NTNU
Norwegian University of
Science and Technology

Data-driven Approach to Information Sharing using Data Fusion and Machine Learning

Lars Christian Andersen

06-01-2016

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Norwegian University of Science and Technology, 2016

Supervisor 1: Prof. Katrin Franke
Supervisor 2: Andrii Shalaginov

Data-driven Approach to Information Sharing using Data Fusion and Machine Learning

Lars Christian Andersen

06-01-2016

Abstract

The number of security incidents worldwide is increasing, and the capabilities to detect and react is of uttermost importance. Intrusion Detection Systems (IDSs) are employed in various locations in networks to identify malicious activity. These sensors produce large amounts of data, which are fused and reduced. It is necessary to determine how to perform such fusion and reduction of data from heterogeneous sources. IDS is known to produce a high amount of false positives which create a high workload for human analysts at Security Operation Center (SOC). To ensure scalability, systems for reducing and streamlining the detection process is critical. The application of Threat Intelligence (TI) in information security for detection and prevention is widespread. When performing sharing of TI, it must be ensured that the data is reliable and trustworthy. Further, it must be guaranteed that the sharing process leaks sensitive data. This thesis has proposed a process model describing the process of fusion and reduction of heterogeneous sensor data and TI in intrusion detection. Our work is based on a literature study and qualitative research interviews with security experts from law enforcement and public and private organisations. Further, an identification of reliable and trustworthy features in such fused and reduced data for use in Machine Learning (ML) is given. We have applied data-driven methods on a real-world dataset from a SOC for this identification, and evaluate our results using well-known performance measure. Our results show that the application of ML can be used for prediction and decision support in the operation of SOC. We also provide an identification of sensitive features from the features selected by our data-driven experiments.

Sammendrag

Antall sikkerhetshendelser i verden øker, og mulighetene for deteksjon og reaksjon er kritisk. Intrusion Detection System (IDS)er blir plassert i forskjellige lokasjoner i nettverk og systemer for å kunne identifisere ondsinnet aktivitet. Disse sensorene produserer store mengder data som må bli fusjonert og redusert. Det er derfor viktig å definere hvordan slik datafusjonering og -reduksjon skal gjøres når man har et stort antall heterogene sensorer. Det er kjent at IDSer produserer store mengder falske positiver, som igjen skaper store mengder unødvendig arbeid for sikkerhetsanalytikere i en Security Operation Center (SOC). For å tilrettelegge skalering er det kritisk med systemer som kan reduserer og effektivisere deteksjonsprosessen. Bruken av trusseletteretning for deteksjon og prevensjon i informasjonssikkerhetsmiljøet er utbredt. Når trusseletteretning blir delt, er det sentralt at den delte informasjonen er pålitelig, og at man unngår å dele sensitiv informasjon. Denne oppgaven foreslår en prosessmodel som beskriver fusjonering og reduksjon av data fra heterogene sensorer og trusseletteretningskilder. Vårt arbeid er basert på en litteraturstudie kombinert med kvalitative forskningsintervjuer med sikkerhetseksperter fra politimyndigheter og offentlige og private organisasjoner. Videre så har vi identifisert attributer i slik fusjonert og redusert data som kan brukes i maskinlæring. Dette ble gjort via en datadrevet fremgangsmåte på et datasett fra en SOC med data fra den virkelige verden. Videre så ble resultatene våre evaluert med kjente metoder for ytelsesmåling. Våre resultater viser at bruken av maskinlæring for prediksjon og beslutningsstøtte i daglig operasjon av en SOC er mulig. Videre så har vi identifisert sensitive attributer fra attributene valgt av våre datadrevne eksperimenter.

Preface

This thesis is original and unpublished work by author, L. C. Andersen.

The motivation for this project was a combination of several factors. The security community, and especially the machine learning community, at NTNU Gjøvik has shown me the potential of data-driven approaches to information security. Further, the professional culture at mnemonic has shown me the challenges, needs, and potential of intrusion detection. Without these two communities, this thesis would not have been.

Oslo, 2016-06-01

Lars Christian Andersen

Acknowledgements

First and foremost, I would like to thank Prof. Katrin Franke and Andrii Shalaginov for the support, ideas, and discussions throughout the thesis. Further, you have both provided great lessons and motivation for the master students.

Secondly, I would like to thank Fredrik Borg and mnemonic for providing valuable and interesting discussion and an experimental environment. Further, you have provided me an excellent dataset for my experiments.

I would like to thank classmates Torbjørn and David for discussions and company during the process. I would also like to thank classmates Espen, Jan, and Lars for valuable discussions, detailed feedback, and proofreading. Finally, I would like to thank family and friends who have persisted during this process. I could not have done this without all of you.

L.C.A

Contents

List of Figures

List of Tables

Abbreviations

AIDS Application IDS.
ANN Artificial Neural Network.

CC Command and Control.
CERT/CC Computer Emergency Response Team Coordination Center.
Cfs Correlation Feature Selection.
CPNI Center for Protection of National Infrastructure.
CSIRT Computer Security Incident Response Team.
CSV Comma-Separated Values.
CybOX Cyber Observable Expression.

DAG Directed Acyclic Graph.
DOS Denial of Service.

ECOC Error Correction Output Codes.
EK Exploit Kit.

FN False Negative.
FP False Positive.
FW Firewall.

GeFS Generic Feature Selection.
GUI Graphical User Interface.

HIDS Host IDS.
HUMINT Human Intelligence.

IDS Intrusion Detection System.
IOC Indicators of Compromise.
IP Internet Protocol.
IPS Intrusion Prevention System.
IRT Incident Response Team.

k-NN k-Nearest Neighbor.

LDA Linear Discriminant Analysis.
LMS Longest Meaningful Substring.
LOO Leave-one-out.

ML Machine Learning.
mRMR minimum Redundancy Maximum Relevance.
MSS Managed Security Service.

NIDS Network IDS.
NLP Natural Language Processing.

OSINT Open Source Intelligence.
OSN Online Social Networking.
OVA One-vs-All.
OVO One-vs-One.
OWA Ordered Weighting Averaging.

PCA Principal Component Analysis.

SC Soft Computing.
SIEM Security Information and Event Management.
SOC Security Operation Center.
STIX Structured Threat Information Expression.
SV Support Vector.
SVM Support Vector Machine.

TAXII Trusted Automated Exchange of Indicator Information.
TI Threat Intelligence.
TOR The Onion Router.
TP True Positive.
TTP Tactics, Techniques, and Procedures.

URI Uniform Resource Identifier.
URL Uniform Resource Locator.
VPN Virtual Private Network.

Glossary

aggregation To collect, combine, and reduce information from various sources.

benign Not malicious, normal activity.

botnet Here: Number of compromised computers controlled by a malicious actor.

CIA triad Model defining Information Security; Confidentiality, Integrity, and Availability.

convergence The property or manner of approaching a limit.

correlation Combining mutual or similar elements.

embedded Here: Combination of filtering and wrapper method.

filtering Feature selection method evaluating features independently.

heterogeneous Different in kind; unlike.

inline Here: Positioned on the network link; traffic passes through.

stratified Here: Containing the class distribution when sampling dataset.

tap Hardware device which copies all traffic flowing through the device.

wrapper Feature selection method evaluating features together.

zero-day A vulnerability which there are no available patch for; previously unknown vulnerability.

1 Introduction

1.1 Topic Covered by the Thesis

The number of security incidents worldwide is increasing, and the security community relies on the ability to detect and react to such threats. Historically, information security is a continuous cycle where vulnerabilities are discovered, exploited by malicious actors, and patched by the information security community. As new vulnerabilities and exploits are observed, signatures or patterns indicating malicious activity is created. These signatures are used by Intrusion Detection System (IDS) to detect malicious activity in networks. The IDSs create alarms for human analysts for which to decide on what action to be taken. Unfortunately, many of these alarms are False Positives (FPs), that is wrongly raised alarms. It has been observed that up to 99% of the triggered alarms are FPs [1], and finding the True Positives (TPs), correctly raised alarms, are labour-intensive. The high work load can lead to errors and thus False Negatives (FNs), that is misclassification of a correct raised alarm. The work load of the human analyst can be decreased by *aggregation* and *correlation* of alarms. However, this is not enough in a large scale Security Operation Center (SOC). The need for systems to reduce and streamline the process is present.

Applying Machine Learning (ML) methods to the alarms raised is a possible solution to this. ML is a field which studies the construction of algorithms that can learn from data, and make new predictions based on this data. By training the ML algorithms using historical classification of alarms, it is possible to create a model which performs similarly to the human analyst who classified the historical alarms. The generated model can be applied to new alarms for noise removal or quality control. Further, ML methods can be applied for identifying hidden trends for prediction of future events.

Sharing of Threat Intelligence (TI) is a central aspect of today's work against malicious actors, and the security community considers TI important [2, 3, 4]. Indicators of Compromises (IOCs) are used and generated by processes such as the SOC operation. Determining how such IOCs should be shared, and to what extent values have to be anonymised are problems arising when such sharing is performed. Data fusion and reduction is also important due to the significant amounts of processed data. Sharing of significant amounts of data is complex, and it is of interest to share the data which are the most valuable.

The author has studied information security for five years and has thus achieved a broad academic understanding of the field. The author has also worked at a

SOC for an information security company for two years, and has thus an understanding of the problems arising in this work.

To successfully accomplish this project, a deep understanding of the ML process is needed. Further, knowledge of IDS and Security Information and Event Management (SIEM) is essential as well. Knowledge of data fusion, sharing of TI and potential IOCs are needed.

1.2 Keywords

Keywords covered in this thesis according to IEEE Computer Society: **I.4.8**{Sensor Fusion},**I.2.6**{Machine Learning}, **H.3.5**{Data Sharing}.

1.3 Problem Description

In security monitoring processes, large amounts of data is collected, correlated, and aggregated for further use in analysis. A various amount of *heterogeneous* data sources are used, and the data fusion must be governed by standardisation to ensure correctness and efficiency in the consecutive phases. Further, the inclusion of TI is central. The fusion and reduction of such data may provide great benefit in information sharing.

Applying ML approaches to event classification can provide great benefits to the daily operation of a SOC. However, several problems are arising when considering the performance of the classification process. Blindly applying ML to data will in most cases not result in desired performance.

Understanding the data is crucial to ensure that the chosen features provide the best classification for the specific problem. Currently, there is little knowledge about which features are the most reliable, hence sufficient classifier performance cannot be guaranteed. Identifying the most reliable features in aggregated and correlated data is needed.

As IOCs are observed and collected, it is of interest for the security community to share such information. Unfortunately, sharing such information may cause damage to the affected companies, and care should be taken when sharing such information. Anonymisation can help solve this problem, thus the identification of features which must be anonymised is needed.

1.4 Justification, Motivation and Benefits

The fusion, processing, and sharing of information related to digital threats are critical processes for fighting the ever-increasing cyber threat. Several efforts of combining data and knowledge have been performed, however, a standardised process-based model would benefit the security community. A process-based model including fusion and sharing of TI is needed in current operation. ML has proven great results in data driven environments, and so the inclusion of ML techniques in such a system is unavoidable. For automation of the security operation, ML is central.

According to security companies [5, 6, 7, 3] attacks against the finance sector and financially motivated attacks is on the rise. The importance of information sharing is noted by several [2, 4, 6, 3]. According to Gartner [8], 60% of digital business infrastructure will, by 2019, rely on TI feeds to ensure operational resilience. By having a standardised process for data fusion and reduction creates the possibilities of increasing the efficiency and quality of the information security processes. By anonymising sensitive features, information sharing can be performed between security actors.

1.5 Research Questions

1. How can data fusion and reduction for intrusion detection at an early stage using various heterogeneous sources be modelled?
2. Which features are reliable and trustworthy in the classification of aggregated and correlated events, and which cannot be shared without anonymisation?

1.6 Contributions

The intended goal of this thesis can be separated into two parts.

(*i*) A model describing the process of fusion and reduction of data at an early stage in intrusion detection. The model should provide an overview of the advantages and disadvantages of fusion and reduction at an early stage.

(*ii*) An identification of reliable and trustworthy features in correlated and aggregated intrusion detection events for use in ML. Further, an overview of sensitive features which cannot be shared without anonymisation.

1.7 Thesis Outline

This thesis is divided into several chapters covering various parts of the project. The following section provides an overview of the organisation of the rest of the thesis.

In *Chapter 2 - Security Operation and Threat Intelligence (p. 5)* an overview of relevant theory related to security operation and TI is given. We provide an introduction to the field of security operation, and describes how TI can be used to increase the efficiency of such an operation.

In *Chapter 3 - Machine Learning and Data Fusion (p. 17)* an overview of relevant theory related to ML and data fusion. We provide an introduction to the field of ML while discussing different techniques for the various phases of the process. An introduction to the field of data fusion is given with definitions from literature and concrete examples of use. Further, we present how data fusion relates to current security operation.

In *Chapter 4 - Related Work (p. 35)* related work and the current state of the

art related to the two research questions is provided. We present an overview of previous work and discuss the advantages and disadvantages.

In *Chapter 5 - Choice of Methods (p. 41)* we present a detailed description of the scientific methods applied when conducting this project. An overview of tools and techniques used is presented, ensuring repeatability for future researchers.

In *Chapter 6 - Reliable and Trustworthy Features in Aggregated Intrusion Detection Events (p. 53)* we present in detail how the experiment for solving research question two is conducted. We provide a presentation of the results, and a discussion of these is given.

In *Chapter 7 - A Model for Data Fusion, Reduction, and Sharing in Financial Sector (p. 67)* we present our findings regarding research question one. Requirements for a data fusion process model is presented based on literature and research interviews, and a proposed process model is presented.

In *Chapter 8 - Implications and discussion (p. 74)*, we discuss the implications and considerations of the thesis, and we provide a summary of work done in thesis.

In *Chapter 9 - Conclusion (p. 78)*, we present a conclusion of our work and results.

Finally, in *Chapter 10 - Further work (p. 79)*, we propose further work based on our research, experiments, and results.

2 Security Operation and Threat Intelligence

In the previous chapter, an introduction to the thesis was given. Research questions were introduced, together with justification and motivation for this thesis. Further, the contributions of the thesis were presented. The following chapter will present relevant theory related to security operation and TI. An introduction to IDS is given, as well as the operation of Computer Security Incident Response Teams (CSIRTs). Further, the concept of TI is discussed, and the application of TI is demonstrated. Finally, the process of information sharing is discussed.

2.1 Intrusion Detection Systems

Defensive security operations are primarily based on the protection of the confidentiality, availability, and integrity of information infrastructure and its data [9]. These elements are commonly known as the *CIA triad*. To protect such infrastructure IDSs can be implemented. These systems monitor and detect potentially malicious activity on, from, and towards the infrastructure. By adding preventive mechanisms such as a Firewall (FW), Intrusion Prevention Systems (IPSs) are created. This system can then stop the malicious activity when detected. In *Figure 1 - Positions for IDS and IPS*, examples of locations for IDS and IPS in networks are presented. In *(1)*, the IDS has a *tap* which copies bit by bit, and can therefore monitor all network activity going through the link. However in *(2)* the IPS is positioned *inline* and can, therefore, monitor and stop malicious activity.

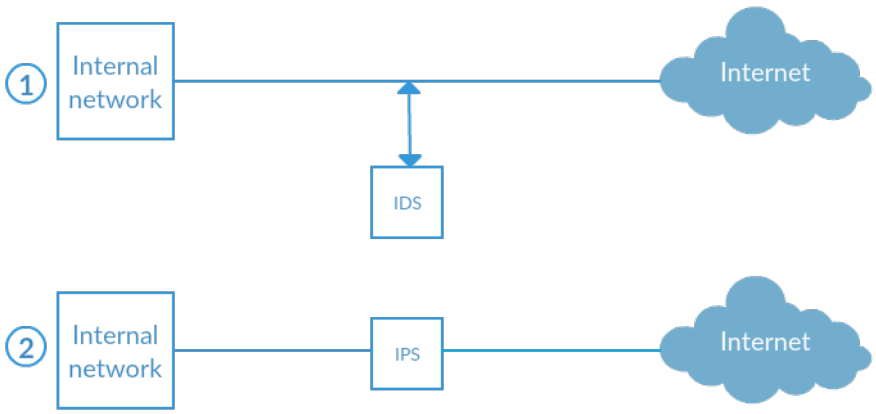

Figure 1: Positions for IDS and IPS

IPSs apply same detection methods as IDSs. The concept of intrusion prevention is outside of the scope of this thesis, and we will therefore not discuss this subject further. As seen in *Figure 1 - Positions for IDS and IPS*, IDS can be used in networks. However there are several other locations, and one common way of classifying IDS is by the scope of protection.

2.1.1 Scope of Protection

By categorising IDSs by which aspect of the information infrastructure the IDS monitors, it can be divided into Network IDS (NIDS), Host IDS (HIDS), and Application IDS (AIDS).

NIDS Network-based monitoring where the IDS monitors activity on the network. A network device is set to capture all traffic on the network, and not just the traffic addressed to the device.

HIDS Host-based monitoring of all actions performed on the host. The system collects data from various internal logs such as system logs and system audit trails.

AIDS Application-based monitoring where the IDS monitors internal data specific for certain applications.

2.1.2 Scope of Model

Another complimentary method of categorising IDSs is by the scope of model. That is, how the system detects potential malicious activity. The two models are misuse-based, where patterns of malicious activity are predefined, and anomaly-based detection, where profiles of normal activity are defined [9].

Misuse-based

By observing malicious activity, security analysts can define patterns accordingly. Pattern matching is then used to determine whether the observed activity matches any known malicious activity. However, there are several disadvantages to this approach. The obvious disadvantage being that it can only detect known bad activity. The unknown bad will not match any patterns. Another downside is that new signatures must be created continuously as new attack methods are developed, and the pattern database expands rapidly making the process of pattern matching more computational complex. However, even with the disadvantages of this approach, misuse-based IDSs is still the most common approach [9].

Anomaly-based

By observing normal activity in the infrastructure, profiles can be generated as a baseline for further activity. The IDS then compares the observed activity towards the previously defined baseline determining whether it is normal or not. The main disadvantage of this approach is the process of defining the baseline of

what is normal. In a complex system, it is difficult to model all possible normal behaviour, while ensuring no abnormal activity is modelled as well. The major reason for using this approach is the fact that this approach can detect previously unknown attacks, i.e *zero-day* attacks.

2.1.3 Challenges

When applying an IDS to a system to monitor and detect malicious activity, it is of interest to measure the performance of the IDS to ensure it performs as expected. Five measures of efficiency have been proposed in the literature [10, 11], which reflect the challenges each implementation of an IDS has. The measures are *accuracy, performance, completeness, fault tolerance,* and *timeliness*.

Accuracy Describes the correctness of classification of *benign* activity. Classifying benign activity as malicious, FP, is an inaccuracy. Currently, large amounts of data pass IDSs, and signature databases increase accordingly. According to the *Base-rate fallacy*, a minuscule small amount of FP is necessary for IDSs to be efficient [12]. A high number of FP is expensive in terms of analyst resources.

Performance The processing performance of the system. Performance must be high to enable real-time detection. Due to the large amounts of data combined with the diversity, IDS is approaching big data problems. If the allocation of sufficient hardware resources is not performed, the IDS may have to queue packets and lose its capabilities for real-time detection.

Completeness Describes the correctness of classification of malicious activity. Classifying malicious activity as benign, FN, is incompleteness. In real networks, it is not possible to have a complete understanding of all attacks, and measuring completeness is, therefore, difficult.

Fault tolerance Describes the resistance to attacks. IDSs can be vulnerable to attacks, and Denial of Service (DOS) attacks ,in particular, can be a problem for such systems. Assume a signature-based IDS. The detection is performed using pattern matching, and an attacker can craft custom packets which trigger the worst case scenario for each pattern matching. That is, the system must compare the activity with all signatures in the database.

Timeliness Similar to the performance measure, but also describes the performance of the propagation of alerts.

2.2 Computer Security Incident Response Team (CSIRT)

A CSIRT provides reactive and proactive services for response and prevention of security incidents [13]. The history of CSIRTs began with the foundation of Computer Emergency Response Team Coordination Center (CERT/CC) at Carnegie

Mellon University as a result of the Morris worm paralysing large parts of the Internet [14]. The purpose of the organisation was to enable quick spreading of notifications and coordinating communication between a network of incident responders during security emergencies. The following section will provide an overview of CSIRTs by presenting the services typically performed by CSIRT, as well as the different types of CSIRTs.

2.2.1 CSIRT services

The services of a CSIRT can be separated into reactive services, proactive services, and security quality management services [15]. Reactive services focus on mitigating notified incidents, however, proactive services and security quality management services focus on preventing future incidents. In the following section, an overview of the three types services is presented. The individual services performed in each of these categories is dependent on the type of CSIRT, which will be discussed in *Section 2.2.2 - CSIRT types (p. 8)*.

Reactive services When CSIRTs are notified of incidents, there are generally four main practices performed [14]: information to constitute a response to a network security problem such as an attacker, vulnerability, or threat campaign is issued via *alerts and alarms*; *incident handling* is performed by receiving, triaging, responding to and analysing incidents; *vulnerability handling* is performed by analysis of vulnerabilities, responding to a vulnerability by producing patches or workarounds, and coordinating broader response by sharing information on how to fix or mitigate; *artifact handling* is performed by analysis of malware and other artifacts, and responding and coordinating by developing patch or detection and prevention mechanisms, on their own or in coordination with others.

Proactive services Continuous services for prevention of future incidents is performed by CSIRTs. General security-related information and information on developments and trends is disseminated, security audits or assessments is performed on organisation's infrastructure, new security tools are developed, and intrusion detection services are performed.

Security Quality Management services CSIRTs may also perform functions which indirectly contribute to the overall security community [14]. Services like product certification, risk analysis, and education and training are proactive activities with the goal of preventing future incidents.

2.2.2 CSIRT types

CSIRTs can be separated into different types depending on the sector or group served [15]. The operation and approach of the various types are slightly different, depending on the constituency they serve. The combination of services performed can also be slightly different.

National CSIRTs The main point of contact for domestic incident responders and other national CSIRTs. National CSIRTs have, according to CERT/CC a "*specific responsibility in cyber protection for the country or economy*" [16].

Regional CSIRTs Also facilitates communication between national CSIRTs as well as information sharing between CSIRTs in the region.

Sectoral CSIRTs The constituency of sectoral CSIRTs are specific sector of society or economy. Banking and education sector are two examples [14].

Organisational CSIRTs The main task of organisational CSIRTs is the monitoring and response to incidents residing in the internal network of an organisation. Academic institutions, private companies and government organisations are examples of organisations where such a CSIRT can exist.

Vendor CSIRTs CSIRTs can also reside within vendor organisations, providing services to individuals and companies. They are often customer-focused [14].

Commercial CSIRTs These types of CSIRTs provide incident handling for hire. The services are either sold as products to other organisations, or, in case of a non-profit organisation, provided for free.

2.3 Threat Intelligence and Information Sharing

TI can in simple terms be described as the knowledge of a threat's capabilities, infrastructure, motives, goals, and resources [17]. These elements are the foundations of the Diamond Model proposed in [18], which will be discussed later in this section. By applying TI to the security operation, organisations seek to understand threats towards the organisation and use the information to change the outcome of potential threats. To understand TI, it is important to understand traditional intelligence. The relationship of data, information, and intelligence is presented in *Figure 2 - Relationship between data, information, and intelligence* as described by the US Department of Defense.

Intelligence begins with the collection of large amounts of environmental attributes ranging from elements such as data regarding civilians, friendly and adversary forces, to data regarding weather. The collected data is then processed and refined to create information. Finally, by analysing the information, specific intelligence is produced. In cyber, intelligence is also produced in a similar refinement process. However, TI per se can also be data, and thus, the comparison between traditional intelligence and TI is vague. To provide an overview of the types of TI, we will use a model proposed by Chismon and Ruks [20] in cooperation with the Center for Protection of National Infrastructure (CPNI) and CERT-UK. This model separates TI based on the consumption of the TI and separates into strategical, tactical, operational, and technical. The separation is visualised in *Figure 3 - Subtypes of TI proposed by Chismon and Ruks*.

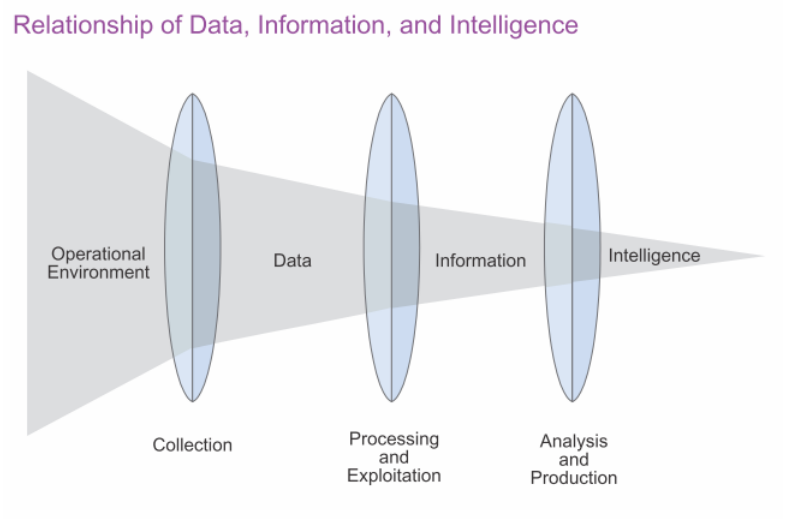

Figure 2: Relationship of data, information, and intelligence as presented by [19]

Strategic Strategic intelligence is high-level information, most commonly consumed by C-level, the Board and senior decision makers. The intelligence is most likely not technical, and is often given in the forms of reports or briefings, be it in meetings or one-to-one. The content of such intelligence focuses on financial impact and trends in cyber. Events, organisations or persons related to cyber activities having an impact on the high-level business of the organisation is an example of such content. The strategic intelligence is created to help strategists understand risks for further decision making, and deals in high-level elements like risk and likelihood. The collection of such intelligence can be collected from open sources, commonly called Open Source Intelligence (OSINT), whitepapers from security related organisations and from other humans within the same field, commonly known as Human Intelligence (HUMINT). These types of intelligence is rarely shared as the can reveal information regarding the organisations plans. On the other hand, if the strategic intelligence is generic, it is most likely not useful for other organisations. Strategic intelligence should be crafted in-house, as it most commonly are created on specific requirements from C-level or the Board.

Tactical Tactical intelligence is mid-level information, most commonly consumed by system administrators, system architects and security staff. The main goal of such intelligence is to describe the tactics used by various threat actors, and will describe the Tactics, Techniques, and Proceduress (TTPs) of threat actors. TTPs being information on how each phase of the operation

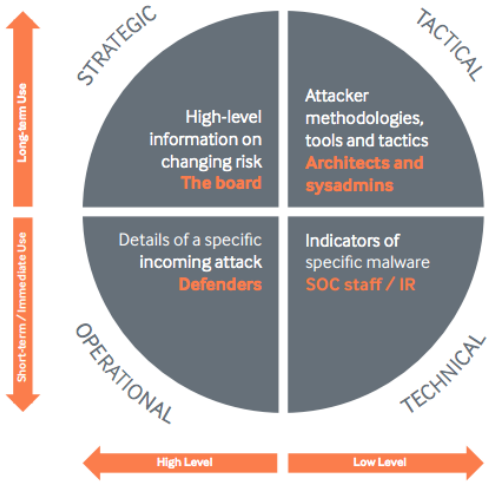

Figure 3: Subtypes of TI proposed by Chismon and Ruks [20]

is performed, be it initial access, lateral movement, or extraction of data. This can be mapped to the Cyber Kill-chain (see *Section 2.3.1 - Application of TI (p. 12)*). It also contains information on tools used in the different phases, as well as techniques used. Collection of such intelligence can be done from several types of sources. Reports on attack groups, campaigns, and incidents can provide tactical intelligence. Analysis of malware, either done in-house or externally can also provide valuable tactical intelligence. Sharing of such intelligence is often encouraged, as it helps the security community. The intelligence is rather specific, but general enough to provide value for other organisations.

Operational Operational intelligence is mid-level information about a possible incoming attack. The intelligence is consumed by defenders who can ensure the required controls are in place in advance, be it removing assets, applying defensive tools, or applying monitoring tools for identification of attackers. The intelligence describes the nature of an upcoming attack, and may also describe the identity and capabilities of the attacker. By combining such intelligence with tactical intelligence, the defenders can ensure a deeper understanding of the threat actor, and possible attack vectors. Traditionally, collection of such intelligence can be done by recruiting persons within the community, or compromise their communication or systems. However, for private organisations such activity is in most cases illegal, and at best immoral. This is a problem, especially if the intelligence is to be used in legal cases. Legal collection of such information can be done by collecting open communication like chat roms, social media and forums.

Technical Technical intelligence is low-level information about the assets of an attacker, be it tools, Command and Control (CC) channels or infrastructure. It is on a technical detailed level as IOCs, and should be rapidly distributed and included in the security systems due to its short lifespan. By adding elements like MD5 sums of files and Internet Protocol (IP) addresses, SOC staff and Incident Response Team (IRT) can rapidly detect new events, or search existing logs for earlier undetected events. However, there are several challenges to such intelligence. Due to the large amount of indicators, resources must be assigned to ensure they are applied to the correct systems. The data often lack contextual information, and is therefore of little use for higher analysis. In the case of targeted attacks, most of the IOCs can be easily changed, and therefore avoid detection. There is also a significant amount of available feeds, and they should be evaluated before use.

2.3.1 Application of TI

The most obvious application of TI is the use of technical intelligence, that is IOCs, in security appliances like firewalls, IDSs and endpoint security products. However, the subtypes of TI as presented in the previous section can be combined to provide a much wider and deeper situational awareness in regards to events in the past as well as in the future. One example of such an application is the Diamond Model proposed by Caltagirone et.al [18].

The Diamond Model is a model describing the atomic elements of any intrusion activity, i.e. any event, and is presented in *Figure 4 - Diamond Model*. In the model, each event consists of meta-features, confidence value, and four core features represented as nodes. The core features are adversary, capability, infrastructure, and victim.

The Diamond model gets its name from the diamond formed by the four core features. By creating diamonds of each event, it is possible to correlate new events easily when some of the core features are the same. This correlation allows for detection of small changes in the TTP of the attacker and therefore supports the collection of new intelligence as well. By combining these diamonds with another well known model for cyber attacks, *The Cyber Kill Chain*, threat actors can be identified across stages and attacks. The combination of these models will be discussed below.

The Cyber kill chain was proposed by Hutchins et.al [21], and describes general stages of an attack. It describes the sequential phases in an attack, and we have presented each phase below. It is loosely based on the military methodology *kill chain* which contains phases for conducting an operation from start to end. U.S Department of Defense defines these as *find, fix, track, target, engage,* and *assess*(F2T2EA) [22].

Reconnaissance The first stage in conducting an attack is gathering available data to understand the target. This would include elements such as brows-

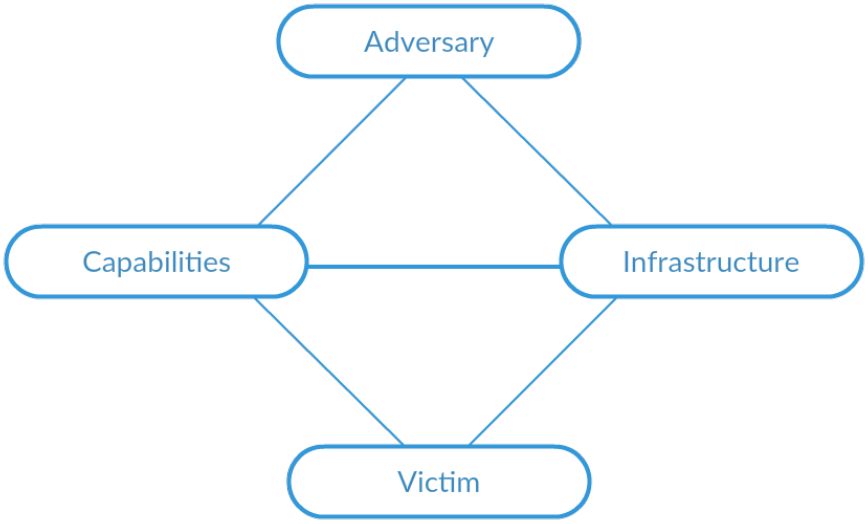

Figure 4: Diamond Model

ing company websites and other open sources accessible for identification and selection of targets. The extensive use of social media makes it possible to create a rather specific social profile for targets before moving on to the next stage. Understanding the behaviour of the targets enables the attacker to customise the campaign for an increased chance of success, and information leakage enabled by Online Social Networking (OSN) is a great resource for this understanding.

Weaponisation In this stage, the malicious payload is added to what appears to be a legitimate file. This stage is often not observed by the target and may be performed before the reconnaissance as well.

Delivery In his stage, the weaponised file is served to the target. This could be done in many different manners, but a common approach is by email.

Exploitation In this stage, a vulnerability on the targeted system is exploited. This enables an attacker the possibility of executing commands on the system which, in the end, may lead to the downloading of arbitrary code to the target.

Installation In this stage, the malware installs itself on the target system. Techniques for achieving persistence is often applied in this step.

CC To be able to continue the infiltration operation, the attackers must be able to communicate with the infected clients. In this stage, a pre-defined connection is established towards CC-servers.

Component	Description	Elements
Meta-features	Features describing the event	Timestamps, phase, result, direction, methodology and resources.
Confidence Value	A value of confidence associated to each element of the event.	None.
Adversary	Information about the adversary	Personalia such as email addresses, phone numbers, language and physical location
Capability	Information about the capabilities of the adversary	Hacker tools, malware, stolen certs and exploits.
Infrastructure	Information about the infrastructure used by the adversary	IP addresses, domain names and email addresses.
Victim	Information related to the target of the attack	Personalia, sector and email addresses.

Table 1: Components of the Diamond Model

Exfiltration In this final stage, data is exfiltrated from the infected system. Several techniques can be used to do this undetected, including steganography and encryption. The data exfiltration stage is, however, not always present in all attacks. This stage is what generally finishes the goal of the attack, and may therefore also be deception, disruption, denial, degradation, or destruction.

These stages are what all attacks have in common, and several of these stages are observable in a cyber attack. *Figure 5 - Two incidents correlated using Diamond Model and Cyber Kill Chain* shows how several attacks observed in different stages can be correlated. Each core feature observed allows for pivoting to other events in the threat actor hunt. By combining these two models, it is possible to correlate easily different incidents and group incidents which are probable to be related to the same threat actor.

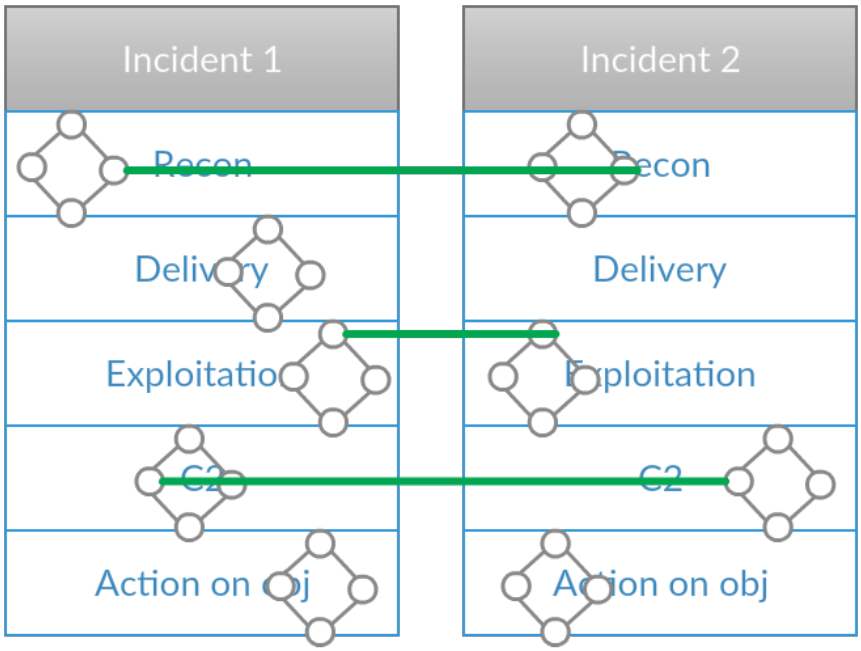

Figure 5: Two incidents correlated using Diamond Model and Cyber Kill Chain

2.3.2 Information sharing

Information sharing is the process of sharing information between various entities such as private and public organisations, with the common goal of improving cyber incident/attack prevention, detection, prediction, response and recovery [23]. In order to maintain successful operations targeting and defending against cybercrime groups, sharing of information between law enforcement, private industry, and academia is necessary [3].

Currently, there are three popular frameworks for standardisation and sharing of TI developed by MITRE. Cyber Observable Expression (CybOX) is a standardised language for communicating information about cyber observables [24], Structured Threat Information Expression (STIX) is a standardised language, which represents structured information about cyber threats [25], and Trusted Automated Exchange of Indicator Information (TAXII) is a collection of services which enables the sharing of TI between partners [26]. By combining these frameworks, standards for the structure of TI and the consecutive sharing of TI is achieved. The relationship between these frameworks is as follows: STIX describes cyber threats using CybOX to describe observations, and TAXII is used for the transportation of this information. They enable automated cyber threat information exchange between defenders, which is crucial due to the current amount of indicators available. In 2015, 431 million new malware variants were

observed by Symantec [27], and it provides an insight into the amount of indicators necessary to keep up with cybercrime.

2.4 Summary

In summary, we have in this chapter discussed the field of security operation. An introduction to the various types of IDSs was given, discussing the scope of protection and model. We also discussed common challenges in IDS like accuracy, performance, completeness and fault tolerance. Further, the concept of CSIRT was discussed. We presented the three types of services CSIRTs provides; reactive services, proactive services, and service quality management services. The various types of CSIRTs types were also introduced. Finally, the concept of TI and sharing of such was introduced. We discussed the four subtypes of TI; strategic, tactical, operational, and technical. Further, we discussed how TI can be applied in threat models like Cyber Kill Chain and the Diamond Model. Frameworks CybOX, STIX, and TAXII for sharing of TI was presented.

3 Machine Learning and Data Fusion

In the previous chapters, an introduction to the thesis has been given. Further, relevant theory on the topics of security operation and TI has been presented. The following chapter will present theory on another field relevant to this thesis, ML and data fusion. The field of ML is discussed, and common steps and challenges in the ML process is demonstrated. Further, an introduction to the field of data fusion is given. Previously proposed models for data fusion is presented. Finally, an introduction to multisensor fusion is given.

3.1 Machine Learning

The following sections will describe the field of ML and briefly present its elements. ML is a field studying the construction of algorithms which can learn from data, and then predict based on this data [28]. It can be divided into three distinct types of learning; supervised learning is where data sets is labelled with class or value, unsupervised learning is where data sets have no label, semi-supervised learning is a hybrid of supervised and unsupervised learning where some data is labelled. The general ML process is presented in *Figure 6 - ML process*.

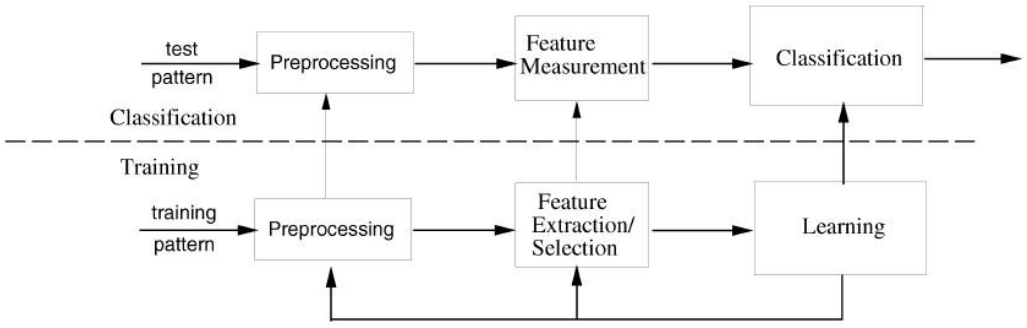

Figure 6: ML process [29]

3.1.1 Preprocessing

In the real world, the available data may not be formatted and ready for feature extraction and selection (described in *Section 3.1.2 - Feature Selection (p. 19)*). Therefore, it is important to perform a preprocessing phase where the dataset is prepared for the next phases in the ML process. The methods applied in this phase includes:

Feature scaling Most ML algorithms behave much better if feature scaling is performed beforehand [30]. Decision trees and random forests are some of the few algorithms where we do not need feature scaling. The two common approaches for feature scaling is normalisation and standardisation. Most commonly, normalisation refers to the transformation of feature values to a range of $[0, 1]$, while standardisation centre each feature column with a mean equal 0, and a standard deviation of 1. Standardisation is often more practical as it maintains information about outliers [30].

Convert continuous attributes into discrete If the applied ML classifiers cannot handle continuous attribute values, the attributes must be discretized. That is, the continuous values must be mapped to discrete values. When performing discretization, there are generally two problems to solve: the optimal number of intervals and the optimal boundaries for each interval [28].

Convert continuous and discrete attributes to binary A specific case of discretisation is when the applied ML classifier is designed for binary attributes only. When performing binarization, the attribute values are mapped to one of two binary values [30].

Convert discrete attributes to continuous Several ML methods assume that all attributes are continuous [28]. Therefore, discrete attribute values must be transformed to continuous values.

Dealing with missing values When performing ML on real-life datasets, the quality may not always be optimal. That is, some attributes can have missing values. When handling missing values, it can either be ignored, replaced with the most probable value, or replace using a probability distribution of the attribute values [28].

Visualisation By visualising the data, data scientists can use the human brain's capabilities for processing visual information. Understanding the problem and the available data is important in ML, and while expert knowledge about the domain is best, data visualisation can provide the data scientist an overview of the data. Visualisation techniques include *histogram, scatter plot, time plot, parallel plot* and *star glyph* [28].

Handling categorial attributes Categorial attributes can be divided into ordinal and nominal attributes [30]. Ordinal attributes have values which can be sorted or ordered like `'small'`, `'medium'` while nominal have values which there is no specific order like `'red'`, `'blue'`. Learning algorithms do not understand this correctly, and the categorical attributes must be mapped to understandable values.

3.1.2 Feature Selection

As the datasets have the correct format, features can be selected for the analysis phase. In doing so, the amount of data to process is decreased, which also decrease the complexity. Selection of feature sets can also help handling common challenges, see *Section 3.1.3 - Learning (p. 20)*. Specifically, the objective of feature selection is three-fold: improving the performance of classifiers, providing faster and more cost-effective classification, and providing a better understanding of the underlaying process that generates the data [31]. Blindly selecting features may not yield an optimal subset of features, which then decrease efficiency. Therefore, methods for feature selection exist, and there are three approaches for feature selection: *filtering*, *wrapper*, and *embedded*. Before discussing these, it is important to understand features and their quality.

Feature Quality and Feature Reliability

When selecting feature subsets, the quality of selected features is important. Generally, higher feature quality allows for more efficient ML. Some of the most common feature quality measures are presented in *Table 2 - Common feature quality measures*.

Quality measure
Information Gain
Gain-Ratio
Distance Measure
ReliefF
Correlation Feature Selection (Cfs)
minimum Redundancy Maximum Relevance (mRMR)

Table 2: Common feature quality measures [28]

By using these measures, the quality of available features can be calculated and chosen accordingly. Of these common quality measures, Cfs and mRMR are those proving best performance in research. However, we cannot rely on previous performance on other datasets. A well-known challenge in ML is that we cannot guarantee method performance without knowledge of the dataset. This is further discussed in *Section 3.1.5 - Challenges (p. 24)*. The feature subsets can also be assessed according to the reliability. Nguyen et. al. [32] proposed feature selection method for reliable feature selection using these two quality measures. The

proposed method and state of the art in reliability in feature selection process is discussed in *Section 4.2 - Reliable Feature Selection and Feature Anonymisation (p. 35)*.

Filtering

Filtering is the quickest and simplest method for feature selection [28]. This method calculates the quality of each attribute and selects the k best attributes. The value of k can either be defined in beforehand or changed dynamically by selecting all attributes with a quality above a certain threshold.

Wrapper

The wrapper method is a more advanced and slower method for selecting features [28]. This method uses a ML algorithm together with cross-validation (described in *Section 3.1.5 - Challenges (p. 24)*). The method searches for the optimal subset of features and applies learning algorithm on each combination. Therefore, the time complexity is larger than filtering method.

Embedded

The embedded method is a combination of filtering and wrapper methods [31]. It incorporates the feature selection as part of the training process and is in many aspects more efficient. According to [31]:

> they make better use of the available data by not needing to split the training data into a training and validation set; they reach a solution faster by avoiding retraining a predictor from scratch for every variable subset investigated.

Feature Extraction

An alternative approach to feature selection exists, namely feature extraction. Feature extraction is the process of transforming the feature set to a new feature subspace with lower dimensionality than the original [30]. Using techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), data compression is performed by summarising the original information into lower dimensions.

3.1.3 Learning

Learning in ML refers to the process of describing or modelling the available data. When performing learning, the algorithm searches for the best description which yields the optimal performance. Thus, learning can be treated as an optimisation problem [28]. Learning can be divided into three distinct types of learning; supervised learning is where data sets is labelled with class or value, unsupervised learning is where data sets have no label, semi-supervised learning is a hybrid of supervised and unsupervised learning where some data is labelled. As supervised and unsupervised is most common, we will not discuss semi-supervised learning further.

Supervised learning

A Common application of supervised learning is in classification and regression problems [28]. In classification problems, each object is assigned a class from a finite set of possible classes, e.g. $\{malicious, benign\}$. The task of the learning model is then to classify new observations to one of these classes based on previous data. Common classifiers include:

- *Decision tree* - The method builds a decision tree using the attribute entropy to decide nodes. For each node, it splits the set using the attribute with the lowest entropy.
- *k-Nearest Neighbor (k-NN)* - The method classifies new samples based on the class of its k nearest neighbour attributes. Distance measures like the Euclidean distance and Hamming distance are used.
- *Naive Bayes* - The method assumes a conditional independence of attributes, given the class. It applies the Bayes' Theorem when building the model.
- *Bayes Net* - The method creates data structures enabling classification using Bayes Network. It creates a representation of the probabilistic relationship between features in the form of a Directed Acyclic Graph (DAG), which then are used for classification.
- *Random Tree* - The method constructs a decision tree by selecting a random attribute for each node. It does not perform any pruning.
- *Random Forest* - The method generates N number of Random Trees, creating a forest of such trees. Then, it applies each tree for classification of sample. Classification of sample is then decided by voting process based on all trees.
- *SVM* - The method creates a hyperplane separating the classes in the most optimal way. When learning, it calculates the hyperplane with the largest difference to the Support Vectors (SVs). In the case of non-linear classification problem, it applies kernel method to convert to linear classification problem.

A two-class classification(binary classification) problem is presented in *Figure 7 - Two-class classification problem: Support Vector Machine (SVM)*. The SVM algorithm defines a hyperplane which separates observations from each class. New observations are then classified based on what side of the hyperplane it is located. Note that this example has only two dimensions for simplicity. In real scenarios, the feature space is much higher.

When there are more than two classes, e.g. $\{malicious, suspicious, benign\}$, we have a multinomial classification problem(multi-class classification). Unfortunately, many classification algorithms were designed for binary classification and therefore not suitable for multi-class classification problems. However, strategies have been developed for reducing the multi-class classification problem into several binary classification problems. These strategies are One-vs-All (OVA),

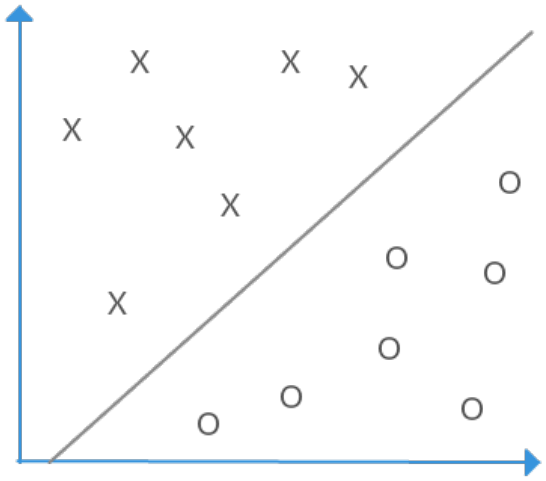

Figure 7: Two-class classification problem: SVM

One-vs-One (OVO), and Error Correction Output Codes (ECOC) [33].

OVA Being the simplest approach, it trains K classifiers where K is the number of classes. The k^{th} classifier is trained with positive examples belonging to the class k, and negative examples belonging to the other $K - 1$ classes.

OVO This approach combines all classes against each other. $\frac{K(K-1)}{2}$ binary classifiers are trained to discriminate between each class [33]. When classifying new samples, a voting scheme is applied to determine winning class. According to [34, 35], OVO is generally better than OVA approach.

ECOC This approach use the concept of codewords to distinguish classes. N binary classifiers are trained between K classes. *Table 3 - ECOC, as presented by Aly* shows an example of codewords where $N = 7$ and $K = 5$. When classifying new samples, the output codeword from the N classifiers are compared to the given codewords. Minimum Hamming distance is used to determine closest match which is used as the class label.

In regression problems, the task of the predictor is to determine the value of the dependent unobservable continuously variable [28]. Most common regressional predictors include regression trees, linear regression, SVM for regression, and Artificial Neural Network (ANN). A regression problem is presented in *Figure 8 - Regression problem: Linear regression*. Linear regression is used to determine the coefficient of the linear function, commonly presented as $y = f(x) = ax + b$, which yields the smallest errors of predictions evaluated on the training data.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
Class 1	0	0	0	0	0	0	0
Class 2	0	1	1	0	0	1	1
Class 3	0	1	1	1	1	0	0
Class 4	1	0	1	1	0	1	0
Class 5	1	1	0	1	0	0	1

Table 3: ECOC as presented by Aly [33]

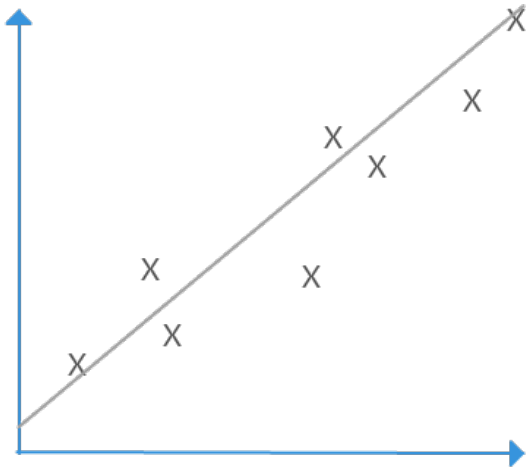

Figure 8: Regression problem: Linear regression

Unsupervised learning

A common application of unsupervised learning is clustering. For these problems, only description of objects is available, not the target variable. The task of algorithm is to determine clusters based on a dissimilarity measure like the Manhattan metric or Euclidean Distance. The number of clusters can either be defined in beforehand, or determined by the learning algorithm. K-means is a well-known clustering algorithm, and is presented in *Figure 9 - Clustering problem: K-means*. The algorithm works by defining a number of centroids equal to K, and then assign each object to the closest centroid, creating K clusters. A new centroid is calculated as the average of all objects in the cluster, and assignment is repeated. This is done until *convergence*, i.e. when the clusters are stable.

3.1.4 Evaluation

Evaluation of the performance of the ML algorithm is done by estimating the quality of the model. That is, how well it solves new problems. For estimation of the quality of supervised learning models, the data is split into two subsets: a learning set and a testing set [28]. The model is then trained using the learning

(a) Initial (b) Processing (c) Convergence

Figure 9: Clustering problem: K-means

set, and tested using the testing set. Performance measures are then applied to the results of the testing. Performance measures for classification and regression problems are shown in *Table 4 - Performance measures*. These measures allow us to estimate how successful the generated model is for solving new problems.

Classification algorithms	Classification accuracy and confusion matrix
	Misclassification cost
	Brier score, information score and margin
	Sensitivity, specificity, ROC curves, precision and recall
Regression algorithms	Mean squared error
	Mean absolute error
	Correlation coefficient

Table 4: Performance measures [28]

3.1.5 Challenges

When applying ML methods, several problems and challenges arise. The following section will describe the most common challenges in ML which are related to our research.

Ugly Duckling theorem

The Ugly Duckling Theorem describes how feature selection should be performed to achieve reliable classification performance. Generally, it states that features that which contribute to classification must be selected.

> Given that we use a infinite set of predicates that enables us to distinguish any two patterns under consideration, the number of predicates shared by any two such patterns is constant and independent of the choice of those patterns. Furthermore, if pattern similarity is based on the total number of predicates shared by two patterns, then any two patterns are equally similar [36].

Curse of Dimensionality

The curse of dimensionality can occur when operating on datasets with a large number of dimensions [28]. With a large number of dimensions, the volume of space increases exponentially, and the data becomes sparse. Generally, an increase in dimensionality causes a decrease in predictive power, commonly known as Hughes effect [37].

The solution for this problem is either collecting more data samples, or reducing the number of dimensions. Feature reduction can be performed either via feature selection or feature extraction as discussed previously in this section.

Overfitting and underfitting

Classifiers can have a high accuracy on the training dataset, however, a low accuracy on testing dataset. That is, the classifier fits the training data too well, and is unable to classify new samples successfully [28]. The classifier fails to create a generalised model of the dataset, and thus overfits the data.

However, if the classifier generalises too well, we have the problem of underfitting. The classifier is too general, and classification performance is low. A solution to these challenges is to separate the data used for training and the data used for evaluation. Three approaches exist for this: splitting the dataset, Leave-one-out (LOO), and k-fold cross-validation [28].

Splitting the dataset

The full dataset is split into a training set and a testing set. A common separation is 2/3 for training, and 1/3 for testing.

LOO

If the number of samples is low, splitting the dataset into two sets may remove relevant samples which cause the model to not be representable for the dataset. This can, however, be solved by using LOO. This method removes one sample from the dataset, and trains the model using the rest of the dataset. The removed sample is then used for evaluation, and then put back into the full dataset. This process is repeated for all samples, and the quality of the model is estimated using all the results.

K-fold cross-validation

LOO method works well, however, it is very time-consuming with larger datasets. Therefore, we can apply a generalised method of LOO called k-fold cross-validation. This method split the dataset into k number of folds, and then use the same process as LOO. This ensures that all samples are used for training and for evaluation, while avoiding a too time-consuming process.

No Free Lunch Theorem

The No Free Lunch Theorem describes how we cannot generally expect certain classifiers performing better than others on a certain dataset. It states:

> The apparent superiority of one algorithm or set of algorithms is due to the nature of the problems investigated and the distribution of data [36].

It is, therefore, apparent that multiple classifiers should be applied to the dataset to ensure the optimal classifier is selected.

3.2 Data Fusion

The field of data fusion has been around for a long time. Data fusion has been used in a various number of areas, including situational awareness in military context, bioinformatics, robotics, medical diagnosis, remote sensing, and manufacturing [38, 39]. One of the earliest definitions of data fusion is given by White [40] 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, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results.*"

Other definitions [41, 42, 43] focus on the use of multiple sensor sources to create an optimal estimate. Later work [44, 45] defines it as the process of combining data from multiple sensors to provide a better understanding of the scenario. That is, performing more specific inference which could not have been performed using single sensors. In [38], the authors present an overview of the various definitions given in literature. They identify common criteria and propose a new definition based on the identified criteria. The proposed definition is:

> "*Information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making*" [38].

By performing this process of refinement, the collected data can be transformed into information, and further into knowledge. More specifically, knowledge can be defined as the interpretation of the information contained in the data [28].

In literature, several models for data or sensor fusion have been proposed. The early work in the 1980's related to data and sensor fusion and situational awareness was in military context [46]. Many of the models reflect this by being extensively oriented towards military domain, both process wise and terminology wise. The following section will present the design of the earlier proposed models together with their capabilities and flaws. Several of the models have common elements, and comparisons will be made.

3.2.1 The Intelligence Cycle

The Intelligence Cycle has its root in military operation. As with much terminology in data fusion, several terms may describe similar elements. The J-P 2.0 Joint Intelligence by the US Department of Defense [19] describes this as the Intelligence Process, and includes 5 phases for the process of creating intelligence from operational environments. The process is presented in *Figure 10 - Intelligence process*.

Figure 10: Intelligence process [19]

Planning and Direction – Includes activities related to the development of plans and the consecutive execution of such. Including, but 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* [19].

Collection Includes activities related to the acquisition of data as defined in the Planning and Direction phase.

Processing and Exploitation Includes activities related to the conversion of collected data into formats readily for entities such as commanders, decision makers, intelligence analysts and other consumers.

Analysis and Production Includes activities related to the production of intelligence from the collected information and from refined intelligence from other parties.

Dissemination and Integration Includes activities associated with the delivery to and use by a consumer. Means of delivery are determined according to needs.

3.2.2 JDL Fusion Model

The JDL Fusion Model was originally proposed by the US Joint Directors of Laboratories Data Fusion Sub-Group in 1985 [40, 47], and has thereafter been updated several times [48, 49]. The JDL Fusion Model is well presented in [50], by one of the authors working on the revisited version. The model describes the sequential flow from (*i*) data, measurements and observations, to (*ii*) information, data placed in context, indexed, and organised, to (*iii*) knowledge, information understood and explained [50]. The proposed model is presented in *Figure 11 -*

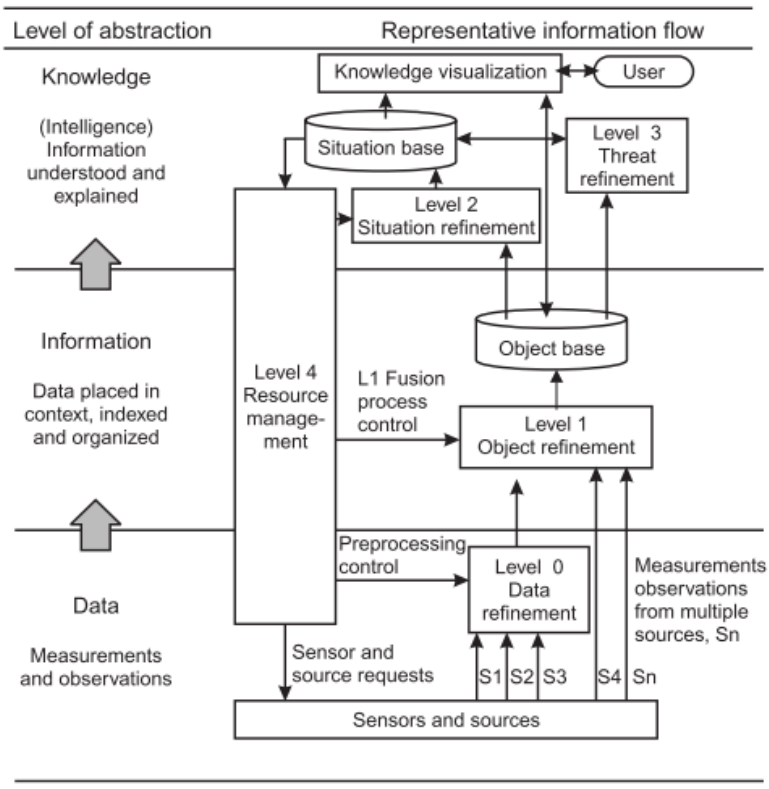

Figure 11: Process of data fusion as proposed by Waltz [50]

Process of data fusion as proposed by Waltz. The model uses five different levels of data refinement.

Level 0 Data refinement Calibration and filtering of raw data, such as bias correction.

Level 1 Object refinement The measures are aligned to a common frame of reference. Correlation is performed based upon an association process indicating which observations from different sensors have common elements.

Level 2 Situation refinement Situational awareness is created based upon the aggregated sets of objects. Elements such as behaviour, common points of origin, common protocols, common targets, and other high-level attributes are used.

Level 3 Threat (meaning) refinement Future possible outcomes are determined using situational knowledge to model and analyse feasible future behaviour.

Level 4 Resource management (process refinement) The whole process is refined in this management level. It refines based on current situational awareness and additional data when required.

This model governs the process of data fusion well. However, the proposed model has a general approach towards data fusion, and more detailed specifications is needed when applied to real life scenarios. The model provides a good basis for this project, which will focus more on a detailed modelling of early data fusion and reduction. The model does also not describe how to define the balance between data reduction and loss of valuable data well.

3.2.3 The Boyd Control Loop

The Boyd control loop [51, 52], commonly known as the OODA loop, contains four phases. Observe, Orient, Decide, and Act as shown in *Figure 12 - Boyd Control Loop*. This process is represents the decision-support for situational awareness commonly used in the military. As situational awareness is one of the goals in data fusion, the Boyd control loop has been used in sensor and data fusion.

The four phases can be mapped to the JDL model. The authors in [46] compare the two models as follows,

Observe Comparable to level 0 of the JDL

Orient Comparable to the functions of level 1, 2, and 3

Decide Comparable to level 4

Act No directly comparable function as the JDL model does not close the loop.

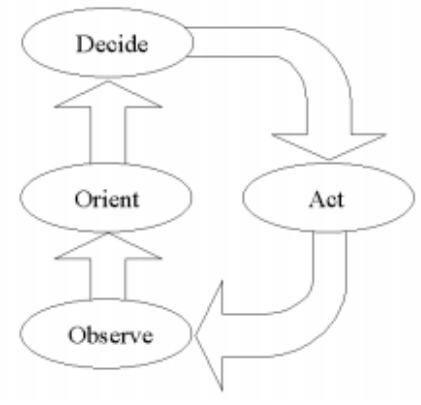

Figure 12: Boyd Control Loop [46]

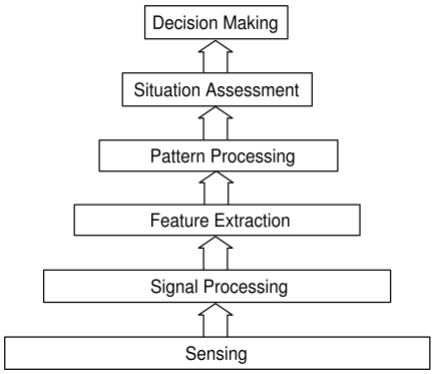

Figure 13: The Waterfall Fusion Model [53]

3.2.4 The Waterfall Model

A waterfall based model proposed by [53] heavily focuses on the lower level processing functions. The stages of this model as presented in *Figure 13 - The Waterfall Fusion Model* corresponds to level 0, 1, 2, and 3 in the JDL model.

Due to its similarities with the JDL model it has many of the same flaws [54]. The waterfall model is more detailed in analysing the fusion process, however, it lacks any feedback data flow. As seen in *Chapter 2 - Security Operation and Threat Intelligence (p. 5)*, security operation is a continuous process and a feedback loop is crucial.

3.2.5 The Dasarathy Model

Dasarathy [55] identifies five possible categories or levels of fusion. The categorisation is dependent on the input and output of the fusion, and the author presents how previous categories can be mapped to this categorisation. The five

categories are presented in *Table 5 - The Dasarathy Model*.

Input	Output	Notation	Analogues
Data	Data	DAI-DAO	Data-level fusion
Data	Features	DAI-FEO	Feature select and feature extraction
Features	Features	FEI-FEO	Feature-level fusion
Features	Decisions	FEI-DEO	Pattern recognition and pattern processing
Decisions	Decisions	DEI-DEO	Decision-level fusion

Table 5: The Dasarathy Model [55]

3.2.6 The Omnibus Model

Bedworth and Obrien [46] states that the existing fusion models are oriented towards military domain, thus the need for a model fitting the extensive data fusion community was necessary. They propose the Omnibus Model, which are based on the advantages of the previous models. It has the cyclic nature from the Intelligence Cycle and the Boyd Control Loop, the detailed definitions of the Boyd Control Loop which all can be mapped to one of the levels in the JDL model and Dasarthy Model.

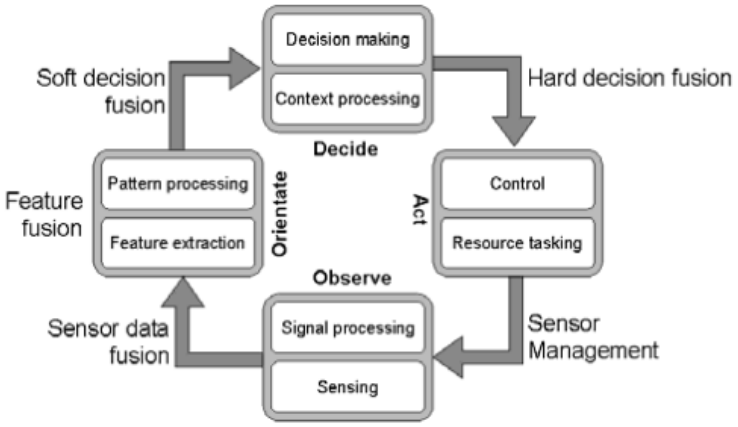

Figure 14: The Omnibus Model [46]

3.3 Multisensor Fusion

Multisensor fusion is a relatively new discipline which combines data from multiple and diverse sensors and sources [39]. The goal is to make an inference of events, activities, and situations using several observations. The following section will present how data from multiple and diverse sensors and sources can

be combined to increase reliability, and thus increase situational awareness. The concept of voting is introduced, and several voting schemes are discussed. Finally, soft computing approach to voting is introduced.

3.3.1 Voting schemes

Von Neumann [56] suggested in 1956 the use of voting to combine unreliable data into a reliable version. In a general voting algorithm, the four main components are input data, output data, input votes, and output votes [57]. Parhami [57] propose a taxonomy for voting algorithms, which we will use to present the different types, or classes, of voting methods. Below, an overview of the possible combinations is presented, as proposed by [57].

Input data Can either be *exact* where the input is viewed as inflexible, i.e. input y must be equal some x_i, or *inexact* where input is viewed as flexible, and input objects represented neighbourhoods.

Output data Can either be a *consensus* where output data is a subset of inputs with votes w supporting y, or *mediation* where output data y is the result of an object function minimising or maximising all input.

Input votes Can either be *oblivious* where input votes are fixed by being built in the voting algorithm, or *adaptive* where input votes can be provided as inputs.

Output votes Can either be a *threshold* where output vote exceeds a given threshold, or *plurality* where output is the sum of votes for the object with most votes.

For simplicity, we have decided to focus on the output votes only. That is, how a winner object is decided. As seen above, this can either be by threshold or plurality.

Threshold voting

As threshold voting selects object with votes exceeding a given threshold, common majority voters can, in fact, fall within the threshold category [57]. Generally, threshold voting is fundamentally simpler than plurality voting [58].

Plurality voting

Plurality voting, on the other hand, counts votes for each object and selects on of the objects with the highest vote. We can, by combining plurality voting and simple comparison of output vote with threshold implement many threshold voters, however, the results may be much less efficient that a direct threshold voter [57].

Ordered Weighting Averaging (OWA)

So far, we have only discussed voting where all votes are assumed equal. In many cases, some of the voters may be more reliable than others, and their votes should ,therefore, weight more. The OWA for aggregation was introduced in 1988 by Ronald R. Yager [59]. The OWA operators can allow a positive compensation between ratings, i.e. they can realise trade-offs between objectives [60]. It allows a higher degree of satisfaction of one criterion to compensate for a low degree of satisfaction of another criterion. The extreme cases of OWA operators can either be full compensation ($Max(a_1, ..., a_n)$) or no compensation ($Min(a_1, ..., a_n)$). The weights w would then be $w = (1, 0, .., 0)^T$) and $w = (0, 0, .., 1)^T$ accordingly. It is important to note that the weights are not connected to specific criteria, but to a specifically sorted ordering of the value of criteria.

A linguistic quantifier [61] $Q_\alpha(r) = r^\alpha, \alpha \geq 0$ is defined and α value is search so that the linguistic quantifier Q_α approximate the criteria as much as possible, be it expert preference or other [60]. The selected Q_α is then applied to an OWA operator $F_Q(a_1, ..., a_n)$ and an aggregated score is calculated.

3.3.2 Fuzzy voting

The application of fuzzy logic has proven successful in many scenarios like the combination of neural networks [62], malware detection [63], and general expert systems [64]. Fuzzy logic is based on the concept variables being part of a set to a certain degree, calculated using a membership function $\mu()$, and is part of Soft Computing (SC), a collection of data-driven computational models [65]. What separates fuzzy logic operations from traditional logical operations is that there are no crisp lines or sets. Let A and B be two intersecting subsets of set X. The membership for x in subsets A and B can then be calculated using $\mu()$, e.g $\mu_A(x) = 0.4$ and $\mu_B(x) = 0.6$.

When applying fuzzy logic to voting, a fuzzy integral is calculated for each object. The fuzzy integral is defined by [62]:

$$h(x) \circ g(\cdot) = \max_{F \subseteq X}[\min(\min_{x \in F}, g(E))] = \max_{\alpha \in [0,1]}[\min(\alpha, g(h_\alpha))] \quad (3.1)$$

where g is a fuzzy measure and h is a density measure. By calculating the fuzzy integral for each object based on all voters, an aggregated score is generated; thus, a winner is decided based on the number of votes as well as how certain each voter is.

3.4 Summary

In summary, we have in this chapter discussed the field of ML, providing an overview of common processes. The process of preprocessing has been presented with its methods commonly used in the ML process. An introduction to feature selection and commonly used measures was presented. Further, we presented

the concept of learning and discussed two common approaches *supervised learning* and *supervised learning*. Evaluation of performance is discussed and common challenges as *ugly duckling theorem*, *curse of dimensionality*, *no free lunch theorem*, and *overfitting* was presented.

Further, an introduction to the field of data fusion was given. Previous work in terms of definitions is presented, and widely used data fusion models were discussed. Models like JDL Fusion Model, Intelligence Cycle, and The Boyd Control Loop was presented providing an overview of the different types of models in terms of granularity and coverage. Where applicable, models were compared either stage by stage or by product.

Finally, multisensor fusion was presented. An overview of how multisensor fusion can be applied to combine data from several unreliable data sources to reliable data was given. Further, we briefly introduced fuzzy voting exploiting great benefits from fuzzy logic.

4 Related Work

In the previous chapters, an introduction to the thesis was given, and theory on several central topics was presented. The following chapter presents previous work related to the expected contributions of this thesis. An overview of the state-of-the-art in data fusion in security operations is presented. Further, state-of-the-art in reliable feature selection is discussed. An overview of the newly proposed reliable feature selection method is given, discussing the results from previous work. Current work in anonymisation is briefly presented. Further, the current use of information sharing in practice is presented.

4.1 Data Fusion in Security Operation

In intrusion detection, a common problem is the high number of FP. As a result, there has been numerous work on decreasing the FP level as well as the general level of alerts [66, 67, 68, 69].

Nguyen et al. [70] identified in 2014 current gaps in existing alert management. Thereafter they propose efficient alert management approach reducing unnecessary alerts from IDS. Their approach uses two modules: alert verification module which validates alerts with vulnerability; aggregator module which removes redundant alerts. The aggregator module reduces the volume of alerts by aggregating alerts belonging to the same attack within a time window. This is performed by sending alerts to predefined sub aggregator for each class of attack. Each of these sub aggregators combines relevant alerts and create a meta alert, efficiently reducing the volume of alerts. Their aggregation approach uses simple fusion by fusing when all features are overlapping. In their experiment, features IP, port, and time were used. The approach also allows for aggregation of meta alerts. For evaluating the effectiveness, they used

$$\mathrm{reduction\ \ rate} = \frac{\mathrm{filtered\ \ alerts}}{\mathrm{total\ \ number\ \ of\ \ alerts}} \tag{4.1}$$

Based on their testbed with three different IDSs, they achieved reduction rate between 44.4% and 59.5% over five attack classes with an average of 50.39%.

4.2 Reliable Feature Selection and Feature Anonymisation

Many studies have focused on the feature selection process, and the measures used. However, many studies focus on the wrapper method for selection. In intrusion detection, the potential number of features makes it inconvenient and resource consuming. The filtering method, on the other hand, allows for a high

number of features. The earliest approach for feature selection in machine learning focused on filtering [71]. Work like Schlimmer [72] in 1984 and Almuallim and Dieterich [73] in 1991 approached the problem by finding the minimal combinations of features which are consistent with the training data. Other filtering methods have been proposed in seminal work such as Kira and Rendell [74].

Previous work focuses on the accuracy of the resulting features when deciding upon feature selection method. A more recent work by Hall and Holmes [75] presents a benchmarking for several feature selection methods. The performance of each method was assessed based on the classification accuracy of two well-known classifiers Naive Bayes and C4.5 implementation of decision tree, size of the three in C4.5, and number of features in the Naive Bayes. The experimental setup consisted of 18 different datasets from the UCI collection [76]. The feature selection methods assessed were the Information Gain Attribute Ranking, ReliefF, Principal Components, Correlation-based Feature Selection, Consistency-based Subset Evaluation, and Wrapper Subset Evaluation. It is important to note that the authors only chose feature selection methods that rank features, not those who evaluate subsets of features. From the results presented in their work, it is clear that none of the feature selection methods discussed produce acceptable accuracy for all datasets and classifiers. Most of the methods increase accuracy on some datasets but decrease accuracy on other.

However, there has not been much work approaching the reliability of the feature selection process. Nguyen et. al. [32] performs an analysis of the main factors affecting the reliability in feature selection: (i) choice of feature selection method and (ii) search strategies for relevant features. A formal definition of a reliable feature selection process is given taking into account the main factors analysed: (i) steadiness of the classifier, and (ii) consistency of the search strategy. The steadiness, β, of a classifier C is defined as

$$\beta = \frac{Acc_F - \frac{1}{M}\sum_{i=1}^{M} \mid Acc_F - Acc_i \mid}{Acc_F} \tag{4.2}$$

given M. The greater the β, greater steadiness of the classifier. The consistency, α, of a search strategy is defined as

$$\frac{\mid X_1 \cap X_2 \cap \ldots \cap X_M \mid}{\mid X_1 \cup X_2 \cup \ldots \cup X_M \mid} = \alpha \tag{4.3}$$

where X_i is the selected subset of features. A method for addressing the main causes of low reliability in feature selection is proposed as Generic Feature Selection (GeFS) measure. The reliable feature selection process can then be seen as a maximisation problem finding $x \in \{0,1\}^n$ that maximises GeFS(x), as seen

in (3).

$$\max_{x \in \{0,1\}^n} GeFS(x) = \frac{a_0 + \sum_{i=1}^{n} A_i(x)x_i}{b_0 + \sum_{i=1}^{n} B_i(x)x_i} \tag{4.4}$$

The newly proposed method is applied to two datasets, the ECML/PKDD 2007 dataset and a new CSIC 2010 dataset created by the authors. The ECML/PKDD 2007 dataset was generated for the ECML/PKDD 2007 Discovery Challenge [77], however according to the authors in [32], it contains attack requests that are constructed blindly. Therefore, they produce their own dataset generated from an e-commerce web application. A comparative analysis is performed between two instances of the proposed GeFS and the heuristic search methods genetic search and Peng's method, max-relevance, min-redundancy (mRMR) [78].

An overview of their results in given in *Table 6 - Classification accuracy using proposed GeFS compared to full set of features* and *Table 7 - Consistency and steadiness of selected features using proposed GeFS compared to genetic algorithm and Peng's method*.

	CSIC 2010			**ECML/PKDD 2007**		
	Full set	$GeFS_{CFS}$	$GeFS_{mRMR}$	Full set	$GeFS_{CFS}$	$GeFS_{mRMR}$
Average accuracy	93.65	93.53	75.67	97.04	86.42	92.93

Table 6: Classification accuracy using proposed GeFS compared to full set of features [32]

	CSIC 2010			**ECML/PKDD 2007**		
	$GeFS_{CFS}$	$GeFS_{mRMR}$	GA_{CFS}	$GeFS_{CFS}$	$GeFS_{mRMR}$	mRMR
Consistency(%)	100	100	25	100	100	27
Steadiness(%)	99.87	80.80	97.33	89.05	95.76	92.14

Table 7: Consistency and steadiness of selected features using proposed GeFS compared to genetic algorithm and Peng's method [32]

From these results, we can see that the proposed GeFS provides good results when applying the GeFS class which is best fitted the data set (linear vs. non-linear relationship between features). Application of the proposed measures for

consistency and steadiness shows that the proposed GeFS provides good results here as well when best fitted GeFS class is applied.

Berg et al. [79] applied the GeFS method to the problem of *botnet* malware detection. The authors conduct their own experiments to construct a botnet malware dataset. Static and dynamical approaches are used creating a dataset of 7308 features. Data analysis shows that many features are linearly correlated, and the authors choose the $GeFS_{CFS}$ instance of the GeFS class. Experiments are conducted comparing the $GeFS_{CFS}$ with GA_{CFS} and BF_{CFS}. In their experiment, the authors use well-known classifiers Naive-Bayes, K-nearest neighbours, C4.5, SVM, and Bayesian Network. An overview of their results are shown in *Table 8 - Detection rate and false positive rate using proposed GeFS compared to genetic algorithm and Best-first*.

	Full-set	$GeFS_{CFS}$	BF_{CFS}	GA_{CFS}
Number of selected features	7308	12	30	2471
Average detection rate	93.76	95.11	93.77	90.74
Average false positive rate	17.96	9.74	7.89	12.83

Table 8: Detection rate and false positive rate using proposed GeFS compared to genetic algorithm and Best-first [79]

From these results, we can see that the proposed GeFS greatly reduce the number of features while on average increase the detection rate. Compared to similar feature selection methods, both the feature reduction and average detection rate are better. There is, however, no comparison of the steadiness and consistency of the resulting features in their work.

One of the pioneering works in anonymization is the work of Samarati and Sweeney [80]. They propose a generalisation method for anonymization where values are replaced with less precise alternatives that are semantically consistent and truthful.

In Burke [81], the authors propose a modified version of k-anonymity. As k-anonymity is an NP-hard problem; they propose a heuristic approach. The heuristic approach is applied to crime data, where good results are achieved. Compared to standard k-anonymity, the information loss is dropped nearly 50%. The largest limitation of their work is the fact that the approach is heuristic. According to literature [82], heuristic approaches that consider numerical and categorical data yield good results in terms of privacy preservation. However, the problem with heuristic approaches is that they cannot guarantee the optimal solution. As opposed to deterministic approaches, the methods may result in suboptimal solutions. Therefore, it is not possible to guarantee privacy preservation when applying heuristic approaches for data anonymisation.

Figure 15: Requirements for threat hunting platform as defined by Sqrrl[85]

4.3 Data Driven TI

When discussing state-of-the-art in data-driven TI, industry is where to look. In the last few years, numerous companies and product lines have surfaced applying Big Data technologies and mindsets to the classical security operation. The common denominator of many of these product lines is that they focus on automation of the process of combining TI and various internal data sources. The following section will describe some of the most prominent products in this field, which all have different focus.

Sqrrl[1] is a security analytics company focusing on a data-driven approach towards detecting and protecting against threats. Their product *Sqrrl Enterprise* unifies Big Data technologies including "Hadoop, linked data analysis, machine learning, Data-Centric Security, and advanced visualization." [83]. Their approach and company slogan is "*Target. Hunt. Disrupt*" [84]. That is, they focus on actively hunting and detecting threats, as opposed to the classical passively detecting threats. For this approach, they apply strong data-driven methods, unifying various data sources as well as external TI. The focus is on the application of data-driven methods on internal sources, while TI is used for support. They define four requirements for a threat hunting platform, as seen in *Figure 15 - Requirements for threat hunting platform as defined by Sqrrl*. These requirements briefly describe their solution.

On the other hand, we have Recorded Future[2]. They also provide a data-driven approach to detection and protection against threats, but with a different focus. Their product for TI teams applies Natural Language Processing (NLP) and machine learning for collecting and representing TI based on sources like the open, deep, and dark web.

[1]http://sqrrl.com/
[2]https://www.recordedfuture.com/

A third approach is provided by Digital Shadows[3]. Their approach focus on collecting and defining what is called a "*Digital shadow*", which can be considered a digital footprint. By understanding this digital shadow for both the organisation and potential adversaries, they achieve a situational awareness which can be used to detect and protect against threats [86].

4.4 Information Sharing

The principle of information sharing has been applied in various fields ranging from military sector to health sector. However, much of the approaches for information sharing is proprietary, and methods and formats used is created on a per scenario.

The government of New South Wales (NSW) in Australia has published for guides for information sharing between different entities [87]. The entities being government agencies, non-government sector, research sector, and the public. The guides are a part of the NSW Government ICT Strategy. They are designed to help entities prepare, manage and capture the benefits of information sharing. NSW government has also created a framework for information management [88]. The framework aims to support the management and use of data and information for the government and contains a set of standards, policies, guidelines, and procedures. It creates a common frame of reference which supports the sharing and re-use of information by other entities.

These previous works by governments provide good guidelines for information sharing, and they have identified the entities often performing information sharing. It is, however, a general approach, and may not be directly applicable to information sharing in regards to TI. Investigation of entities in such sharing must be performed.

4.5 Summary

In summary, we have discussed the current state-of-the-art related to the expected contribution of this thesis. We have discussed measuring of performance feature selection methods, and presented the newly proposed feature selection measure Generic Feature Selection (GeFS) by Nguyen et. al. [32]. Performance has been demonstrated by discussing results from previous work. Further, an overview of some common anonymisation techniques was presented. Finally, the application of information sharing in industry was discussed. It was demonstrated how government as well as organisations perform information sharing.

[3] https://www.digitalshadows.com/

5 Choice of Methods

In the previous chapters, an introduction to the problem, relevant theory, and current state of the art have been presented. In the following chapter, an overview of the methodology applied in answering the research questions is defined. This chapter clearly states how activities are performed, ensuring repeatability for future researchers.

Restating the research questions from *Chapter 1 - Introduction (p. 1)*.

1. How can data fusion and reduction for intrusion detection at an early stage using various heterogeneous sources be modelled?
2. Which features are reliable and trustworthy in classification of aggregated and correlated events, and which cannot be shared without anonymisation?

Section 5.1 - Interview (p. 41) will be used to answer research question one, and partly research question two. *Section 5.2 - Data Analysis and Experiment (p. 42)* will be used to answer research question two fully.

5.1 Interview

Selecting interview as part of the methodology was done for several reasons. It is, in information security, important to have communications between academia and industry. The continuous process of research, implementation, application, and feedback allows for new technology and techniques to be developed and used in the current and future fight against cyber criminals. By interviewing security experts in industry, feedback can be collected which then are used for further research.

5.1.1 Research Interview

A part of solving the research questions is to gather the experience of security experts. It is important to state questions without limiting their response to ensure as much information as possible is collected. Therefore, qualitative interview is best fitting [89]. When performing the interviews, best practices from literature where used, and an interview guide was created. An overview of the interviews is presented below, and the interview guide is presented in *Appendix A - Interview Guides*. It is important to note that the interviews were open, as the primary goal of these were not to compare the results. Instead, they were used as one of several information sources when answering the research questions.

We decided it was important to obtain as much information as possible regarding the relevant topics from the security experts, and thus relevant topics were also included in the interview. The interview was divided into three main parts;

Information Sharing discussed topics related to the sharing of information, focusing on sharing partners, trust, and technologies; *Threat Intelligence* discussed topics related to what and how intelligence was used in the organisations, how advanced current use was, as well as the effect of such intelligence; *Data Fusion* discussed topics related to how current fusion processes were designed, the potential requirements for such a system, as well as how such processes can be designed more efficiently.

The interview subjects selected are from various fields of the information security community. More specifically, we interviewed experts from private organisations, public organisations, and legal enforcement. The group of interview subjects consist of experts in both technical and operational positions, as well as strategic and tactical positions, allowing us the collect information and opinions from all levels of the information security community.

The interview process was performed in a combined effort with Ringdal [90]. Due to the small size of the security community in Norway, it was decided to cooperate to collect as much information as possible. Therefore, the interview guide presented in *Appendix A - Interview Guides* and the interview summaries presented in *Appendix B - Interview subject 1*, *Appendix C - Interview subject 2 and 3*, *Appendix D - Interview subject 4*, *Appendix E - Interview subject 5*, *Appendix F - Interview subject 6*, and *Appendix G - Interview subject 7* are also published in [90].

5.1.2 Method discussions

By applying qualitative research interviews as methodology, we seek to collect knowledge, opinions and experiences which are not easily captured using other interview methods. This approach complements our data-driven approach also applied in this research, and provides a wider understanding of the field, including its trends and challenges.

5.2 Data Analysis and Experiment

As data-driven security is main focus of this thesis, experiments on live data is central. In the following section, we will describe the methodology for our data analysis and experiments, providing insight on how and why decision were made.

5.2.1 Experimental Design

The experimental phase is based on the ML process as presented in *Figure 6 - ML process*. The specific design of this experiment is presented in *Figure 16 - Methodology for classification of intrusion events*.

Data Acquisition

The acquisition of data is already performed by the monitoring system at mnemonic as part of the Managed Security Service (MSS). Alerts is generated by various

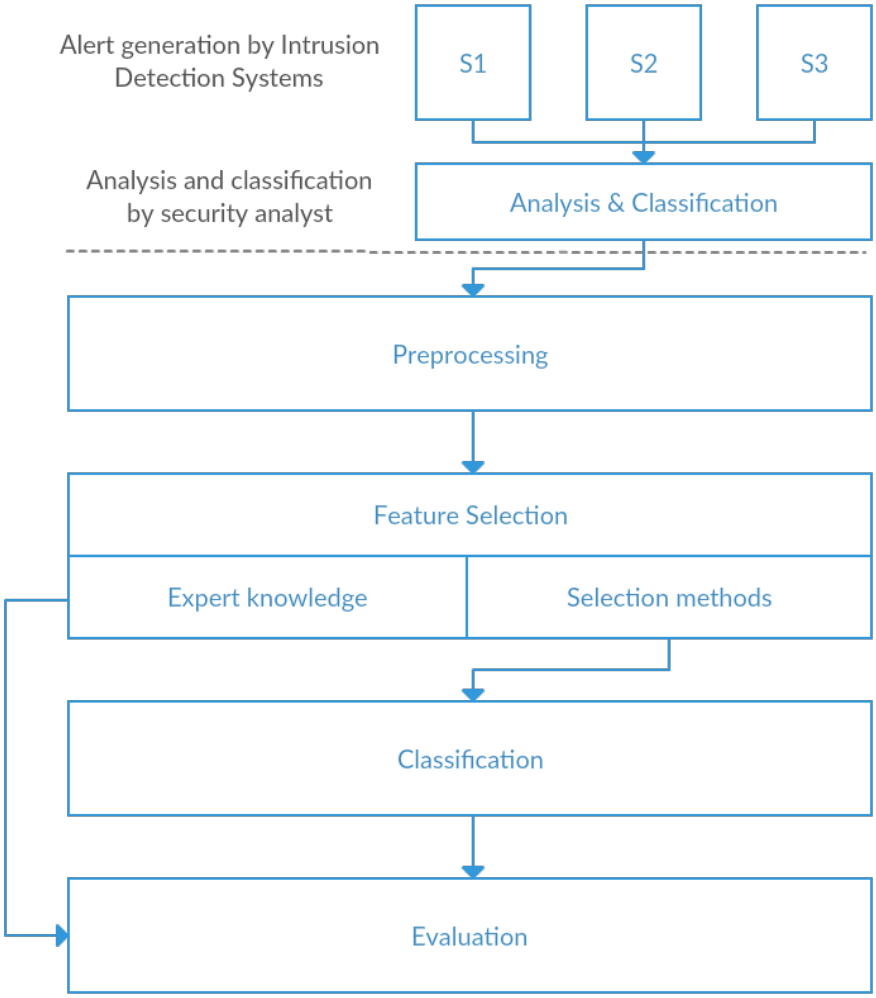

Figure 16: Methodology for classification of intrusion events

IDSs and other information sources, aggregated and correlated, and then analysed and classified by a security analyst. Classified events are then exported to Logstash[1] for the sake of this project. Characteristics of the dataset are presented in *Section 5.2.2 - Dataset (p. 45)*.

Preprocessing

As presented in *Section 3.1.1 - Preprocessing (p. 17)* preprocessing must often be performed to ready the data for feature extraction and selection. The following section describes how we prepared the acquired data for later phases.

Data is first acquired from Logstach using ElasticSearch Queries. Unfortu-

[1]https://www.elastic.co/products/logstash

nately ElasticSearch Queries requires specific attribute values defined, and so all features had to be predefined. Feature names were manually collected from Kibana web Graphical User Interface (GUI) and converted to Comma-Separated Values (CSV) files using script *Appendix H.1 - convert_features_to_csv.py*. The acquired data is by default presented as JSON data, and some preliminary processing must be performed for easier analysis. More specifically, the JSON data were converted to Pandas Dataframe. Thereafter feature names and values were cleaned for characters that which cause problems for Weka[2]. Weka is an Open-Source ML tool which allows for easy visualisation. Finally, some redundant features related to class were removed. These operations are all presented in script *Appendix H.2 - convert_clean.py*. One of the biggest challenges when processing data from large amounts of heterogeneous sources is the sparse nature of the data. With the heterogeneous output, creating a common frame of reference for all events cause high dimensions with sparse data, which are discussed in 5.2.2.

We decided to convert the dataset to .ARFF file format used by Weka. We applied the Weka function *weka.core.converters.CSVLoader* to convert the dataset to .ARFF format. Preliminary analysis was performed to create an understanding of the dataset. The findings from the visualisation and preliminary analysis is presented in *Section 5.2.2 - Dataset (p. 45)*.

Feature Selection

The feature selection process in this thesis is based on data-driven approach. However, we have included the application of expert knowledge in the methodology as well. The collection of expert knowledge was discussed in *Section 5.1 - Interview (p. 41)*, and will be incorporated when evaluating feature subsets. For data-driven approach, common feature selection methods implemented in Weka were applied. These feature selection methods were chosen on a combination of availability in Weka and from results and findings in the literature [91, 28, 31, 32, 78]. More precisely, we applied the following methods:

- *Infogain* (class `weka.attributeSelection.InfoGainAttributeEval` in Weka) - Implementation of the information gain measure. It calculates the information gained with the attribute with respect to the class. Let H be Shannon entropy [92], c be class, and A be attribute. Information gain can then be presented as $IG(c, A) = H_c - (H_c|H_A)$. This is a filter method, and evaluates attributes in isolation from another. It has therefore an information-theoretic point of view.
- *Correlation-based Feature Selection (Cfs)* (class `weka.attributeSelection.CfsSubsetEval` in Weka) - Implementation of the Correlation-based Feature Selection method proposed by Hall [93]. It is based on the idea that feature sets of high quality contain features that are highly correlated with the class while being

[2]http://www.cs.waikato.ac.nz/ml/weka/

uncorrelated with each other. It selects those attributes who have high correlation with the class, and low correlation with other attributes. Cfs has shown good performance in previous experiments [93]. It is a filter method, however, it has given comparable results to wrapper methods.
- *ReliefF*(class `weka.attributeSelection.ReliefFAttributeEval` in Weka) - Implementation of the ReliefF algorithm proposed by Kononenko [91, 94] which is an updated version of the Relief algorithm proposed by Kira and Rendell [74]. ReliefF takes into account attributes with strong dependencies, the difference in attribute values, difference in class, and the distance between the examples [28].

Classification

For the classification of the datasets, we decided to apply the classifiers implemented in Weka. We chose these classifiers based on availability in Weka and on the results and findings literature [28, 30, 32]. More precisely, we applied the following classifiers as introduced in *Section 3.1.2 - Feature Selection (p. 19)*:

- *J48* (Class `weka.classifiers.trees.J48` in Weka)
- *IBk* (Class `weka.classifiers.lazy.IBk` in Weka)
- *Naive Bayes* (Class `weka.classifiers.bayes.NaiveBayes` in Weka)
- *Bayes Net* (Class `weka.classifiers.bayes.BayesNet` in Weka)
- *Random Tree* (Class `weka.classifiers.trees.RandomTree` in Weka)
- *Random Forest* (Class `weka.classifiers.trees.RandomForest` in Weka)
- *SVM* (Class `weka.classifiers.functions.LibSVM` in Weka)

Evaluation

For evaluating the performance of each classifier on each dataset, we used classification accuracy as introduced in *Chapter 3 - Machine Learning and Data Fusion (p. 17)*. This allows us to easily compare classifiers on the same dataset, as well as the feature selection methods on same classifiers.

5.2.2 Dataset

The dataset acquired for this thesis consist of security incidents over 60 days analysed and classified by an analyst. It has originally 66621 vectors with 667 number of features and 10 classes. An overview of all available features in the dataset is presented in *Appendix I - Features*. The dataset is generated as part of the security monitoring where various sensors and log sources are correlated and aggregated. Correlation is done to provide as much information about each event as possible while aggregation is performed to ensure the analyst is presented with an acceptable number of events.

Classes

The possible classes in the dataset are presented below with a short description of each. The class separation is already being used by the analysis system which

the data was acquired from, thus not defined by the author. Examples of central features for each class is presented. These can be considered as features interesting to share related to each type of class. We have also presented an example of all interesting features for the class *'Exposure to malicious code'* in *Table 9 - All interesting features for class* 'Exposure to malicious code'. All customer relevant records which must not be shared without anonymisation techniques applied is marked in red. Fields which may be a problem is marked in yellow. These are mostly features which, depending on traffic direction, can be both victim and attacker. Extra caution must therefore be taken to ensure that traffic direction is known and only attacker information is shared.

Exposure to malicious code

Download of malicious code, or access to a site hosting malicious code. Malicious code is computer code or web scripts designed to perform malicious actions on target systems. When sharing data on such events, elements like domain, IP, malware classification, and source country is of interest. From our dataset, features like `destination.network- Address.address`, `properties.domain`, `attackInfo.attackIdentif` and `destination.geoLocation.countryCode` is features that valuable when sharing. However, these are not the only interesting features in such events. *Table 9 - All interesting features for class* 'Exposure to malicious code' presents all features of interest in 'Exposure to malicious code' events. These features are extracted based on feedback and discussions with professionals working with security analysis. Features which have business sensitivity concerns are marked with red.

Unauthorised Access or Intrusion

Unauthorised users accessing system either by benign methods or exploitation. This is a successful attempt of an attacker actively avoiding implemented security measures to access unauthorised systems. Such activity can be either automated or manually. When sharing data on such events, elements like source IP, access technique, and destination is of interest. From our dataset, features like `source.network- Address.address`, `destination.port`, and `customerInfo.name` are some of the features valuable for sharing.

Malicious code infection

A malicious code infections that is verified. Activity which indicates that the client or server is infected has been observed. Such activity may be e.g CC traffic, port scan, or DOS traffic. When sharing data on such events, elements like destination domain and IP, communication channel and timestamp is of interest. From our dataset, features like `destination.networkAddress.address`, `properties.domain`, `destination- .port`, and `timestamp` are some of the features valuable for sharing.

Poor practice or policy violation

Unsafe use of systems, or violation of company policy. The use of technologies often associated with malicious behaviour can be classified as this. E.g use of The Onion Router (TOR) from company clients. This can also be an activity which violates the policy defined by the company, e.g access of websites with pornographic content, the use of Virtual Private Network (VPN) or other tools for proxy avoidance. When sharing such events, elements like technology, communication channel, and destination domain and IP is of interest. From our dataset, features like `destination.port`, `protocol`, `destination.networkAddress.address`, and `properties.domain` are some of the features valuable for sharing.

Reconnaissance

Reconnaissance activity either external or internal. Activities often associated with reconnaissance activity such as port scan and automated exploitation attempt. When sharing such events, elements like technique, source IP, and destination domain and IP is of interest. From our dataset, features like `attackInfo.attackIdentifier` `source.network- Address.address`, `destination.port`, `protocol`, `count`, `destination-.networkAddress.address`, and `properties.domain` are some of the features valuable for sharing.

Data leakage

Leakage of information. Information can be leaked either by an attacker actively exploiting a vulnerability in the target system, making the system return potential sensitive information, or by target users performing actions which leak sensitive information. This can be activities like accessing phishing sites, responding to phishing emails, or sending emails to the wrong recipients. When sharing such events, elements like organisation, destination information, source information and technologies is of interest. From our dataset, features like `customerInfo.name`, `destination.networkAddress.address`, `properties.domain`, `source.- networkAddress.addr` and `protocol` are some of the features valuable for sharing.

Suspected or confirmed targeted attack

Activity related to targeted attacks. Such activity is often hard to detect due to its low profile. The activities can be anything ranging from reconnaissance, emails containing malicious content, to phishing emails. Therefore, elements interesting for sharing is often on a per case basis, however elements like organisation, techniques, technologies, and source information is some of the interesting elements. From our dataset, features like `customerInfo.name`, `destination.port`, `attackInfo.attackIdentifier`, `source.port`, `protocol`, `source.network- Address.address`, and `properties.domain` are some of the features valuable for sharing.

Failed authentication attempts
Failed attempts to log into a system or service. This can either be attributed to a wrong username password combination, or to an attempt to access resources the user are not authorised to access. Such activity can often be associated with bruteforce attacks. When sharing such events, elements like user information, source information, destination information and technology is some of the interesting elements. From our dataset, features like `properties.ad_src__user__name`, `source.networkAddress.address`, and `protocol` are some of the features valuable for sharing.

Misconfigured device
Activity related to devices functioning incorrectly. Misconfigured devices can cause network problems by not operating as expected, or by using more resources than it should. These types of events contain little information that are interesting to share. They provide little value for other organisations.

Adware
Activity related to software presenting users with ads. This type of software is often harmless, however it can be annoying. It has been observed that such software can create vulnerabilities which can be exploited by attackers. When sharing such events, elements like destination information and communication technique is of interest. From our dataset, features like `destination.networkAddress.address`, `properties.domain`, `destination- .port`, and `protocol` are some of the features valuable for sharing.

No incident
Benign activity which have been wrongly classified by monitoring systems. This is the most common type of events, as current security tools produce large amounts of FP. Information regarding these types of events may be interesting to share as part of a feedback loop if the TI has been collected from external sources.

From the overview of classes and their central features, it is clear that many classes can be identified using the same features. Features that are common are those describing attacker infrastructure and techniques, like IP, domain, and Uniform Resource Locators (URLs). However, these types of features are often based on reputation; therefore, they provide little value by itself when classifying using ML. Reputation sources are necessary, and will by itself be a feature.

The class distribution of the original dataset is presented in *Figure 17 - Class distribution: original dataset*. It is clear that the dataset is imbalanced in regards to 'No incident' versus all the other classes. However, this is not surprising. It is known that current IDS produce large amounts of FP [1]. When generating the datasets, we could collect events so that the class distribution were uniform;

Feature	Description	Example
destination.networkAddress.address	IP address of destination	192.168.0.1
destination.geoLocation.-country	Country hosting malicious code	Russia
properties.domain	Domain	google.ru
attackInfo.attackIdentifier	Signature triggered	Snort x:xxxxx
reputation.count	No. of reputation sources containing IP or domain	4
reputationRoles	Behaviour according to reputation sources	Malware-server
source.networkAddress.-address	IP address of source	192.168.0.2
destination.port	Port number on destination	80
customerInfo.name	Name of customer	mnemonic
properties.ad_requestURL	The requested URL	http://www.google.ru/index.php
properties.estreamer_malware_filesize	Filesize of downloaded code	74019
properties.reputationRoles	Reputation on IP or domain (enrichment)	malware-server, cc-server

Table 9: All interesting features for class *'Exposure to malicious code'*

however, we decided not to do this as that would create a clearly different scenario than what is currently observed in intrusion detection. When security analysts perform analysis of events, several classifications are performed. It can be described as a two-step process: *(i)* Is the activity malicious or suspicious? *(ii)* If so, what type of activity is it? Therefore, we decided to generate two new datasets based on the original. The datasets are presented *Section 5.2.2 - Dataset generation (p. 49)*. By doing so, we hoped to counter for the skewed distribution of classes. It also allows us to investigate whether features have different value depending on whether it is to classify malicious or benign, or if it is to classify what type of malicious activity event is.

Dataset generation

The first dataset, coined *Binary dataset*, is a binary classification problem. The two classes available are `{no incident, malicious}`. The dataset is generated by changing all events not having the class *'no incident'* to *'malicious'*. The class distribution of the binary dataset is presented in *Figure 18 - Class distribution: binary dataset*.

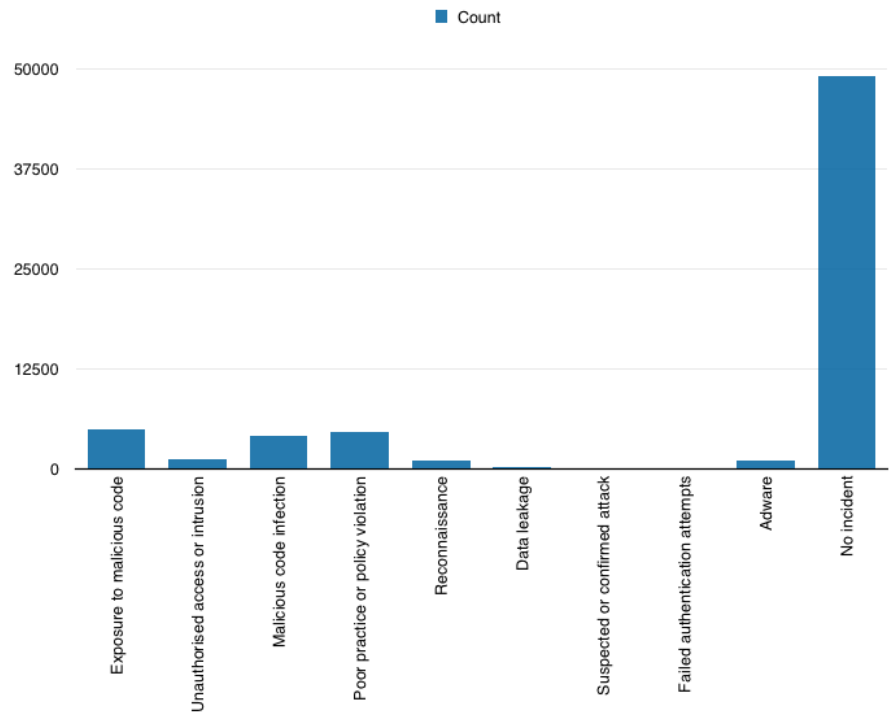

Figure 17: Class distribution: original dataset

The second dataset, coined *Malicious dataset*, is a multiclass classification problem. It has all the same classes as the original dataset except *'no incident'*. Because of this, it has significantly fewer samples, namely 17580. The dataset is generated by dropping all events having the class *'no incident'*. The class distribution of the malicious dataset is presented in *Table 19 - Class distribution: malicious dataset*.

5.2.3 Method discussions

Our method for experiments adhere to the common ML process as described in *Section 3.1 - Machine Learning (p. 17)*. The datasets acquired is from real world networks, and can, therefore, be assumed to be a good representation of real world IDS events.

5.3 Summary

In summary, we have in this chapter presented our methodology for this thesis. Interview methodology was presented and justified, and the interview guide was discussed. Further, we presented detailed methodology of experiments. Finally, descriptions of thesis datasets were given.

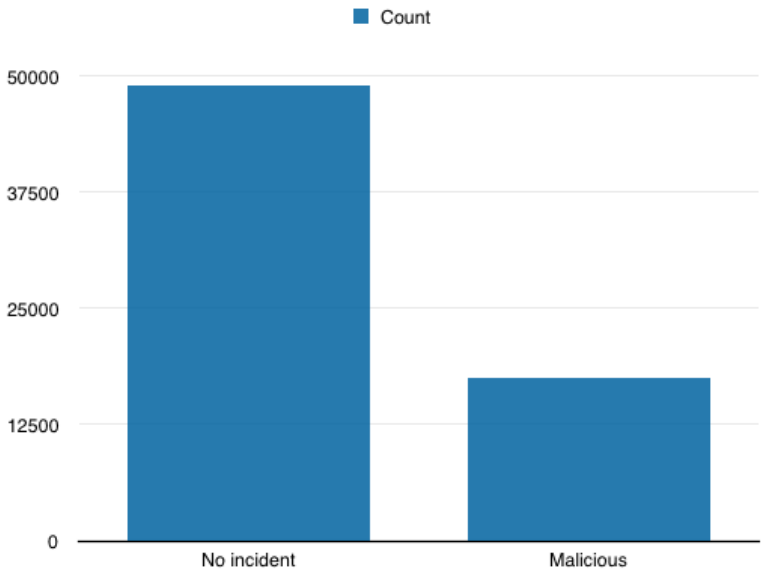

Figure 18: Class distribution: binary dataset

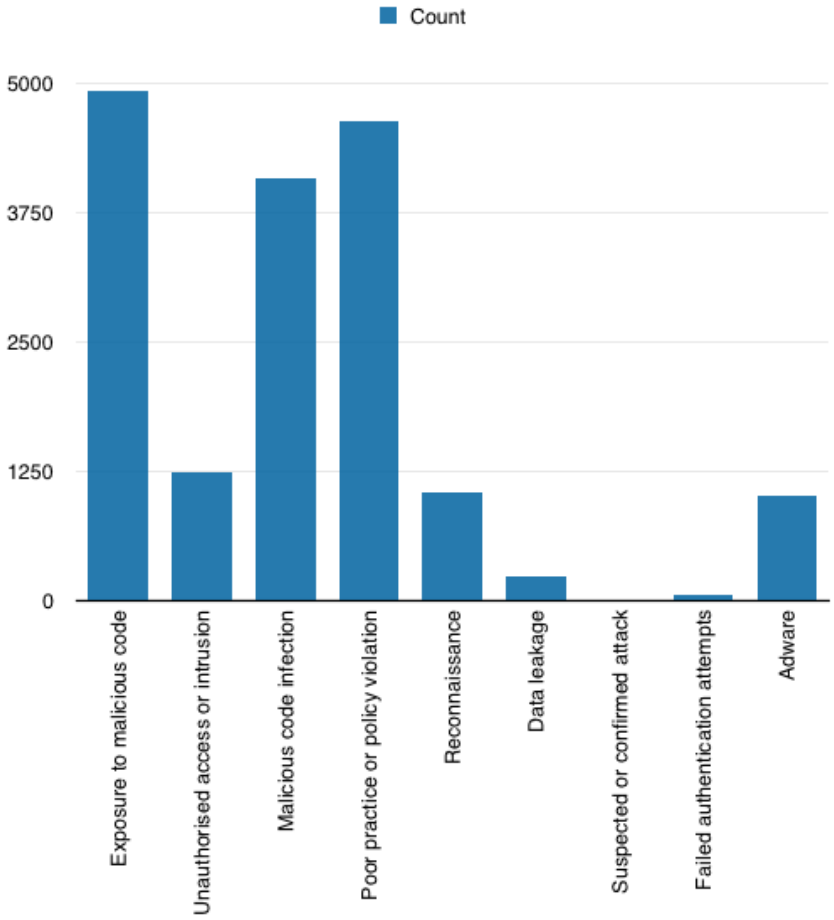

Figure 19: Class distribution: malicious dataset

6 Reliable and Trustworthy Features in Aggregated Intrusion Detection Events

In the previous chapters we have discussed the motivation and expected contribution of the thesis. Further, we presented relevant theory as well as state-of-the-art in topics related to our research questions. Finally, we presented our methodologies for solving the research questions. In the following chapter we will discuss the experiments conducted including results. An overview of the experimental environment is given, providing guidelines for future research. The results from the feature selection process is demonstrated, and the quality of feature subsets are evaluated. Further, the results related to features from the interview process is compared against experimental observations.

6.1 Experimental Environment

The following section describes the physical and logical environment in which the experiments have been conducted.

6.1.1 Physical Environment

Some preliminary testing and visualisation were performed on the Macbook Air, however feature selection process and consecutive training and evaluation process were performed on the HP DL360.

Macbook Air 2015

Processor 2.2 GHz Intel Core i7

Memory 8 GB 1600 MHz DDR3

Storage Flash Storage 121 GB

Operating System OS X El Capitan Version 10.11.4

HP DL360 Gen9 2x10 cores

Processor 2x Intel Xeon E5-2650 v3 (10 core, 2.3 GHz, 25MB, 105W)

Memory 64GB RDIMM

Storage 2x300GB + 6x900GB

Operating System CentOS Linux release 7.2.1511

6.1.2 Logical Environment

Below, the software used for experiments are presented. The computational heavy tasks like feature selection, training, and evaluation were performed using Weka. Rest of software were used for data acquisition and formatting.

- Python 3.5.1 [95]
- Pandas 0.7.1 [96]
- Scikit-learn 0.17 [97]
- Pip 7.1.2 [98]
- Weka 3.6 [99]
- Logstash 2.2 [100]

6.2 Experimental Scenarios

In the following section, we present the results and analysis based on our work on the data used in the thesis. Feature subsets are presented and discussed. Further, a data-driven evaluation of the feature subsets is presented.

6.2.1 Feature Selection

Adhering to methodology presented in *Chapter 5 - Choice of Methods (p. 41)*, we performed feature selection. As we generated three datasets from the original dataset, we will discuss the feature selection analysis separate. For each dataset, we applied feature selection methods with same option for Weka command. Methods and switches are listed in *Table 10 - Switches for feature selection using Weka*. Default settings from Weka were used.

Method	Options
ReliefF	-M -1 -D 1 -K 10
Ranker	-T -1.7976931348623157E308 -N -1
InfoGain	-M
Ranker	-N 50
Cfs	-M
BestFirst	-D 1 -N 5

Table 10: Switches for feature selection using Weka

Original dataset

The contribution of each feature according to the three chosen feature selection methods is presented in *Table 11 - Feature contribution: Original dataset(1)* and *Table 12 - Feature contribution: Original dataset (2)*. All customer relevant records which must not be shared without anonymisation techniques applied is marked in red. Fields which may be a problem is marked in yellow. These are mostly features which, depending on traffic direction, can be both victim and attacker.

Extra caution must, therefore, be taken to ensure that traffic direction is known and only attacker information is shared.

Cfs
Attribute
attackInfo.attackCategoryID
attackInfo.attackIdentifier
customerInfo.id
destination.geoLocation.locationID
priority
severity

Table 11: Feature contribution: Original dataset(1)

ReliefF		Infogain	
Merit	Attribute	Merit	Attribute
0.79924	source.geoLocation.locationName	1.54244	timestamp
0.72169	destination.geoLocation.countryCode	1.54244	id
0.72169	destination.geoLocation.countryName	1.54244	lastUpdatedTimestamp
0.62089	customerInfo.name	1.53428	startTimestamp
0.62089	customerInfo.shortName	1.53428	startTime
0.60000	destination.networkAddress.address	1.53171	endTimestamp
0.59371	destination.geoLocation.locationName	1.10600	source.networkAddress.address
0.40000	source.networkAddress.address	1.08195	destination.networkAddress.address
0.22088	destination.geoLocation.locationID	0.91611	attackInfo.attackIdentifier
0.18779	customerInfo.id	0.79801	attackInfo.alarmDescription
0.15707	source.geoLocation.latitude	0.78151	attackInfo.alarmID
0.13345	attackInfo.attackIdentifier	0.54262	attackInfo.attackCategoryName
0.13345	attackInfo.alarmDescription	0.54208	attackInfo.attackCategoryID
0.11509	properties.reputationCount	0.51875	customerInfo.shortName
0.10505	source.port	0.51875	customerInfo.name
0.10000	lastUpdatedTimestamp	0.51810	customerInfo.id
0.09509	attackInfo.attackCategoryName	0.48083	destination.geoLocation.locationID
0.06956	destination.geoLocation.latitude	0.29464	destination.port
0.05366	destination.geoLocation.longitude	0.28948	destination.geoLocation.countryName
0.03752	source.geoLocation.countryName	0.28948	destination.geoLocation.countryCode
0.03752	source.geoLocation.countryCode	0.28892	destination.geoLocation.locationName
0.02968	destination.port	0.27912	source.geoLocation.locationID
0.02330	source.networkAddress.public	0.26869	priority
0.02277	source.geoLocation.longitude	0.26869	severity
0.02128	location.name	0.25388	location.name
0.02128	location.shortName	0.25388	location.shortName
0.02122	protocol	0.25345	location.id
0.01935	source.geoLocation.locationID	0.19137	source.port
0.01673	attackInfo.attackCategoryID	0.18168	source.geoLocation.locationName
0.01477	destination.networkAddress.public	0.15982	protocol
0.01252	attackInfo.alarmID	0.15978	protocolID
0.00213	location.id	0.14461	source.geoLocation.countryName
0.00209	priority	0.14461	source.geoLocation.countryCode
0.00209	severity	0.13956	normalizedURL
0.00186	destination.networkAddress.host	0.10401	destination.geoLocation.latitude
0.00081	protocolID	0.10366	destination.geoLocation.longitude
0.00016	count	0.10078	count
0.00006	destination.networkAddress.maskBits	0.08723	properties.reputationCount
0.00000	normalizedURL	0.06551	source.geoLocation.longitude
0.00000	detailedEventIDS.aggregated	0.06409	source.geoLocation.latitude
0.00000	destination.networkAddress.multicast	0.03817	destination.networkAddress.public
0.00000	detailedEventIDS.deviceEventID	0.03755	source.networkAddress.public
0.00000	destination.networkAddress.ipv	0.02837	destination.networkAddress.maskBits
0.00000	detailedEventIDS.customerID	0.01889	destination.networkAddress.host
0.00000	detailedEventIDS.deviceID	0.01709	srcDstGeoDistance
0.00000	attackInfo.auditCategories.key	0.00006	destination.networkAddress.ipv
0.00000	source.networkAddress.maskBits	0.00006	source.networkAddress.ipv
0.00000	source.networkAddress.multicast	0.00003	source.networkAddress.host
0.00000	comments.user.group	0.00000	properties.:pam_dns_tunnel_idle_timeout
0.00000	source.networkAddress.ipv	0.00000	properties.:statement

Table 12: Feature contribution: Original dataset (2)

Binary dataset

The contribution of each feature according to the three feature selection methods is presented in *Table 13 - Feature contribution: Binary dataset(1)* and *Table 14 - Feature contribution: Binary dataset (2)*. All customer relevant records which must not be shared without anonymisation techniques applied is marked in red. Fields which may be a problem is marked in yellow. These are mostly features which, depending on traffic direction, can be both victim and attacker. Extra caution must, therefore, be taken to ensure that traffic direction is known and only attacker information is shared.

	Cfs
Attribute	
attackInfo.attackIdentifier	
lastUpdatedTimestamp	
normalizedURL	
priority	
severity	

Table 13: Feature contribution: Binary dataset(1)

ReliefF		Infogain	
Merit	Attribute	Merit	Attribute
1.00000	destination.geoLocation.countryName	0.83226	timestamp
1.00000	destination.geoLocation.countryCode	0.83226	id
0.79856	source.geoLocation.locationName	0.83226	lastUpdatedTimestamp
0.60000	destination.networkAddress.address	0.82583	startTimestamp
0.59860	destination.geoLocation.locationName	0.82583	startTime
0.40000	source.networkAddress.address	0.82322	endTimestamp
0.28000	destination.geoLocation.locationID	0.51719	source.networkAddress.address
0.14375	properties.reputationCount	0.48174	destination.networkAddress.address
0.10000	lastUpdatedTimestamp	0.36203	attackInfo.attackIdentifier
0.09061	destination.port	0.27836	attackInfo.alarmDescription
0.00000	count	0.25836	attackInfo.alarmID
0.00000	source.geoLocation.locationID	0.16503	customerInfo.shortName
0.00000	normalizedURL	0.16503	customerInfo.name
0.00000	destination.geoLocation.latitude	0.16482	customerInfo.id
0.00000	destination.geoLocation.longitude	0.16078	destination.geoLocation.locationID
0.00000	customerInfo.name	0.13282	attackInfo.attackCategoryName
0.00000	customerInfo.shortName	0.13253	attackInfo.attackCategoryID
0.00000	comments.user.userName	0.12025	priority
0.00000	comments.user.timezone.offset	0.12025	severity
0.00000	comments.userID	0.09808	destination.geoLocation.locationName
0.00000	destination.networkAddress.ipv6	0.09786	source.geoLocation.locationID
0.00000	customerInfo.id	0.08496	location.shortName
0.00000	destination.networkAddress.host	0.08496	location.name
0.00000	attackInfo.alarmDescription	0.08470	location.id
0.00000	destination.networkAddress.maskBits	0.07673	normalizedURL
0.00000	detailedEventIDS.loggerID	0.07379	source.geoLocation.locationName
0.00000	detailedEventIDS.type	0.07023	source.port
0.00000	detailedEventIDS.writable	0.06694	destination.geoLocation.countryCode
0.00000	location.name	0.06694	destination.geoLocation.countryName
0.00000	location.shortName	0.06468	destination.port
0.00000	priority	0.04402	protocolID
0.00000	detailedEventIDS.timestamp	0.04402	protocol
0.00000	detailedEventIDS.eventID	0.04232	source.geoLocation.countryName
0.00000	destination.networkAddress.multicast	0.04232	source.geoLocation.countryCode
0.00000	detailedEventIDS.deviceID	0.03889	properties.reputationCount
0.00000	destination.networkAddress.public	0.02936	destination.geoLocation.longitude
0.00000	comments.user.timezone.description	0.02858	destination.geoLocation.latitude
0.00000	detailedEventIDS.aggregated	0.02046	source.networkAddress.public
0.00000	detailedEventIDS.customerID	0.01634	count
0.00000	detailedEventIDS.deviceEventID	0.01254	source.geoLocation.longitude
0.00000	comments.user.timezone.id	0.01223	source.geoLocation.latitude
0.00000	location.id	0.00577	srcDstGeoDistance
0.00000	comments.user.realName	0.00553	destination.networkAddress.public
0.00000	source.geoLocation.countryCode	0.00233	destination.networkAddress.maskBits
0.00000	attackInfo.attackCategoryName	0.00013	destination.networkAddress.host
0.00000	source.geoLocation.countryName	0.00003	source.networkAddress.host
0.00000	severity	0.00000	source.networkAddress.ipv6
0.00000	source.geoLocation.longitude	0.00000	destination.networkAddress.ipv6
0.00000	protocolID	0.00000	properties.:stats_interval
0.00000	protocol	0.00000	properties.:pam_dns_tunnel_detection_rate

Table 14: Feature contribution: Binary dataset (2)

Malicious dataset

The contribution of each feature according to the three feature selection methods is presented in *Table 15 - Feature contribution: Malicious dataset(1)* and *Table 16 - Feature contribution: Malicious dataset (2)*. All customer relevant records which must not be shared without anonymisation techniques applied is marked in red. Fields which may be a problem is marked in yellow. These are mostly features which, depending on traffic direction, can be both victim and attacker. Extra caution must, therefore, be taken to ensure that traffic direction is known and only attacker information is shared.

	Cfs
Attribute	
attackInfo.alarmID	
attackInfo.attackCategoryID	
attackInfo.attackCategoryName	
attackInfo.attackIdentifier	
customerInfo.id	
destination.networkAddress.address	
lastUpdatedTimestamp	
priority	

Table 15: Feature contribution: Malicious dataset(1)

ReliefF		Infogain	
Merit	Attribute	Merit	Attribute
1.00000	destination.networkAddress.address	2.700061	timestamp
1.00000	attackInfo.attackIdentifier	2.700061	lastUpdatedTimestamp
1.00000	lastUpdatedTimestamp	2.700061	id
1.00000	startTimestamp	2.693663	endTimestamp
1.00000	startTime	2.693483	startTimestamp
0.92395	location.name	2.693483	startTime
0.92395	location.shortName	2.281698	destination.networkAddress.address
0.92386	customerInfo.name	2.239209	source.networkAddress.address
0.92386	customerInfo.shortName	2.10298	attackInfo.attackIdentifier
0.69168	destination.geoLocation.countryCode	1.97171	attackInfo.alarmDescription
0.69168	destination.geoLocation.countryName	1.942402	attackInfo.alarmID
0.58358	source.networkAddress.public	1.553765	attackInfo.attackCategoryName
0.42406	attackInfo.alarmID	1.551974	attackInfo.attackCategoryID
0.23515	source.port	1.343407	customerInfo.name
0.23257	attackInfo.attackCategoryID	1.343407	customerInfo.shortName
0.16778	destination.geoLocation.locationID	1.337916	customerInfo.id
0.14828	customerInfo.id	1.202608	destination.geoLocation.locationID
0.13627	location.id	0.857031	destination.port
0.10167	protocol	0.848468	destination.geoLocation.countryName
0.06700	properties.reputationCount	0.848468	destination.geoLocation.countryCode
0.05172	destination.networkAddress.public	0.724944	destination.geoLocation.locationName
0.04272	destination.geoLocation.latitude	0.665828	source.geoLocation.locationID
0.03933	destination.port	0.640931	location.shortName
0.03344	destination.geoLocation.longitude	0.640931	location.name
0.00343	destination.networkAddress.host	0.640079	location.id
0.00279	protocolID	0.564548	priority
0.00162	priority	0.564548	severity
0.00162	severity	0.471622	source.port
0.00030	count	0.439556	protocol
0.00009	destination.networkAddress.maskBits	0.439556	protocolID
0.00000	destination.geoLocation.locationName	0.409402	source.geoLocation.locationName
0.00000	attackInfo.alarmDescription	0.388225	source.geoLocation.countryCode
0.00000	source.geoLocation.countryCode	0.388225	source.geoLocation.country_Name
0.00000	source.geoLocation.countryName	0.311059	count
0.00000	normalizedURL	0.282621	destination.geoLocation.longitude
0.00000	properties.:tcp_hynacks	0.280898	destination.geoLocation.latitude
0.00000	properties.:tcp_connections_timeouts_synfin	0.240832	normalizedURL
0.00000	properties.:tcp_events_attack	0.201629	source.geoLocation.longitude
0.00000	properties.:tcp_events_audit	0.194413	source.geoLocation.latitude
0.00000	properties.:tcp_hyndups	0.18448	properties.reputationCount
0.00000	properties.:tcp_connections_fullduplex	0.123982	destination.networkAddress.public
0.00000	properties.:tcp_connections_timeouts_abort	0.098172	destination.networkAddress.maskBits
0.00000	properties.:tcp_connections_timeouts_data	0.071127	destination.networkAddress.host
0.00000	properties.:tcp_connections_embryonic	0.06519	source.networkAddress.public
0.00000	properties.:tcp_connections_flushed	0.04346	srcDstGeoDistance
0.00000	properties.:tcp_packets	0.000223	destination.networkAddress.ipv6
0.00000	properties.:tcp_connections_onesided	0.000223	source.networkAddress.ipv6
0.00000	properties.:tcp_hyns	0	properties.:queue_full
0.00000	properties.:totalMessages	0	properties.:score
0.00000	properties.:tcp_rsts	0	properties.:recordLen

Table 16: Feature contribution: Malicious dataset (2)

6.2.2 Evaluation

When feature selection methods have been applied, it is of interest to measure the performance of each feature subset. In the following section, we present our findings when applying the previously discussed classifier methods on our subsets. For each dataset, we applied classifier method with same options for Weka. Method and switches are listed in *Table 17 - Switches for classifier using Weka*.

Method	Options
J48	-C 0.25 -M 2
IBk	-K1 - W0 - A <linearsearch> -A EuclideanDistance
NaiveBayes	N/A
BayesNet	-D -Q K2- P 1 -S BAYES -E -A 0.5
RandomForest	-I 10 -K 0 -S 1
RandomTree	-D -K 0 -M 1.0 -S 1
SVM	-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 \ -P 0.1 -seed 1

Table 17: Switches for classifier using Weka

Our results from the application of classifiers on datasets are presented in *Figure 20 - Classification results*. For each dataset, we have applied feature selection method and applied classifier methods on each new dataset based on feature selection process. Classifier performance is determined using classification accuracy and k-fold with K = 10.

Because of problems related to computational complexity, we decided to split the datasets into several parts, therefore only applying classification methods on a subset of the whole dataset. When doing this, maintaining class distribution in the new subsets is important to ensure results are representable. Therefore, we applied the Weka filter *weka.filters.supervised.instance.StratifiedRemoveFolds* which prepares the dataset for cross-validation. The folds created are *stratified*, and so class distribution is preserved. The dataset was split into four stratified folds. Experiments marked with an * (asterisk) have been performed on two stratified folds of the full dataset, i.e. 50% of the dataset, while experiments marked with ** (two asterisks) have been performed on one stratified fold of the full dataset, i.e. 25%.

6.3 Discussion

As discussed previously, reliability is important in the classification of events. To achieve reliable security operation using ML, reliable feature selection methods must be applied. In the following section, we will discuss how reliable the methods applied in our research are. Further, we will discuss the results of our exper-

Dataset	Feature Selection	J48	IbK	NaiveBayes	BayesNet	RandomForest (I = 10)	RandomTree	SVM	Average score (feature set)
Original	No feature selection	*90.93%	**91.96%	*54.06%	*67.78%	*91.49%	**89.07%	**9.97%	70.75%
	ReliefF	*90.94%	*91.43%	*61.60%	*74.41%	*95.25%	**93.88%**	*25.07%	76.08%
	InfoGain	*90.94%	*93.05%	*54.06%	*67.87%	*88.79%	*93.26%	**9.79%	58.12%
	Cfs	*91.59%	*88.22%	85.84%	*85.05%	*91.72%	*91.52%	10.59%	77.79%
Malicious	No feature selection	92.00%	**94.73%**	78.80%	86.84%	93.91%	*71.60%	**48.20%	80.87%
	ReliefF	91.84%	**94.73%**	73.61%	84.33%	94.53%	75.11%	37.50%	78.81%
	InfoGain	91.97%	**94.73%**	78.80%	86.85%	92.19%	68.64%	**48.12%	80.19%
	Cfs	90.96%	94.60%	85.90%	88.33%	94.29%	*84.30%	**85.07%	89.06%
Binary	No feature selection	*88.70%	*93.43%	*91.23%	*90.47%	**91.11%	**87.86%	**84.96%	89.68%
	ReliefF	*83.29%	*93.27%	*92.41%	*92.12%	*95.00%	**93.43%	**82.57%	90.30%
	InfoGain	*88.72%	*93.43%	*91.08%	*90.41%	*89.43%	*89.57%	**85.37%	89.72%
	Cfs	*83.29%	**95.03%**	*92.49%	*93.49%	*94.59%	*94.66%	*82.11%	90.81%
Average score (classifier)		89.60%	93.22%	78.32%	84.00%	92.69%	86.08%	50.78%	

Figure 20: Classification results

iments, and compare these against features identified in the interview process identifying common elements. Finally we discuss the complications the difference in the feature sets cause.

From the results in *Figure 20 - Classification results*, we can see that on average the Cfs method provides the best result for all three datasets. It has been observed in literature that feature sets generated using Cfs equalled or bettered the accuracy of using the full feature set [101], and our experimental results reflect this well. In almost all cases, classification accuracy waw increased using Cfs. Looking at average accuracy score on the Cfs feature set and full feature set we observe a significant increase as shown in *Table 18 - Performance increase using Cfs*. From *Table 12 - Feature contribution: Original dataset (2)*, *Table 16 - Feature*

Dataset	Full feature set	Cfs feature set	Increase
Original	70.75%	77.79%	7.04%
Malicious	80.87%	89.06%	8.19%
Binary	89.68%	90.81%	1.13%

Table 18: Performance increase using Cfs

contribution: Malicious dataset (2), and *Table 14 - Feature contribution: Binary dataset (2)* we observe that number of selected features is in range 5 - 9 features. Comparing these number against the total number of features, $n = 667$, a significant increase in computational performance is expected as well. When classifying security events for decision support for analysts, it is of interest to perform this in real-time or near real-time; thus, computational performance is important.

Regarding classifier performance, IBk performed best on average with an accuracy of 93.22% with RandomForest only 0.53% points behind with an accuracy of 92.69%. However, we should note that of these only RandomForest had an increase in accuracy for all three datasets when applying Cfs feature set compared to the full feature set.

The highest classification accuracies for each dataset have been colourized in green, and the lowest classification accuracies for each dataset have been colourized in red. From this, we can observe which combination of feature selection method and classifier method perform best on aggregated intrusion detection events in each of the three classification problems. We can observe from these results that the classifier performing best on each dataset varies. This is a good representation of the No Free Lunch Theorem discussed in *Section 3.1.5 - Challenges (p. 24)*, and shows why we should apply different classifiers depending on which classification problem we are solving.

To discuss how reliable our feature selection methods are, we will apply previously proposed definition of reliability in the feature selection process by Nguyen [32]. As described in *Section 4.2 - Reliable Feature Selection and Feature Anonymisation (p. 35)*, the reliability of the feature selection process can be defined as $(\alpha, \beta)_{\text{reliable}}$, where α is the steadiness of the classifier, and β is the consistency of the search method. Due to the nature of our experimental design, we are not able to empirically prove whether our results are reliable or not. However, based on the feature selection methods applied, we have several assumptions. From our results, we have seen that wrapper methods provide better accuracy than filter methods. This has also been observed in the literature [32, 31, 28, 71]. However, our application of wrapper methods use a heuristic approach, and it is expected that it will not result in the optimal subset of features every time it is applied. Due to the extent we compared several feature selection methods and classifier methods, we were not able to perform feature selection multiple times to calculate β. Similarly, we cannot calculate α. However, we can assume that our approach for feature selection is not reliable. We expect that, by performing new experiments, we would observe a low reliability score because of the low consistency in the search method. If we were to ensure a reliable feature selection process, we could apply GeFS proposed by Nguyen [32]. This is left for further research.

Research interviews have been performed with security experts on topics in-

formation sharing, TI, and data fusion. The summaries of these interviews are presented in *Appendix B - Interview subject 1*, *Appendix C - Interview subject 2 and 3*, *Appendix D - Interview subject 4*, *Appendix E - Interview subject 5*, *Appendix F - Interview subject 6*, and *Appendix G - Interview subject 7*. Our key findings in regards to what is of most value for information sharing is presented in *Table 19 - Key findings: Valuable elements for information sharing*. Since the Cfs method pro-

Uniform Resource Identifier (URI)
IPs
Domains
Detection rules
Hashes
Malware samples
Methods
Tools
Procedures

Table 19: Key findings: Valuable elements for information sharing

duced best results on average, we will use those features when comparing the selected features and *Table 19 - Key findings: Valuable elements for information sharing*. From this, we observe some overlap. However, there are also elements which analysts define as important in the decision-making that is not selected by the feature selection process. Below, we will discuss each of the findings, and whether they can be included into the current ML process.

URI Such indicators can often be used for detection of activities like Exploit Kit (EK) landing pages and callback. For an analyst, comparing two URIs for determining whether the activity is an EK landing page is often easy. However, this is unfortunately tough for ML classifiers without extracting features from the URI. Hence in our current experiment, URIs should provide little value. However, the feature `normalizedURL` was selected by Cfs on the binary dataset. This indicates that there was a high correlation between the URIs and classes. From interview process and experience, we assume that attribute can be of even more value if correct features are extracted.

IP This indicator is often used for reputation purposes, and is a commonly shared indicator according to interview process. Observing a specific IP can indicate malware callback. Intuitively, the value of an IP feature should contribute little. However, Cfs on malicious dataset selected the `destination.networkAddress.address` feature which is the destination IP. From this, we can deduct that certain IPs were observed several times as either malicious or benign, and trends were observed.

Domain Similar to IPs, this indicator can also be used for reputation purposes, and is also a commonly shared indicator according to the interview process. Features related to domains were not selected by Cfs in our experiment. However, domain names have previously been proved to contribute to detection of malware not only on reputation [102]. Extracted features like the number of numerical characters, length or Longest Meaningful Substring (LMS) can be used in ML.

Detection rules Static and dynamic behavioural signatures like signatures for Snort, Suricata or Yara[1] are predefined detection methods. Sharing of such signatures helps analysts avoid the time-consuming process where deep domain knowledge is often necessary. A related feature was selected in our experiments, namely `attackInfo.attackIdentifier`.

Hashes File hashes can be used for whitelisting or blacklisting of samples as it creates a unique id for each sample. For automated detection and response, such measures are simple but effective for low fruit malware. However, according to security trend reports [103, 27] threat actors often modify samples to create new unobserved hashes for each attack; therefore, hash is not as reliable as before. Such a feature is of little use in automated classification using ML methods. Our data-driven approach did not select features related to file hashes either.

Malware samples According to feedback from interviews, sharing of samples is rather common. Participants appeared to be willing to share samples, and saw great value in receiving such information. Unfortunately, this is not something which can be directly used in ML methods. Features must be extracted either statistically, dynamically, or both.

Methods, tools, and procedures Participants agreed on technical indicators providing some value in the detection of malicious activity; however, there was also much interest in receiving more refined intelligence like methods, tools, and procedures of specific actors. Understanding these elements allows for potential attribution, and also the prediction of future attacks towards similar sector or targets. These types of features were not in our dataset, as such information are collected from other sources.

From the discussion above, we see that few of the elements security experts consider relevant is selected by the ML methods. However, there are also some specific elements which were selected by the ML methods that were not mentioned by the security experts. One of the most central elements were those related to the customer and the location. Understanding the industry, sector, and country

[1] https://github.com/plusvic/yara

of residence of the target can provide much information on the threat actor. On the other hand, understanding the country of the threat actor is beneficial in attribution.

The general trend when comparing experimental results and interview results is that there are only a few common elements. Also, in the case of overlap, there are several cases where current implementation uses the features differently than security analysts. Generally, combining the results from the feature selection method and the research interviews requires several feature extraction processes applied on attributes before it can be used in ML classifiers.

6.4 Summary

In summary, we have in this chapter presented our experiments. Environments were discussed, and specific switches for software commands was given. Further, the results from our experimental process were presented. It was shown that the Cfs method performed best on average. We also showed that the Ibk classifier performed best on average.

Finally, we performed a comparison between experimental results and interview process results. We observed that the security experts considered several of the features selected by the data-driven feature selection. However, there are also several elements discussed by the experts that were currently not in our dataset. Further work on feature extraction is necessary to combine the findings from our data-driven experiments and the research interviews.

7 A Model for Data Fusion, Reduction, and Sharing in Financial Sector

In the previous chapters, introduction and relevant theory have been presented as well as state-of-the-art related to the research questions. Our methodology has been shown, and results on research regarding one of the research questions have been presented. In the following chapter, the results of the research related to the second research question are presented. Requirements for a process model is identified, and a process model for data fusion and sharing is proposed.

7.1 Requirements

Based on the literature study and the interview process, requirements for a data fusion, reduction, and sharing process model is identified. By identifying the advantages of previously proposed fusion process models, we seek to design a process model decreasing or removing identified flaws. Further, by identifying how industry performs fusion and sharing, combined with the current flaws in these approaches, we seek to design a process model based on both academia and industry. The following requirements for a process model have been identified:

Cyclic Ensuring that the model clearly describes a cyclic process is important. The fusion process should be a continuous cycle to ensure optimal situational awareness.

Detailed definitions According to Bedworth and O'Brien [46], a process model should provide a sub-division of the problem which is rich and detailed enough to allow reuse of specific knowledge. By breaking the problem into sub-problems, and those into smaller sub-problems, we can create a set of problems which are easily solvable and implementable.

Automation With the ever increasing amount of potential sensors and log sources, the amount and diversity of available data is increasing drastically. To ensure situational awareness, it is of interest to acquire as much relevant data as possible to facilitate a correct analysis. Human analysts can only do so much, and including automation for increasing efficiency as well as providing decision-support is imperative. Automation in terms of sharing and inclusion of data allows for an efficient system which are continuously up-to-date with the existing threat environment. Automation in terms of analysis and decision-support allows for more efficient and accurate decision-

making, and can be done by introducing ML and pattern recognition to the analysis phase.

Sharing Sharing of TI to trusted external parties is important in the current fight against cyber criminals. According to Gartner [8], 60% of digital business infrastructure will rely on TI to ensure operational resilience by 2019. The sharing process should be a two-way flow which allows for the inclusion of new TI into the fusion process. The standardisation of sharing is necessary to allow for automation.

Feedback As in most of the earlier proposed fusion models, an explicitly defined feedback process must be included. A feedback flow should be at all levels to ensure findings are used continuously to increase the quality of the fusion process.

Concurrent processes The fusion processes should be concurrent. By having concurrent fusion processes, we can enable independent and parallel operation, which are critical in complex systems computing large amounts of data.

Intelligence-driven The model should include the acquisition, consumption, analysis, and distribution of intelligence.

TI fusion When including TI from trusted external parties, the quality of the TI may vary. There may be overlap in the provided data, and fusion of TI from various sources should be performed. The content and format of TI also vary depending on the level of TI. Therefore, the fusion of TI is essential to increase situational awareness.

Centralised management With requirements for a cyclic process as well as a feedback process, centralised management is preferred for managing this. Centralised management is necessary with the increasing amount of sensors and log sources.

Distributed fusion With the increasing amount of sensors and log sources we are approaching Big Data. More specifically, the velocity, volume, and variety of data are increasing. Centralised storage and fusion demands costly resources in data storage and computational power, and so fusion process should be performed distributed. This is especially important when designing for scalability.

7.2 Proposed Model

The proposed model is shown in *Figure 21 - Proposed process model*. The proposed model is an attempt to adhere to the previously defined requirements,

and is a step towards full automation of data fusion and information sharing in the financial sector. Based on the popularity and advantages of the JDL Fusion Model, we decided to apply this model as foundation for our proposed process model. More specifically, the separation of levels of abstraction and the five levels of fusion are used, while we propose new processes enabling automation of fusion and information sharing.

The rest of the section will describe each of the components and functions performed in detail.

7.2.1 S1-S3 - Sensors

IDSs which monitor and alerts on suspicious or malicious activities, and other log sources. These components are of heterogeneous nature and output is, therefore, different depending on component type.

7.2.2 T1-T3 - Threat Intelligence

TI from internal and external sources. The nature of these sources can range from technical feeds to more tactical intelligence.

7.2.3 Data Refinement - Sensors (L0)

This fusion process calibrates and filters raw data. Preprocessing methods are applied, where bias correction and other data cleaning activities are performed if necessary.

7.2.4 Data Refinement - Threat Intelligence (L0)

Similarly to *data refinement - sensors (L0)*, this process calibrates and filters the raw data collected from TI sources.

7.2.5 Object Refinement - Sensors (L1)

Measures from various sensors are correlated to a common frame of reference. Different correlation methods can be applied, like association process selecting observations with common elements. This process is governed by the *process refinement (L4)* which also enrich observations based on previously observed situational and predictive intelligence. Data have currently been refined to information by creating a context and index.

7.2.6 Object Refinement - Threat Intelligence (L1)

Similarly to *object refinement - sensors (L1)*, correlation is performed combining observations with common elements. In terms of TI, this may be elements like amongst other threat actor, country, sector, industry, vulnerability, and attack technique. A distinct difference between sensor data and TI is that is may already be of a higher level of abstraction. If the incoming data is on a strategic or tactical level, it can be directly applied to the predictive analytics database. Data have currently been refined to information by creating a context and index.

7.2.7 Object database

This is a database or collection of normalised observations. Each element in this database represents an event which have been observed by one or more sensors. Where intelligence is available, each element have been enriched with e.g IP or domain reputation and geolocation for source and destination.

7.2.8 Situation Refinement (L2)

This fusion process seeks to create a situational awareness. The process would in current operation be a combination of automation and human interaction. The events from the object database is aggregated to create a better understanding of the situation, while human analysts perform decision making based on this situation. ML methods can be used either in cooperation with the analyst, or directly replacing the analyst. The output from the process is situational data which either can be acted on, and or stored for further analysis.

7.2.9 Threat Refinement (L3)

The fusion process performed here seeks to create data for future predictions. In terms of cyber crime, this could be to predict trends in targets for a specific threat actor, or predict how a specific threat actor will attack a specific target. This process is also currently a combination of automation and human intuition. By applying data-driven methods like ML and data mining, the vast amounts of data can be used to create predictions previously not possible by manual work only. The predictions are based on data from the situational database and the object database. The output from the process is prediction data which either can be acted on and or stored for further analysis.

7.2.10 Situational Database

This is a database of situational data where some decision making have been applied. The content is data which represents the current situation based on the observations from sensor networks and TI sources.

7.2.11 Predictive Analytics Database

This is a database of predictive analytics data based on the refinement performed in *object refinement - threat refinement* process and from strategic and tactical TI. The content is predictions based on the observations from sensor networks and TI sources, and is ready to be acted on either automatically or manually.

7.2.12 Information Sharing

This process governs the sharing of information from the data fusion process. Generally, it can operate in two different ways: it can continuously export feeds with specific TI, e.g IPs related to a specific botnet; it can export TI based on requests, e.g a TI partner can request TI on a specific IP or threat actor. The process should also handle problems related to anonymisation to the extend that it defines what attributes or elements are sensitive. The process of anonymising

is out of the scope for this model, and is left for future work on the sharing of TI. It should also handle the classification schemes which are applied in previous fusion processes.

7.2.13 Process Refinement (L4)

The process refinement governs the whole fusion and classification process. This management level ensures that the process is a continuous cycle, and implements feedback to previous processes as new findings are discovered in later processes. By communicating with the *situational database*, it can automatically configure sensors to ensure new intelligence are applied in near real-time. The timeliness of TI is often important, and this management process ensures all components of the fusion process is up-to-date with latest intelligence. By communicating with the *predictive analytics database*, it can automatically apply those configurations which enable preventive approach. Collection of specific elements can be performed based on the predictions in this database, e.g. enabling full capture of network traffic when expecting an attack. Further, a preventive measure can be applied to mitigate the potential threat, e.g. blocking access to and from a specific IP range associated with a threat actor.

This management process also enriches objects fused in the *object refinement - sensors (L1)* process with internal and external TI. The enrichment process can either be based on the TI in the *situational database* and *predictive analytics database*, or on external TI e.g. geolocation of an IP.

7.3 Model Discussions

The proposed model adhere to the requirements identified in 7.1. It enables an automated fusion process with a cyclic nature. Further, it defines how TI and consecutive sharing of TI should be included in this fusion process, which has not been done in previous fusion process models. It also defines centralised management with distributed fusion to enable future scaling of operation. The general process can be described as follows:

Observation

A new security event is observed by sensor S1. The event is preprocessed using L0 and then correlated to a common frame of reference in L1. At this point, it is stored in the object database for further analysis.

Analysis

The event is processed by the L2 to create situational awareness. The analysis can be performed by an analyst, an automated process, or in cooperation. The situational knowledge is stored in the situational database.

Prediction

Based on the event, the situational knowledge, and other related events, L3 can perform predictions. These predictions are stored in the predictive analyt-

ics database.

Sharing
Finally, the situational knowledge and potential predictions can be shared. It is important to note that the information sharing process should be able to receive feedback, which is then handled by the process refinement.

Intelligence gathering
Similarly to how the model shares information, it can receive information from other sources. A new TI object is collected by T1. The object is then preprocessed in L0 where the abstraction level is decided. If the TI is of lower, more technical nature, it is pushed to L1. There, it is correlated and added to the situational database. If on the other hand, the TI is of higher, more strategic and tactical nature, it is pushed directly to the predictive analytics database.

Intelligence processing
As new TI is added to the system, L4 manage the distribution of information to the various levels based on the collected information. This information is either used for enabling detection and prevention capabilities to sensors, or for the enrichment of security events.

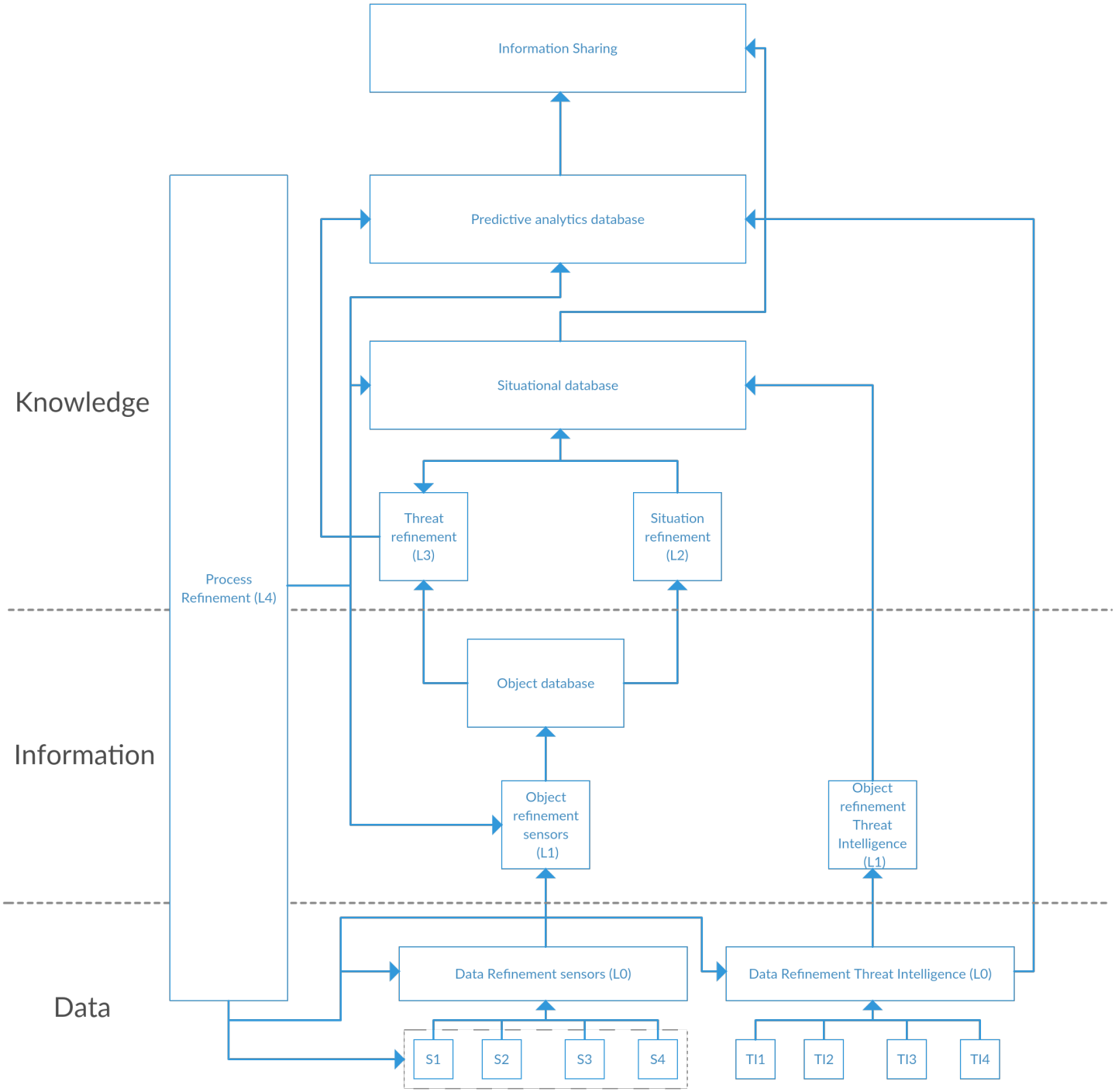

Figure 21: Proposed process model

8 Implications and discussion

In the previous chapters, we have presented theory and state of the art related to our research questions. Methodologies for research interviews and experiments have been presented, and experimental results were discussed. Further, requirements for a process model for fusion was given, and we proposed a fusion process model based on these requirements. This chapter provides discussions of the implications of the thesis, and a summary is given.

8.1 Theoretical implications

In this thesis, we sought to demonstrate how information sharing in security operation can be automated to a higher degree than current solutions. Much research has been performed on the usage of ML methods for classification of security events; however we applied ML methods to real-world data demonstrating that it does, in fact, provide good results. By combining previous work in data fusion and experiences and opinions from security experts, we proposed a data fusion process model enabling automation in information sharing.

> Research question 1: How can data fusion and reduction for intrusion detection at an early stage using various heterogeneous sources be modelled?

To identify the requirements for a data fusion process model, we have performed a literature review of previously proposed data fusion models identifying the advantages and disadvantages of each model. Further, research interviews with security experts were performed identifying common use cases and problems with the current solutions. Based on these findings, we proposed a data fusion process model which focus on automation using data fusion and ML. The use and sharing of TI are central in the proposed model, as interviews and literature study identify TI as critical in security operation.

The proposed data fusion process model is based on a literature study and has, therefore, a strong theoretical foundation combined with the findings from our research interviews. However, based on the research interviews, we observed a difference in how organisations wanted to apply data fusion and information sharing. The proposed model combines all findings and may, therefore, contain processes or elements which are not of interest to all organisations. Also, even though security experts from law enforcement and public and private organisations have been interviewed, we cannot guarantee that our findings is representable for the rest of the security community.

> Research question 2: Which features are reliable and trustworthy in the classification of aggregated and correlated events, and which cannot be shared without anonymization?

To identify the reliable and trustworthy features we applied commonly used ML methods to real world data. Feature selection methods were applied, and feature subsets were then evaluated using classification accuracy of common classifiers. Due to the problem related to the No Free Lunch theorem, 7 different classifiers were applied for evaluation. The features identified in the feature selection process were demonstrated relevant by achieving classification accuracies mostly between 80% and 95%. Based on the research interviews, we identified sensitive features which cannot be shared without anonymization. The general opinion was that features that which can be linked to a specific person or organization are sensitive. To demonstrate how well ML methods can utilise commonly shared data, we compared the feature subsets generated by the data-driven approach and the elements identified in the interview process.

However, due to the No Free Lunch theorem, we cannot argue that the selected features will perform similarly on another dataset. Similarly, we cannot argue that classifiers IbK and RandomForest will perform best on another dataset. Moreover, the threat environment is dynamic and the most relevant features cannot be expected to be static. The applied dataset consist of 60 days of data from real networks, and while our results are representable for this period, the dataset and its trends may be very different from our dataset.

A common problem with IDSs is the large amount of FP. As a result, our dataset is very skewed towards one of the classes, namely 'no incident'. This may cause problems when evaluating the classification performance, as the majority class often represents a large percentage of the class distribution. This challenge was attempted solved by separating the original classification problem into two subproblems where class distribution was slightly better.

The dataset used in our experiments were classified by human analysts before data-driven methods were applied. Therefore, some of the events in our dataset may be wrongly classified and therefore including errors into our models.

8.2 Practical considerations

The recreation of the experimental phase of this thesis is mostly feasible. Software like Logstash, Python, and Pandas which were used for storage and acquisition of features are all available for free. Further, software used for feature selection and classification, Weka, is free for use. We have described commands and command options for our applied tools where necessary.

The main problem with recreating these specific experiments is the availability of the dataset. The dataset applied in this thesis is from real networks and thus contains sensitive information which cannot be shared outside the organisation, however, an overview of all available features is presented in this thesis. Much of the experiments can be recreated using similar datasets. While the results

may not be the same, we argue that this is expected. Because of the data-driven approach, the results are expected to be different depending on the dataset.

Several of the experiments performed in this thesis were very computational complex. Particularly the classification methods require large amounts of RAM and CPU resources. As presented, stratified folds had to be created for the experiments to be feasible on the available equipment.

Similarly, the research interviews can easily be recreated. We have provided our interview guide as well as the summaries of the interviews. We argue that our findings based on the interview guide describe a general trend in the security communities. Outliers are expected towards both ends, however since our interview subjects represent communities from legal enforcement and private and public organisations, we argue that general trends were discovered.

The proposed requirements for a process model is based on literature and research interviews. Recreation of these requirements can easily be done by pursuing literature and the summaries of our research interviews. Similarly, the proposed process model is based on these requirements and research interviews.

8.3 Summary

The goal of this thesis was to enable more automation in the security operation and information sharing. The motivation for this was the rapid increase in security events combined with the continuous increase in the velocity, volume, and variety of data, making automation an essential part of security operations. The number of security threats increases each year, and the use of TI is central for the cooperation between security communities. Problems arise when data are collected from an increasing amount of heterogeneous sensors and log sources, combined with the heterogeneous TI data. Information security has become a field where the timeliness of information and action is critical. More specifically, we have two problems: Large amounts of data in various formats cannot be used for decision support without reduction and fusion because of the complexity; The increase in volume and velocity of threats makes the decision-making process performed by security analysts a daunting task. We cannot expect security analysts to keep up with the increasing amount of events. Because of these problems, we sought to propose data fusion process model for better reduction and fusion of security events and TI. Further, we sought to demonstrate that ML methods can be applied to real-world networks for decision support or decision making.

To achieve this, we investigated literature on data fusion identifying advantages and disadvantages of current models. Further, we performed research interviews to investigate current trends and challenges in automation and information sharing in security communities. Based on our findings, we proposed requirements for a data fusion process model, and also proposed process model based on these requirements.

Further, we created a dataset by collecting aggregated and correlated events from real world networks. Events were classified by human analysts, and thus ready for supervised ML methods. Preprocessing were performed for standardisation, before new datasets were created. The problem of classification was separated into two subproblems to investigate whether different methods performed better on this subproblems. Then, three feature selection methods were applied from Weka; ReliefF, InfoGain, and Cfs. Extensive evaluation of feature subsets was performed using seven common classifiers from Weka; J48, IbK, NaiveBayes, RandomForest, RandomTree, and SVM.

A best classification accuracy of 93.88% on the original problem, and 94.73% and 95.03% on subproblems were provided, and we prove that ML methods can provide a great advantage in decision making and decision support in the classification of IDS events.

9 Conclusion

In this thesis, we have shown that feature selection methods on aggregated IDS events increase the performance of ML classifier methods notably. The dataset applied in this thesis consist of aggregated IDS events from real world networks; thus, we have demonstrated that ML classifier methods yield good results when applied to real-world data. We have identified two subproblems based on the problem of IDS event classification and demonstrated how ML can solve these with acceptable performance. For each subproblem, we identified the best performing feature selection method as well as the best performing classifier method. More specifically, we have identified the Cfs method as best performing feature selection method. Further, we identified IbK and RandomForest as best performing classification methods. We have achieved a classification accuracy of 93.88% on the original problem, and 94.73% and 95.03% on the subproblems. Our results show that the applied ML methods for feature selection and classification perform well both for multinomial classification and binomial classification. Information security experts have been interviewed in research interview process, and we have demonstrated the difference between features selected by data-driven approach and features selected by security experts. Our observations are that while there are some common features, there is a distinct difference between features selected by the data-driven approach and features chosen by security experts.

We have performed a literature review of data fusion process models and proposed requirements for a data fusion process model enabling automation in the security operation and information sharing based on literature and research interview findings. Further, we proposed preliminary data fusion process model based on requirements and research interview findings. The proposed model defines how TI and sharing of TI should be included in the data fusion process, and is, therefore, a contribution towards the automation of information sharing and security operation. To the authors knowledge, no previous fusion process models incorporate TI in the way we have proposed.

Our work is a contribution towards the much-needed automation in IDS event classification and security operation. We have bridged the gap between academia and industry by applying ML methods on real-world security events, and by performing research interviews with security experts from information security community.

10 Further work

Based on our experimental phase, experimental results, and proposed requirements and model for data fusion, we propose several future research areas. We hope that our research motivates future work in these areas.

Separation of classification tasks
When performing multinomial classification, One-vs-One or One-vs-All is generally used. As a result, each class is trained using same classification method. We propose the investigation on whether different classes of security events in security operation can be classified with higher performance by using different classifiers for different classes. Resulting classification can then be calculated using methods like voting or weighted voting.

Class specific features
In our experiments, we assumed that all classes are best identified using the same feature set. However, based on personal experience as an incident handler and our findings in research interview process, we observe that human analysts use different features for decision support, depending on what class they are considering. Therefore, we propose to investigate feature contribution per class. We recommend applying data-driven approach combined with research interviews or questionaries of security analysts.

Optimising method parameters
For our experiments, we applied default parameters for both feature selection methods and classification methods in Weka. We suspect that the tweaking of parameters can provide better classification results. We propose to investigate whether other parameters provide better classification results.

Non-heuristic search methods
In our experiments, we applied heuristic search methods. This was the default search method by Weka, and we chose to apply this due to the computational complexity of using non-heuristic search methods. We propose to apply non-heuristic search methods were applicable. Especially in the feature selection process, non-heuristic search methods should be used. We suggest the application of the GeFS [32] on aggregated IDS events.

Trend-based classification
The dataset applied in our experiments consist of 60 days worth of IDS events. As discussed, the results of our research may not be applicable for a new dataset

in the future. We propose to investigate whether there is, in fact, a distinct difference over time. Based on these findings, we also propose to investigate whether some features are better for classification based on trends.

Feedback-based improvements of data fusion process model

Our proposed process model for data fusion is based on previous work in literature combined with experience and challenges from industry. We propose to investigate further improvements to this model. More specifically, the model can be improved by creating more detailed and technical specifications of each process. Further, suggestions for data flow and a format is needed.

Bibliography

[1] Julisch, K. & Dacier, M. 2002. Mining intrusion detection alarms for actionable knowledge. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 366–375. ACM.

[2] mnemonic AS. 2014. Security report 2015. http://www.mnemonic.no/Global/PDF/mnemonic_security%20report_2015.pdf. [ONLINE] Accessed November 22. 2015.

[3] Europol. 2015. The internet organised crime threat assessment (iocta). https://www.europol.europa.eu/sites/default/files/publications/europol_iocta_web_2015.pdf. [ONLINE] Accessed December 2. 2015.

[4] Arnfinn Strand, C. P. 2015. Check point mobile threat prevention. http://www.mnemonic.no/Global/Presentasjoner/2015-10-20-MTP.pdf. [ONLINE] Accessed December 2. 2015.

[5] Micro, T. 2015. The invisible becomes visible - trend micro security predictions for 2015 and beyond. https://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt-the-invisible-becomes-visible.pdf. [ONLINE] Accessed December 2. 2015.

[6] Websense. 2015. Websense - 2015 threat report. https://www.websense.com/assets/reports/report-2015-threat-report-en.pdf. [ONLINE] Accessed November 25. 2015.

[7] Vormetric Data Security. 2015. 2015 vormetric insider threat report. http://enterprise-encryption.vormetric.com/rs/vormetric/images/CW_GlobalReport_2015_Insider_threat_Vormetric_Single_Pages_010915.pdf. [ONLINE] Accessed December 3. 2015.

[8] Contu, R. & McMillan, R. 2014. Competitive landscape: Threat intelligence services, worldwide, 2015. http://www.gartner.com/technology/reprints.do?id=1-23HXD07&ct=141023&st=sb%29#h-d2e258. [ONLINE] Accessed December 10. 2015.

[9] Bace, R. G. 2000. *Intrusion detection*. Sams Publishing.

[10] Porras, P. A. & Valdes, A. 1998. Live traffic analysis of tcp. In *IP Gateways,"To appear in Internet Society's Networks and Distributed Systems Security Symposium*.

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

[12] Axelsson, S. 1999. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, 1–7. ACM.

[13] Tesink, S., MIM, L. R., & Leune, C. 2005. Improving csirt communication through standardized and secured information exchange.

[14] Skierka, I., Morgus, R., Hohmann, M., & Maurer, T. 2015. *CSIRT Basics for Policy-Makers: The History, Types & Culture of Computer Security Incident Response Teams*.

[15] Carnegie Mellon University. Csirt services. `http://www.cert.org/incident-management/services.cfm`. [ONLINE] Accessed April 18. 2016.

[16] Carnegie Mellon University. National csirts. `https://www.cert.org/incident-management/national-csirts/`, note = [ONLINE] Accessed April 18. 2016.

[17] ThreatConnect. 2015. Threat intelligence platforms - everything you've ever wanted to know but didn't know to ask. `http://cdn2.hubspot.net/hubfs/454298/ebook/Threat-Intel-Platform-ebook-ThreatConnect.pdf?__hssc=258496277.1.1448652718422&__hstc=258496277.e6254306715b10ee0605a48445fb4be3.1447195565490.1448646587314.1448652718422.53&hsCtaTracking=ab06d884-140d-4263-ad46-cb4ee60f805a%7Ca3cc5a32-4894-4c6a-a671-f189c757708e`. [ONLINE] Accessed February 6. 2016.

[18] Caltagirone, S., Pendergast, A., & Betz, C. The diamond model of intrusion analysis. Technical report, DTIC Document, 2013.

[19] US Department of Defense. 2013. J-p 2.0 joint intelligence. `http://www.dtic.mil/doctrine/new_pubs/jp2_0.pdf`. [ONLINE] Accessed Mars 13. 2016.

[20] Chismon, D. & Ruks, M. 2015. Threat intelligence: Collecting, analysing, evaluating. `https://www.cpni.gov.uk/documents/publications/`

2015/23-march-2015-mwr_threat_intelligence_whitepaper-2015.pdf?epslanguage=en-gb. [ONLINE] Accessed February 4. 2016.

[21] Hutchins, E., Cloppert, M., & Amin, R. 2010. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Proc. 6th Int'l Conf. Information Warfare and Security (ICIW 11)*, Academic Conferences Ltd., 113–125. [ONLINE] Accessed December 5. 2015. URL: http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf.

[22] U.S. Department of Defense. April 2007. Joint publication 3-60 joint targeting. [ONLINE] Accessed April 18. 2016. URL: http://www.bits.de/NRANEU/others/jp-doctrine/jp3_60(07).pdf.

[23] Computer Incident Response Center Luxembourg. Information sharing and cyber security - the benefits of the malware information sharing platform (misp). https://www.circl.lu/assets/files/infosharing.pdf. [ONLINE] Accessed Mars 5. 2016.

[24] MITRE. About cybox. http://cyboxproject.github.io/about/. [ONLINE] Accessed May 3. 2016.

[25] MITRE. 2012. Standardizing cyber threat intelligence information with the structured threat information expression (stix™). https://www.mitre.org/sites/default/files/publications/stix.pdf. [ONLINE] Accessed January 12. 2016.

[26] Homeland Security System Engineering and Development Institute. 2014. Threat-intelligence-sharing-using-stix-and-taxii. http://secure360.org/wp-content/uploads/2014/05/Threat-Intelligence-Sharing-using-STIX-and-TAXII.pdf. [ONLINE] Accessed April 15. 2016.

[27] Symantec. 2016. Internet security threat report - volume 21, april 2016. https://www.symantec.com/security-center/threat-report. [ONLINE] Accessed April 20. 2016.

[28] Kononenko, I. & Kukar, M. 2007. *Machine learning and data mining: introduction to principles and algorithms*. Horwood Publishing.

[29] Jain, A. K., Duin, R. P., & Mao, J. 2000. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1), 4–37.

[30] Raschka, S. 2015. *Python Machine Learning*. PACKT Publishing.

[31] Guyon, I. & Elisseeff, A. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3, 1157–1182.

[32] Nguyen, H. T., Franke, K., & Petrović, S. 2012. Reliability in a feature-selection process for intrusion detection. In *Reliable Knowledge Discovery*, 203–218. Springer.

[33] Aly, M. 2005. Survey on multiclass classification methods. *Neural Netw*, 1–9.

[34] Allwein, E. L., Schapire, R. E., & Singer, Y. 2001. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1, 113–141.

[35] Hsu, C.-W. & Lin, C.-J. 2002. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2), 415–425.

[36] Duda, R. O., Hart, P. E., & Stork, D. G. 2012. *Pattern classification*. John Wiley & Sons.

[37] Oommen, T., Misra, D., Twarakavi, N. K., Prakash, A., Sahoo, B., & Bandopadhyay, S. 2008. An objective analysis of support vector machine based classification for remote sensing. *Mathematical geosciences*, 40(4), 409–424.

[38] Boström, H., Andler, S. F., Brohede, M., Johansson, R., Karlsson, A., Van Laere, J., Niklasson, L., Nilsson, M., Persson, A., & Ziemke, T. 2007. On the definition of information fusion as a field of research.

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

[40] White, F. 1987. Joint directors of laboratories-technical panel for c3i, data fusion sub-panel. *San Diego: Naval Ocean Systems Center*.

[41] Durrant-Whyte, H. F. 1986. Integration, coordination and control of multi-sensor robot systems (sensors, robotics).

[42] Llinas, J. 1988. Toward the utilization of certain elements of ai technology for multi sensor data fusion. *Application of artificial intelligence to command and control systems, Peter Peregrinus Ltd*.

[43] McKendall, R. & Mintz, M. 1988. Robust fusion of location information. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, 1239–1244. IEEE.

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

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

[46] 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.

[47] White, F. E. et al. 1988. A model for data fusion. In *Proc. 1st National Symposium on Sensor Fusion*, volume 2, 149–158.

[48] Steinberg, A. N., Bowman, C. L., & White, F. E. 1999. Revisions to the jdl data fusion model. In *AeroSense'99*, 430–441. International Society for Optics and Photonics.

[49] Llinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E., & White, F. Revisiting the jdl data fusion model ii. Technical report, DTIC Document, 2004.

[50] Waltz, E. L. 1998. *Information warfare principles and operations*. Artech House, Inc.

[51] Boyd, J. 1987. A discourse on winning and losing (report no. mu43947). air university library, maxwell afb. *AL. An unpublished briefing*.

[52] Osinga, F. A discourse on winning and losing. `http://www.au.af.mil/au/awc/awcgate/boyd/osinga_boydconf07_copyright2007.pdf`. [ONLINE] Accessed December 7. 2015.

[53] 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*.

[54] Elmenreich, W. 2002. Sensor fusion in time-triggered systems.

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

[56] Von Neumann, J. 1956. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata studies*, 34, 43–98.

[57] Parhami, B. 1996. A taxonomy of voting schemes for data fusion and dependable computation. *Reliability Engineering & System Safety*, 52(2), 139–151.

[58] PARHAMI, B. 1994. Threshold voting is fundamentally simpler than plurality voting. *International Journal of Reliability, Quality and Safety Engineering*, 1(01), 95–102.

[59] Yager, R. R. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1), 183–190.

[60] Fullér, R. 1996. Owa operators in decision making. *Exploring the limits of support systems, TUCS General Publications*, 3, 85–104.

[61] Zadeh, L. A. 1984. A computational theory of dispositions. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, 312–318. Association for Computational Linguistics.

[62] Cho, S.-B. & Kim, J. H. 1995. Combining multiple neural networks by fuzzy integral for robust classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 25(2), 380–384.

[63] Shalaginov, A. & Franke, K. 2014. Automatic rule-mining for malware detection employing neuro-fuzzy approach. *Norsk informasjonssikkerhetskonferanse (NISK)*, 2013.

[64] Zadeh, L. A. 1983. The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy sets and systems*, 11(1), 197–198.

[65] Franke, K. *The influence of physical and biomechanical processes on the ink trace. Methodological foundations for the forensic analysis of signatures*. PhD thesis, University of Groningen, 2005.

[66] Kruegel, C., Robertson, W., & Vigna, G. 2004. Using alert verification to identify successful intrusion attempts. *Praxis der Informationsverarbeitung und Kommunikation*, 27(4), 219–227.

[67] Julisch, K. 2003. Clustering intrusion detection alarms to support root cause analysis. *ACM transactions on information and system security (TISSEC)*, 6(4), 443–471.

[68] Valdes, A. & Skinner, K. 2001. Probabilistic alert correlation. In *Recent advances in intrusion detection*, 54–68. Springer.

[69] Ning, P., Reeves, D., & Cui, Y. 2001. Correlating alerts using prerequisites of intrusions.

[70] Nguyen, T. H., Luo, J., & Njogu, H. W. 2014. An efficient approach to reduce alerts generated by multiple ids products. *International Journal of Network Management*, 24(3), 153–180.

[71] Langley, P. et al. 1994. *Selection of relevant features in machine learning*. Defense Technical Information Center.

[72] Schlimmer, J. C. et al. 1993. Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. In *ICML*, 284–290. Citeseer.

[73] Almuallim, H. & Dietterich, T. G. 1991. Learning with many irrelevant features. In *AAAI*, volume 91, 547–552. Citeseer.

[74] Kira, K. & Rendell, L. A. 1992. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, 249–256.

[75] Hall, M., Holmes, G., et al. 2003. Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 15(6), 1437–1447.

[76] UCI. 2015. Uci machine learning repository. `https://archive.ics.uci.edu/ml/datasets.html`, note=[ONLINE] Accessed January 23. 2016.

[77] Raıssi, C., Brissaud, J., Dray, G., Poncelet, P., Roche, M., & Teisseire, M. 2007. Web analyzing traffic challenge: description and results. In *Proceedings of the ECML/PKDD*, 47–52.

[78] Peng, H., Long, F., & Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8), 1226–1238.

[79] Berg, P. E., Franke, K., & Nguyen, H. T. 2012. Generic feature selection measure for botnet malware detection. In *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, 711–717. IEEE.

[80] Samarati, P. & Sweeney, L. 1998. Generalizing data to provide anonymity when disclosing information. In *PODS*, volume 98, 188.

[81] Burke, M.-J. & Kayem, A. V. 2014. K-anonymity for privacy preserving crime data publishing in resource constrained environments. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, 833–840. IEEE.

[82] Ghinita, G., Karras, P., Kalnis, P., & Mamoulis, N. 2009. A framework for efficient data anonymization under privacy and accuracy constraints. *ACM Transactions on Database Systems (TODS)*, 34(2), 9.

[83] Sqrrl Data Inc. Sqrrl architecture. `https://sqrrl.com/product/architecture/`. [ONLINE] Accessed April 15. 2016.

[84] Sqrrl Data Inc. Sqrrl datasheet. `https://sqrrl.com/media/Overview-Datasheet.pdf`. [ONLINE] Accessed April 15. 2016.

[85] Sqrrl Data Inc. What is a threat hunting platform? `http://info.sqrrl.com/framework-for-hunting-wp-download-0`. [ONLINE] Accessed April 16. 2016.

[86] Digital Shadows. 2015. Cyber situational awareness - gain an 'attacker's eye view' of your organisation.

[87] NSW Government. Nsw ict strategy, priorities: Information sharing. `www.finance.nsw.gov.au/ict/priorities/managing-information-better-services/information-sharing`. [ONLINE] Accessed December 15. 2015.

[88] NSW Government. Nsw ict strategy, priorities: Information management framework. `https://www.finance.nsw.gov.au/ict/priorities/managing-information-better-services/information-management-framework`. [ONLINE] Accessed December 15. 2015.

[89] McNamara, C. General guidelines for conducting research interviews. `http://managementhelp.org/businessresearch/interviews.htm`. [ONLINE] Accessed February 1. 2016.

[90] Ringdal, E. A conceptual framework for sharing of threat intelligence. unpublished thesis, 2016.

[91] Kononenko, I. 1994. Estimating attributes: Analysis and extensions of relief. In *European Conference on Machine Learning*, Bergadano, F. & Raedt, L. D., eds, 171–182. Springer.

[92] Shannon, C. E. 2001. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1), 3–55.

[93] Hall, M. A. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[94] Robnik-Sikonja, M. & Kononenko, I. 1997. An adaptation of relief for attribute estimation in regression. In *Fourteenth International Conference on Machine Learning*, Fisher, D. H., ed, 296–304. Morgan Kaufmann.

[95] Python Software Foundation. Python 3.5.1. `https://www.python.org/downloads/release/python-351/`. [ONLINE] Accessed April 7. 2016.

[96] Foundation, P. S. Pandas 0.7.1. `https://pypi.python.org/pypi/pandas/0.17.1/`. [ONLINE] Accessed April 7. 2016.

[97] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

[98] PyPA. Pip. `https://pip.pypa.io/en/stable/`. [ONLINE] Accessed April 7. 2016.

[99] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1), 10–18.

[100] Elasticsearch BV. Logstash. `https://www.elastic.co/products/logstash`. [ONLINE] Accessed Mars 9. 2016.

[101] Hall, M. A. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[102] Bilge, L., Kirda, E., Kruegel, C., & Balduzzi, M. 2011. Exposure: Finding malicious domains using passive dns analysis. In *NDSS*.

[103] Verizon. 2016. 2016 data breach investigations report. `http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/`. [ONLINE] Accessed April 27. 2016.

Appendices

A Interview Guides

A.1 Information sharing

Information Sharing
1. How is Information Sharing performed in your organisation?
2. Do your organisation produce and share TI?
3. Who are your external sharing partners?
4. What is the size of the sharing communities you are a part of?
5. How do you obtain trust in these communities?
6. Is there a vetting process? Please describe.
7. Do you use any specific methods for establishing trust with your sharing partners?
8. What are your major concerns and challenges with regards to sharing data?
9. What are special considerations with regards to sharing of TI data which have been enriched with internal tools?
10. What kind of information would you be reluctant to share with external parties?
11. What is your impression on how information is shared today in industry?
12. Is the current solutions adequate, or do they need improvement?
13. Do you apply information classification system on TI?
14. If so, what are the requirements for each classification level?
15. What are your opinion on the mindset of "Everyone should share."?

Table 20: Interview guide: Information Sharing

A.2 Threat Intelligence

Threat Intelligence
1. In Threat Intelligence, what indicators of compromise is most relevant and valuable?
2. In the current threat environment, what types of attacks do you see as most prominent and damaging?
3. Please explain where and how you obtain TI.
4. Roughly how many TI feeds do your organisation subscribe to?
5. What kind of TI, guidance, and security related information do you share?
6. Please describe the process of selecting TI feeds and sources.
7. Have you observed difference between publicly available free feeds as opposed to commercial feeds?
8. Do you enrich TI with internal tools? Please describe what types of information is added.
9. Would you say your organisation is automation-centric or analyst-centric when it comes to TI?
10. Currently, many organisations are analyst-centric. As scaling must be done due to increasing amount and diversity of sensors and sources, how do you believe organisations can move towards automation-centric operation?

Table 21: Interview guide: Threat Intelligence

A.3 Data Fusion

Data Fusion
1. Can you briefly describe the architecture of your data fusion system?
2. Roughly, how many heterogeneous sensors / data sources do you have in your organisation?
3. What is the quality of the input data from your sources?
4. In the data fusion process, do you keep raw data?
5. Briefly describe the techniques and methods used in fusion of data.
6. When designing data fusion process, did you adhere to existing models on data fusion?
7. When designing a process model for fusion, automated decision support, and sharing, what do you consider to be the most important requirements?
8. When performing fusion, do you remove elements with privacy concerns early in the process?
9. Is there data or elements which you do not fuse?

Table 22: Interview guide: Data Fusion

B Interview subject 1

Sharing is currently divided into four different categories: at my place of work we differentiate between what we share to the public, within the information security community, within other security experts in closed forums, and what is shared through formalized partnerships. When sharing information with the public we use different measures such as: our blog and our newsletters. Analysts also share advisories to friends and families over social media. Within the information security community most communication is performed over instant messaging services such as internet relay chat (IRC). Within the closed forums information is usually obtained and shared through moderated email lists. Formalized partnerships entails that the sharing obligation is bidirectional, and in these partnerships other types of assets may be shared, such as competence and not only IOC and threat intelligence.

We initially want to share as much as we can to organizations which can act on the received information, however some information is too sensitive to share. Among sensitive information we find information which may identify the victim or single individuals, one could view this as customer data or membership data. This includes information such as: names of victim in e.g. configuration files, employees in the victim organization, information regarding systems connected to the victim.

We participate in several sharing forums and communities, where the size of the group varies. Some are rather large, while others are small e.g. 15-20 members. While some contain 100s of people, who have been vetted. Although I think it is a strong correlation between the size of the group and the degree of trust within these communities. The larger community groups tend to share less information, and several members are present only to consume. Many of these communities require that you know someone who are already a member, usually 1-2 members from different organizations need to recommend you prior to obtaining membership. Trust is thus more or less obtained since the recommenders risk sanctions by approving the wrong people. Some communities require clearance or background checks, although this is usually limited to the formalized partnerships.

When sharing with mutual partners it is often required to meet face to face, mulltiple times. The degree of trust is hard to measure, although meeting your partners multiple times helps to establish some degree of trust. A good motivator for sharing is a mutual benefit with regards to parties both receiving quality data. It isn't always that we expect to receive information back from the community,

sometimes there is a wish to share and enlighten the community, and to raise the security level. This may also be in the form of tools which are made available to the public, and not necessarily information.

Most of the data that we receive through different sharing communities can't be conceived as threat intelligence, it is mostly raw data in the form of indicators of compromise, this data needs to be enriched with internal tools and further analysis, before it can be called intelligence. This goes for the information seen in intrusion detection systems as well. Enrichment may be performed through contextual information such as domain reputation, behavioural traits, and so on. When a sufficient amount of contextualisation is performed we may be able to tie an attack or campaign against a specific threat actor, which makes it a lot easier to aggregate on the intelligence and perform the right mitigation.

When we share with customers there are rarely any specific requirements to protect the information which has been enriched with internal tools, unless the attack is suspected of being targeted.

There may be a need to restrain sharing, or anonymize if the threat intelligence contains customer data or otherwise targeted information. And sometimes we have a wish to conceal our capabilities and methods, this is usually only true for targeted operations where the attacker is advanced, e.g. nation sponsored or other forms of APT groups. This may differ from incident to incident.

Information observed in IDS environment is considered owned by the organization which has the sensor equipment installed in their network. Although information observed from these sensors which may be regarded as public information is unproblematic to share. E.g. a binary retrieved from a specific location. Information from customer network may not always be shared, some customers are eager to share, others are more reluctant. If information is shared it is usually done so under a classification scheme, more specifically the traffic light protocol. It is of course important to differentiate between opportunistic attacks and targeted attacks, where the latter one requires more discretion if IoCs and threat intelligence is to be shared.

Sharing is usually done over IRC, closed email lists, and some dedicated platforms. STIX and TAXII are both gaining ground as mechanisms for expressing and sharing threat intelligence. The issue today is not necessarily the sharing platforms, rather the amount of data. The issue arises when trying to determine which of these data is of interest for my organization, and how can I act on these data in my own environment, and thus use it to increase the overall security level within my organization.

We use classification schemes for data, we have an internal one. As for external communication we mark the data that is to be shared with the appropriate TLP color. TLP is a good protocol for classifying information, however I get the impression that not everyone knows how to use it appropriately. There are different opinions on what TLP yellow and red means, which can create some issues.

More specifically, people tend to over classify the information they share, making it hard to act upon, when it is received.

Most of the sharing that is performed today is analytics-based sharing, this means it is shared between analysts and to a less degree automatically. Although we automate some of the sharing.

In order to approach a more automated sharing environment, we need the following: The community needs to improve its analysis methods, and then we need to have flexible and scalable solutions which can receive data in a wide range of forms, over several protocols. These solutions also need to be able to normalize, and correlate the data and present them to the analyst in such a way that the analyst can automate as many steps of his work as possible. E.g. the analyst should not have to remove commas from csv-files. This would enable us to better contextualise and visualize the information we receive.

It isn't easy to define which indicator of compromise has the most value, or is the most prominent one. This will usually vary, and depend on the context. Opportunistic attack such as bank trojans, and exploit kits will have other interesting traits than other forms of attacks. With exploit kits we are most interested in knowing the parameters in the URI-strings, we want to know which ones are changing over time so that we may customize detection rules that can stand the test of time. For banking trojans we are interested in improving our capability to decipher the configuration files, in order to identify targets and to see if it is within our scope. Targeted attacks means that IP addresses, domains, and so on are less important, since the actors are great at staging infrastructure for specific operations. This means that infrastructure known from one campaign is unlikely to be seen in another campaign performed by the same adversary. In the case of targeted attacks it is more interesting to understand their methods, tools and procedures, as well as their behavioural traits within a compromised network.

With regards to what level one should use threat intelligence will vary, it depends on the maturity of the organization. For some organizations it's more interesting to use technical threat intelligence, while others, more mature organizations may use strategical threat intelligence to invest in security in a long term manner.

We use a vast amount of different sources for threat intelligence, in the form of threat intelligence feeds. We don't have any specific criteria for starting to use a new feed, although if they don't give us any increased value, we will stop using them. Some feeds overlap a lot with others, and thus is not very interesting in use. It seems to me that the public feeds have come along way, and in many cases deliver just as good threat intelligence and indicators of compromised as the commercial ones.

When we want to share information which may be viewed in anyway as sensitive, we first contact the information stakeholders and get approval to share. We then share it within the community as TLP:yellow to the ones that have a need

to know. And of course if there are any information we deem as sensitive which the information stakeholder has not identified themselves, we give them advice on how to handle such information.

We anonymize through manually, by removing sensitive information. Some of this is done through redaction, or through scripts. Although a second pair of eyes is always handy when dealing with anonymization.

As for improvement of the anonymization I have nothing to add. Although it is interesting to add that in EU nations, privacy regulations describe IP-addresses as personal information which makes it hard to share with some european countries.

C Interview subject 2 and 3

Currently, a secure channel is hosted. It can be viewed as an information sharing platform linked to every member. The platform allows for exchange of personal data according to legal frameworks. The platform is secure, and only trained staff has access. Everything is logged and audited, and there is full control of who has access and to which information.

The issue of trust is easy when it comes to member states. The staff operating the platform is from law enforcement and have security clearance. They also receive extensive training for use, as well as procedures for handling and classification of information. Currently, there are no concerns with regards to sharing data. Mindset is that "if you don't share, you don't receive".

There are little considerations when sharing TI which have been enriched with internal tools. A intelligence product is created with findings, hypotheses, and recommendations. The enrichment can be based on information from various members.

Information sharing between Law Enforcement and private parties is quite good. It is not structured, and exchange often happens via unstructured channels such as secure e-mail. Often, teams work together in ad-hoc operations. However, precisely due to the ad-hoc nature of such operations, team members tend to change frequently. Despite the fact that there are no general structured way of sharing, there are clear points of contacts on both Law Enforcement and private parties sides. With both private sector and Law Enforcement, trust is keyword.

Classification and handling codes is used for TI also between LE and Private Parties. To improve the current methods of sharing, a change of legal framework is needed. The legal basis doesn't allow for work in bidirectional way with private parties, and collaboration is difficult. Change is however happening, which will allow for sharing bidirectional with private companies. Currently, formal procedures via other law enforcement is needed.

The most relevant and valuable IOCs are IP, domains, file signatures, hash, and malware samples. These are most commonly used. The most prominent and damaging attack at the moment is ransomware and financial malware in general. Financial malware can cover many types like trojan, botnet, etc. Underreporting is a problem, and many overviews of prominent attacks/threats may only be of what is reported. IPs and generally everything that can be linked to a physical person is personal data. With a strict legal framework, all personal data have sensitivity concerns.

D Interview subject 4

As of today we share both internally and externally, the sharing is facilitated by technologies such as IRC and emails. The threat intelligence we produce is only shared internally, and is primarily in the form of indicators of compromised observed in our own sensor network. We are currently participants in two different smaller sharing communities, both categorized as analytic-driven communities. As for how the vetting process is performed in these communities, I am not sure. I reckon that this is performed prior to invitation, and the sharing community we participate in within our own sector is mostly open, since everyone knows everyone. We plan to expand our function in the future, one of the plans is to utilize Splunk in order to correlate data, and enrich them, although this process is still immature. We currently do not utilize any information sharing platform, although this is something we will work towards in the future. When we share or receive information we mark it according to the traffic light protocol, although among other parties we don't classify information. TLP seems to me as too general, I wish there was more specific guidelines for usage, and more concrete guidelines on issues like sharing information further internally.

We primarily collect NetFLOW data, although some of us are looking on more contextual information. We use heuristics to expand on the contextual value of the observed issues, and thus aim to react more sufficiently to incidents. The most serious attacks we see towards our organization is compromised and stolen equipment, due to the fear of proprietary information being stolen. As for threat intelligence feeds, we currently don't use any commercial feeds, although we subscribe to several open source feeds such as ETopen. As for how we choose the different sources we usually base our decisions on the experiences of others, what others are saying about the sources. We also try out the sources for a limited time to see if the yield good and non-overlapping threat intelligence. As for comparing commercial and open-source feeds I can't say anything, since we do not use any commercial sources. As for sensitive elements, we currently see personal identifiable information as the most sensitive, and we do not share this with anyone. Most other types of information are shared, since we try to be as open as possible. Although there may be sensitivity issues with regards to information disclosed and projects with other partners.

We mostly share in an analytic-centric method as of today, and to some degree automatic. I think we need a better infrastructure in order to enable us to share in an more automated manner.

As of today we do not use anonymization when sharing externally, this is due to we share very little with external parties. Although, if we are to start sharing more externally, I would use redaction as an anonymization method.

We do some data fusion, at least in the sense that we correlate data with splunk, making us able to search through correlated data. Apart from this we don't perform any data fusion. In order to prepare for data fusion, we need to centralize more, it is possible that we can solve this in the future. As for how many differents formats we use, it is mostly json and syslog. We have a few different sensor types. We don't use firewall logs, but we use hostbased IDS systems, network IDS, system logs, netflow, and we have defined TI as a sensor. I definitely see the need to fuse data from the different sensors. The quality of the data from the different sensors seem to me as good, and reliable. We retain logs, but most other raw data is deleted. This is due to limited capacity. Enrichment is performed after necessity.

E Interview subject 5

Currently, information sharing is performed using mostly automated lists. When performing automated sharing, web GUIs and other protocols are used. STIX and TAXII have been considered. Sharing in a 1:1 relationship is easy, but more difficult when more sharing partners. A format must be agreed upon. We are also performing manual exchange of information which could have been more automated. Generally, the information shared manually is on a higher level than current automated lists. Modus operandi is shared manually. Threat intelligence is produced in-house and some of it is shared.

The size and type of sharing communities we are part of varies. Generally, smaller and closed communities are of higher quality than large, open communities. External sharing partners range from CERTs, government organisations, and private organisations.

The trust in such communities are often built upon previous contacts. New members are introduced by old members, and a chain of trust is built. Another approach used is background check. Establishing trust often takes time.

A major concern regarding sharing data is that it depends on who own the data. Is it from customer networks, or internal networks? Norwegian Personal Data Act must also be followed. Deciding whether the data must be sanitised or anonymised must also be done, as well as classify using TLP-classification.

Threat intelligence is enriched using reputation lists, where IPs, URLs, and domains are grouped. E.g. crimeware cc, crimeware download, etc. Some of these internal sources is not shared, and some are shared only manually in specific cases. Other are shared automatically.

We are using TLP-classification internally. The classification method works well, however there are problems regarding over classification. TLP:Red provides little use, as we cannot apply information. TLP:Amber cause problems regarding subcontractors. It is important to ensure that recipient understand the classification. Written contracts are often used.

Threat Intelligence is used as a large part of operation. It allows us to detect relevant threats. Most Threat Intelligence we use is on a technical level. Higher levels of information have huge value, however they are also more uncertain. Operational Threat Intelligence is used to detect threat actor across sector.

The most damaging threat is APT with large amount of resources. Ransomware also cause much damage. CEO fraud is also economical threat. Often such attacks cause large losses (10 million - 100 million NOK).

We are collection Threat Intelligence from hundreds of feeds and sources. Generally we collect everything we can. The quality of sources if assessed, and is

weighted according to reliability. If there are much false positives it can be tuned down. Various sources have different focus and often complement each other. Some can focus on scan and reconnaissance, other on crimeware, or even on a specific malware source. Generally, the more specific sources are more reliable.

The organisation is currently very analyst-centric where new threats are detected by analyst and shared analyst to analyst in sharing community. Research is ongoing on how to automate sharing of Threat Intelligence, as well as automating the data fusion to achieve higher level of intelligence.

If there are business sensitivity concerns we mostly redact data. Anonymisation with current techniques is currently difficult. Hash algorithms is not valid due to few combinations (e.g. social security number). A solution is to create random value and map these to values. However this is very complex. Another solution is to not share data sources which contain such information. Even with anonymisation, meta data can leak. Current approaches is legal where Non-disclosure Agreements and other contracts are signed. Legal approach is better than data washing. We provide public data sources, however they does not contain any sensitive data at all.

F Interview subject 6

Information sharing is performed as a service. An important factor is what we can share and what not to share. Generally, no customer data should be shared, and only metadata which cannot be identified to a customer can be shared. The sharing performed by us can be separated into three levels. The first level is automated sharing of intelligence. The second is on-request sharing where intelligence regarding specific cases or topics is shared on-request. The third level is personal sharing. This type of sharing often provides the most value, but is difficult to automate. Information is exchanged with key personnel in organisations, often based on a more informal/private relationship (friends etc). It is difficult to achieve an overview of such personnel networks. The format of such exchange are chat, sharing of samples and e-mail.

There are two issues in regards to sharing of information: trust and format. Current situation is that someone experienced must read and interpret Threat Intelligence when received. As people have different backgrounds, the evaluation is not consistent and uncertainties may arise.

Sharing is performed on various levels. Some sharing agreements based on personal contacts, and which have been more formalised later on. Generally, sharing performed here can be separated into: Sharing with Legal Enforcement like the police, Norwegian Defence, Kripos, Europol, etc. These types of sharing activities are often only sharing out from organisation to Legal Enforcement, and not the other way around; Information security communities like CERTs and other private organisations; FIRST where incident response related topics are discussed. In such a community, much can be shared as members have to be accepted to join. This solve the issue of trust. The sharing of information is done via channels like meetings and IRC. The information that is shared is pretty technical, however due to experience analysts can “read between the lines”.

Trust is often established by knowing people. Another approach is to have central actors like NorCERT creating sharing platform. Communities where membership is necessary is also good (like FIRST). Sharing communities created and operated by private actors is often a problem. Written agreements must be signed.

We are using TLP internally. Current implementation does not work. There should be a problem between TLP:Red and TLP:Amber. It is important to create a separation between organisation and group. E.g we want to share information with IRT in organisation X, but not the whole organisation. Custom TLP classifications is possible. Another problem is that the classification is based on trust. Also problem when receivers do not understand TLP.

The most damaging threat for our organisation is threats related to information leakage. Confidentiality of customer data and the availability of services provided is central. Loss of confidentiality is worst, as it is easier to detect loss of availability. Slow, targeted attacks are a problem.

Threat Intelligence is used for detection, however experience and new information is collected for each incident and valuable knowledge and skills are built over time. By experience, the smaller the sharing community, the easier trust is achieved. Important to note that smallest achievable community is 1:1.

A problem with automated sharing of Threat Intelligence is that experience based decisions based on intelligence is not possible. Further, bringing such automated Threat Intelligence to C-level is often difficult. A possible solution to this is to apply machine learning. A standardisation of how certain observations are is needed.

When performing early fusion of data, there's a problem regarding loss of information. Can we tell what type of information we removed/lost? It is known what information is extracted, but not what is lost when extracting only these elements.

G Interview subject 7

For internal sharing of information we primarily use Sharepoint and Lync. Two instances of sharepoint are running, one open for everyone internally, and one confidential. In some cases special tools are used for heightened security. There are no standardised way of sharing when sharing with external parties. E-mail are mostly used. Also some ad-hoc solutions.

We produce and distribute a newsletter internally on current trends. This is on a tactical level. We also sell similar newsletter as a product. Sharing communities includes international sector focused forums and communities with CERTs. Sector focused forums are mostly tactical intelligence. CERT communities focus on operational intelligence.

Establishing trust is done by meeting in person. Trust is built over time. Strict policy on what and how sharing is performed helps increasing trust in communities. When new members join community, previous relationships often basis. Start with sharing of agreed upon data, with more trust, more sharing is performed and more details are shared.

Unstructured channels are used for sharing externally, however applied methods cover current needs. First step towards more automated sharing would be standardisation. If sharing methods is defined, automation is easier. Defining classification levels for sharing is also important step.

Currently, we use three levels of classification. Open, which are open for everyone. Internal, which are for internal employees and subcontractors with NDA. Confidential, which are only for a predefined group. Currently, same classification is used when sharing externally. NDA is often used.

For us, strategic, tactical, and operational TI is important. IOCs are valuable for detection in monitoring systems, however there's also a need to understand behaviour of threat actor and potential trends.

The most damaging attacks towards our organisation is fraud and abuse of service. Targeted attacks for accessing information or monitoring is also serious. Denial of Service is also critical.

We use various sources, both commercial and open. Try to collect data from different types of sources to ensure wide awareness.

When sharing information, adhering to Norwegian Personal Data Act is important. Often, data fields must be anonymised. This is mostly done by removing fields manually. We do not have an automatic anonymisation method. Currently too many problems. Anonymisation is a problem in cases like APT investigations, where valuable information can be lost if anonymised. Currently, we do not have clear policy for anonymisation when sharing detection data across countries.

Sharing of information is very important to ensure organisations are best prepared for cyber incidents. However, there are often legal problems. To be able to automate sharing, a high level of professionalism is needed. Further, high level of standardisation is necessary. Consistency is also important in what to monitor and what to share, instead of ad-hoc solutions.

H Code

H.1 convert_features_to_csv.py

```
#!/usr/bin/env python2.7
# Converts raw list of features to csv format for data acquisition

IN_FEATURES = "features.txt"
OUT_FEATURES = "features.csv"

tmp = []
with open(IN_FEATURES, "r") as infile:
   for line in infile:
        tmp.append(line.rstrip().replace(" ", "_"))

with open(OUT_FEATURES, "w") as outfile:
    for element in tmp:
        outfile.write(element + ",")
```

H.2 convert_clean.py

```
#!/usr/bin/env python
# coding=utf 8
import pandas as pd
import numpy as np
import sys
from sklearn.preprocessing import LabelEncoder
import json

# Script for initial preprocessing of events. Events are loaded from
# json files, transformed to Pandas Dataframe, cleansed, and stored
# as csv files for further computation using Weka

FEATURE_FILE="X.json"
CLASS_FILE="y.json"
CSV_FILE='dataset_soc_senior_ack_20d.csv'

```

```
17 def read_data(FEATURE_FILE, CLASS_FILE):
18     """ Read data from json files using Pandas.
19     FEATURE_FILE: json file with feature values
20     CLASS_FILE: json file with class values
21 
22     return: Pandas Dataframe with feature and class value
23     """
24     with open(FEATURE_FILE, "r") as f:
25         data = json.load(f)
26     df_X = pd.DataFrame(data)
27 
28     with open(CLASS_FILE, "r") as f:
29         data = json.load(f)
30     df_y = pd.DataFrame(data)
31     df_y.columns = ['class']
32 
33     df_data = pd.concat([df_X, df_y], axis=1)
34     return df_data
35 
36 
37 def write_data(df, CSV_FILE):
38     """ Write data to csv file using Pandas.
39     df: Pandas Dataframe with feature and class value
40     CSV_FILE: Output file
41 
42     return: 0
43     """
44     df.to_csv(CSV_FILE, sep=',', encoding='utf 8', index=False)
45     return 0
46 
47 def clean(df):
48     """ Removes unwanted features and characters. Redundant features
49         related to class is removed. Characters which are known to
50         cause problems with Weka is replaced.
51         df: Pandas Dataframe with feature and class value
52 
53         return: Pandas Dataframe
54     """
55     unwanted_features = ['associatedCaseCategoryID',
56                          'associatedCaseCategoryName',
```

```
                          'associatedCaseID']
    df = df.rename(columns=lambda x: x.replace('\n', '  '))

    for feature in unwanted_features:
        df = df.drop(feature, 1)
    for i, col in enumerate(df.columns):
        try:
            df.iloc[:, i] = \
                df.iloc[:, i].str.replace(',|"|%|\*|\+|\'', ';')
        except AttributeError:
            pass
    return df

def run():
    df = read_data(FEATURE_FILE, CLASS_FILE)
    df = clean(df)
    write_data(df, CSV_DATASET)
    return 0


if __name__ == "__main__":
    run()
```

I Features

properties.:suspicion
properties.:pam_tcp_synflood_window
comments.timestampDate
properties.:TunnelSource
properties.ad_browse__time
properties.ICMPType
properties.estreamer_malware_userName
properties.requestAction
properties.ad_to
properties.:tcp_connections_dropped
properties.:clientEncStoC
properties.ad_server__inbound__interface
properties.nitroPolicy_Name
properties.dayOfWeek
properties.ad_Comment
properties.:tcp_events_attack
properties.:pam_javascript_fre_replaces
properties.ad_app__rule__id
properties.ad_voip__method
properties.ad_DCE-RPC_
Interface_
UUID
detailedEventIDS.customerID
properties.:IssuerOrg
properties.ad_serverGroup
properties.deviceSeverity
properties.estreamer_HTTP_URI
properties.:ipv6_packets
properties.ad_UserCheck__Confirmation__Level
properties.ad_Industry_
Reference
properties.categorySignificance
properties.ad_Attributes:Logon_
Hours
properties.:name
properties.nitroDevice_IP

properties.ad_Additional_
In0c2UQw_~_~ntication_
Type
properties.:pam_tcp_synflood_update
properties.sntdom
properties.:queue_full
properties.:supercedes-event
properties.websense_security_category
properties.:stats_interval
properties.:splits
destination.geoLocation.geoJSON
properties.bytes
properties.minTS:
destination.geoLocation.locationName
properties.:intruder
comments.user.parentIDs
properties.:Domain
properties.requestContext
source.geoLocation.countryCode
properties.ad_app__id
properties.nitroTrust
properties.nitroUUID
properties.ad_Attributes:Account_
Expires
destination.networkAddress.ipv6
properties.:tcp_fin_total
properties.submissionLink_v2
properties.ad_inzone
customerInfo.name
properties.nitroURL
properties.:Host-Extra
associatedCaseCategoryID
properties.:refFrames
properties.categoryBehavior
properties.ad_line
properties.:PacketsPerInterval
properties.ad_rawEvent
properties.:evasions
properties.:objStmNest
properties.argus_filter_comment
properties.externalID
properties.destinations_blocked_count

properties.:tcp_connections_timeouts_data
properties.dntdom
properties.levenshtein_distance
properties.IssueID
properties.:MTU
properties.ad_client__outbound__interface
properties.fname
properties.:Window
properties.ad_reject__category
source.networkAddress.multicast
customerID
properties.:pam_quicktime_set_nal_unit_limit
properties.:tcp_checksum_errors
properties.:additionalRRs
properties.:HighWaterMark
properties.:pam_dns_tunnel_detection_rate
source.geoLocation.locationID
properties.ad_file__size
properties.estreamer_malware_eventDescription
properties.:service
properties.estreamerEventID
eventID
properties.:clsid
properties.ad_frequency
properties.:givenPropName
properties.ad_user__status
properties.:moniker_len
properties.argus_transformed_by_filterid
properties.:HSlen
properties.ad_proxy__src__ip
properties.cs5Label
properties.ad_data
properties.ad_WindowsParserFamily
properties.nitroSignature_Name
properties.:compMethod
properties.ad_from
properties.:replaces
properties.:~sch
properties.ad_s-supplier-country
properties.:questions
properties.args
properties.ad_guid__t

properties.:actual
properties.:func
properties.ad_Protection_
name
properties.ad_Signature_
Info
properties.:channel
properties.ad_configuredName
properties.Info
properties.ad_src__user__name
properties.ad_email__session__id
properties.:victim
domain.currentPart
properties._cefVer
properties.ad_dropped__outgoing
properties.destination_ports
properties.:zip_filename_len
properties.:chaff_count
properties.:status
attackInfo.attackCategoryID
properties.:Range
properties.:tcp_rst_total
properties.:len_queue_max
properties.ad_mem_
utilization_
percent
properties.ad_Destination__Interface
properties.:encrypted
location.id
properties.ad_client__outbound__packets
properties.:useragent
properties.bandwidth
properties.ad_app__desc
properties.duid
properties.ad_cs-uri-extension
properties.filenName
properties.ad_subscription__stat__desc
properties.ad_vendor__list
_source
properties.nitroInterface
properties.snort_signature_release
properties.destinationDnsDomain

properties.ad_server__inbound__packets
properties.ad_file__sha1
properties.cs4Label
properties.:stgty
properties.ad_Summary
properties.nitroVPN_Feature_Name
properties.ad_Failure_
Information:Sub_
Status
properties.ad_service__id
properties.ad_TimeBatched
properties.ICMPCode
properties.period_maxTS:
properties.sproc
properties.period_minTS:
properties.:icmp_xsum_errs
properties.ad_email__control
properties.QuarantineEndTime
properties.deviceCustomDate1
properties.deviceCustomDate2
properties.:icmp_packets
properties.ad_Update_
Status
properties.ad_encryption_
fail_
reason:
properties.deviceInboundInterface
properties.ad_Certificate_
I0tc7xg_~_~te_
Issuer_
Name
properties.deviceCustomDate1Label
source.networkAddress.address
properties.:server
properties.:sac
properties.:pdfNest
properties.:mem_max
properties.reputationComment
properties.ad_Key[20]
properties.ad_origin__sic__name
properties.ad___id
properties.:tcp_segments_dropped

```
properties.ad_Email_
Subject
source.geoLocation.longitude
properties.:command
properties.ad_ObjectCanonical
properties.ad_information
properties.Namespace
properties.ad_time-taken
properties.:Archiver
detailedEventIDS.type
properties.ad_TE__verdict__determined__by
properties.ad_requestContext
properties.nitroFile_Path
destination.networkAddress.host
properties.FusionVulnStatus
properties.:oidMatchedDSA
properties.nitroQuery_Response
properties.ad_authenticationResult
properties.ad_src__country
properties.:ConnectTo
properties.:CertsCount
properties.:flushed_age
properties.source_ports
properties.ad_dst_
phone_
number
properties.sid
properties.Code
_id
properties.ad_Streaming_
Engine
properties.maxTS:
properties.ipsid
properties.ad_segment__time
properties.catdt
properties.source_ips
properties.ad_apcnt2
properties.ad_snid
properties.ad_apcnt1
properties.ad_db__ver
properties.:tcp_synacks
properties.ad_resource__shortage
```

properties.sourceTranslatedAddress
properties.destinationTranslatedZoneURI
properties.:malformedMsgs
properties.:mem_current
properties.:options
properties.:Host
properties.:boolean
properties.:LowWaterMark
properties.group
properties.ad_ip__offset
properties.protocol
properties.:Chunk-Size
properties.EPS
properties.:victimip
properties.PeriodCount:
properties.event_role_comment
location.locationID
properties.:unescapes
properties.:out_audits
properties.ad_Additional_
InagWMqg_~_~ncryption_
Type
domain.fqdn
properties.:info
properties.name
properties.nitroSource_Zone
properties.event_blocked
properties.ad_action__details
properties.:object
properties.argus_request_modified
properties.url
properties.ad_protection__id
properties.:frames
properties.:obj
properties.:flushed_size
properties.ad_Detailed_
AuthbXaM4Q_~_~ion:Key_
Length
properties.Rule_Order
comments.user.group
properties.httpStatusCode
properties.destinationServiceName

attackInfo.auditCategories.id
properties.argus_assessed_by_userid
properties.nitroAppID
properties.:threshold
properties.dmac
properties.ad_app__sig__id
properties.reputationSource
properties.:ipv4_xsum_errs
properties.filePath
properties.:pam_dns_tunnel_idle_timeout
properties.argus_source_aggr_bits
properties.ad_Attributes:User_
Account_
Control
properties.eventTime
properties.argus_aggregation_key
properties.levenshtein
properties.ad_Audit_
Policy_
7ICX7w_~_~bcategory_
GUID
properties.ad_UserCheck__incident__uid
source.port
properties.ad_Attributes:SID_
History
properties.ad_dst__machine__name
properties.ad_CollectionHost
properties.:pam_dns_tunnel_detection_total
attackInfo.alarmID
properties.Sensor_Severity
normalizedURL
properties.argus_intruder_aggr_bits
properties.target-ip-addr-end
properties.:hsLen
properties.:component
properties.:pam_msrpc_lsass_limit
properties.:no_pending
properties.deviceFacility
properties.ad_Attributes:Old_
UAC_
Value
properties.estreamer_malware_parentFileName

properties.:totalMessages
properties.:duration
properties.:pam_injection_sql_boolean_triggers
properties.ad_IKE:
timestamp
properties.:clientHostKey
properties.:sid
properties.:closeness
properties.:result
properties.:proxyport
comments.user.imageURL
properties.:clientEncCtoS
properties.:tagType
properties.ad_reject__id
comments.user.flags
properties.ad_PanOSPacketsSent
destination.networkAddress.multicast
properties.:pam_injection_argument_token_limit
properties.ad_rcode
properties.:sawServerKexInit
properties.BaselineCount:
properties.instanceName
properties.ad_ActionID_l
properties.:pam_javascript_rue_unescapes
properties.dtz
properties.at
comments.userID
properties.nitroCategory
properties.av
properties.:text
properties.ahost
properties.ad_p__dport
properties.ad_Changed_
AttriQGj68Q_~_~History_
Length
properties.ad_Failure_
Information:Status
properties.ad_malware__family
properties.deviceProcessName
properties.:length
properties.ad_Code
properties.ad_email__recipients__num

```
comments.user.userName
properties.:pam_tcp_synflood_dstport
properties.:score
properties.:header_name
properties.:docname
properties.submissionLink
properties.ad_appi__name
properties.ad_scheme:
properties.clientIP
properties.nitroResponse_Code
properties.suid
source.geoLocation.geoJSON
properties.:tcp_segments
attackInfo.alarmDescription
properties.nitroEvent_Class
properties.ad_eventlogindex
properties.ad_SslTls_i
properties.originalAttackIdentifier
severity
properties.ad_ResultID_l
properties.locality
associatedCaseCategoryName
properties.ad_app__properties
properties.:httpsvr
comments.user.timezone.id
properties.cefSignatureID
properties.ad_version
properties.ad_Changed_
Attributes:SAM_
Account_
Name
properties.dvc
properties.:classes
properties.ad_malware__action
properties.argus_destination_aggr_bits
properties.:Seconds
properties.:fusion-intruder-ip-addr
properties.deviceTranslatedAddress
properties.:Content-Length
properties.:threat_threshold
properties.oldAttackIdentifier
properties.:src
```

properties.:moniker_limit
properties.:method
properties.:xml_URI
properties.ruleMessage
properties.ad_Certificate_
INW93RA_~_~_
Serial_
Number
properties.:SEQnum
properties.:tcp_connections_timeouts_abort
priority
properties.fsize
properties.failed_login_count
properties.categoryOutcome
properties.:tcp_connections_onesided
properties.deviceProduct
properties.:serverUserName
properties.iprlicensed
properties.:dataDesc
destination.port
properties.:referer_field
properties.:~eo
flags
properties.:udp_xsum_errs
properties.:protected
properties.AnalyzedBy
properties.:prefix
properties.:pam_http_request_limit
startTime
properties.sensor_alarm
properties.:chaff
properties.fileId
properties.ad_OSVersion
properties.ad_Group:Security_
ID
properties.:authorityRRs
properties.:fragment
properties.estreamer_malware_parentFileShaHash
properties.:pam_tcp_synflood_limit
properties.:prog
properties.ad_x-bluecoat-application-name
properties.:pam_http_apache_bo_chunksize

properties.ad_SmartDefense_
Profile
properties.:product
properties.start
properties.:fragLen
properties.secondaryAction
properties.ad_scope
properties.:CentralDirLen
properties.ad_x-exception-id
properties.ad_Attributes:Allowed_
To_
Delegate_
To
properties.ad_nsons
properties.:connections_rate
properties.ad_UserCheck
properties.cs1Label
properties.:PATH
properties.ad_received__bytes
properties.:pam_content_zip_uncompressed_min
properties.eps:
protocol
comments.user.timezone.offset
properties.ad_matched__category
source.networkAddress.maskBits
properties.:netbios_server
properties.repeat-count
properties.:pam_javascript_suspicious_hex_string_limit
destination.geoLocation.countryCode
properties.:pam_html_attribute_length_limit
properties.smac
properties.:tagLen
properties.:referer
properties.ad_server__outbound__packets
properties.:accepted
properties.:pam_conficker_p2p_report_interval
associatedCaseID
properties.block
properties.:id
properties.:pam_dns_tunnel_min_data_length
properties.hourOfDay
properties.estreamer_malware_fileSize

properties.:version
properties.ad_Interface
_index
properties.Check_Point_blade
properties.nitroDestination_Zone
properties.ftype
properties.:serial
properties.:numParamsLeft
properties.argus_severity_reduced_by_filterid
properties.c6a3Label
comments.user.timezone.description
properties.user
properties.:reference
properties.ad_TimeOfDay_l
properties.event_role
properties.ad_dropped__total
properties.multiple_uri_strings
properties.:reflection
properties.:tcp_connections
properties.target-ip-addr-start
properties.:flags
properties.argus_assessed_timestamp
properties.:opnum
properties.:tcp_connections_active
detailedEventIDS.timestamp
properties.ad_web__server__type
comments.user.id
destination.networkAddress.maskBits
properties.:storedCRC
properties.ad_reverseDNSHostName
properties.:offset
properties.ad_More_
Sources
properties.:user
properties.:pam_login_maxpass
properties.:difference
properties.ad_limit__requested
properties.ad_NAT__addtnl__rulenum
properties.ad_authProfile
properties.ad_web__client__type
properties.nitroObject_Type
properties.baseline_events_per_hour

properties.:threat
properties.:segment_offset
properties.deviceVendor
properties.ad_Attributes:Password_
Last_
Set
properties.:password
properties.:expected
properties.ad_packet__capture__time
properties.:tcp_hynacks
properties.:hsSeq
properties.:cumLen
properties.ad_WindowsKeyMapFamily
properties.:in_attacks
properties.ad_ESSID
properties.:tcp_synack_total
attackCategoryName
properties.:SeqNumExpected
properties.:login
destination.geoLocation.locationID
properties.ad_Attributes:SAM_
Account_
Name
detailedEventIDS.eventID
properties.ad_dropped__incoming
properties.dhost
properties.ad_precise__error
properties.snort_classification
properties.:LatestQuery
properties.:reason
properties.ad_New_
Account:Security_
ID
properties.:RawDataLength
properties.ad_AD__UserPrincipalName
properties.ad_fileName
properties.ad_src_
phone_
number
properties.ad_Changed_
Attributes:SID_
History

properties.ad_Account_
For_
WiJfqpA_~_~ed:Security_
ID
properties.:pam_injection_sql_chaff_limit
properties.ad_TimeReceived
properties.ad_SiteName
properties.ad_destinationTranslatedIPv6Address
properties.:MessageCount
properties.cfp1Label
properties.c6a2Label
properties.flexString2
properties.ad_packet__capture__name
properties.flexString1
properties.ad_ip__len
properties.ldapSourceUser
properties.iprsrcstate
properties.:tcp_connections_embryonic
properties.classificationName
source.networkAddress.public
properties.ad_CookieR
properties.ad_CLF__LogGenerationTimeZone
properties.:pam_injection_shell_score
properties.Rule_UUID
properties.:arp_packets
properties.ad_arcSightEventPath
properties.ad_CookieI
properties.:len_queue
properties.ad_WindowsVersion
destination.networkAddress.address
properties.applicationName
properties.:in_events
properties.ad_antivirus-engine
properties.returnCode
properties.dvchost
properties.:pam_tns_tochar_limit
properties.:maskl
properties.:type
properties.user_role
properties.:content
properties.ad_Attributes:Home_
Directory

```
properties.ad_bytesIn
properties.ad_resultIndex
properties.ad_outzone
properties.port_blocked_count
properties.:ValidityTime
properties.:expression
properties.malware
properties.ad_CLF__LogReceivedTimeZone
aggregationKey
count
properties.:msg
properties.shost
properties.ad_stationName
properties.:compressed
properties.:tag
properties.:maskh
properties.event-type
properties.:verdict
endTimestamp
properties.ad_sshVersion
properties.cat
properties.ad_requestUrl
properties.:hsType
properties.cn1Label
properties.:standAlone
properties.ad_EventIndex
source.geoLocation.countryName
properties.:fusion-victim-ip-addr
properties.ad_function
properties.nitroNAT_Details-NAT_Port
properties.:pam_html_max_params
properties.cn2Label
properties.baseline_minTS:
properties.:to
properties.:Version
properties.ad_voip__call__id
properties.:X-Forwarded-For
properties.fileMagic
properties.:movie
properties.:tcp_syn_total
properties.count:
properties.:pam_dns_tunnel_report_interval
```

properties.deviceOutboundInterface
properties.ad_Detailed_
Authsvs3BA_~_~sited_
Services
srcDstGeoDistance
properties.reputationRoles
properties.ad_limit__applied
comments.user.language
properties.:max_attacks/sec
properties.ad_Protection_
Name
properties.ad_Attributes:Home_
Drive
properties.:pam_script_suspicious_score_limit
properties.:statement
properties.ad_packet__capture__unique__id
encodedFlags
properties.iprdststate
properties.ad_CategoryID_l
properties.nitroInterface_Dest
properties.ad_TCP_
flags
properties.:gen
properties.:pam_injection_sql_pedantic
startTimestamp
properties.:dstPort
properties.ad_RequirementID
properties.fileType
properties.nitroPCAP_Name
properties.:host
properties.:origfile
comments.user.name
properties.:cookie
properties.RST_Sent
protocolID
properties.:header_value
properties.:requestLength
properties.estreamer_malware_fileShaHash
properties.severityUpdated
properties.ad_Message__Category
properties.appid
properties.deviceDirection

```
id
source.geoLocation.locationName
properties.multiple_domains
properties.:tcp_xsum_errs
properties.ad_Update_
Version
comments.comment
properties.ad_Log_
delay
properties.classificationDescription
properties.ad_message
properties.:tcp_packets
properties.reputationScore
properties.:IntervalCount
properties.:netbios_client
properties.:Certificate_Table
properties.dest_host
properties.estreamer_malware_fileName
properties.ad_rs(Content-Type)
properties.ad_NAT__rulenum
properties.:pam_javascript_activexObfuscate_split_limit
properties.:pam_injection_sql_score
properties.ad_voip__config
source.geoLocation.latitude
properties.sourceDnsDomain
properties.:action
properties.:file
properties.filterComment
comments.user.disabled
properties.:trons_rules_count
properties.ad_file__sha256
properties.ad_client__inbound__packets
properties.cfp1
properties.:ipv4_checksum_errors
properties.ad_Access-group
properties.:clientUserName
properties.VLAN
properties.:pam_pkzip_nesting_limit
properties.:certSLen
properties.ad_subscription__stat
properties.ad_Detailed_
AuthN28meg_~_~me_
```

(NTLM_
only)
properties.ad_Impersonal_
Level
_type
uri
properties.deviceExternalId
properties.act
properties.attack_identifiers
properties.:~ws
properties.:tcp_hyndups
properties.ad_app__risk
properties.ad_Performance_
Impact
properties.ad_Errors
properties.sslmethod
properties.nitroNormID
comments.timestamp
properties.ad_interval
properties.end
properties.ad_Total_
logs
properties.flexString2Label
properties.ad_DetectionSource_i
properties.AdapterID
properties.:index
properties.ad_stcnt2
properties.ad_stcnt1
properties.cs2Label
properties.:archivedFile
properties.ad_Attributes:User_
Principal_
Name
properties.:ipv4_packets
properties.domain
properties.:Rev
properties.nitroDirection
properties.:heartbeatLen
properties.ad_ID
properties.ad_app__category
properties.ad_Attributes:Profile_
Path

properties.ad_PanOSPacketsReceived
properties.:tcp_events_audit
properties.:tcp_rsts
location.shortName
comments.user.realName
properties.:code
properties.ad_packets
properties.ad_analyzed__on
attackInfo.auditCategories.value
properties.English_Description
properties.cn3
properties.:propName
properties.ad_sourceTranslatedIPv6Address
properties.:pam_tns_username_limit
properties.ad_Source
properties.ad_fragments__dropped
properties.:len_pending
properties.ad_ip__id
properties.:computedCRC
properties.ad_capture__uuid
attackInfo.auditCategories.key
properties.ad_TimeZoneOffset_l
properties.ad_Attributes:Script_
Path
properties.ad_Severity
properties.ad_special__properties
properties.:pam_tcp_outside_window_max
properties.categoryObject
properties.ad_src__machine__name
properties.:remainder
comments.user.customerID
properties.destinationTranslatedAddress
properties.ad_resource__probing
properties.:EventType
properties.:LUSER
properties.:max_bits/sec
properties.:expLen
properties.signatur_id
properties.ad_table
properties.:pam_html_hex_text_limit
properties.ad_contract__name
properties.ad_Packet_

```
info
properties.eventCount
properties.ad_VLF__VirusLogType_l
properties.agt
properties.:pam_injection_shell_pedantic
properties.ad_portal__message
properties.:out_events
destination.geoLocation.latitude
properties.:~spot
properties.ad_Certificate_
Id5IXSg_~_~ate_
Thumbprint
attackInfo.attackCategoryName
properties.cnt
acknowledgedMode
properties.ad_content__type
properties.:token
properties.:tcp_client_ack_total
source.networkAddress.ipv6
properties.categoryDeviceGroup
properties.:count
properties.ad_DCE-RPC_
Interface_
UUID-2
properties.ad_DCE-RPC_
Interface_
UUID-3
properties.:Section
properties.ad_DCE-RPC_
Interface_
UUID-1
properties.aid
properties.:recordLen
properties.deviceCustomDate2Label
properties.ad_Domain:Domain_
ID
properties.:digits
properties.:~crc
properties.categoryTechnique
properties.:port
properties.:StartCode
properties.ad_has__accounting
```

properties.:udp_packets
properties.c6a4Label
properties.ad_email__id
properties.:connections_rate_max
properties.:Set-Cookie
properties.:udp_checksum_errors
properties.ad_sent__bytes
properties.:c
properties.ad_TimeDetected
properties.argus_victim_aggr_bits
properties.:rev
properties.argus_severity_increased_by_filterid
properties.topdomain
properties.:keyword
customerInfo.shortName
properties.ad_vpn__feature__name
properties.:applet
properties.deviceNtDomain
properties.actualAction
properties.ad_Account_
Information:User_
ID
properties.:field
properties.:intruder_syns
properties.:innerFile
properties.ad_count
properties.ad_New_
Logon:Security_
ID
comments.user.mobile
properties.ad_SeverityID_1
properties.flexString1Label
properties.ad_TCP_
packet_
out_
of_
state
properties.workingHours
properties.estreamer_malware_filetype
properties.:host-length
properties.ad_community
properties.ad_tcp__flags

properties.ad_PendingActions_l
properties.requestMethod
properties.:answerRRs
properties.nitroNAT_Details-NAT_Type
properties.ad_peer_
gateway
properties.:className
properties.:extent
properties.:X-Sinkhole
properties.ad_malware__rule__id
properties.outcome
properties.:server-type
properties.ad_session__id
properties.:IssuerCN
properties.argus_event_last_severity
properties.ad_rpc__prog
properties.ad_Service_
Information:Service_
ID
properties.:ACKnum
properties.ad_encryption_
failure:
properties.:client
properties.:icmp_checksum_errors
properties.:pam_js_fromcharcode_multi_radix_required
properties.ad_CollectionEventLog
properties.nitroDevice_Action
properties.ad_x-bluecoat-application-operation
properties.:out_status
properties.:expectedTag
properties.cs2
properties.:hash
properties.:tcp_connections_fullduplex
properties.c6a1Label
properties.ad_s-supplier-failures
properties.sensor_address
properties.:rangeCount
properties.cs4
properties.cs5
properties.:in_status
properties.ad_Parameter
properties.estreamer_malware_detectionName

destination.geoLocation.countryName
properties.app
destination.networkAddress.public
properties.:max_frames/sec
properties.ad_Attributes:User_
Parameters
properties.nitroMethod
source.networkAddress.host
properties.ad_Additional_
Inc1+cZg_~_~sited_
Services
properties.destination_ips
properties.:nickname
properties.:len_pending_max
properties.:TopIntruders
properties.StatusSource
properties.:MDIR
properties.ad_connectionType
properties.:URI
properties.dpid
properties.ad_file__md5
attackInfo.attackIdentifier
properties.:tod
properties.:pam_javascript_unescape_limit
properties.ad_fw__subproduct
properties.ad_dcid
properties.argus_severity_adjusted_by_filterid
properties.ad_client__inbound__interface
properties.start-time
properties.:~len
customerInfo.id
properties.:FTP
properties.userid
properties.period_events_per_second
properties.argus_exploit_filter
properties.ad_during__sec
properties.argus_created_by_filterid
properties.:comment
properties.ad_returncode
properties.:dstIP
properties.:c-size
properties.:issueId

properties.:URL
properties.dproc
properties.baseline_events_per_second
properties.:out_attacks
properties.Message
properties.:date
properties.:from
properties.:data
properties.HostOSName
properties.ad_voip__log__type
properties.ad_Attributes:Display_
Name
properties.ad_SL__LogType_l
type
properties.ad_Attributes:New_
UAC_
Value
location.name
properties.:tcp_connections_flushed
properties.ad_methods:
properties.ad_cpu_
utilization_
percent
properties.:RUSER
properties.:string
properties.cefName
properties.ad_Confidence_
Level
properties.:pam_javascript_fromcharcode_limit
properties.nitroUniqueId
properties.pps
properties.c6a4
properties.flexNumber1
properties.count
properties.nitroThreat_Name
properties.ad_Attributes:Primary_
Group_
ID
properties.fileHash
properties.:prior
properties.:TotalCount
properties.cs6Label

properties.:argType
properties.:len
domain.domainParts
properties.:pam_script_unescape_eval_limit
properties.estreamer_malware_filePath
properties.:in_audits
properties.atz
properties.originator
comments.user.email
properties.:proxy
properties.ad_Additional_
InMbHgGQ_~_~Ticket_
Options
properties.ad_Suppressed_
logs
properties.:context
properties.dpriv
properties.ad_next__update__desc
properties.riskName
properties.:server_protocol
properties.c6a2
lastUpdatedTimestamp
properties.:ident
properties.:remLen
properties.:no_update
properties.argus_usermap
destination.geoLocation.longitude
properties.:tcp_syns
properties.estreamer_HTTP_Hostname
properties.cn3Label
properties.ad_Log_
ID
properties.:crc
properties.type
properties.ad_server__outbound__interface
properties.ad_UserID_l
properties.ad_s-supplier-ip
properties.lastUpdateTime
properties.:classtype
properties.:tcp_hyns
properties.:~flag
properties.ad_log__id

properties.:protocol
properties.:max_tcp_connections/sec
properties.:oidMatchedECC
properties.:max_audits/sec
properties.ad_Attributes:User_
Workstations
properties.:pdfObj
properties.:encoded_count
properties.nitroDevice_Port
properties.:cipherNumber
properties.nitroNAT_Details-NAT_Address
properties.:size
properties.ad_cs-threat-risk
properties.:RawData
properties.Protocol_Name
properties.exploitkit
properties.:value
properties.:ID
properties.ad_Protection_
Type
properties.Factor_above_baseline:
properties.cs3Label
properties.:pam_zip_executable_encrypted
properties.:line
properties.baseline_maxTS:
properties.:octets
properties.argus_domain_whitelist
properties.:depth
properties.nitroCommandID
properties.ad_ts_l
properties.:tcp_connections_timeouts_synfin
properties.ad_dst__user__name
properties.period_events_per_hour
properties.:sublength
properties.ad_verdict
properties.ad_emulated__on
properties.ad_pos
properties.:tcp_segments_gaps
properties.ad_ProcessId_i
properties.:value2
properties.:mem_limit
properties.:arg

```
properties.:TunnelDest
properties.:pos
properties.:cve
properties.ad_Group__SiteID_c
properties.period.maxTS:
properties.ad.PanOSPacketsSent
properties.ad.PanOSPacketsReceived
properties.:Connection
properties.scale
properties.:vp3VersionNo
properties.:atom_size
properties.javaVersion
properties.:Header
properties.:pam.script.suspicious.score.limit
properties.ad_sourceDnsDomain
properties.spid
properties.:pam.js_fromcharcode.multi_radix.required
properties.ad.p__dport
properties.ad.browse__time
properties.ad_MessageID
properties.ad.precise__error
properties.ad.Interface
properties.:intruder.syns
properties.ActionTaken
properties.argus.severity.adjusted.by.filterid
properties.ad.methods:
properties.:tcp_syn.total
properties.ad_MessageType
properties.ad.resource__shortage
properties.:fieldLength
properties.:rout
properties.ad_AverageRate
properties.ad.version
properties.estreamer.malware.fileSize
properties.ad_msgid
properties.:fieldType
properties.ad.TimeZoneOffset.l
properties.:no.pending
properties.argus.usermap
properties.ad_Object
properties.ad_connection__uid
properties.argus.filter.comment
```

```
detailedEventIDS.writable
properties.ad_scan__hosts__week
properties.Check_Point.blade
properties.ad_IKE_
IDs:
properties.spriv
properties.ad_webID_c
properties.:arp.packets
properties.:zip.filename.len
properties.:tcp_synack.total
properties.v
properties.:tcp.syns
properties.u
properties.:mem.limit
properties.a
properties.ad.encryption_
fail_
reason:
properties.:tcp.events.audit
properties.application/x-www-form-urlencoded;charset
properties.:yJpg
properties.:pam.javascript.suspicious_hex_string.limit
properties.ad_Site__ID_c
properties.argus.victim.aggr.bits
properties.:icmp.packets
properties.ad_Web__Title
comments.user.languageID
properties.ad.time-taken
properties.:out.audits
properties.ad.Certificate_
INW93RA_~_~_
Serial_
Number
properties.:userid
endTime
properties.:pam.tcp.synflood.limit
properties.:pam.quicktime.set.nal.unit.limit
properties.ad.interval
properties.:len.pending.max
properties.multiple.domains
properties.ad.TE__verdict__determined__by
properties.:ctlWord
```

properties.sourceServiceName
properties.:pam.script.unescape.eval.limit
properties.argus.transformed.by.filterid
properties.ad_unique__detected__hour
properties.event.blocked
properties.ad_STATEMENT
properties.:pam.tcp.synflood.dstport
properties.argus.severity.increased.by.filterid
properties.:xml.URI
properties.:pam.http.request.limit
properties.ad_Role__ID_1
properties.multiple.uri.strings
properties.user.role
properties.:funclen
detailedEventIDS.loggerID
detailedEventIDS.aggregated
properties.ad_vlanTag
lastUpdated
properties.:max_tcp.connections/sec
properties.:pam.injection.argument.token.limit
properties.ad.WindowsVersion
properties.ad.app__id
properties.:pam.content.zip.uncompressed.min
properties.:flushed.size
properties.ad.mplsTag
properties.:ipv4.xsum_errs
properties.:tcp_fin.total
properties.ad.NAT__rulenum
properties.:pam.tcp.outside.window.max
properties.ad_voip__duration
properties.ad.x-bluecoat-application-operation
properties.port.blocked.count
properties.source.ports
properties.ad_Counts
properties.:pam.javascript.rue.unescapes
properties.ad.matched__category
properties.:ipv6.packets
properties.:tcp_rst.total
properties.ad_mplsRD
properties.:flushed.age
detailedEventIDS.deviceID
properties.:~flavor

properties.ad.cs-uri-extension
properties.:pam.injection.sql.pedantic
properties.:share
properties.:pam.http.apache.bo.chunksize
properties.:pam.pkzip.nesting.limit
properties.ad.resource__probing
properties.:connections_rate.max
properties.destination.ports
properties.ad.snid
properties.:height
properties.period.events.per.hour
properties.ad.SmartDefense_
Profile
properties.argus.intruder.aggr.bits
properties.:pam.tcp.synflood.update
properties.ad.scope
properties.event.role
properties.failed.login.count
properties.ad_MaxAverageRate
properties.ad.appi__name
properties.ad.Access-group
properties.:no.update
properties.destination.ips
properties.:version_id
properties.attack.identifiers
properties.ad_bandwidth
properties.:pam.tcp.synflood.window
properties.ad_userType
properties.:pam.tns.username.limit
properties.ad_unique__detected__week
properties.:Rangecount
properties.estreamer.malware.filePath
properties.ad.TCP_
packet_
out_
of_
state
properties.ad_webDescription
properties.ad.has__accounting
properties.ad.EventIndex
properties.:pam_script_percentn_limit
properties.:queue.full

properties.:ver
properties.baseline.minTS:
properties.ad.ip__id
properties.argus.destination.aggr.bits
properties.ad_mplsTag
properties.:TLSlength
properties.:cookieLen
properties.ad.from
properties.ad.service__id
properties.:pam.html.hex.text.limit
properties.ad.MessageID
properties.ad_severity
properties.ad.Signature_
Info
properties.:pam.dns.tunnel.report.interval
properties.ad.Total_
logs
properties.argus.assessed.by.userid
properties.:pam.injection.shell.pedantic
properties.ad_codeDescription
properties.ad.requestUrl
properties.:font_length
properties./filformat.asp?wci
properties.:pam.dns.tunnel.detection.total
properties.:stats.interval
properties.:cksum
properties.baseline.events.per.hour
properties.URL
properties.:pam.javascript.fre.replaces
properties.ad_scan__mail
properties.:pam.injection.shell.score
properties.:recordlen
properties.destinations.blocked.count
properties.ad.src__machine__name
properties.ad.New_
Logon:Security_
ID
comments.user.blocked
properties.:pam.injection.sql.chaff.limit
properties.:y
properties.:x
properties.:tcp.xsum_errs

properties.:pam.http.php.mem.hdr.limit
properties.estreamer.HTTP_URI
properties.:xJpg
properties.ad.UserID.l
detailedEventIDS.deviceEventID
properties.ad.Code
properties.:pam.javascript.fromcharcode.limit
properties.argus.source.aggr.bits
properties.ad.dstkeyid
properties.:udp.packets
properties.ad_scan__hosts__day
properties.ad_srckeyid
properties.ad.Suppressed_
logs
properties.ad.Protection_
Type
properties.ad.Message__Category
properties.:certRLen
properties.ad_MaxBurstRate
properties.period.events.per.second
properties.:pam.html.max.params
properties.ad.log__id
properties.ad.Performance_
Impact
properties.ad.bytesIn
properties.c6a3
properties.ad.Group:Security_
ID
properties.ad.sourceTranslatedIPv6Address
properties.:in.status
properties.:len.queue
properties.ad.sshVersion
properties.argus.aggregation.key
properties.:nick
properties.argus.created.by.filterid
properties.argus.assessed.timestamp
properties.ad_BurstRate
properties.ad.reverseDNSHostName
properties.ad.Detailed_
AuthN28meg_~_~me_
(NTLM_
only)

properties.ad_dstkeyid
properties.:version_length
properties.:trons.rules.count
properties.:out.attacks
properties.ad_Webs__ID_c
properties.ad.Attributes:Old_
UAC_
Value
properties.ad.rpc__prog
properties.ad.client__inbound__packets
properties.ad_scan__hosts__hour
properties.:tcp_client_ack.total
properties.ad.eventlogindex
properties.argus.event.last.severity
properties.URL_Categories
properties.:pam.javascript.unescape.limit
properties.ad_unique__detected__day
properties.estreamer.malware.fileShaHash
properties.ad.IKE:
properties.ad_SubmessageToken
properties.ad_signal
properties.:out/audits
properties.:tcp/connections/onesided
properties.ad/time-taken
properties.ad/Protection_
Type
properties.ad/information
properties.:ICMP-Unreachables
properties.ad.message
properties.:in/audits
properties.ad/Performance_
Impact
properties.ad/SmartDefense_
Profile
properties.ad/Impersonal_
Level
properties.ad/server__inbound__packets
properties.ad/dropped__total
properties.period/events/per/second
properties.ad/service__id
properties.ad/TimeBatched
properties.ad/WindowsKeyMapFamily

properties.ad/Detailed_
Authsvs3BA_~_~sited_
Services
properties.ad/app__id
properties.:pam/dns/tunnel/detection/total
properties.ad/Comment
properties.:pam/javascript/fromcharcode/limit
properties.failed/login/count
properties.:server_offered
properties.ad/serverGroup
properties.:len/queue
properties.:ipv6/packets
properties.ad/mplsRD
properties.:CVSROOT
properties.:pam.javascript.activexObfuscate.split.limit
properties.ad/SeverityID/l
properties.ad/ESSID
properties.:udp/xsum_errs
properties.:tcp/connections
properties.ad/TimeOfDay/l
properties.:len/queue/max
properties.ad/sourceTranslatedIPv6Address
properties.destination/ips
properties.ad/protection__id
properties.:subjectLength
properties.ad/PanOSPacketsReceived
properties.:client_sent
properties.:tcp/syns
properties.ad/pos
properties.:tcp/connections/timeouts/synfin
properties.:max_tcp/connections/sec
properties.ad/AD__UserPrincipalName
properties.:tcp/segments/gaps
properties.ad/app__desc
properties.argus/severity/increased/by/filterid
properties.ad/special__properties
properties.ad/capture__uuid
properties.:tcp/events/audit
properties.ad/Failure_
Information:Sub_
Status
properties.ad/cs-uri-extension

```
properties.:icmp/packets
properties.argus_exploit_filter_url
properties.ad/subscription__stat
properties.ad.app__properties
properties.:pam_content_xml_entity_expansion_limit
properties.:tcp/hyns
properties.ad/NAT__addtnl__rulenum
properties.ad/web__client__type
properties.ad/Message__Category
properties.ad/browse__time
properties.ad/app__rule__name
properties.ad/inzone
properties.ad/ID
properties.:tcp/segments/dropped
properties.ad/SslTls/i
properties.ad/src__machine__name
properties.ad/db__ver
properties.period/maxTS:
properties.ad_auth-db-add-entry-async-name
properties.:icmp/xsum_errs
properties.ad/app__sig__id
properties.ad/ResultID/l
properties.ad.ProcessId.i
properties.ad/Signature_
Info
properties.ad/has__accounting
properties.ad/fragments__dropped
properties.:mem/current
properties.ad/arcSightEventPath
properties.ad/Streaming_
Engine
properties.ad/during__sec
properties.argus/domain/whitelist
properties.ad.dstApplication
properties.argus/filter/comment
properties.ad/Log_
delay
properties.:tcp/connections/fullduplex
properties.ad/origin__sic__name
properties.argus/aggregation/key
properties.:udp/packets
properties.ad/x-bluecoat-application-name
```

properties.argus/usermap
properties.ad/limit__applied
properties.:out/events
properties.ad_sourceApplication
properties.user/role
properties.:tcp/hyndups
properties.:Type1
properties.nitroDescription
properties.ad/next__update__desc
properties.argus/event/last/severity
properties.multiple/uri/strings
properties.ad/Severity
properties.ad/dropped__outgoing
properties.ad/PanOSPacketsSent
properties.ad.s-supplier-ip
properties.ad/Update_
Status
properties.ad/resource__probing
properties.:pam.conficker_p2p.report.interval
properties.:pam/html/attribute/length/limit
properties.ad/cs-threat-risk
properties.:pam/dns/tunnel/idle/timeout
properties.ad.dcid
properties.:pam/zip_executable/encrypted
properties.period/events/per/hour
properties.ad/client__outbound__packets
properties.ad_refid
properties.:pam/dns/tunnel/detection/rate
properties.ad.ESSID
properties.ad/Additional_
InagWMqg_~_~ncryption_
Type
properties.:tcp/connections/embryonic
properties.ad/UserID/l
properties.:out/attacks
properties.ad/ObjectCanonical
properties.ad/dst__user__name
properties.ad/CollectionEventLog
properties.period/minTS:
properties.:tcp/connections/dropped
properties.ad/DCE-RPC_
Interface_

```
UUID
properties.ad/Failure_
Information:Status
properties.estreamer/malware/filePath
properties.ad_detected_
port
properties.:pam/javascript/unescape/limit
properties.estreamer.malware.eventDescription
properties.ad/SiteName
properties.ad/nsons
properties.ad/DCE-RPC_
Interface_
UUID-1
properties.argus/severity/adjusted/by/filterid
properties.ad/DCE-RPC_
Interface_
UUID-3
properties.ad/DCE-RPC_
Interface_
UUID-2
properties.argus/severity/reduced/by/filterid
properties.:tcp/connections/timeouts/data
properties.ad/More_
Sources
properties.estreamer/malware/eventDescription
properties.:stats/interval
properties.Check_Point/blade
properties.:tcp/rsts
properties.ad/New_
Logon:Security_
ID
properties.ad/client__inbound__interface
properties.ad/UserCheck__incident__uid
properties.ad/eventlogindex
properties.estreamer/HTTP_URI
properties.:len/pending/max
properties.:pam/conficker_p2p/report/interval
properties.argus/assessed/timestamp
properties.ad/appi__name
properties.destinations/blocked/count
properties.ad/TimeZoneOffset/l
properties.:tcp/hynacks
```

```
properties.ad/email__control
properties.baseline/events/per/hour
properties.ad/dropped__incoming
properties.ad/Update_
Version
properties.ad/__id
properties.:tcp/connections/flushed
properties.ad/log__id
properties.ad/mem_
utilization_
percent
properties.ad/outzone
properties.ad/stationName
properties.argus/intruder/aggr/bits
properties.ad/app__rule__id
properties.ad/ProcessId/i
properties.ad_auth-db-query-entry-async-name
properties.argus/source/aggr/bits
properties.ad/Additional_
In0c2UQw_~_~ntication_
Type
properties.ad/cpu_
utilization_
percent
properties.ad/bytesIn
properties.ad.dropped__incoming
properties.ad/PendingActions/l
properties.ad/resultIndex
properties.ad/ts/l
properties.:VirtualAddr
properties.ad/connectionType
properties.ad/limit__requested
properties.ad/client__inbound__packets
properties.ad/TCP_
packet_
out_
of_
state
properties.ad/ip__id
properties.ad/server__outbound__interface
properties.:flushed/size
properties.:tcp/packets
```

properties.:hits
properties.:in/attacks
properties.ad/Industry_
Reference
properties.ad/s-supplier-country
properties.ad/mplsTag
properties.:pam/dns/tunnel/min/data/length
properties.argus/assessed/by/userid
properties.baseline/maxTS:
properties.ad/web__server__type
properties.ad/SubmessageToken
properties.event/role
properties.ad/snid
properties.argus/victim/aggr/bits
properties.ad/Certificate_
INW93RA_~_~_
Serial_
Number
properties.:tcp/xsum_errs
properties.ad/WindowsParserFamily
properties.ad/Protection_
Name
properties.:threat/threshold
properties.ad/CollectionHost
properties.ad/EventIndex
properties.ad/app__risk
properties.ad/matched__category
properties.ad.authenticationResult
properties.ad/Packet_
info
properties.ad/requestUrl
properties.:tcp/connections/timeouts/abort
properties.:ipv4/packets
properties.ad/resource__shortage
properties.ad_app__name
properties.ad/email__session__id
properties.ad/Source
properties.ad/DetectionSource/i
properties.ad/p__dport
properties.ad/Parameter
properties.ad_protocol
properties.ad/tcp__flags

properties.ad/ip__len
properties.ad/reject__id
properties.ad.configuredName
properties.ad/TimeReceived
properties.baseline/minTS:
properties.destination/ports
properties.source/ports
properties.ad.connectionType
properties.ad.data
properties.ad/Additional_
Inc1+cZg_~_~sited_
Services
properties.:arp/packets
properties.multiple/domains
properties.ad/Access-group
properties.ad/server__outbound__packets
properties.ad/rs(Content-Type)
properties.ad/bandwidth
properties.:pam/content/zip/uncompressed/min
properties.ad_dstApplication
properties.ad/x-bluecoat-application-operation
properties.port/blocked/count
properties.ad/message
properties.ad/proxy__src__ip
properties.ad/client__outbound__interface
properties.:tcp/synacks
properties.ad/x-exception-id
properties.:queue/full
properties.ad/app__properties
properties.ad/rawEvent
properties.ad/TimeDetected
properties.ad/returncode
properties.ad/packets
properties.ad/from
properties.:pam/dns/tunnel/report/interval
properties.ad/data
properties.ad/Destination__Interface
properties.attack/identifiers
properties.baseline/events/per/second
properties.ad/received__bytes
properties.:pam.html.attribute.length.limit
properties.ad/Detailed_

AuthbXaM4Q_~_~ion:Key_
Length
properties.:len/pending
properties.ad/contract__name
properties.ad/Key[20]
properties.:ipv4/xsum_errs
properties.ad/sent__bytes
properties.ad/Service_
Information:Service_
ID
properties.:no/pending
properties.argus/transformed/by/filterid
properties.ad.authProfile
properties.ad/CategoryID/1
properties.ad/s-supplier-failures
properties.event/blocked
properties.:pam/script/suspicious/score/limit
properties.:mem/max
properties.ad/Confidence_
Level
properties.:in/events
properties.ad/segment__time
properties.ad/app__category
properties.ad/WindowsVersion
properties.ad/dst__machine__name
properties.ad.rawEvent
properties.:out/status
properties.argus/created/by/filterid
properties.ad/RequirementID
properties.:GridFit
properties.argus/request/modified
properties.ad/authenticationResult
properties.ad.sourceDnsDomain
properties.:no/update
properties.ad/subscription__stat__desc
properties.ad/Certificate_
Id5IXSg_~_~ate_
Thumbprint
properties.ad/version
properties.ad/ip__offset
properties.ad_newPID
properties.:tcp/events/attack

properties.ad/Additional_
InMbHgGQ_~_~Ticket_
Options
properties.:mem/limit
properties.ad/src__user__name
properties.ad/Total_
logs
properties.ad/portal__message
properties.ad/Suppressed_
logs
properties.ad/Detailed_
AuthN28meg_~_~me_
(NTLM_
only)
properties.:pam/js_fromcharcode/multi_radix/required
properties.ad/Code
properties.argus/destination/aggr/bits
properties.ad/Account_
Information:User_
ID
properties.ad/interval
properties.ad/server__inbound__interface
properties.:origin
properties.ad/frequency
properties.ad/Certificate_
I0tc7xg_~_~te_
Issuer_
Name
properties.ad/s-supplier-ip
properties.:tcp/connections/active
properties.:error
properties.ad/UserCheck__Confirmation__Level
properties.:trons/rules/count
properties.ad/NAT__rulenum
properties.estreamer.malware.parentFileShaHash
properties.:in/status
properties.ad/UserCheck
properties.:pam/http/request/limit
properties.ad/vlanTag
properties.ad/Account_
For_
WiJfqpA_~_~ed:Security_

```
ID
properties.:tcp/segments
properties.:flushed/age
properties.:PercentFromIntruder
properties.ad/OSVersion
properties.ad/ActionID/1
properties.estreamer/HTTP_Hostname
properties.ad.destinationTranslatedIPv6Address
properties.ad_auth-db-update-entry-async-name
properties.:numIndices
properties.:fileSize
properties.:dPath
properties.:INDXsize
properties.:Accept-Encoding
properties.ad_minutes
properties.:FILE
properties.computerName
_score
```