. How large is your constituency?

The Term Constituency

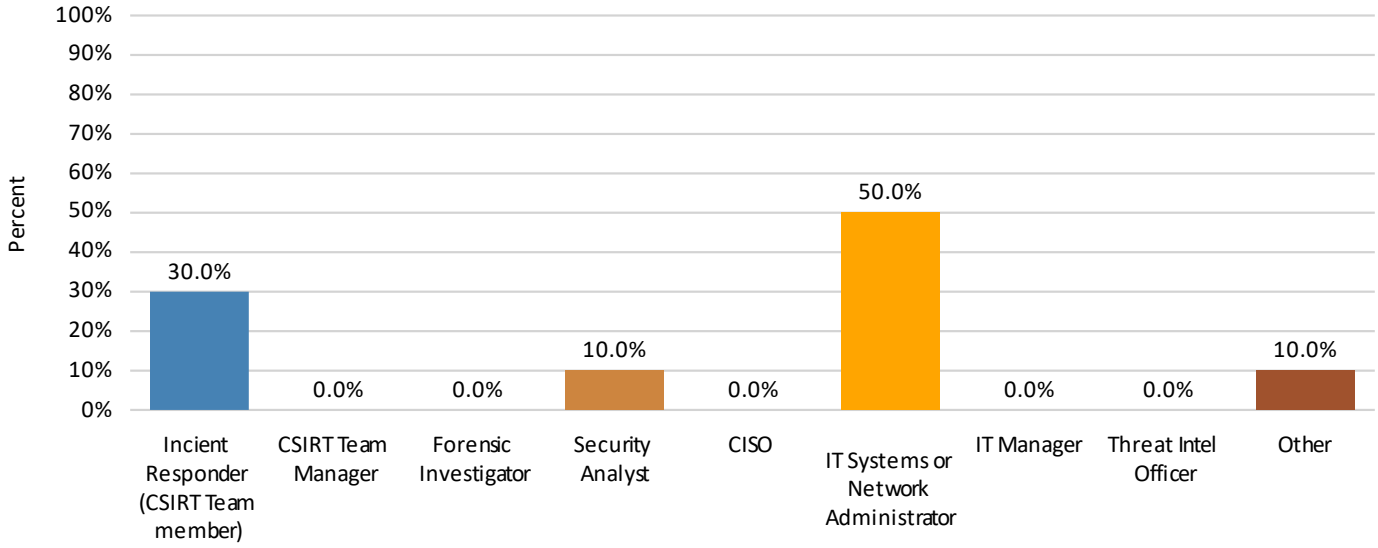
From now on the (in the CERT communities) well established term 'constituency' will be used to refer to the customer base or the served group of users of a CERT. A single customer will be addressed as ‘constituent’, a group as 'constituents'.

Ref: ENISA



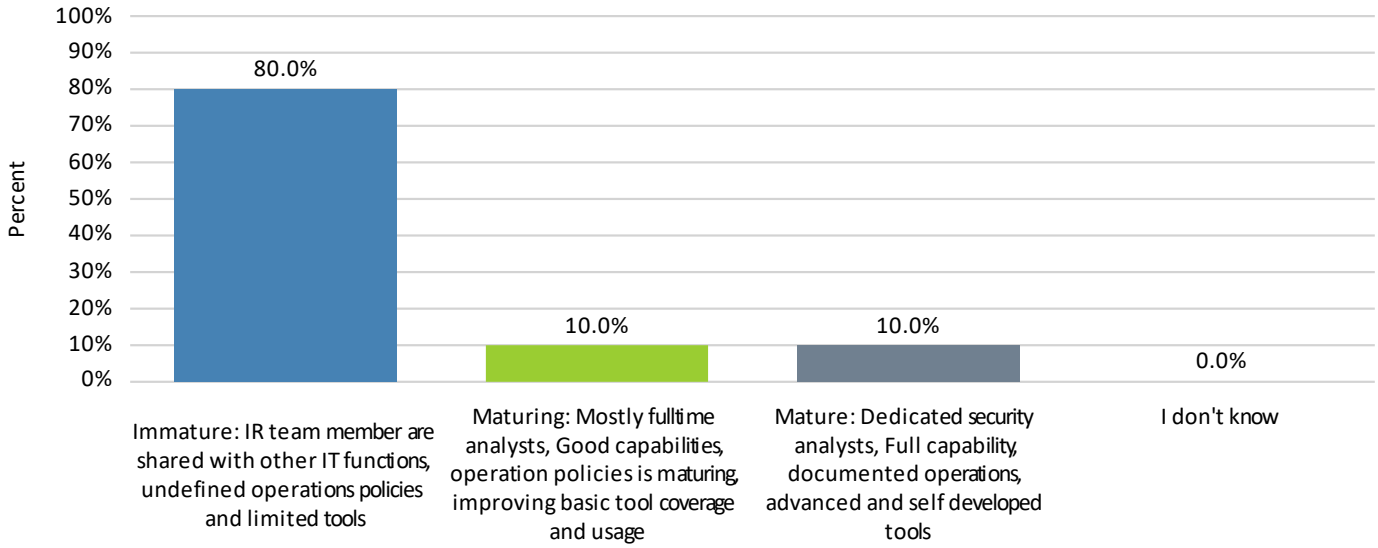
Name	Percent
Fewer than 50	10.0%
51 - 100	0.0%
101 - 1000	10.0%
1001 - 5000	0.0%
5001 - 10000	20.0%
10 001 - 15 000	0.0%
15 001 - 20 000	0.0%
More than 20 000	50.0%
I don't know	10.0%
N	10

3. What is your domain expertise?



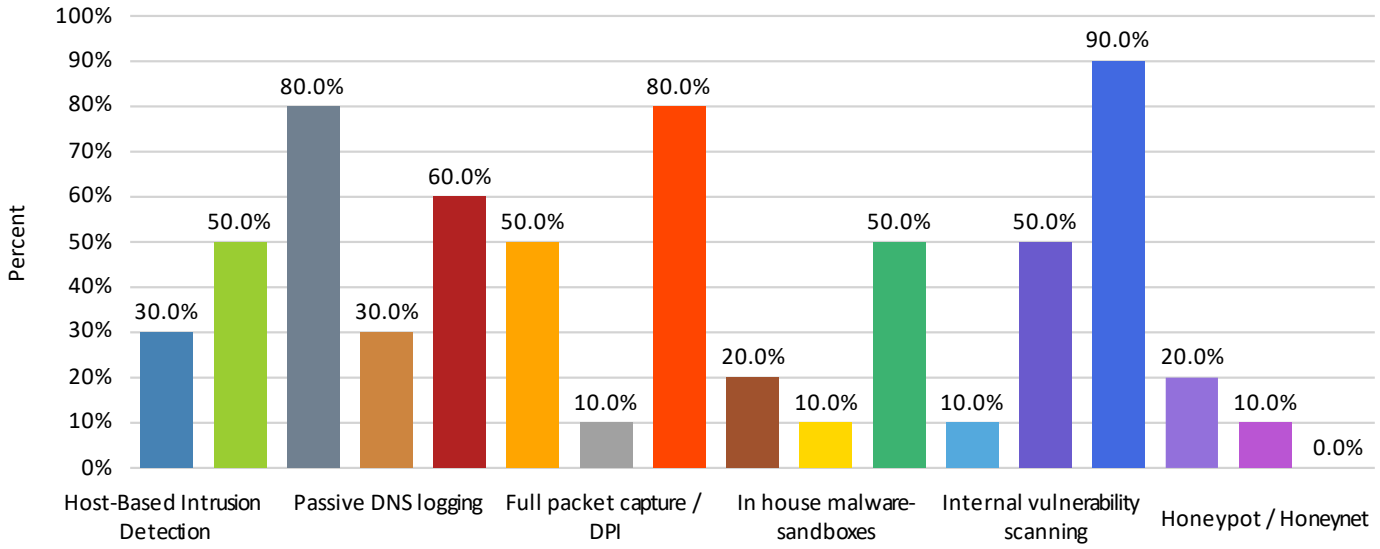
Name	Percent
Incient Responder (CSIRT Team member)	30.0%
CSIRT Team Manager	0.0%
Forensic Investigator	0.0%
Security Analyst	10.0%
CISO	0.0%
IT Systems or Network Administrator	50.0%
IT Manager	0.0%
Threat Intel Officer	0.0%
Other	10.0%
N	10

4. How mature is your security team/operations?



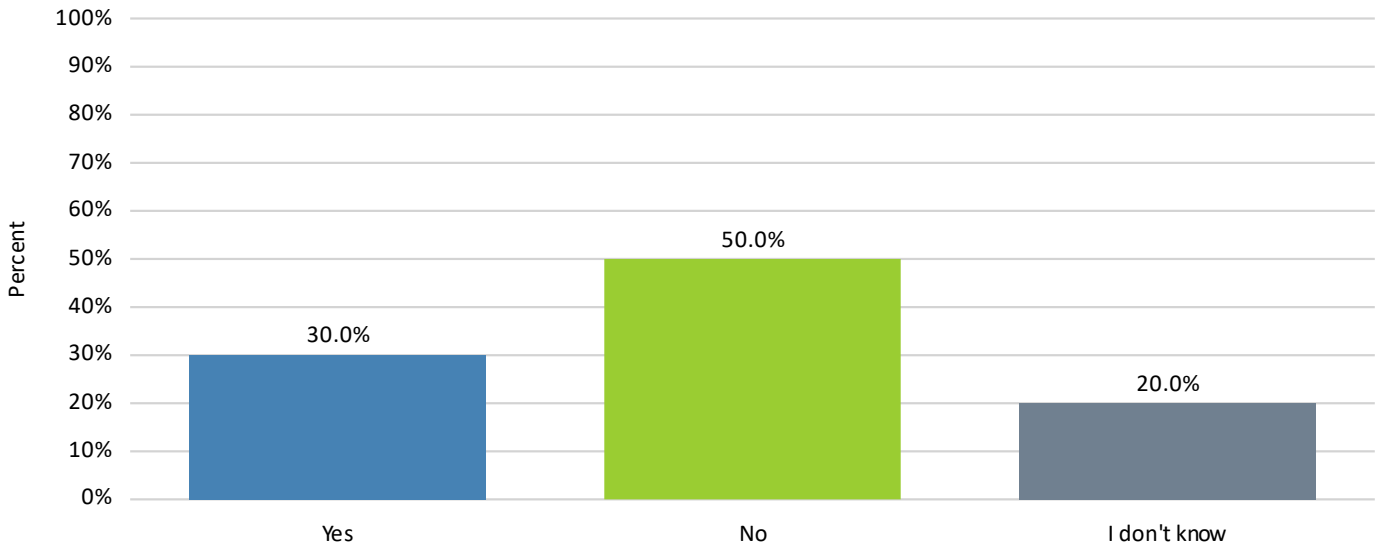
Name	Percent
Immature: IR team member are shared with other IT functions, undefined operations policies and limited tools	80.0%
Maturing: Mostly fulltime analysts, Good capabilities, operation policies is maturing, improving basic tool coverage and usage	10.0%
Mature: Dedicated security analysts, Full capability, documented operations, advanced and self developed tools	10.0%
I don't know	0.0%
N	10

5. Does your organization use:



Name	Percent
Host-Based Intrusion Detection	30.0%
Network-based Intrusion Detection	50.0%
Netflow/sflow etc	80.0%
Passive DNS logging	30.0%
Central Syslog/Eventlog collection	60.0%
Central Application log collection	50.0%
Full packet capture / DPI	10.0%
Firewall (Packetfilter)	80.0%
Firewall (Next-gen)	20.0%
In house malware-sandboxes	10.0%
Endpoint protection software (Traditional Antimalware)	50.0%
Endpoint protection software (Next-Generation)	10.0%
Internal vulnerability scanning	50.0%
Network Monitoring system (E.g. Nagios)	90.0%
Threat Intelligence feeds	20.0%
Honeypot / Honeynet	10.0%
Other	0.0%
N	10

6. Do you combine data from these tools?



Name	Percent
Yes	30.0%
No	50.0%
I don't know	20.0%
N	10

7. How do you combine data from these tools?

Correlation and aggregation.

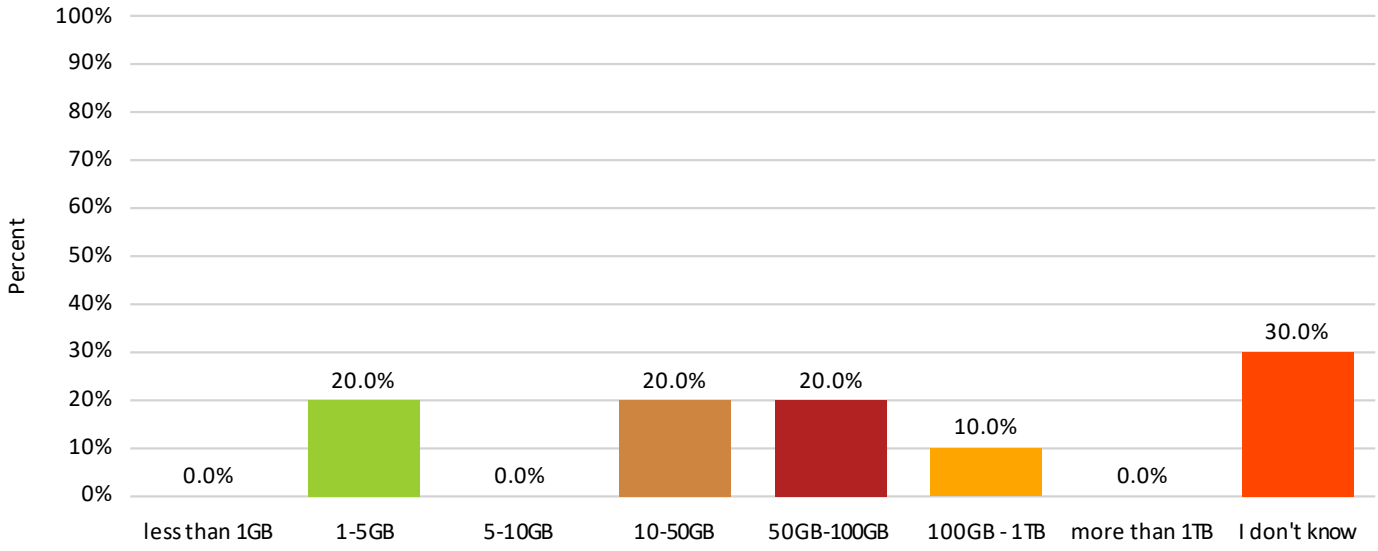
Some automated combination tools are used, some combination is done manually for the case at hand.

8. For what purpose do you combine data from these tools?

Increasing situational awareness. More efficient analysis.

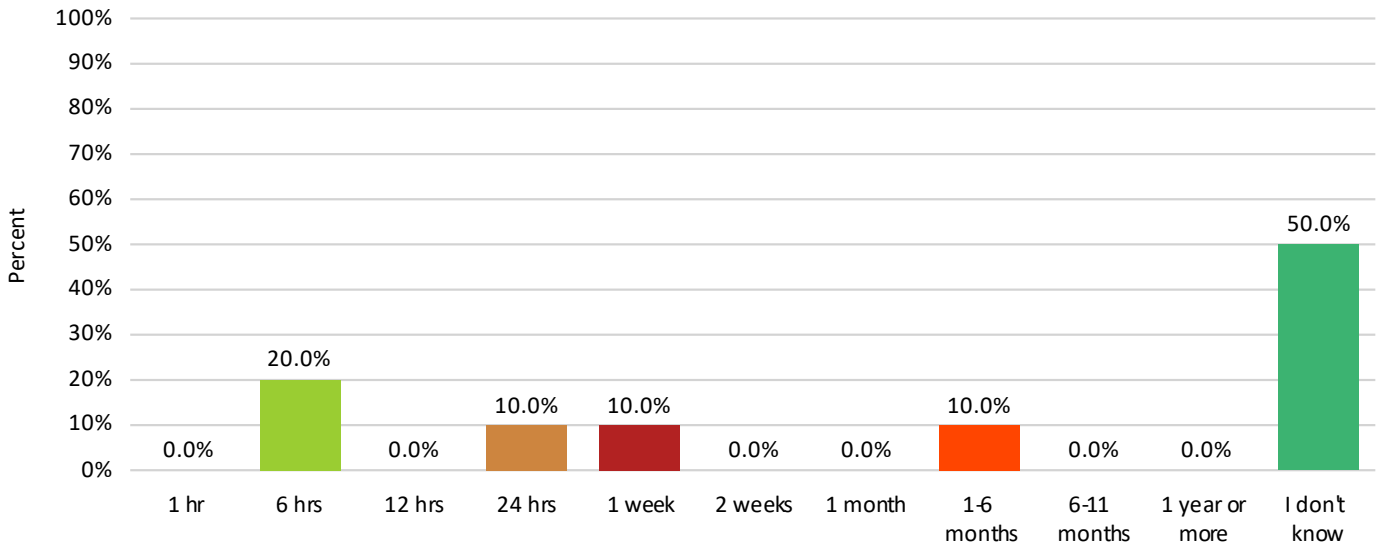
For identifying ongoing malicious activity and stopping it, and for identifying patterns that can be used to identify installed backdoors etc., so that future malicious activity can be preventet.

9. What is the daily estimated storage needs for event collection and analysis?



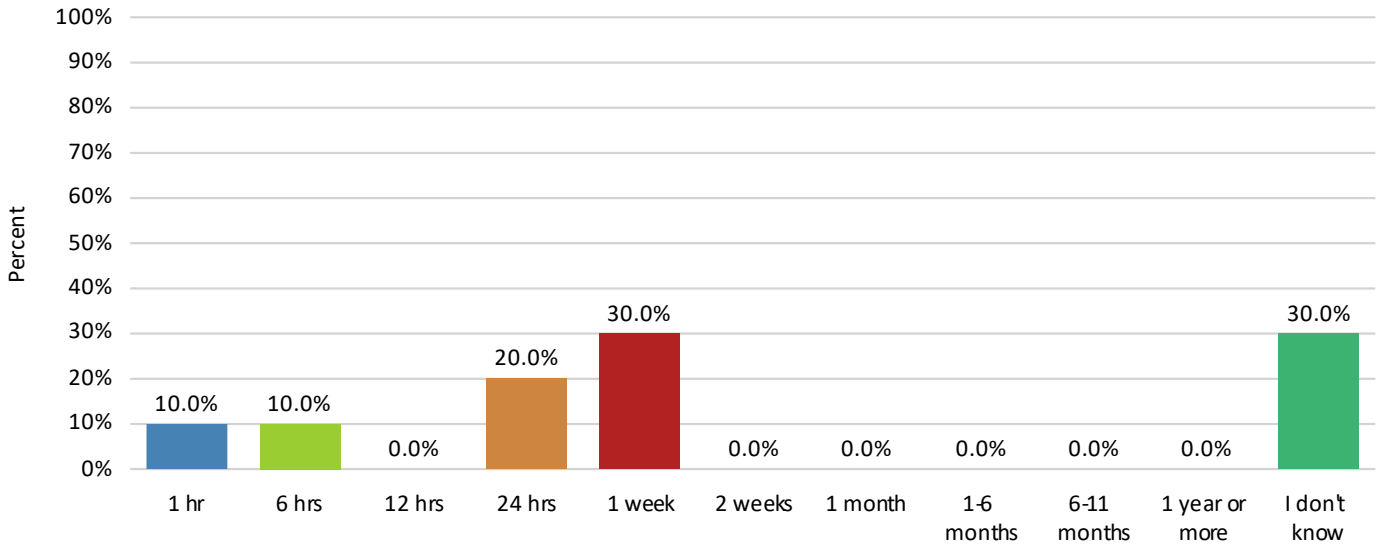
Name	Percent
less than 1GB	0.0%
1-5GB	20.0%
5-10GB	0.0%
10-50GB	20.0%
50GB-100GB	20.0%
100GB - 1TB	10.0%
more than 1TB	0.0%
I don't know	30.0%
N	10

10. Average time of detection for a compromise (up to)



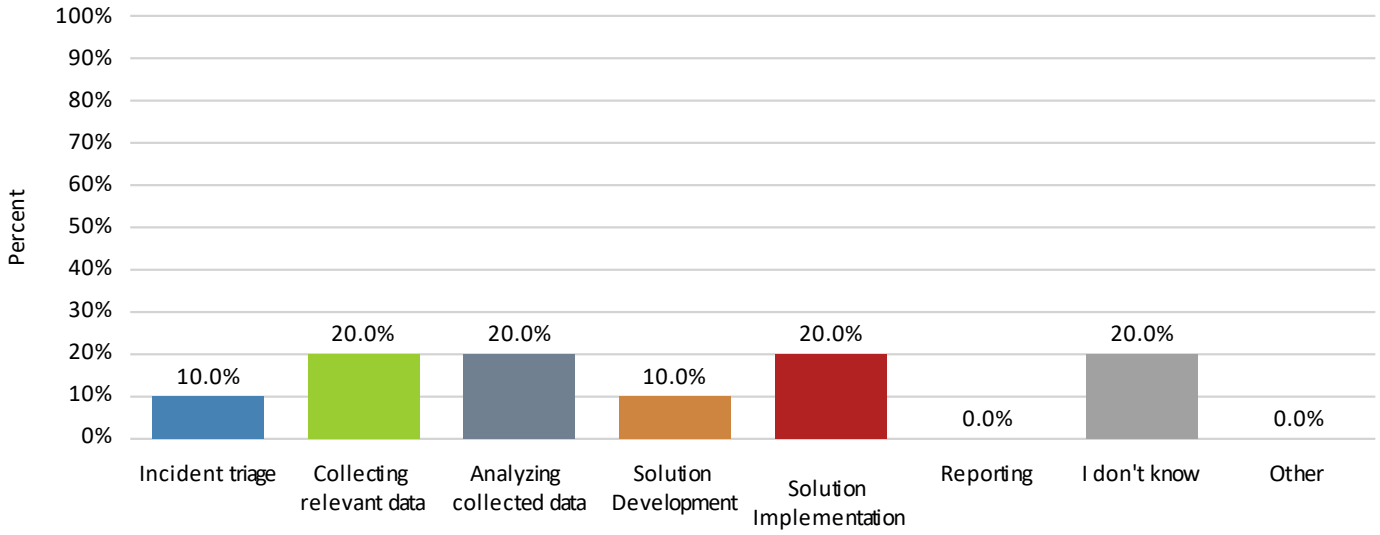
Name	Percent
1 hr	0.0%
6 hrs	20.0%
12 hrs	0.0%
24 hrs	10.0%
1 week	10.0%
2 weeks	0.0%
1 month	0.0%
1-6 months	10.0%
6-11 months	0.0%
1 year or more	0.0%
I don't know	50.0%
N	10

11. Average time from detection to remediation (up to)



Name	Percent
1 hr	10.0%
6 hrs	10.0%
12 hrs	0.0%
24 hrs	20.0%
1 week	30.0%
2 weeks	0.0%
1 month	0.0%
1-6 months	0.0%
6-11 months	0.0%
1 year or more	0.0%
I don't know	30.0%
N	10

12. What is the most time-consuming part of your incident resolving activities today?



Name	Percent
Incident triage	10.0%
Collecting relevant data	20.0%
Analyzing collected data	20.0%
Solution Development	10.0%
Solution Implementation	20.0%
Reporting	0.0%
I don't know	20.0%
Other	0.0%
N	10

13. Which challenges do you currently have with intrusion detection?

Not dedicated time for it.

lack of tools
lack of time

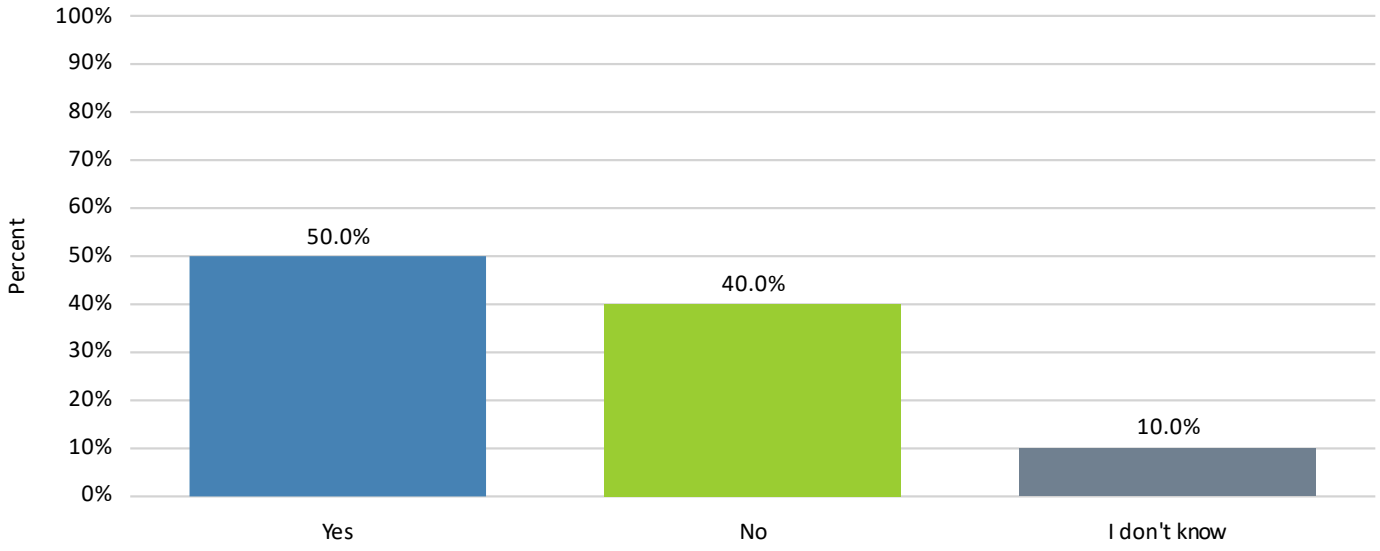
The feeling that we're not good enough and there are intrusions that we don't have control over.

High amount of false positives. Difficult to estimate false negatives.

lack of a complete system that has all relevant information in it. Instead we have to use many different systems around eachother and it makes for a very disorganized way of working

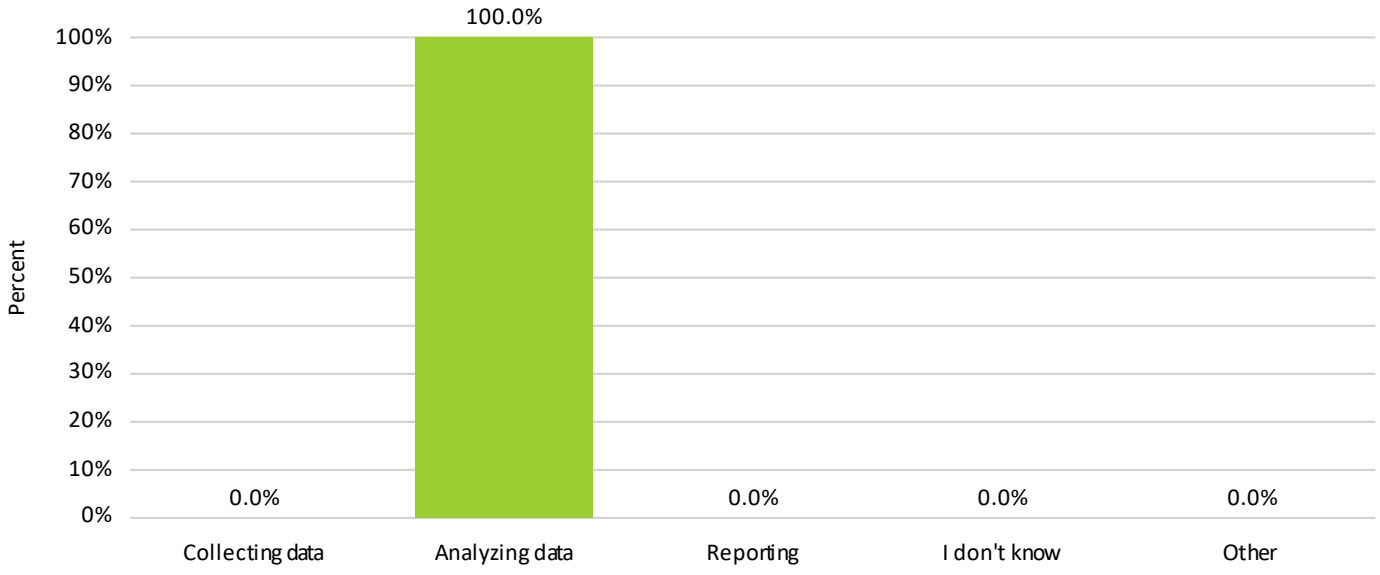
A very heterogenous environment with extreme variety in users and usage patterns, and ongoing developments in malicious activity, means that a lot of malicious activity may successfully be camouflaged as legitimate.

14. Do you perform post-intrusion forensics?



Name	Percent
Yes	50.0%
No	40.0%
I don't know	10.0%
N	10

15. What is the most time-consuming part of post-intrusion forensics for you?



Name	Percent
Collecting data	0.0%
Analyzing data	100.0%
Reporting	0.0%
I don't know	0.0%
Other	0.0%
N	5

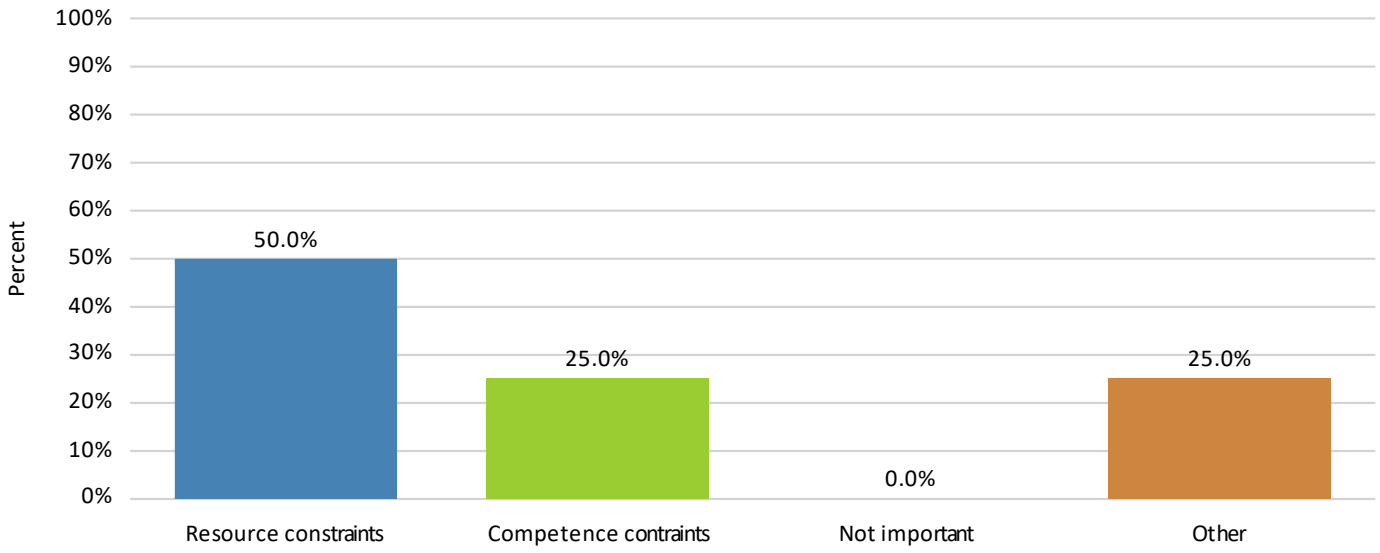
16. What kind of data do you often need when performing post intrusion forensics?

Netflow, loggfiler, minnedump

logfiles
filesystem metadata

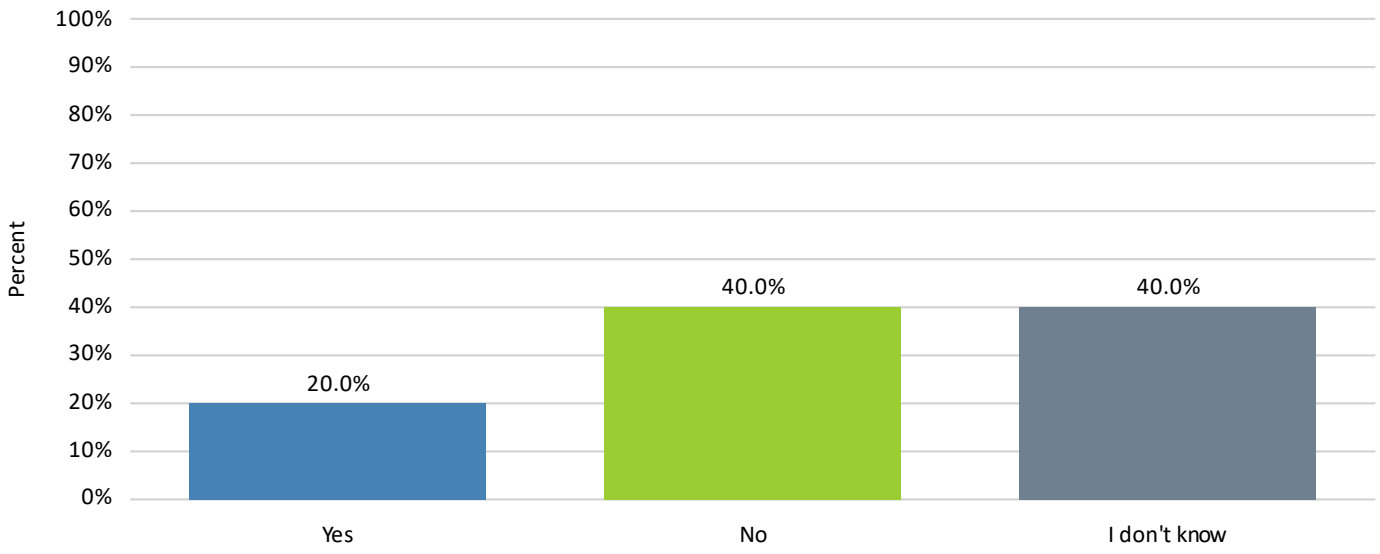
Logs, filesystem and database metadata

17. Why not?



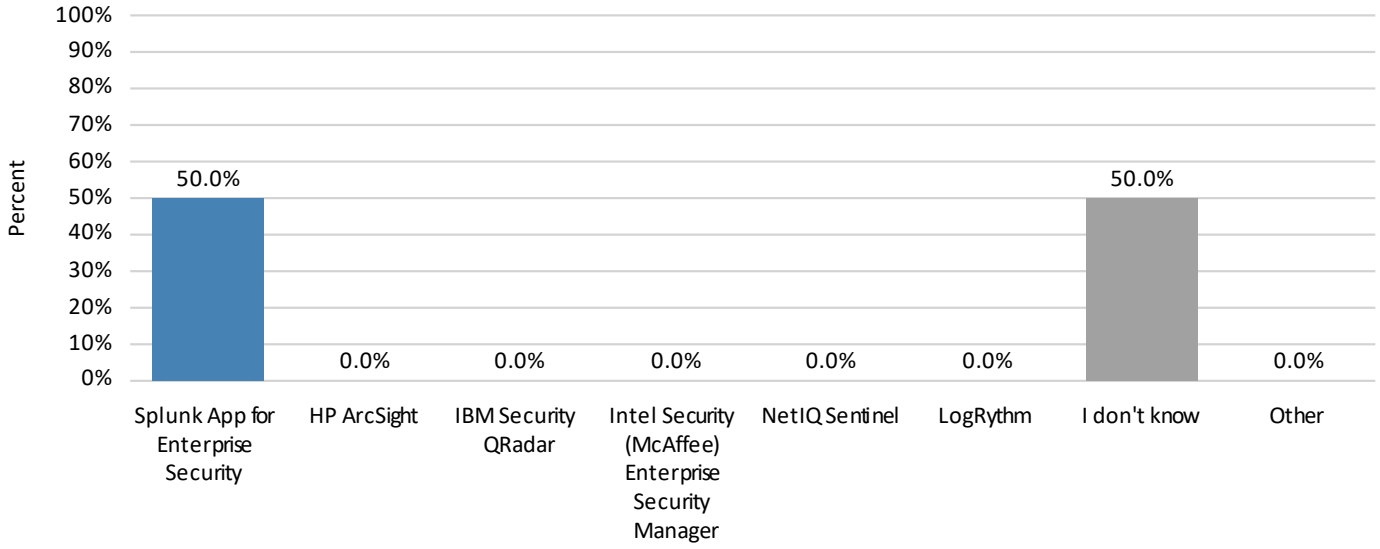
Name	Percent
Resource constraints	50.0%
Competence contraints	25.0%
Not important	0.0%
Other	25.0%
N	4

18. Do you use a SIEM Solution?



Name	Percent
Yes	20.0%
No	40.0%
I don't know	40.0%
N	10

19. Which SIEM Solution?

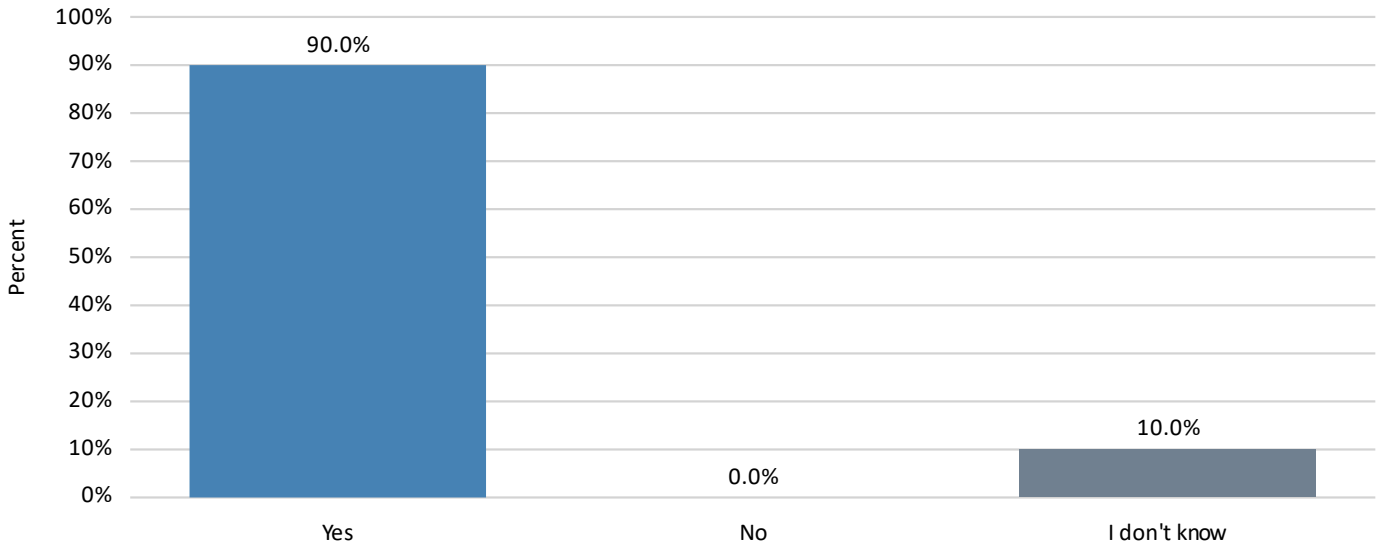


Name	Percent
Splunk App for Enterprise Security	50.0%
HP ArcSight	0.0%
IBM Security QRadar	0.0%
Intel Security (McAffee) Enterprise Security Manager	0.0%
NetIQ Sentinel	0.0%
LogRythm	0.0%
I don't know	50.0%
Other	0.0%
N	2

20. What are the biggest advantage with this tool?

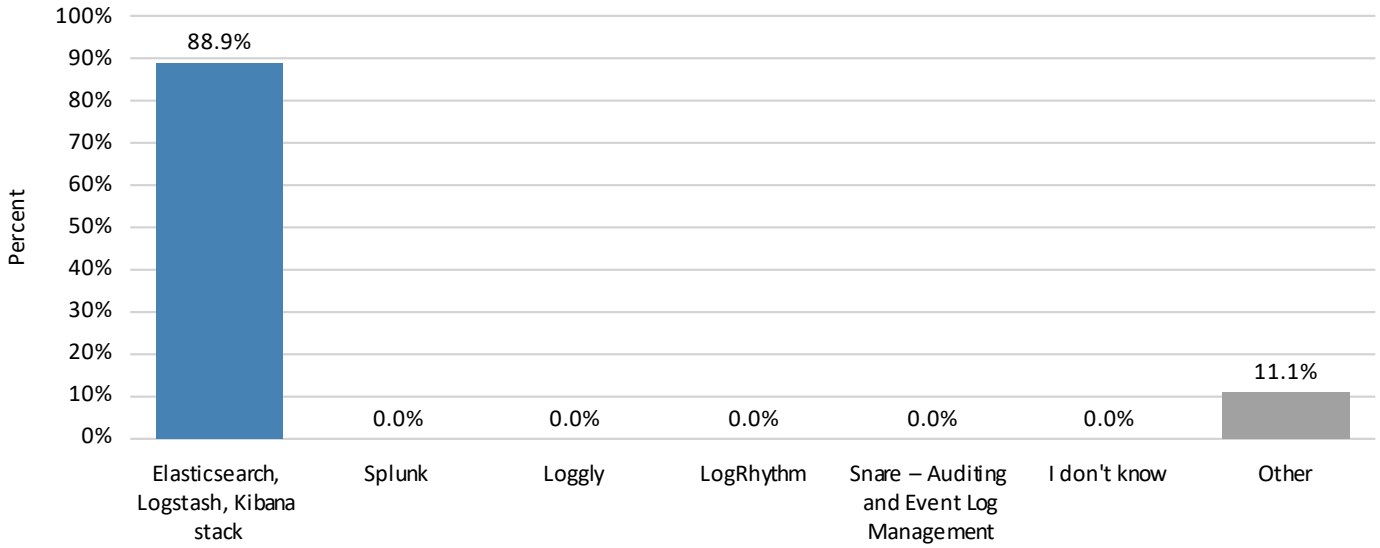
21. What are the biggest disadvantage with this tool?

22. Do you use a Log management tool?



Name	Percent
Yes	90.0%
No	0.0%
I don't know	10.0%
N	10

23. Which log management tool?



Name	Percent
Elasticsearch, Logstash, Kibana stack	88.9%
Splunk	0.0%
Loggly	0.0%
LogRhythm	0.0%
Snare – Auditing and Event Log Management	0.0%
I don't know	0.0%
Other	11.1%
N	9

24. What are the biggest advantage with this tool?

mye data på ett sted

Ok tuned

One instance

Unified access to windows and linux logs. Centralizes windows logs.

Quicker searching, collecting the data one place, cheap.

We have full control over the software.

25. What are the biggest disadvantage with this tool?

(for) mye data på ett sted

still too much false-positives

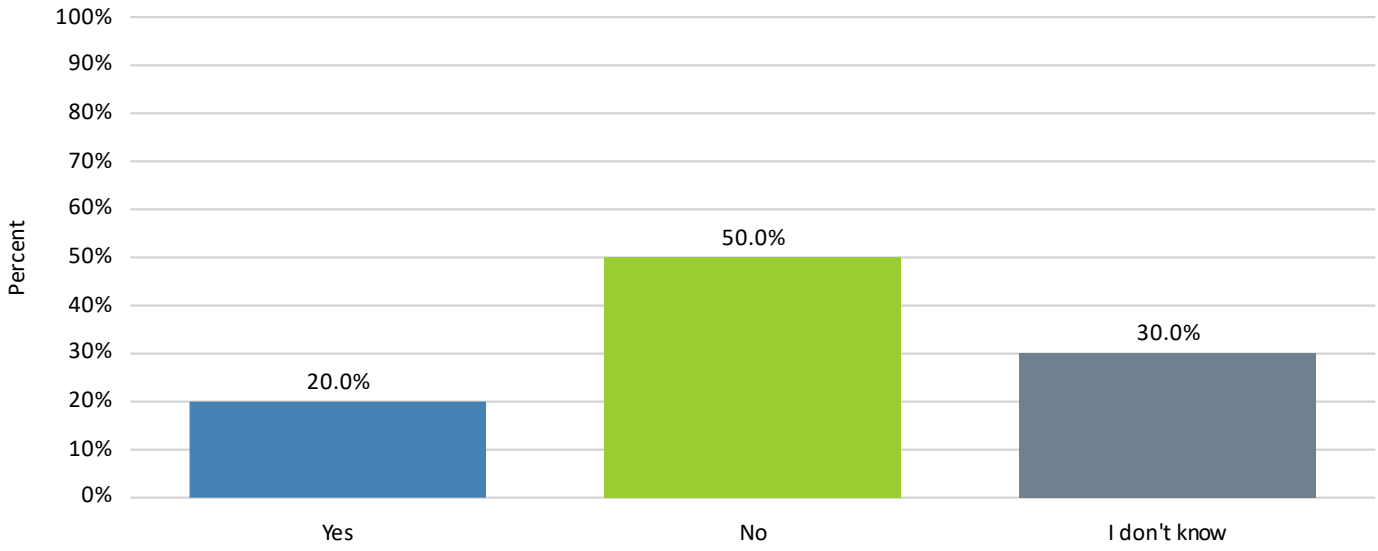
Need high competence for getting started

Web interface feels clunky.

Implementing the filtering, teaching others to use it and not possible to querry the API directly yet.

We cannot rely on external competence in the same degree.

26. Do you develop your own tools for advanced event processing?



Name	Percent
Yes	20.0%
No	50.0%
I don't know	30.0%
N	10

27. How do you do event correlation?

By finding and identifying common denominators and patterns, sometimes merely on correlation in time.

28. How do you perform outlier detection?

Manual inspection

29. How do you deal with conflicting data?

Manual inspection

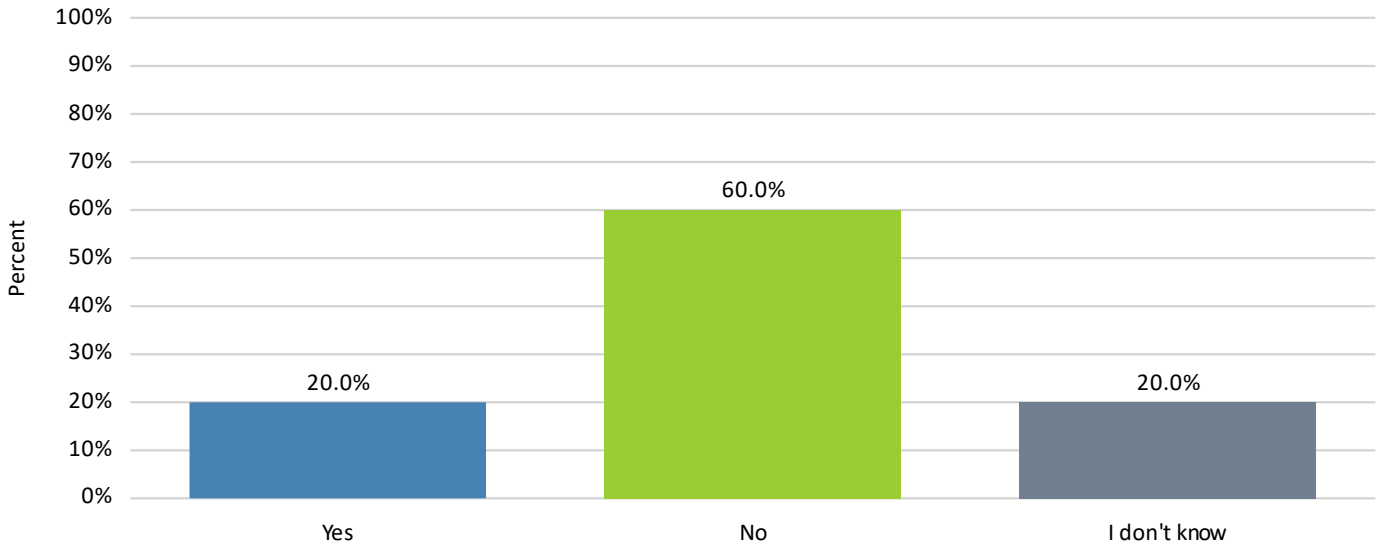
30. How do you do event association?

Similar to correlation.

31. Do you use Machine Learning? and what kind of algorithms?

No.

32. Do you use a distributed computing platform for processing events, performing security analythics or forensic investigation?

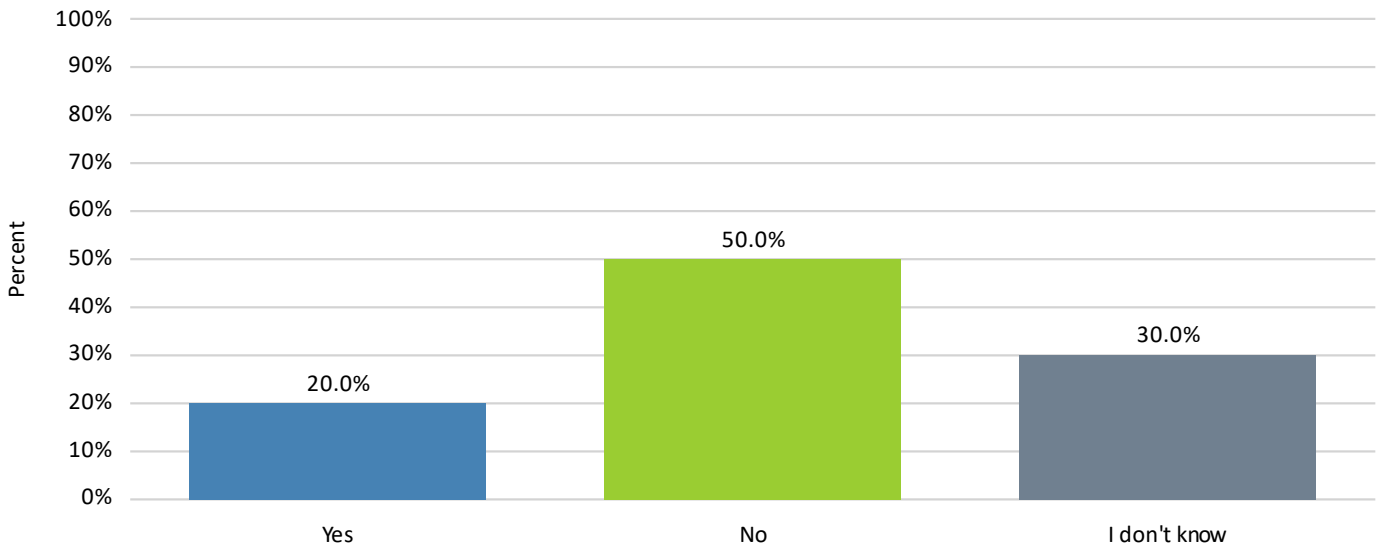


Name	Percent
Yes	20.0%
No	60.0%
I don't know	20.0%
N	10

33. Which one?

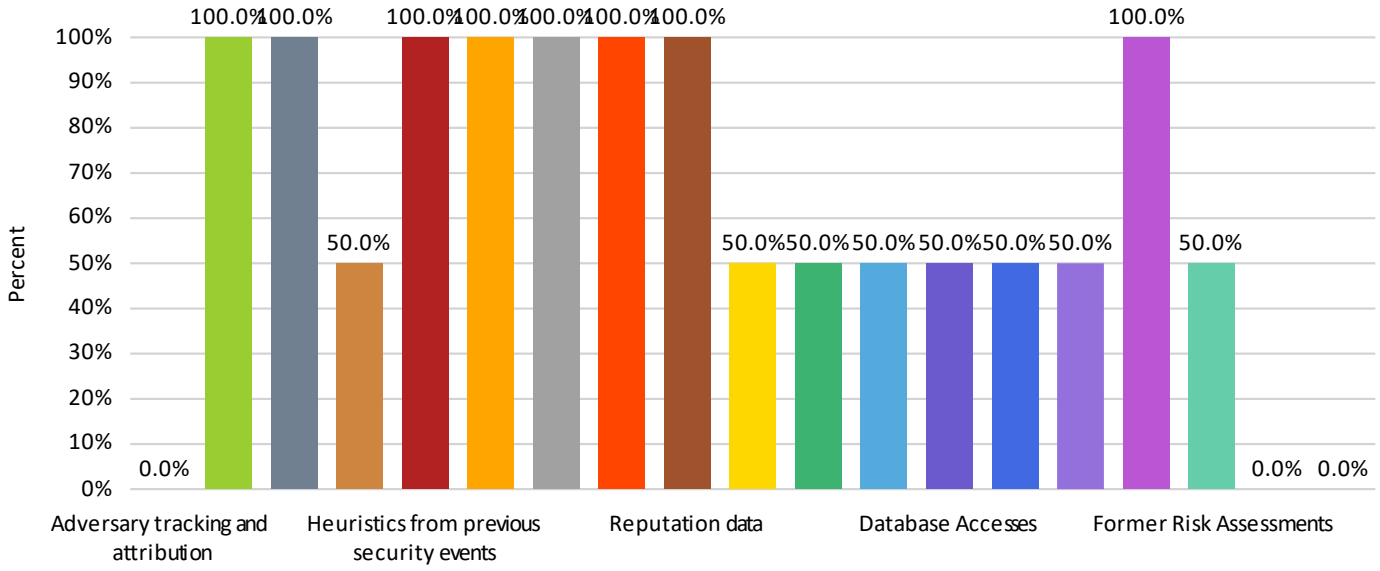
Developed in-house.

34. Do you use internally produced threat intelligence?



Name	Percent
Yes	20.0%
No	50.0%
I don't know	30.0%
N	10

35. What kind of internal threat intelligence sources do you use?



Name	Percent
Adversary tracking and attribution	0.0%
Communication with known malicious hosts (IP/DNS)	100.0%
DNS query data	100.0%
Endpoint security logs	50.0%
Heuristics from previous security events	100.0%
Host and network indicators of compromise	100.0%
User Authentication Failures / Successes	100.0%
Network history / Traffic patterns	100.0%
Reputation data	100.0%
Execution of unknown files	50.0%
Suspicious files	50.0%
File/Register integrity checks	50.0%
Database Accesses	50.0%
Honeypots	50.0%
IDS events	50.0%
Vulnerability scanning	100.0%
Former Risk Assessments	50.0%
I don't know	0.0%
Other	0.0%
N	2

36. Describe how you use this threat intelligence?

37. Please describe how your current tools help you to keep situational awareness?

getting mail-alert from detections at the Kibina when something is detected.

We have different tools. The Intranet, The service desk, Jabber and a Sharepoint site

I look the other way.

They don't. I rely on external warnings.

By providing a more complete picture across a multitude of hosts of clients, we can see some kinds of low-volume malicious behavior that would otherwise come across as non-malicious, for instance because a one-time probe against one customer is not necessarily suspicous, but one-time probes against several individuals.

38. What would improve intrusion detection, incident triage, incident response or post-incident forensic in your organization?

Mer tid dedikert til slikt arbeid...

More people working dedicated with the topic of security. More collection of what is "normal", making finding "unnormal" easier

Giving security/CSIRT leader the right to demand preventive measures from the organization.

People, tools and a plan.

A combination of better, more wide-spanning tools and more dedicated resources would improve.

39. If I can contact you for follow-up questions (if needed), please leave your email