Raimo Heikkinen

Information Security Case Study with Security Onion at Kajaani UAS Datacentre Laboratory

BBA

Business Information Technology

Spring 2018

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Author(s): Heikkinen Raimo

Title of Publication: Information Security Case Study with Security Onion at Kajaani UAS Datacentre Laboratory

Degree Title: Bachelor of Business Administration

Keywords: Information Security, Datacentre, Networks, Security Onion, Intrusion Detection & Prevention Systems, Unix, Virtualization

This graduation thesis' goal was to evaluate and improve the information security stance of the Datacentre Laboratory of Kajaani University of Applied Science which hosts its own datacentre for student governed administration to facilitate teaching and project work. This work aims to improve this by the means of deploying an intrusion detection solution to the datacentre production environment which prior to this thesis works has no such solution for network security monitoring.

The theoretical portion of this thesis describes the key functions and common features of intrusion detection and prevention solutions in non-vendor specific manner with the aim of giving the reader a clear view of what constitutes as an intrusion detection & prevention solution and what requirements there are for them to be an effective and how and where they operate regards to networks and datacentres.

The practical portion of this thesis describes the chosen security tool, its characteristics, and requirements for deployment along with the actual plan of deployment and deployment process for the UAS Datacentre Laboratory. The deployment's initial configuration and tuning takes into account to match the hosting production environments' characteristics such as network topology and requirements for effectiveness.

With the completion of the deployment the administrators of the laboratory can use the deployed security solution to improve visibility into the monitored network traffic of the laboratory and have full packet captures available for the data retention period for forensic and analysis tasks. This deployment will produce a solution that can be further adjusted and expanded to meet the future needs of the datacentre laboratory and its administration.

TIIVISTELMÄ

Tekijä(t): Heikkinen Raimo

Työn nimi: Tietoturva Case Study Security Onion avulla Kajaanin AMK:n konesali laboratoriossa

Tutkintonimike: Tietojenkäsittelytieteiden tradenomi

Asiasanat: Tietoturva, Konesali, Tietoverkot, Security Onion, Tunkeutumisen Havainnointi- ja Estojärjestelmät, Unix, Virtualisointi

Tämän opinnäytetyön tarkoituksena on arvioida sekä parantaa tietoturvaa Kajaanin ammattikorkeakoulun konesali-laboratoriossa jonka itsenäisessä konesalissa opiskelijoiden ylläpitämänä tarjotaan ympäristö opetukselle sekä projektityölle. Työn tarkotuksena on saavuttaa tämä tavoite asentamalla tunkeutumisen havainnointi järjestelmä konesalin tuotantoympäristöön jossa tätä työtä aiemmin ei ole ollut vastaavaa monitorintiratkaisua.

Työn teoreettinen osuus käsittelee tunkeutumisen havainnointi- ja estojärjestelmien keskeisiä ominaisuuksia sekä toiminnallisuuksia ilman tukeutumista tuote tai valmistaja kohtaisiin ratkaisuihin. Näin lukijalle jää selvä käsitys mitä ominaisuuksia ja vaatimuksia näillä järjestelmillä on jotta ne olisivat tehokkaita ja kuinka nämä toimivat tietoverkkojen sekä konesalien saralla.

Käytännön osuudessa käydään läpi valitun tietoturvatyökalun ominaisuuksia, vaatimuksia sekä itse asennuksen suunnitelma sekä asennusprosessi AMK:n konesali laboratoriossa. Asennus ottaa huomioon asennuksen aikasen konfiguroinnin sekä säädöt niin että nämä ottavat huomioon laboratorion tuotantoympäristön ominaisuudet kuten tietoverkkorakenteen sekä vaatimukset tehokkuudelle.

Työn valmistuttua laboratorion ylläpitäjillä on mahdollisuus asennetun tietoturvaratkaisun käyttämiseen tietoturvan näkyvyyden parantamisessa laboratorion monitoroidun liikenteen osalta sekä saatavuus täyteen pakettikaappauksiin laboratorion liikenteestä analyysiä sekä tutkintaa varten. Tämä asennus tuottaa ratkaisun jota voidaan jatkossa laajentaa ja säätää konesali-laboratorion sekä sen ylläpidon tulevaisuuden tarpeita varten.

TABLE OF CONTENTS

SYMBOL LIST

AP = Access Point

BBA = Bachelor of Business Administration

CPU = Central Processing Unit

CSP = Cloud Service Provider

DDoS = Distributed Denial of Service

DMZ = Demilitarized Zone

DNS = Domain Name System

DPB = Data Protection Bill

FTP = File Transfer Protocol

GB = Gigabyte

GDPR = General Data Protection Regulation

GUI = Graphical User Interface

HIDS = Host Intrusion Detection System

HTTP = Hypertext Transfer Protocol

ICMP = Internet Control Message Protocol

IDPS = Intrusion Detection & Prevention System

IDS = Intrusion Detection System

IETF = Internet Engineering Task Force

IP = Internet Protocol

IPS = Intrusion Prevention System

ISMS = Information Security Management System

ISO = International Standard Organization

ISP = Internet Service Provider

KVM = Keyboard Video Mouse

LTS = Long-term Support

MAC = Media Access Control

NBA = Network Behavioural Analysis

NDA = Non-Disclosure Agreeement

NIC = Network Interface Controller

NIDS = Network Intrusion Detection System

NIST = National Institute of Standards & Technology

NSM = Network Security Monitor

NTP = Network Time Protocol

OS = Operating System

RAID = Redundant Array of Independent Disks

RAM = Random Access Memory

SIEM = Security Information & Event Management

SSH = Secure Shell

SSID = Service Set Identifier

SSL/TLS = Secure Socket Layer/Transport Layer Security

STA = Station

TCP/IP = Transmission Control Protocol/Internet Protocol

UAS = University of Applied Sciences

UDP = User Datagram Protocol

UFW = Uncomplicated Firewall

UPS = Uninterruptible Power Supply

URL = Uniform Resource Locator

VLAN = Virtualized Local Area Network

VM = Virtual Machine

WAN = Wide Area Network

WLAN = Wireless Local Area Network

1 INTRODUCTION

This work aims to give its readers basic theoretical understanding of information security and information assurance surrounding the case study. The case study employs the use of intrusion detection systems, network security concepts, and how they affect datacentre security. The case study portion of this work relies on the infrastructure of Kajaani University of Applied Sciences datacentre lab and the tool for this case study is Security Onion. The purpose of the thesis work is to study both, the theoretical and the practical aspects of the tool to examine Kajaani UAS datacenter lab environment and to recommend security guidelines for daily operations.

The data gathering suite for this will be deployed in a process-oriented case study tailored to meet the needs of the hosting infrastructure and its current day-to-day configuration. The deployment will be made in a way to maximize deployment security and monitoring scope while also masking the presence of an intrusion detection related monitoring. This study will generalize and obfuscate some portions of the practical case study to avoid disclosure of vital configuration(s) and/or layout of the datacentre lab.

The business world—technology included—is rapidly changing and the pace is only getting faster as of last two decades. This speeding up of technological advancement is like the law of accelerating returns, introduced by Ray Kurzweil in his book “The Singularity Is Near.” This means increasing challenges for information security and assurance, especially in current times when there’s increased demand for data retention and privacy by signatory governments and countries; with sanctions to be meted for failures to uphold them. Thus, there is an incentive for any organization to have visibility and control over their information security. (Berman & Dorrier, 2016)

1.1 Research Motivations

Research motivations for this study are partly due writer's personal interest but the biggest reason for tackling information security specifically in our university's BBA degree is the ever-evolving need of information security and assurance. Added to this are administration of the two; both in private life, between customers and businesses, and ubiquitous security that has no specific vendors.

In current times, the knowledge and knowledge of information management is essential to everyone, regardless of one's background. The quickly advancing technology makes it apparent that a "merely get by" mentality will no longer suffice. This requires actors to think of the ramifications of information security for the assets and tools they employ in their daily lives of work. (Kini, 2013)

On the rudimentary level, when an actor—be it researcher or an organization—introduces an isolated security principle into practical use, the design is no longer isolated from outside influence anymore. When this design is a single part which incorporates into a bigger system, it will have other weaknesses and aspects to it that an organization needs to evaluate and address accordingly by their security impacts. Real use cases and experience dictates that when the average downtime period for a service is 90 minutes, it is not about "if" in terms of designing security or availability but a matter of when the downtime happens. (Perlin, 2012)

Information security is of upmost importance and is always present on technology field, especially with datacentres when it comes to operating costs. Information security aspects evolve over time in response to the needs of industry and at times even juridical governance; e.g. the new data retention and privacy laws are causing organizational concerns and advocating new positions in today's businesses like data confidentiality representative. When one ignores charting risks and forgoes preparedness, it will not be a question of if, but when an unwanted event or catastrophic failure plays their full course with the administrators and owners reduced to be mere spectators. These downtimes come attached with averaged costs reaching thousands of dollars for every passing minute during downtime or handling a security escalation. (Perlin, 2012)

Prior Research & Definitions

International Standard Organization (ISO) has produced a standard series 27000, and included in this series is the standard 27001, which has defined information security thusly: *"Preservation of confidentiality, integrity and availability of information; in addition, other properties such as authenticity, accountability, non-repudiation and reliability can also be involved* (Taylor, Alexander, Finch, & Sutton, 2013, p. 9)*".*

UK Cabinet Office on the other hand has defined information assurance as: *"The confidence that information systems will protect the information they carry and will function as they need to, when they need to, under the control of legitimate users* (Taylor, Alexander, Finch, & Sutton, 2013, p. 9)*".* With this the quoted institution defines information assurance to be of bigger scope—meaning that the information is already secure by definition—and assures that the information systems and its functions, with the information contained within are in the control of only, and only by the legitimate users.

Taylor et al (2013) in their book "Information Management Principles" outline the notion that neither information assurance nor information itself isolate from outside influence. For information assurance to be a properly functioning part of a modern business, the assurance cannot be on the shoulders of single actor or employee. Information security and its processes, functions, and planning are on the shoulders of the necessarily qualified staff but for information assurance to reach assurance itself it will need the involvement of all organization's actors and assets. (Taylor, Alexander, Finch, & Sutton, 2013, p. 9)

Taylor et al. (2013) lays risks factors for information security such as disillusioned employees, ex-employees or outsiders with ill intent. The multiple risk factors, causes for damages, and problems with these factors are critical for an organization's risk assessment. Business models change, organizations and business deal with less and less inside one set of borders or a single country and thus the requirements for flexible enough, yet adaptable information security and assurance are only getting increasingly complex and costly.

Information security and assurance cannot be extraneous or skippable steps in an organization's operations nor in its business practices and for that information security and assurance both need consideration and evaluation in company's business processes and daily operations. This translates into information assurance being a consideration or necessity to actors and designs on all levels of the operating business; with both the users and the devices. (Taylor, Alexander, Finch, & Sutton, 2013, p. 12)

With afore mentioned connections, both information security and assurance policies are by necessity for an organization's risk management and planning due to legislatures like UK's DPB and European Union's GDPR. On organizations' levels, information assurance should be there to take care of the mismanagement of information containing assets. Information assurance examples can be proper handling of organizational documents and company assets like hardware and optical media, the handling of Bring-Your-Own-Device scenarios and actor training. (Taylor, Alexander, Finch, & Sutton, 2013, p. 13)

To highlight the need for proper information assurance, security, and management of these aspects there is always the legal aspect. At times criminals are even further or specialized in information security than the best practices or policies when compared to the expertise employed or available to businesses or organizations. With the rise of data retention directives and legislation which govern the management of personally identifiable information, being well prepared with proper information assurance is important to all businesses. (Taylor, Alexander, Finch, & Sutton, 2013, p. 14)

Authors of "A Comprehensive Guide to Secure Cloud Computing" consider the following seven principles to be the main supports of information security and assurance. The first three are confidentiality, integrity, and availability while the four supplementary principles are authentication, authorization, auditing, and accountability. (Krutz & Vines, 2010, pp. 91-94)

- Confidentiality principle is defined by the prevention of both intentional and unintentional information disclosure to unintended actors.

- Integrity of data requires the following three requirements to be met; information is consistent both internally and externally at the same time, unauthorized actors do not make modifications, and that authorized actors cannot make unauthorized modifications.

- Availability of data means that the data is accessible at request and that the systems providing the data are functioning properly.

- Authentication guarantees that the actor's identity is that of the one they claim to be, e.g. actor provides a username and system requires them to submit the corresponding correct password stored in the system.

- Authorization grants the privileges of an actor's access to resources and assets and govern the extent of rights the actors possess for the session.

- Auditing is the ensuring of operational assurance and the verification of audit logs and trails of an organization. Auditing can be performed either internally or externally with both not being mutually exclusive.

- Accountability is the action of verifying and determining the behaviour and actions by a single actor which cannot repudiate the actions it has done.

When computing systems seek to achieve multiple objectives ranging from cost and performance to reliability and maintainability and with the need of achieving security then trade-offs are a necessity to meet all these objectives. The realistic goal is to have an assurance of secure enough solution that is tolerable for daily operations while meeting the four other objectives to reasonable degree. (Krutz & Vines, 2010, p. 94)

Similarly, the authors relying on 1974 publication by Saltzer & Schroeder of the University of Virginia address the information security of information stored within a computer system with the following eleven security design principles. (Krutz & Vines, 2010, pp. 94-98)

- Principle of least privilege demands the allocation of minimum privileges, resources, and time to complete the required task.
- Separation of duties requires multiple actors to achieve authorization of action, thus requiring collusion or cooperation between actors for a breach.
- Defence in depth necessitates multiple layers of nested protections where each layer is isolated from the breach of another layer.
- Fail safe dictates that if a system should fail then it should fail into a state that will not compromise the security or data by further escalation.
- Economy of mechanism offers a principle of mitigating unintended access paths from promoting simple and comprehensible design.
- Complete mediation demands that an authorization procedure cannot be suspended or bypassed and must run the full authorization path.
- Open design governs the question of security by obscurity versus security by verification; an open design can be verified and improved by experts.
- Least common mechanism sets that the minimum amount of protection mechanisms should be shared between actors to avoid covert channels.
- Psychological acceptability relates to the ability of a systems users to rely on intuitiveness to operate without needing to follow complex instructions.
- Weakest link dictates that the security is as strong as its weakest link thus necessitating the need of all links being at acceptable security level.
- Leveraging existing components approach allows for secure system to be divided into partitions of independently defended sub-partitions.

1.2 Research Method & Questions

The research method for this study is process-oriented case. Due to time and resource constraints this thesis will have to focus on singular solution to provide the data for analysis and testing and is thus vendor specific and will not consider or compare to other security solutions.

Limiting the research method observation to single process is chosen primarily to avoid introducing unwanted uncertainty to the security design of the hosting infrastructure, e.g. overlapping security solutions that all perform same tasks and can interfere with either each other or the underlying systems.

The research questions are chosen to evaluate and support the security solution and its deployment and set a path of research goal as to what the practical portion of this study must meet with its deployment planning and objectives in Chapter 5 Case Study: Security Onion Deployment at Kajak DC. This is to ensure the connection between the practical with the theoretical portions of this research.

The research questions set for this study are:

- Does the chosen security solution and its deployment configuration fulfil the theoretical requirements for an intrusion detection solution set by NIST?
- What is the optimal method of deployment and configuration and why is it chosen for this unique infrastructure scenario?
- Did the chosen security solution improve the information security and assurance of target organization with its deployment?

2 INTRUSION DETECTION & PREVENTION SYSTEMS

As the case study part of this work relies heavily on Security Onion and its various methods to analyse and interpret network traffic and interactions, this chapter will strive to lay down the basic principles and methods on which most intrusion detection and prevention systems (henceforth, IDPS) rely on. As an example, a security solution or a framework might rely just on intrusion detection capabilities, thus leaving the actual prevention or mitigation to the system administrators or security engineers.

An IDPS solution might incorporate both, the methods and capabilities of an intrusion detection system and expand further by being an intrusion prevention system—automating the tasks of prevention in a solution that handles incidents and policy violation events—leaving the system administrators and engineers the task of creating and configuring the security policies, and the monitoring of the logs and alerts generated by the solution and to act upon the reported events and incidents.

Most IDPSs work either as an intrusion detection systems (IDS) or combining the capabilities of IDS with an intrusion prevention system (IPS) and it is up to the deploying organization to decide if they want to use only the monitoring capabilities of an IDPS or to include it as a more permanent asset of infrastructure that prevents intrusions. If deployed as latter option, then such a system must have appropriately adequate authority and access over the security infrastructure and a design which allows the solutions to monitor and uphold the security policies and if violations are detected, take corrective actions to ensure compliance with these policies.

If an IDPS is to monitor an organization’s network and its hosts, it will be handling the task of monitoring the traffic and activities of tens, hundreds, or up to thousands of users and devices connected to the network. This requires for the IDPS solution to be a design consideration in the construction of an organization’s infrastructure or part of a later makeover if it aims to be a growth-feasible solution when it comes to an organization’s daily processes and users and its future needs.

The primary focus of IDPSs is identification of incidents, e.g. either breaches in security or violations of security policies set by security administrators or engineers. These events have many sources, as explored by Scarfone et al, in their 2007 publication “Guide to Intrusion Detection and Prevention Systems (IDPS)”, along with the basic principles for what constitutes as requirements of an IDPS according to National Institute of Standards and Technology (NIST). The events’ origins could be actual from the Internet, activities initiated by malware (e.g., worms, trojans, backdoors) in internal network, violations stemming from unauthorized access attempts from (un-)authorized users, or users trying to gain access to assets for which they have no authorization for. (Scarfone & Mell, 2007, p. 15)

With information security solutions, when the scope of data monitored is the entirety of data from multiple sensors and sources, the possibility of errors and inaccuracies is greater. (Scarfone & Mell, 2007, p. 15) point out, a legitimate user might be mistyping an Internet protocol (IP) address or a uniform resource locator (URL) address by accident or has simply forgotten his own password and is trying different variations. Processing these kinds of aberrations is an objective for the security administrators as they provide insight or details on activities that need proper categorization, so that henceforth they are discernible and separate from the monitored flow and not producing false positives. This also necessitates the need of different tiers when it comes to security events, and the more verbose they are, the better the administrators can act on these events.

This logging of data is of importance in fine-tuning an IDPS by its administrators and at the same time could provide critical help to incident handlers if an event escalates or when a security event reveals that the damage has already occurred and there is a need to find out all trails and relations to the originating incident. These connections and the related events are critical in incident handling with verifying the final damage and finding out the reach of the incident. Incident handlers and security engineers need to be able to recreate scenario of the incident and its relations from the event logs as to allow to be able to follow the incident. (Scarfone & Mell, 2007, p. 15)

Many IDPSs are configurable in comparable manner to what other network solutions are, e.g. making rulesets and the blocking/allowing of separate activities when compared against security policies. At the same time IDPSs are not merely retroactive but sophisticated enough to offer proactive protection and security in the form of analysing and looking out for malign activity in organizations' the internal network or infrastructure. Administrators should not disregard reconnaissance activity originating from the outside as it should be a consideration of how much, what parts of it, and in what way an organization's services and assets are visible to the public. At the same time, organizations have found other uses for IDPSs; e.g. the use in identifying security policy failures when IDPS rulesets duplicated with existing firewall rulesets, or as a tool for documenting existing threats to organization as a utility of executive reporting, and even showing the security policies in use as a deterrent for users in organization. (Scarfone & Mell, 2007, p. 15)

Reconnaissance activity like port scanning and probing from outside an organization network is in many cases against acceptable use policies of Internet service providers (ISP) and this is an activity that happens on regular basis on the Internet, and administrators on the receiving end classify this activity as background noise. This background noise is present with any network connected to public networks and is a notable concern as a grey area of legality, and the legality of this activity differs from nation to nation. For example, in Finland there has been a ruling by Finnish Supreme Court in an incident originating from 1998 that port scanning a bank is punishable offense as it shows an intent of breaking into a secured environment. Another example from the other end of the legal spectrum comes from Israel, from 2004, where a judge acquitted a person from vulnerability scanning the Mossad—Israel's national secret service—when he had not shown malicious intent in doing so. Recent juridical changes are from United Kingdom and Germany, with both nations opting to sign into effect broader cybercrime laws in 2007 and 2008 respectively. Both changes were meant to ban the distribution, use, and even possession of broadly categorized hacking tools. This does not take into consideration the use of security tools like Nmap (network mapping) which can be used both for legal purposes without malicious intent (white-hat hacking) and with malicious intent (black-hat hacking). (Lyon, 2016)

2.1 Key Functions

(Scarfone & Mell, 2007, p. 16) in their NIST publication consider few key function requirements that an IDPS needs to fulfil for it to an appropriate solution or a product according to NIST guidelines, and for this the authors list 3 key functions, separately for both intrusion detection and intrusion prevention.

An IDS solution can have the capability of changing their behaviour when it detects a new threat, or an incident event is triggered. This can result in behaviour where an IDS solution will start gathering additional information concerning the trigger's environment. These are the three main functions that an IDS must fulfil for it to be capable of intrusion detection: (Scarfone & Mell, 2007, p. 16)

- *"Recording information related to observed events."* The solution must record information which it stores either locally or stored in a separate system such as log server, security information and event management solution, or remotely off-site. This collected information is critical in incident response and handling when the handlers or investigators need to re-construct a scenario with the retained information.

- *"Notifying security administrators of important observed events."* The solution can term this as an alert, which it can deliver through several methods, e.g. an e-mail, a phone page, a dashboard update, a regular update on an IDPS's console/interface, a system log message or user-defined programs or scripts. These kinds of alerts could be sent to either system administrators, security engineers or supervisors depending on the previously configured rules for system behaviour and security policies.

- *"Producing reports."* This function links to the first function listed here but offers a broader view into the IDS's key capability. The aim could be an overview much akin to what a web-hosted dashboard would give to an on-site actor like an administrator or an engineer. A report can be basis for executive report or long-term evaluation depending on the measured statistics, which could be alert levels, frequencies, locations, users, incidents, or response times and so on.

These are three functions IPS solution must fulfil for it to be capable of intrusion prevention. All three functions share the characteristic of attempting to respond to a detected threat or security event, and either prevent it from succeeding or to mitigate its effects. The authors divide the response techniques employed into the following three categories: (Scarfone & Mell, 2007, pp. 16–17)

- *"The IPS stops the attack itself."* The IPS could achieve this objective with either terminating the malign network connection or user session that it has detected to be malicious. If termination is not possible then the next step could be containment by blocking access routes to and from the originating user session/device.

- *"The IPS changes the security environment."* This kind of technique is loosely related to the afore mentioned method of stopping an attack in which both share the common objective and the requirement of sufficient integration into the infrastructure of an organization. A change in security environment could mean a change in network security (via a firewall, a router, or a switch) or in host devices deemed to be at risk with the detected intrusion. E.g. action would be to block network access to and from of the affected network zones or host devices, alteration of network zones or domains or even applying pre-configured patches when a device joins the company infrastructure but is unpatched or unsafe according to security policies.

- *"The IPS changes the attack's content."* An IPS technology or solution could try to make an attack or malicious activity benign by removing the malign portion. E.g. the IPS would let an e-mail pass in the network but removes the malign attachment and modifies the e-mail in a way that shows that it has altered the message as per security policies.

A common trait between IDPS technologies is the principle that they cannot provide completely accurate detection rate. Results can be either false positives, benign but triggers a response, or false negative where malign activity bypasses detection. An IDPS manages this limitation by being adjustable, leaving the tuning to the supervising administrators and engineers. (Scarfone & Mell, 2007, p. 17)

2.2 Common Detection Methods

(Scarfone & Mell, 2007, p. 17) consider three main categories of detection methodologies with IDPSs, with most solutions using multiple technologies working separately or combined to provide broader or more effective detection. These different techniques are based on signatures, anomalies, or on stateful protocol analysis.

Signature based analysis is detection method that compares patterns in observed traffic or activities against known threats or policies. Known threats can form from any combination of sources, e.g. IDPS vendor supplied signature database, or a third-party lookup service, a database of known vulnerabilities (e.g. outdated systems or system services being offline), a malware signature database (e.g. executable masked as an e-mail attachment), company security policies that prohibit user actions like attempting to login as a root user. Detection based on signatures is the simplest method of detection since it will only consider observed events in a limited scope, such as a singular e-mail, a network transmission packet, or a log entry. This limited scope means that this kind of detection will not consider relations between events or protocols, meaning that it cannot draw a connection between a log file entry and a HTTP request returning a status code of 403; to signature-based detection method these events are two separate events where it analyses both separately. As the simplest, yet an effective detection method it is also the least resource intensive operation employed by an IDPS. (Scarfone & Mell, 2007, p. 18)

Detection based on anomalies relies on IDPS comparing definitions of activities that it considers normal activity and those that it considers to be triggers for an event where an activity goes above a threshold into to be either a risk or a suspicious activity. Anomaly-based methodology relies on either predetermined or learned profiles that represent normal behaviour for entities such as users, hosts, network sessions, and applications on top of which the IDPS relies on statistical methods to perceive if the monitored activity by it is deemed benign or suspicious. (Scarfone & Mell, 2007, p. 18)

Examples of profiles perceiving events to be against normal behaviour can be any of the these scenarios; a user logging on or accessing a company asset from a unidentified host device, user's network traffic from or to a secured asset is outside of normal operating hours, an application tries to generate a database transaction that's against security policies, user tries access or copy a secured asset to an unsecured location, a server resource is either operating under uncommon load or can be unavailable when it should be available. Anomaly-based detection is very efficient in detecting threats that are previously unknown or are using new vectors but at the same time they rely heavily on building profiles, this constitutes as training. Profile training is the activity of IDPS learning what constitutes normal activity for an observed entity, and its related thresholds as to what would trigger an alert or an event for suspicious activity. E.g. a server could be monitored for a month and then this period's observed activities are deemed to be its normal behaviour (user logins, running processes, performance, etc.) but if there suddenly is a new process or routine maintenance on the server which causes outage, this activity would be flagged as deviation and trigger a false positive in the IDPS. (Scarfone & Mell, 2007, p. 19)

Stateful protocol analysis relies on monitoring protocol usage against predetermined profiles of benign protocol activity where IDPS compares usage events against common deviations or misuse. Where anomaly-based detection uses either built profiles to monitor activity, stateful protocol analysis relies on vendor supplied universal-profiles that base on how the common protocols should be and how they should not be. These vendor-supplied profiles base on either software vendors or standards governing bodies for how protocols would operate normally. Stateful in this methodology means that the IDPS's capabilities allow it to assess relations between or across the observed network, transport, and application protocols. An example of this would common File Transfer Protocol (FTP), where in a communication session an unauthenticated user normally should only perform certain commands at certain points of the exchange but if at any point there is a deviation akin to inputting a command uncommonly (e.g. too long input string) then this kind of activity is a deviation. Major downside for this methodology is its resource intensiveness. (Scarfone & Mell, 2007, pp. 19–20)

2.3 Architectural Components

Typical IDPS consists of four components; an utility capable of monitoring and analysing activity in either network infrastructure where it would be a sensor or in a host device where it's termed as an agent, a management server which is a centralized device meant to manage and receive information from the sensors and agents deployed across the infrastructure and assets, database server for further collecting and organizing of information and statistics coming from afore mentioned sources, and a controller interface—usually called console—that is meant for configuring or management of the IDPS. (Scarfone & Mell, 2007, p. 23)

These four components can offer vendor specific capabilities above their basic tasks, with consoles branching out to be for configuration of components and applying updates, and another console for monitoring and analysis. Management servers can also branch out across vendors or solutions where the servers can perform additional analysis and identification of events that the individual monitoring components cannot, or they could match information from multiple sources to generate an alert or to trigger an event. A smaller scale IDPS deployment might not have a management server and a larger deployment might have either multiple management servers or they can be tiered above each other. (Scarfone & Mell, 2007, p. 23)

These components can exist within an isolated management network where these components have additional network interface connecting to the management network and the original interface connecting to the production network. Network sensors and host agents would be unable to pass any traffic between their two interfaces and management servers and consoles would only connect to the management network. This increases the security by IDPS being harder to detect for attackers, and to increase operational/bandwidth capacity under a spreading malware infection or a distributed denial of service (DDoS) attack but this comes with the cost of increased networking and administrative gear. Between these approaches would be a virtualized local area network (VLAN) within single production network, offering some protection to avoid exposing the IDPS devices, its traffic, or its management. (Scarfone & Mell, 2007, pp. 23–24)

2.4 Security Capabilities

IDPSs come with security capabilities that provide information and logging capabilities. Information gathering capabilities refer to the ability of solutions to identify and probe their operating environment for data through their components; information such as network characteristics or network layout or the operating system and its running processes or installed applications. Logging capabilities refer to the actions taken with detected events where the IDPS will gather related data for later use in incident response where the data can help ascertaining validity of the events, or provide further information to analysts monitoring the incidents, or allow correlation between monitored sources. IDPSs usually gather at least the following fields for events; date and time, event or alert type and their importance or priority rating, actions taken. Some IDPSs may log additional information such as network traffic's packet captures, logging of IDs relevant to the user or host sessions which related to the event or alert. For an IDPS to avoid outside compromise, logs need storing both locally and outside so that attackers cannot simply delete or alter logs to avoid detection or fetter subsequent incident handling. For accurate logs it is important for the infrastructure and the IDPS to employ network time protocol (NTP) or a similar approach to keep log entries and components in-sync so that they use accurate timestamps. (Scarfone & Mell, 2007, p. 24)

IDPSs usually use combinations of detection techniques and methods to provide security capabilities and increased accuracy of detection with allowing more tuning customization. The categories of the events and accuracies of the detections vary between IDPS technologies and vendors, where most solutions require at least some tuning and customization to improve the usability, accuracy, and effectiveness of generated events and alerts. These solutions vary in capabilities and typically the more powerful the solution's capabilities are, the better the accuracy improvement when compared to the default configuration. As such organizations should carefully consider these capabilities when comparing products between vendors against the needs and limitations of their infrastructure. (Scarfone & Mell, 2007, p. 25)

Typical detection capabilities categorized by (Scarfone & Mell, 2007, p. 25) are thresholds, blacklists and whitelists, alerts, and code viewing/editing. As the name threshold implies, it is a value that dictates the limit between normal and abnormal behaviour, where this threshold value is the upper limit and records going over this limit are abnormal. Examples of use would be an event where there are X amount of failed login attempts in pre-set interval, or when an input or filename exceeds threshold length. Anomaly-based methodologies and stateful protocol analysis employ the use of threshold-based capabilities.

Blacklists and whitelists employ the use of lists of discrete entries which are known to be related to malicious or benign activity in the scope of security policies. Blacklists can contain entries such as hosts, network ports, Internet control message protocol (ICMP) messages, applications, executables, known abnormal usernames, addresses, or extensions. Whitelists are the counterpart where administrators adjust the list on granular basis where needed, an example would be a program in in-house use that uses certain protocol(s) or known network ports for traffic and this activity in a known configuration would be whitelisted to lower false positives. Stateful protocol analysis and signature-based detection employ the use of black– and whitelists. (Scarfone & Mell, 2007, p. 25)

Events which generate alerts can do it so in multiple priorities and the administrators can customize these settings as needed for elevation or prioritization. Customizations for alerts include the subsequent prevention actions that should be taken, what information should be recorded and forwarded, what notifications should be sent and by what means, setting of the event's severity or priority, or even toggling the alert on or off in certain scenarios where it would be possible for the IDPS and its administrators to be overwhelmed and slowed down by the subsequent redundant alerts. An example of a redundant alert for the solution to display would be during an infectious spread of a malware or a DDoS attack where only the bigger picture matters, not the individual alerts. The information and related events produce logs and information for the centralized management, but the IDPS should suppress the unneeded alerts and refrain from taxing resources or encumbering of operation. (Scarfone & Mell, 2007, p. 25)

Some IDPSs allow the administrators to view parts or all the detection method's program code related signatures and some vendors allow the administrators to see additional code, e.g. applications or code used for stateful protocol analysis. The benefit of code viewing ability comes from allowing the engineers or administrators to determine why and how alerts generate and as aid in validation of events and false positives. (Scarfone & Mell, 2007, pp. 25–26)

IDPSs usually come with prevention capabilities that combine various detection technologies with the administrators having the possibility of configuring and adjusting prevention/corrective actions related to the alerts or events in question. Prevention capabilities for an IDPS would employ the functions laid out in chapter 2.1 Key Functions to stop malign activity with IDPS trying to follow the actions configured by administrators. As with stateful protocol analysis' profile building and training, some IDPSs' sensors come with a similar learning/simulation mode in which administrators can for a period of time train and verify their prevention configuration to tune responses to alerts/events generated by components in question. (Scarfone & Mell, 2007, p. 26)

2.5 Operational Layers

IDPS vendors usually divide their products into categories that is based on layers of the Transmission Control Protocol/Internet Protocol model (TCP/IP) or host device range, with these four categories being network-based IDPS, host-based IDPS, network behavioural analysis (NBA), and wireless-based IDPS. An IDPS solution might operate on one of these layers, multiple layers, or on all the layers mentioned. Organizations have to compare products and vendors and weigh them against the needs and requirements of their organization and infrastructure; a smaller organization might only need a network-based IDPS where as a bigger organization spanning multiple sites might need a more comprehensive solution. (Scarfone & Mell, 2007, pp. 20–21)

The difference between the four categories is their maturity as network-based IDPSs have been around for decades and host-based IDPSs arrived bit later with a newer approaches being NBA solutions (originally developed for detection and mitigation of DDoS attacks and analysis of network traffic flows) and newest being wireless-based IDPSs, partly in response to the growing deployment and use of wireless local area networks (WLANs) and WLAN clients (e.g. phones, pads, tablets, laptops). (Scarfone & Mell, 2007, p. 21)

Network-based IDPSs monitor network traffic for designated network segments, zones, or boundaries and they analyse both the network and application protocol layers for suspicious activity. Host-based IDPSs monitor and analyse the information of single host device (e.g. office computer) and its logs and characteristics of its activities. NBA-based solutions examine subsets of organization's network infrastructure, network zones, network sites for unusual traffic flows originating from DDoS-attacks, probing, malware infections, or unusual traffic that is against company policies. Wireless-based IDPSs perform the same tasks as network-based IDPS but they target the wireless access points, wireless endpoint devices and the traffic between these. (Scarfone & Mell, 2007, p. 21)

2.5.1 Network-based IDPSs

IDPSs working on network level will monitor designated network segments (zones, demilitarized zones, boundaries of subnets) and devices (routers, bridges, switches, etc.) and the traffic there-in to detect malicious activity. For reader to understand network-based IDPSs, one needs to familiarize the TCP/IP-model which facilitates much of the network communications in world, detailed below. (Scarfone & Mell, 2007, p. 35)

Communications facilitated by TCI/IP-model compromise of four layers that work together; when a user sends data as traffic across networks, the data passes from the highest layer to the lowest layer with each layer encapsulating the data from the previous layer with the new layer's information. This data then passes from layer to layer, location to location, with each layer and device examining the capsulated information and forwarding it until it reaches its destination and rises back with the data losing its encapsulating layers. The relevant four layers of TCP/IP-model from top to bottom are the application layer, the transport layer, the network layer, and the hardware layer with physical devices. (Scarfone & Mell, 2007, pp. 34–37)

Network-based IDPSs usually run their analyses at the application layer but also analyse network activity at the transport/network layers to identify malicious activity and to further facilitate the application layer's additional information needs. Some IDPSs also perform limited analysis and monitoring on the hardware layer. The components used in a network-based IDPS are similar with the architectural component categories mentioned in 2.3 Architectural Components except for the sensors. The monitoring sensors in network-based IDPS are in *promiscuous mode* that accepts all incoming traffic and passes it along to their destination and thus facilitates a mid-point that houses the needed monitoring, with most IDPS deployments using multiple sensors and large deployments capable of having hundreds of sensors. (Scarfone & Mell, 2007, pp. 35–37)

These sensors in the network-based IDPSs come in two variants, appliance variant and software only variant. An appliance-based sensor comes as specialized hardware and sensor software and thus adjoin to be part of a network traffic's flow whereas software only variant from a vendor comes as a sensor software that installs onto a host that meets vendor's specifications. Sensors in network-based IDPS's deploy either as an inline sensor or as a passive sensor. Inline sensors deploy to facilitate the monitoring of network traffic as it passes through, with the primary benefit from inline sensors being that they can provide better capabilities to prevent intrusions. Inline sensors usually situate along with the rest of network security devices, at the boundaries between subnets, at edges of networks where traffic needs crossing networks. Passive sensors, when deployed, monitor a copy of the actual traffic that passes by the sensors and no real traffic goes through them. Passive sensors monitor the network through either a network gear's spanning port, network tap or via IDS load balancer that feeds multiple sensors according to its configuration to balance network loads. Passive sensors cannot stop intrusions as effectively as inline sensors do. (Scarfone & Mell, 2007, pp. 37–40)

Network-based IDPSs offer the same categories of security capabilities as outlined in 2.4 Security Capabilities; information gathering, logging, detection, and prevention. Information gathering examples consist of identifying hosts and enabling the listing of organization's hosts based on IP/MAC addresses, identifying OSs which allows passive fingerprinting by analysing traffic headers, identification of applications by their versions and the subsequent monitoring of application communication and usage. (Scarfone & Mell, 2007, pp. 41–43)

Network-based detection capabilities usually interconnect to the use of combination of techniques outlined in 2.2 Common Detection Methods, e.g. would be of a stateful protocol analysis engine parsing activity into requests and responses and these being examined for anomalies and compared to signatures of malicious activity. The most commonly detected events at network level are the following types: reconnaissance and/or attacks facilitating different operational layers, unexpected application services, and policy violations. (Scarfone & Mell, 2007, pp. 43–45)

Detection accuracy for network-based IDPSs has in the past been known to produce high rates of false positives and false negatives but newer technologies using combinations of detection methods have caused an increase in accuracy but also caused the need of considerable tuning and customization. The deficiencies in detection accuracy are the result from the amount of traffic; a single sensor can often simultaneously monitor the traffic generated by hundreds or even thousands of internal/external hosts with extensive variety of operating systems and applications. (Scarfone & Mell, 2007, pp. 43–45)

Prevention capabilities of a network-based IDPSs vary between the sensor types, with inline sensors having the advantage over passive sensors as mentioned before. Inline sensors have the following three capabilities in preventing malicious activity: inline firewalling where the IDPS's sensors act as firewalls that can reject or drop traffic, bandwidth throttling to conserve network resources when the infrastructure is under Denial of Service (DoS) attack or malware infection, and alteration of malicious content by sanitization. Prevention capabilities available to both inline and passive sensors are the reconfiguration of other network security devices by instructing them to block or re-route analysed malicious activity. (Scarfone & Mell, 2007, pp. 46–47)

Network-based IDPSs offer comprehensive detection capabilities but suffer from prominent limitations with three of the most important being; encrypted network traffic, high traffic loads, and withstanding attacks against the IDPS itself. Network-based sensors can analyse the initiating negotiation in an encrypted session, but it cannot affect the encrypted traffic itself. High traffic scenarios are a limitation where network-based IDPSs are unable to perform at full capability, causing incidents to pass undetected. High load also causes disruption in network availability with inline sensors and to avoid this IDPSs sensors should come with load balancing or mitigative features to perform selectively under high loads. IDPSs are also a target for attacks, with the most common attack vectors being the use of DDoS attacks or anomalous network traffic to debilitate the sensors. Another vector of attack is so-called *blinding technique*, in which a diversionary attack generates so many alerts that the real attack goes unnoticed by the overwhelmed administrators. (Scarfone & Mell, 2007, pp. 45–46)

2.5.2 Host-based IDPSs

Host-based IDPSs monitor host devices and the activities generated by the host with the monitoring targets consisting of wired and wireless traffic, system/event logs and entries, process activities, file accesses/modifications, and system configuration changes. Host-based IDPS components consist of sensors, in this context called *agents*, which provide the detection and prevention capabilities within the singular host and communicate with the management/database server(s) and console(s) that exist outside of the host device. Host-based IDPSs' agents have two variants, an agent application running on single host that monitors devices such as servers and client hosts (end-user devices) and as a network appliance that shields multiple host devices. (Scarfone & Mell, 2007, pp. 73–74)

Host-based agents can further differentiate either as an agent application which monitors the host's activity events such as log entries and file accesses/modifications and has less impact on the host's normal operations, or they can deploy deeper into the host as *shims* which act as an intercepting layer on the OS which allows the monitoring and interception of processes and their operations (e.g. network traffic, filesystem activities, system calls, Windows registry activities). Difference between the two variants is that the former provides less prevention capabilities at a light cost on the host's resources, and the latter can provide better prevention capabilities at a heavy cost to host's resources due the need of monitoring and analysing all of OS/process. (Scarfone & Mell, 2007, p. 75)

The security capabilities of a host-based IDPSs split into four categories: logging, detection, prevention, and other capabilities. Information logged by host-based IDPSs contain the following: timestamps, event/alert types, effect ratings, event details, preventive actions taken. Host-based IDPSs use both signature-based and anomaly-based detection techniques to identify known attacks and previously unknown attacks respectively. Detection capabilities divide into four categories: event types detected, accuracy of detections, tuning/customization, and limitations. (Scarfone & Mell, 2007, pp. 75–76)

The types of events detected further divide to 6 types of techniques: code analysis, network traffic analysis, network traffic filtering, file system monitoring, log analysis, and network configuration monitoring. Code analysis allows the host-based shim agents to use one or multiple techniques to analyse host for malicious activity at and before code execution. Network analysis provides similar capabilities as network-based solutions but at the host level; the host agent analyses network layer, transport layer, and application layer protocols for suspicious activity. Filesystem monitoring may employ multiple techniques, including; file integrity checks, calculated hash checksum, and shim agent(s) which monitor access attempts and can block users and applications from inappropriately accessing system critical files. Log analysis allows agents to monitor OS/application logs to spot malign activity from system events. Monitoring of network configuration changes allows host agents to monitor for changes in host network devices. (Scarfone & Mell, 2007, pp. 76–78)

Host-based IDPSs' accuracy rate with false positives and false negatives is influenced by the challenges posed to detection techniques like log analysis and filesystem monitoring that do not account for the context where the detected events occurred. Normal host events like reboots, installations and critical file replacements can by their nature be normal host device operations or initiated by malicious actor. When the agent detects these events without the relevant context, it causes the agent to be unable to analyse the nature correctly. Some vendor products mitigate this by prompting the host device user for context, e.g. is the current event initiated by the user or related to an activity known to the user such as a new installation or maintenance. (Scarfone & Mell, 2007, p. 78)

Host-based IDPSs require significant tuning and customization as they rely on observing host activity and development of profiles for expected behaviour and a need of configuration by detailed policies set by administrators about how known applications installed to an organization's host behave. This is made more taxing by the changes in host environments, either from updates or by new installations, for which administrators have to address with adjustments in behaviour policies that are delivered as white- and blacklists by a management server. (Scarfone & Mell, 2007, pp. 78–79)

Host-based IDPSs provide intrusion prevention capabilities divided by the employed detection technique. Code analysis techniques can prevent suspicious code from execution and if configured properly also protect from previously unknown code from execution. Host-based traffic analysis and filtering—comparable to their counterparts with detected events section—allow host-based agent to employ firewalls in stopping incoming network traffic and outgoing traffic from exiting the host. Filesystem monitoring techniques can prevent the user, an application, or a process from accessing, modifying, replacing, or deletion of files which can stop malware infections, trojans, rootkits and other attacks from taking place on the monitored host. (Scarfone & Mell, 2007, p. 80)

Few host-based IDPS products offer capabilities outside of the usual IDPS technologies, combining the IDPS product with endpoint protection technologies such as antivirus, antispam, web/e-mail for endpoint protection with IDPS approaches such as removable media handling, audio/visual device monitoring, host hardening, process status monitoring and network traffic sanitization. Host hardening monitors system critical security settings and tries to reactivate them if they are offline. Appliance based agents can perform network traffic sanitization by operating as a proxy between host devices and their destination. Sanitization like this is very effective in lessening the effects of malicious reconnaissance in organization's network. (Scarfone & Mell, 2007, pp. 80–81)

Host-based IDPSs' have five notable technological limitations*;* delays in both alert generation and centralized reporting, host resource use, security control conflicts, and rebooting of hosts. Delays with alert generation occur on host-based IDPS agents because of some techniques run in set intervals or at set time of day to save resources. Centralized reporting delays result from IDPS solution conserving network resources; instead of sending alert data as they generate in real-time, the data transfers in interval batches. In the same manner the installing of agents on hosts can cause the installation to disable existing host security controls if they provide duplicate services. Reboots also may prove to be cause of concern if the agents are unable to detect latest threads when crucial hosts cannot reboot. (Scarfone & Mell, 2007, pp. 79–80)

2.5.3 Network Behavioural Analysis

IDPSs utilizing network behavioural analysis (NBA) examines network traffic, network traffic statistics, and traffic flows for unusual and suspicious activity like DDoS attacks, malware (e.g. worms, trojans, backdoors) and policy violations (hosts offering network services to other devices). NBA-based solutions customarily consist of sensors and console(s) and some vendor products offer management server(s) that they label as analysers. Notable difference to network-based and host-based IDPS solutions are the sensors of an NBA solution that are ordinarily available as appliances only. NBA sensors place similarly to how network-based IDPS solutions, but the difference with these is that NBA sensors do not monitor the target network directly but rely on flow information provided by the routers and similar networking devices. In this kind of NBA setup, the flow refers to the communication sessions occurring between hosts. These *flows* have few standards to them, including NetFlow by Cisco and sFlow by InMon Corporation and a typical data flow pertinent to an IDPS solution contains the following data sets: source/destination IP address, source/destination TCP/UDP ports or ICMP types/codes, number of packets and bytes transmitted in session, timestamps for the start and the end of the session. (Scarfone & Mell, 2007, p. 65)

Like network-based IDPSs, NBA solutions can operate either in an organization's standard production network or separate management network. When sensors are used to gather data from network devices, the NBA solution (console and the management server) can be logically separated from the standard production network. NBA sensors operate in passive mode using the same method of connection as the passive sensors of a network-based IDPS, e.g. router's network tap or a spanning port on switch. NBA solutions' passive sensors most efficiently place in locations where they can monitor key locations such as network boundaries, network segments, DMZ subnets, and near the location of perimeter firewalls, often even between the firewall and router bordering Internet so that they may protect the firewall from incoming attack that could overwhelm it. (Scarfone & Mell, 2007, pp. 65–66)

NBA solutions offer security capabilities split into same four categories: information gathering, logging, detection, and prevention. Out of all the IDPSs, NBA offers the most extensive information gathering abilities due to the knowledge of organization's hosts and their characteristics that are for most of an NBA product's detection techniques. Sensors in an NBA solution automatically generate and upkeep lists of hosts communicating in the range of its monitored networks and this monitoring covers port usage, passive fingerprinting, and mapping of host information. (Scarfone & Mell, 2007, p. 67)

Logging capabilities of an NBA solution are similarly extensive as its information gathering capabilities. The logged data can further utilize validation of alerts, investigation of incidents, and correlation of events between the NBA solution and other log sources. Certain NBA sensors, when monitoring network traffic directly, are also capable of logging limited payload information from packets which for an example allows the tying of actions to specific user accounts. (Scarfone & Mell, 2007, p. 68)

The detection capabilities of an NBA solution mostly rely on detection based on anomalies along with stateful protocol analysis techniques to analyse flows and most solutions do not offer detection based on signature capabilities. Detection capabilities divide to: event categories detected, detection accuracy, tuning and the related technological limitations. The types detected are DDoS attacks, scanning activities, malware, unanticipated application, and policy violations. (Scarfone & Mell, 2007, p. 68)

NBA solutions' sensors work by measuring substantial deviations in normal behaviour and are at their best in detecting attacks that generate unusual amounts of network traffic in short periods of time (e.g. DDoS attacks) or employ uncharacteristic flow patterns (e.g. worm/trojan malware propagating between hosts). As a downside, NBA solutions' sensors are less accurate to detect slowly culminating attacks if this sort of attack does not trigger alerts for violating administrator-set policies. NBA solutions' detection sensitivity against smaller scale attacks can increase but this will also increase the rate of false positives as a downside. (Scarfone & Mell, 2007, p. 69)

NBA solutions' technologies base dominantly on network traffic observation, developed baselines of flow expectations, host characteristic inventories, and NBA products automatically updating their baselines over time, leaving administrators the task of adjusting thresholds, black-/whitelists, and the adjustment of infrastructure environmental changes to reflect in the NBA operations. NBA solutions' prevention capabilities work by its sensors which are in majority of configurations passive deployments, meaning the effective intrusion prevention capabilities are the as limited as network-based passive sensors' which is the reconfiguration of monitoring area's other network security devices. (Scarfone & Mell, 2007, pp. 69–70)

NBA solutions have a significant limitation of delay in detecting attacks or anomalous behaviour as the data source is a monitored flow that first needs a delivery by a networking device like router and further compound by the flow data arriving in batches, where depending on NBA product's capabilities and network infrastructure, the batches could deliver every minute, every two minutes or up to fifteen and thirty minutes. (Scarfone & Mell, 2007, p. 70)

2.5.4 Wireless

Wireless networking devices enable devices to use computing resources of a network without physical connection to one, with the devices need to be within the operational range of a wireless network infrastructure. A WLAN is composed from a group of wireless networking access points in a limited geographical area and is capable of data exchange by employing communications over radio frequencies. A WLAN is typically composed of two elementary architectural components, which are stations and access points. A station is also known as wireless endpoint device, e.g. laptop, mobile phones, personal pads and tablets whereas access points logically connect the roaming endpoint devices with a distribution system which in most cases is an interface connected to organization's wired network infrastructure. (Scarfone & Mell, 2007, pp. 51–52)

Wireless sensors perform in the same manner as network-based IDPS sensors but function fundamentally differently due the complexities of wireless communication monitoring, since where traditional network-based IDPS sensor can see all the packets on the network it is covering (both inline and as passive), the wireless IDPS sensors work by sampling traffic on wireless frequency bands that separates into channels. A dedicated sensor is usually either fixed or mobile deployment, former being appliance-based and latter being mobile administration device that can be appliance-based or software-based. (Scarfone & Mell, 2007, pp. 53–55)

Selecting wireless IDPSs' sensor locations is completely different problem compared to the other types IDPS sensors. Wireless sensors should deploy to cover the entirety of organization's WLAN's range and even the physical boundaries, so that the sensors can find channels/bands not in use by an organization which mean rogue APs and/or ad hoc WLANs. Other considerations for locations include the physical security of a sensors deployed (e.g. are they tamper resistant, or positioned into a security camera's view), sensor ranges that are effected by the surrounding environment (e.g. walls, windows, construction elements). (Scarfone & Mell, 2007, pp. 56–57)

Wireless IDPS security capabilities offer information gathering, logging, detection, and prevention. Wireless IDPSs collect information on WLAN devices and build a maintainable inventory of observed devices including APs, WLAN clients, ad hoc clients/hosts where the inventory is based on SSIDs and the MAC addresses of the wireless network cards of devices. Some wireless IDPS sensors can use fingerprinting techniques on observed traffic to discover vendors instead of relying on the spoof-vulnerable MAC addresses. IDPS sensors also keep track of observed WLANs and categorize them by their SSIDs which allows the administrators to sort and tag them, e.g. as authorized, benign neighbours, or rogue WLANs. The logging capabilities of wireless IDPS perform extensive logging of information like the other IDPS solutions do and fields logged by these are: timestamps, event/alert type, priority/severity rating, source MAC address (vendor-specific first half, susceptible to spoofing), channel number, observing sensor's ID, preventive action taken. (Scarfone & Mell, 2007, pp. 57–58)

Wireless IDPSs can detect malicious activity at the WLAN protocol level (Wireless IDPSs do not examine communications at higher layers), with some products performing simple signature-based detections, and others use combinations of signature-based detection, anomaly-based detection, and stateful protocol analysis. To achieve broader and more accurate detection organizations should choose the wireless IDPS products accordingly. The types of events detected by wireless IDPS are: unauthorized WLANs and WLAN devices, poorly secured WLAN devices, unusual usage patterns, the use of wireless network scanners (war driving tools), DDoS attacks and conditions (network interference), impersonation and man-in-the-middle attacks (spoofing, session hijacking). Most wireless IDPS sensors can pinpoint the physical location or origin of a threat by triangulation when in the overlapping range of multiple sensors. (Scarfone & Mell, 2007, pp. 58–59)

Detection accuracy for wireless IDPSs is better than that of other IDPSs layers which attributes to the focus on single protocol layer and the scope within it. Wireless IDPSs require tuning and customization to improve the accuracy of detection with the main effort from administrators going to specifying WLANs, APs, STAs and their states. Due to the limited scope of WLAN protocol the alert types generated by wireless IDPS are not diverse or high in number. Customization for wireless IDPSs comes from anomaly-based detection thresholds and black-/whitelists to specify known malign and benign devices and vendors. (Scarfone & Mell, 2007, pp. 59–60)

Wireless IDPS sensors offer capabilities to prevent intrusions both, wireless and wired networks; sensors can terminate connections between malign endpoint devices and authorized access points over-the-air which is reminiscent of network-based TCP session sniping with upon successful de-association at AP the sensor would refuse new connections. Some sensors can even instruct a switch on an organization’s wired network to block activity involving malign endpoint device or access point based on the MAC address or employed switch port. Significant limitations of wireless IDPS are their susceptibility to evasion, inability to detect some wireless protocol attacks, and their inability to withstand attacks against the IDPS itself. (Scarfone & Mell, 2007, pp. 60–61)

3 DATACENTRE REGULATIONS, STANDARDS, SECURITY

As the European Union's GDPR is replacing the earlier Data Protection Directive 95/46/EC, it will also harmonize and unify the regulations between member states into a single market encompassing regulation when it comes to information security and assurance. As the May 2018 enforcement date looms in horizon it is important to note that the regulation is for protecting European Union's subjects' information, regardless of the data's physical location. With the GDPR's the broadened scope regarding data controllers (handlers, e.g. datacentres) and data processors (owners) means that a data processor operating in the United States must be compliant with the GDPR if they handle information or data pertaining to a European data subject, they are liable under the GDPR. (Gartner, 2017)

This scope means that all businesses and organizations handling data must appoint a representative to be a contact point for European data protection authorities. Depending on the scale of the controlling or processing of this data businesses can be required to appoint a data protection officer and demonstrate accountability in their activities pertaining to intra-organization processes where personal data of a subject is handled. The GDPR also requires higher standard of transparency and process identification in processes that handle this data so that data subjects may exercise their rights to be forgotten, withdraw data collected from them, and examine what has been collected from them upon request. (Gartner, 2017)

For European Union commercial datacentres are also affected by other regulatory developments outside of GDPR such as the Safe Harbor and Privacy Shield agreements pertaining to data flows between EU and US and the Digital Single Market strategy of 2015, Network and Information Security Directive that's part of EU's Cyber Security Strategy. As EU's Emission Trading Scheme phases and other environmental regulations. Other such regulations that govern European datacentres are those operating or facilitating processes that serve the financial sector, upon which they need compliance with payment processing standards such as Payment Card Industry Data Security Standard (PCI DSS). (TechUK, 2016)

For datacentres' information security needs, International Organization for Standardization (ISO) has published a series of standards that can be viewed as a standard series for datacentres to attain or to follow as a role model in their production and operating design. Some of these relevant ISO standards include ISO 27K series, namely 27001-27006. The ISO 27001 provides the specification for an information security management system (ISMS) through an iterative Plan-Do-Check-Act cycle and the activities for this cycle are as follows: (Krutz & Vines, 2010, pp. 248–250)

- **Plan** phase starts with the establishment of scope to precede the development of an ISMS policy which is followed by risk assessment and treatment planning. This is followed by determining control objectives and controls for security with documentation describing why these controls were selected specifically.

- **Do** component activities include the operation of selected controls, detection and response to incidents, security awareness training, and management of resources needed to accomplishing security tasks.

- **Check** activities include the operation of intrusion detection and incident handling and conducting both internal audits for the built ISMS and doing management review.

- **Act** stage, depending on the findings of the Check step is for implementing amendments to the ISMS, taking corrective and preventive actions before returning to the start of the cycle.

ISO 27002 is for best practices in information security management and a range of controls and guidance for most situations via high-level voluntary guidance. The standard presents requirements for ISMS design, maintenance, documentation while also serving as a certification assessment. The standard covers the areas of structure, risk assessment/treatment, security policies, organization of information security, asset management, human resources security, physical security, communications & operations management, access control, acquisition, development, maintenance of information systems, information security incident management, business continuity and compliance. (Krutz & Vines, 2010, p. 250)

ISO 27003:2017 is there for the guidance in development of an ISMS according to the paradigm introduced by 27001:2013 and features examples that aim to help achieving these requirements. ISO 27004 standard provides guidance on specification and measuring of the effectiveness of non-vendor specific information security management systems and their processes and controls to support decisions in ISMS management. ISO 27005 handles the requirements for information security risk management systems outlined in ISO 27001, whereas ISO 27006 has the guidelines that see to accreditation of organizations concerned with certification and the related registration of ISMSs. (Krutz & Vines, 2010, pp. 250–252)

IDPS solutions benefit from if an organization also has a proper incident response and handling management as a part of their daily operations. Incident response has also the benefit of raising accountability, thus covering organization's role under the GDPR and its 72-hour window for breach notifications to the affected parties. Setting up an incident response process requires the presence of at least an IDS solution, and the personnel or a process that allows personnel to act on the incidents, analysis of the events, responses if they're warranted, escalation procedures and resolution and reporting as follow-up. (Krutz & Vines, 2010, p. 259)

For incident response there are few guidelines published for example by both NIST's Special Publication 800-61 "Computer Security Incident Handling Guide, Recommendations of the National Institute of Standards and Technology" and Internet Engineering Task Force's (IETF) Site Security Handbook (RFC 2196). NIST's publication sets the process of incident response as: 1. Preparation, 2. Detection & Analysis, 3. Containment, eradication, and recovery, 4. Post-incident activity. IETF's guidelines are similarly: 1. Preparation & Planning, 2. Notification, 3. Incident identification, 4. Handling, 5. Aftermath. Both of the guidelines offer similar process but with GDPR coming, the IETF's guidelines and process is recommended due to it having the notification phase that details internal and external communication guidelines (authorities, public relations, customers). (Krutz & Vines, 2010, pp. 259–262)

3.1 Threats to Cloud Computing & Virtualization

The risk and threats to cloud computing traditional information systems are similar and comparable with risks like eavesdropping, fraud, theft, sabotage, and external attacks. The difference with cloud computing, private cloud especially is almost indistinguishable from traditional security architecture with cloud computing and virtualization only adding more layers on top or between the physical layers and zones. With public cloud comes the issues with Cloud Service Providers (CSP) and other customers residing in same cloud or virtualization platform. The customers and organizations have no way to control their neighbours or hosting circumstances so the internal processes of the customer organization that are going to employ virtualization on a public cloud require a rethink the issues of public cloud into consideration in their own production and security architectures. (Krutz & Vines, 2010, pp. 169–170)

The risks faced by CSPs from the hosting viewpoint are, according to Burton Group's "Attacking and Defending Virtual Environments" the fact that all traditional existing attacks work, hypervisors are a risk additive layer, separate systems running as virtual machines (VM) increase risk. Working from these parameters the authors of "Cloud Security: A Comprehensive Guide to Secure Cloud Computing" list several risk areas for virtualized systems: complexity of configurations, privilege escalations, dormant virtual machines, duty segregation and poor access controls. (Krutz & Vines, 2010, p. 175)

The common risks to cloud computing infrastructure are backdoors from external networks to internal networks, identity/network spoofing (e.g. MAC spoofing with WLANs or IP spoofing at TCP level), man-in-the-middle attacks where traffic between known points is either eavesdropped on or redirected to a near-identical destination that tries to pass as authentic destination which would capture the input from the legitimate user or session, social engineering where actors of an organization are used to gain access into information systems and looking for discarded information that should have been secured (optical media, discarded papers, memos, etc.). (Krutz & Vines, 2010, pp. 176–178)

The operational threats solely faced by virtualized environments and platforms are all usually either hypervisor or virtual machine or virtualized networking related. One of these risks is caused by ease and speed of deploying virtual machines, especially in an environment used for testing. This virtual machine sprawling is a risk without proper process control and monitoring for provisioning when a single unpatched VM is needed for the environment or a cluster to be compromised and thus provide an entry point when combined with inadequate access control. (Shackleford, 2013, p. 30)

Another risk factor to VMs is the lack of visibility into virtual environments with virtualized traffic routing where internal communication between hypervisors and VMs and between VMs and the hosting hypervisors is not monitored by the security infrastructure (firewalls, IDPS network or host sensors). Added on these risks comes the risk from personnel and separation of duties when there's no clear responsibility or supervising between the actors using the hosting environment with the worst case being that nobody knows who manages what and it's remedied by giving too broad responsibilities or access to single actors. (Shackleford, 2013, p. 31)

Some of the newer threats against virtualized environments are vectors where the payload is aware or can distinguish virtualization of hardware based on their characteristics, meaning that the payload either refuses to run and lays dormant or behaves differently when it detects that it is inside a deployed virtual machine, making quarantine and defensive actions such as analysis or research harder. It is not just malware that has gained protection against virtualization, malware authors and hackers can use other means such as code packers and obfuscation to hide from detection. (Shackleford, 2013, p. 32)

The newest vector of threats acting against virtualized platforms and cloud computing is malware and hackers that try to escape from the hosting guest operating system to the layer below it, meaning the hypervisor and attack the hypervisor running on physical datacentre hardware. A specific attack would then be able to sniff out the neighbours of the original guest VM and act on the hypervisor in a way that wouldn't be detected by a host-based security measure. (Shackleford, 2013, pp. 32–33)

3.2 Networks, Physical & Virtualized

Secure virtualization networking calls for considering both virtualized networking configurations based on vendor and integration of these virtualized NICs, switches and routers with the underlying physical networking. The considerations for physical networking switches is the need of planning of what will host the virtualized networking and the volumes and loads they generate that need to be handled by the underlying physical layer. Virtualized switches perform the task of carrying and segmentation of generated traffic that can include management traffic, production traffic and specialized traffic between storages or migrations. Virtualization platforms and hardware will come with physical NICs that need to be part of consideration in designing the needs for redundancy, network speed, and the segregation of virtualized traffic flows. Virtual NICs come in two forms, the first as part of the virtual machines like regular physical NICs are part of a computer or a server and the other form is that of a virtualized NICs on the hypervisor that act as links between the VMs, hypervisors and the underlying physical NICs. Physical network security devices can be leveraged to consider virtualized traffic flows if configured properly (firewalls, IDPSs). The vendors of enterprise network security devices may employ virtualized models of their physical counterparts to act as virtualized appliances that can then be joined to the virtualization vendors' platform. (Shackleford, 2013, pp. 119–120)

The similarities between physical and virtual switches is the ability to tag traffic based on VLANs so that traffic can be broken down into different broadcast domains and routing. Another common feature with physical switches is the ability to segregate traffic based on the throughput and speed based on the ports they use. Whereas virtualized switches do not offer much granularity in this category, as they might offer different configurations based on vendor for mirroring similar needs from physical switches. With virtualized networks come virtualized security considerations where the issues that need consideration for an organization are the isolation of management networks and virtual switches, monitoring capabilities, and security policies/controls. Isolation allows for a layered defence approach from information security standpoint and help the organization to harden their security environment. (Shackleford, 2013, pp. 124–126)

4 INTRODUCTION TO SECURITY ONION

Security Onion is an open source tool suite that comes with a customized Unix distribution based on Ubuntu while serving as a Network Security Monitor (NSM) with IDS capabilities and comprehensive tools for in one package. As prefaced by the “Introduction to Security Onion” page on the project’s GitHub wiki—IDPS solutions and at larger scope NSM solutions—are not a silver bullet one can buy and deploy and walk away with the belief that now they’ve security. Monitoring and tuning is an essential part to any NSM and this takes dedication and the willingness to learn from the administrators and analysts of the system. (Security Onion Solutions, 2018, p. Introduction)

As the core Security Onion offers full packet capture accomplished via Net sniff-ng, both network-based intrusion detection (NIDS) and host-based intrusion detection (HIDS) accomplished by the incorporated tools like Snort/Bro IDS/Ossec HIDS, and a suite of tools meant for analysis with the likes of Squert and Sguil. As Security Onion comes with full packet capture capabilities and retention configurations for this, this also serves as a hardware requirement for the deployment scenarios. As an example, the hardware requirements needed for a full packet capture on a small-sized corporate network link that has the average traffic of 10 Mbps directly translates to that a full day’s worth of packet captures take 108 gigabytes of storage. (Security Onion Solutions, 2018, p. Requirements)

Deployment scenarios for Security Onion are varied and the main three deployment scenarios are either standalone server that comes with sensors, or a server that gets its data from distributed sensors, and hybrid mode that relies on standalone deployment receiving additional sensor data from extraneous sources needing monitoring (e.g. service critical hosts, database servers, domain controllers). Security Onion setup and deployment supports all three types and has documentation to support all three types, and Security Onion Solutions also offers enterprise support for a price. (Security Onion Solutions, 2018, p. Introduction)

4.1 Suite Architecture & Requirements

The architecture of Security Onion tool suite is comparable to the requirements outlined earlier in Chapter 2.3 . The architecture consists of sensor(s) as data sources, management server, database(s) and a console interface(s) meant for controlling the operation of the IDPS. Security Onion comes with three deployment scenarios and the architecture of a standalone deployment is depicted below in Figure 1.

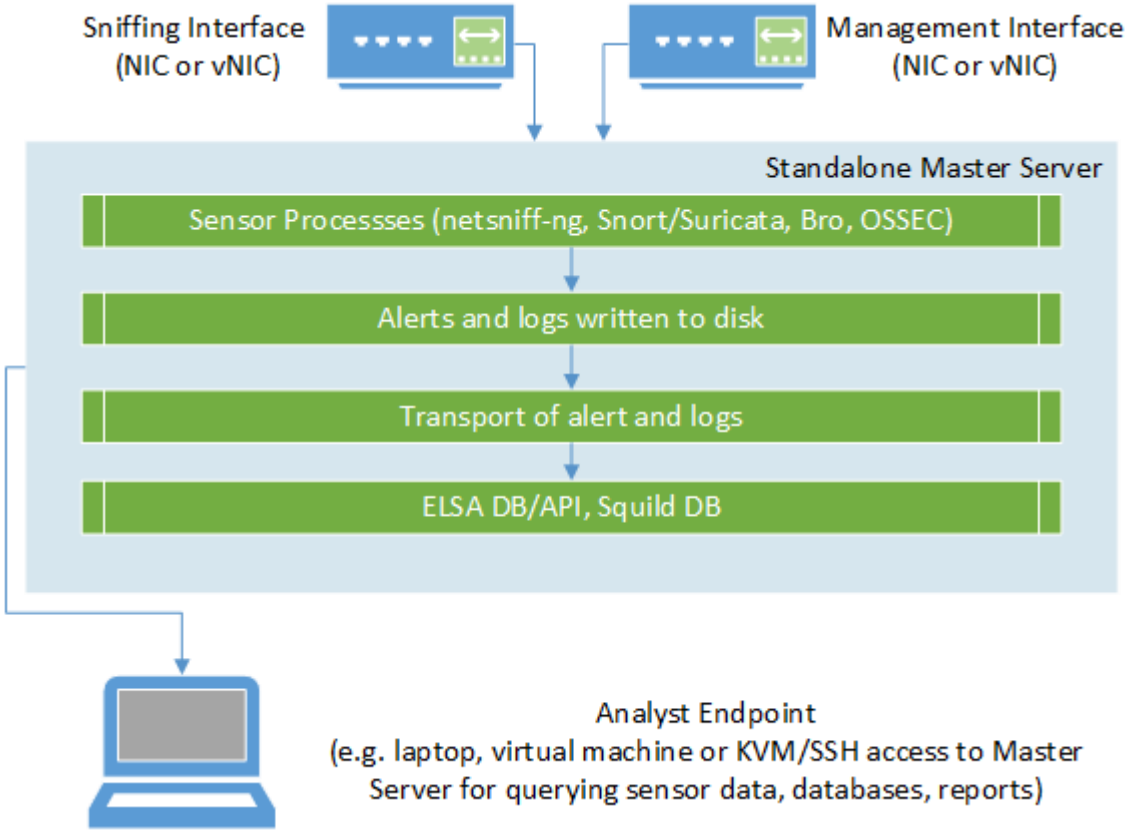

Figure 1. Process and architecture of a Standalone Master Server operation

The architecture depicted above is that of the master server when it is installed either as a physical installation on an appropriate server hardware or virtualized as part of a production environment. The process architecture is the same for both instances; the server comes with 2 network interfaces, sensors running as internal processes, log and packet capture saving on the server's hard drive, and the database storage with graphical user interfaces (GUIs) to control and query the suite and the stored sensor data from the databases. (Security Onion Solutions, 2018, p. Architecture)

The overall process itself for the sensors is covered in Figure 1. Expanding from there, the sniffing interface provides netsniff-ng (full packet capture), Snort/Suricata and Bro (IDS engine/analysis) with traffic, whereas management interface communicates with the OSSEC host-based sensor with data in the case of OSSEC agent is deployed independently. After this the sensors write the captured packets and data to their local disk if the sensors are independent and into the specified hard disk/storage of master server if standalone. The storing locations can be configured and partitioned to be separate from each other, but this increases failure points. Typical storage paths are "/nsm/sensor_data/[hostname-interface]/daily logs" for netsniff-ng, "/nsm/sensor_data/[hostname-interface]/" for Snort/Suricata, "/nsm/bro/logs" for Bro, and "/var/ossec/logs/" for OSSEC. From storage these alerts and logs are transported to ELSA/Sguil databases by pcap_agent (netsniff-ng), barnyard2 (Snort/Suricata), syslog-ng (Bro and management interface), and ossec_agent for the OSSEC. ELSA database is located with the sensors and master server hosts the Sguil database and these can be then queried by either by the master server or an analyst endpoint (virtualized or physical) hardware by using Sguil, Squert, ELSA, CapMe included in Security Onion. (Security Onion Solutions, 2018, p. Architecture)

The difference between the standalone master server deployment and the server-sensor deployment is that master server holds just the management and does no sniffing or packet capture. This is preferable if the need for monitoring is comparatively exhaustive and process heavy, e.g. extensive corporate networks with larger average throughput above hundreds of megabits per second. With comprehensive full packet capture the master server deployment has heavier hardware requirements, e.g. by Security Onion Solutions lists the need of 128-256 gigabytes of random access memory (RAM) and at least 10 central processing unit (CPU) cores for the monitoring of 500-1000 megabytes per second traffic flow. If the monitoring requirements are beyond single 500 Mbps or 1 Gbps port, then a hybrid installation or server-sensor installation would be more practical solution to avoid packet loss and competition of resources. (Security Onion Solutions, 2018, p. Hardware)

4.2 Analyst Tools

The main analyst tools in Security Onion suite are Squert, Sguil, Elsa, CapMe, Xplico. These tools rely on the data fed to the storage and databases by the deployed sensors with the storage paths detailed in Chapter 4.1 . The analyst tools rely on analysts and administrators to monitor and tune the Security Onion's configurations so that the alerts and events displayed are properly sorted and categorized to maximize monitoring efficiency and alert relevancy. As preface, Xplico is reaching end of life support from Security Onion Solutions on June 5, 2018 and will not be supported in the future distributions or deployments and can be removed after EOL and in current modern setups is disabled. Xplico itself is an extraction tool for traffic capture where the target of extraction is the application layer data contained in the full packet capture.

The main analyst tool in Security Onion is Squert, which is a web interface originally developed by Paul Halliday and a fork hosted by Security Onion to ensure stability and distribution. Squert enables the analyst to view and query the underlying Sguil database storing the even and alert data. The default interface provides contextual view to the alert database by using metadata, time presentation, weighted, and logically grouped result sets. The web interface gives the analyst access to NIDS and HIDS alerts, Bro logs and asset data from PRADS. If an event needs tracing, Squert allows the analyst to pivot from a Squert's event ID into the related full packet capture in CapMe or pivot from an IP address, port, or signature to ELSA. (Security Onion Solutions, 2018, p. Squert)

Sguil client and its database are a network security monitoring solution developed by Bamm Visscher, with slogan *"from network analysts to network analysts."* As with Squert above, Sguil offers database-oriented approach to the same data as Sguil. Sguil also allows the user to pivot to transcript/Wireshark/NetworkMiner or ELSA. Proper management and classification of events is important in ensuring the reliability of the database, e.g. autocat rules, alert notifications by email and alert retention settings for Sguil. Sguil as a database frontend allows heavy customization by analysts and requires proper rule management to be an effective monitoring solution. (Security Onion Solutions, 2018, p. Squil)

CapMe as an analyst tool works akin to Squert, a web interface that allows an analyst to pivot from Squert alert or ELSA log entry to CapMe and then access the whole transcript of the events surrounding the alert/log entry with tcpflow or Bro. CapMe also allows the analyst to download the whole packet capture itself for further analysis and works with the same credentials as the rest of the analysis tools when deployed with Security Onion. (Security Onion Solutions, 2018, p. CapMe)

ELSA is abbreviation for “Enterprise Log Search and Archive,” a three-tiered log receiver, archiver, indexer, and web frontend developed by Martin Holste. As mentioned on the project’s GitHub page, it leverages syslog-ng’s parser pattern-db to provide indexing and log normalization. With Security Onion 14.04 ELSA comes with dynamic dashboard and charting abilities to provide visibility for analysts. Security Onion Solutions provides support for best practices with ELSA queries, custom parsers, and tuning. The primary use of ELSA in Security Onion is to comb through the logs provided by the chosen NIDS engine, Bro, OSSEC, and syslog and allows the analyst to pivot into CapMe when needed. (Security Onion Solutions, 2018, p. ELSA)

As of writing and deploying Security Onion distribution’s latest stable version, 14.04 based on Ubuntu 14.04 LTS (Long Term Support), Security Onion Solutions is moving towards integrating Elastic Stack to Security Onion distribution. The aim of this integration is to add a layer to the architecture where sensors and services that store logs will in future be parsed by Logstash from where the logs will be ingested and indexed by ElasticSearch which allows analysts visibility into domain data, frequency analysis, alerts, and index management. On top of ElasticSearch will be Kibana that provides visualization and query tools like dashboards to analysts like ELSA does in the current distribution. This integration work is highly experimental and is not fit for production deployment yet, but all these components will be deployed and run as Docker images built on CentOS 7; meaning containerization of services which will help the distribution to avoid the common problem scenario where code or services run on different platforms or by different users will behave differently and in unexpected ways. (Security Onion Solutions, 2018, p. Elastic)

4.3 Data Sources

Snort and Suricata are the main data sources in Security Onion deployment as the sensors' NIDS engine that will monitor the incoming traffic and generate IDS alerts which then will be transported to their respective database. Both Snort and Suricata in Security Onion are compiled to comply and work with the PF_RING configuration which allows the use of multiple instances as workers to better handle with incoming traffic loads. The deployment of Security Onion will ask the deployment to use either one of the IDS engines. (Security Onion Solutions, 2018, p. Snort)

Bro is a companion to the other NIDS engine(s) used by Security Onion deployment and is the network analysis framework for more general analysis approach to generate logs on the activity it monitors. The logs generated by Bro are transported by syslog-ng for storing into ELSA database. The activities monitored and categorized by Bro are TCP/UDP/ICMP connections, DNS/FTP activity, HTTP requests and replies, SSL/TLS handshakes and internal Bro notices among others. Bro provides customization to analysts by allowing the use of custom scripts and intel storing with third-party integration from outside sources. (Security Onion Solutions, 2018, p. Bro)

OSSEC is the provided tool of Security Onion distribution for HIDS and as a host service will monitor all system activity like file integrity, host events and logs, root, and process monitoring. The primary use of OSSEC is to shield Security Onion master server itself from intrusion but can be also deployed as sensor service to production critical hosts. OSSEC service can perform active response and depending on rule tuning it can produce false negatives by blocking legitimate activities as potentially malicious and terminate the connections utilized by the event. OSSEC manager on the Security Onion master server is configured to support maximum number of 1024 OSSEC agents, meaning 1024 client installations reporting back to the manager on the default alert level of 5. OSSEC agents are available as server and client agents for Unix, Windows, virtual appliances and as docker container. Automated deployment of OSSEC agents is possible by third-party Auto-OSSEC. (Security Onion Solutions, 2018, p. OSSEC)

Both Sysmon and Autoruns are Microsoft applications and services that provide visibility into Windows operating system internals. Autoruns as the name implies provides visibility and monitoring into what services, programs, and drivers are run at system boot, user logins while also reporting on multiple other avenues of interaction between systems and its users. Autoruns monitors the registry and shell extensions such as toolbars and browser helper objects and context menu entries. (Security Onion Solutions, 2018, p. Autoruns)

Sysmon is abbreviation of System Monitor, a Windows system service and a device driver that can be installed onto Windows OS hosts that need monitoring and will then remain as a service to be run across system reboots. The main interest of running Sysmon as part of Security Onion distribution is its ability to monitor process creations, network connections, file integrity checking for critical host files and integrates itself with Windows Event Collection which is then reported upon either by OSSEC or by exfiltrating the logs with third-party solutions to another SIEM. (Security Onion Solutions, 2018, p. Sysmon)

Syslog-ng is the main syslog collector in Security Onion distribution and sends the logs to the ELSA database on master server and can be configured to forward the logs from Bro/OSSEC/IDS to third-party solutions or SIEMs as needed. As syslog-ng listens to port 514 on TCP/UDP for incoming syslogs from sensor hosts, Security Onion master server's Uncomplicated Firewall—shortened as "ufw"—and its configuration can be done from master server's terminal with "sudo so-allow" script which will ask the details of new rule addition and the source address needed. (Security Onion Solutions, 2018, p. Syslog)

Security Onion distribution allows for third-party integration for the data it produces and its transportation to another SIEM framework and Security Onion Solutions offers commercial support for this. Basic log forwarding from Bro and OSSEC can be done by modifying and then restarting the syslog-ng service on master-server. IDS alerts can be forwarded to external system via barnyard2 instances which require extra configuration and a service restart. Commercial support is meant for higher level third-party integration. (Security Onion Solutions, 2018, pp. Third-Party Integration)

5 CASE STUDY: SECURITY ONION DEPLOYMENT AT KAJAK DC

Prefaced in the introduction, the case study with Security Onion is done at Kajaani University of Applied Sciences' Datacentre Laboratory to serve the needs of the laboratory by being a permanent addition to the laboratory's information security measures. This case study and deployment aims to provide visibility into the teaching and production clusters' hardware. This virtualized environment and the hardware rely on the laboratory's datacentre which facilitates parts of the teaching and the student project needs of students from the business administration and engineering departments. The case study itself fulfils the requirement of author's graduation thesis for BBA degree and the documentation with the deployed service will be handed over to the datacentre administrators and project staff who may then expand and use the service for daily operations.

The outline for this case study is the deployment of Security Onion into the datacentre laboratory, the choosing of the deployment style, its configuration and creation of the documentation and a planned weeks' worth of measured traffic to see that the system works as it should and is ready for further use. This also contains the detection tuning and configuration with checking the datacentre environment against best practices published by the creators of Security Onion. The case study ends on the analysis of measured traffic and future considerations for the laboratory staff and senior students.

This case study portion expects the reader to be familiar with the basic technical knowledge of servers and networking principles and will not go in-depth with the steps made during installation nor with basic operation of the Security Onion distribution's operating system. Furthermore, to avoid NDA and confidentiality issues, this case study will generalize the environment configurations and charts, and will not publish any server names, addresses, services, devices, or manufacturers unrelated to the operation of Security Onion deployment itself.

5.1 Baseline

The baseline situation at KajakDC laboratory is that of an optical fibre connection to the Kajaani UAS itself and to the public internet outside, protected and monitored by enterprise firewall which acts as a barrier and security measure to the laboratory datacentre which constitutes of multiple server racks and cooling units.

The racks host the uninterruptible power supplies (UPS), virtualization hardware, cluster hardware, data storages, assorted server rack units ranging from 1U to 4U. These are all connected by network switches and fibre optics to facilitate the teaching and production clusters running on virtualized platforms and the student projects.

The baseline security measures rely heavily on the enterprise firewall that comes with basic antivirus and threat detection capabilities and the best practice use of virtualization and Microsoft Active Directory domain. Physical access to the datacentre hardware is regulated and administration rights along with network access is done case by case. The basic configuration can be seen below in Fig 2.

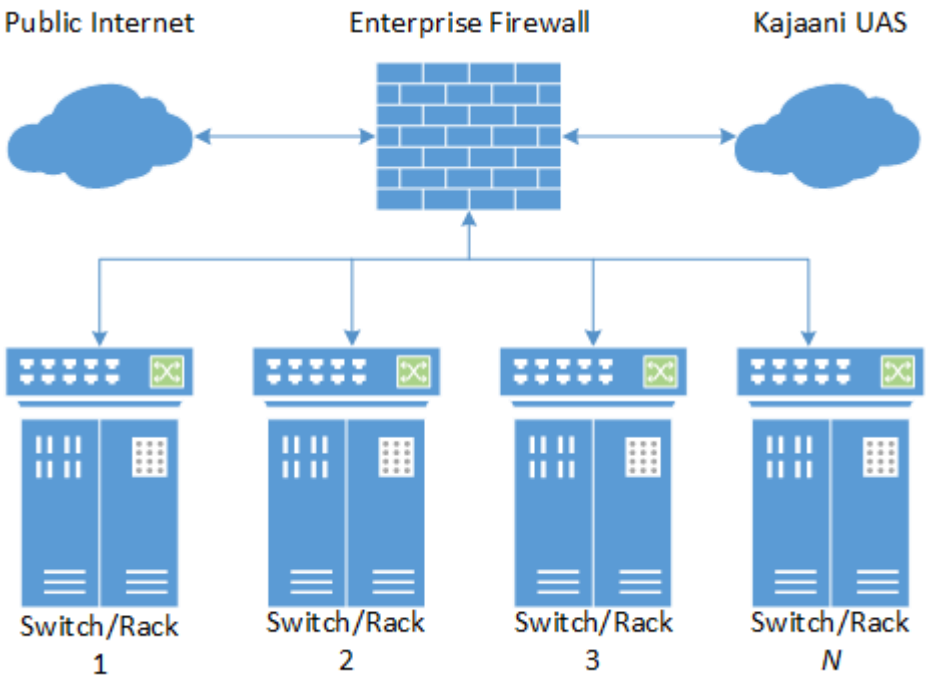

Figure 2. Baseline hardware and connection generalization of KajakDC

5.2 Plan of Deployment

The plan of the deployment for Security Onion into KajakDC laboratory must consider the hardware requirements and the traffic that will get monitored by the deployment. The primary objective is to choose the most sensible deployment type to have the best information security increase possible. This approach takes into consideration the earlier theoretical parts for physical and virtual networks in the manner of layered defence explained earlier in chapter 3.2 . Secondary objective for the deployment is to affect the current infrastructure, hardware, and resource expenditure as little as possible but still leave room for future considerations and expansion.

As considered in chapter 2, an IDPS sensor can be installed either as an inline sensor or as a passive sensor and the planned placement for Security Onion sensor in this case study is passive deployment so that the solution will monitor a copy of the network traffic that needs monitoring. Inline sensor would require hardware investments which would further require multiple install points for sensors and impact the current infrastructure more heavily than needed. With passive deployment of the sensors the master server of Security Onion can be isolated from other production and networking when it is in operation. Using standalone deployment for Security Onion keeps the sensor, management, database, and interfaces in all-in-one solution that is customizable for access and further sensor deployments in the future if needed by the laboratory.

Physical and virtualized deployment considerations lean on the needs of the laboratory and so the chosen the deployment mode is a master server standalone installation on physical hardware and on physical network connections. The physical network will set the requirement for the hardware to have two free NICs which will facilitate ports for the management network and the data source (sniffing). Virtualization of the deployment and the use of virtualized NICs and switches is considered layered defence approach, but this would require the practical portion to redesign and modify the current virtualized environment for best information security capabilities and is thus outside of the scope of this work for being disruptive.

The hardware requirements for the deployment are dependent on the amount of traffic being monitored since the plan is to deploy the solution as standalone master server installation and data is stored on the server. If an organization wants to deploy Security Onion in server-sensor mode where the master server has no sensor processes running, then the requirements would be according to Security Onion 1-4 CPU cores, 8-16 GB RAM, and 100GB-1000GB storage. For hybrid deployments organizations would have to consider these hardware requirements for every sensor they wish to deploy. Sensor installations (or agent installations for HIDS) require hardware resources similarly. Storage space needs are based on the amount of days to keep for full packet captures whereas CPU and RAM needs are based on the traffic amount monitored. (Security Onion Solutions, 2018, p. Hardware)

Monitored Traffic Amount	**CPU Cores**	**RAM Memory**	**Full Packet Capture Storage (per Day)**
0–50 Mbps	1–4 cores	8–16 GB	0–540 GB
50–500 Mbps	4–8 cores	16–128 GB	540–5400 GB
500–1000 Mbps	8–16 cores	128–256 GB	5400–10800 GB

Figure 3. Table of hardware requirements for standalone deployment with one monitoring interface.

Above you can see a table combining the hardware requirements for a standalone master server deployment-based Security Onion documentation where the requirements base on the monitored traffic. The measured traffic for this case study's data source during office hours is 5 Mbps average with peaks of 10 Mbps when measured from the data source during active teaching. Thus, the minimum hardware requirements for this case study are on the top row, with additional storage lengthening the data retention span. This case study will be thus run on a server with 4-core CPU with 16GB RAM and 1200 GB local storage. (Security Onion Solutions, 2018, p. Hardware)

5.3 Deployment

The deployment itself can be done by downloading the latest stable Security Onion ISO image, which is a 1:1 disk image containing the operating system distribution and the tools and settings. Security Onion can be installed on top of a standard Ubuntu 14.04 distribution at the time of writing this work if it is needed for quick evaluation. Since the case study will try to accommodate best practices and developer guidelines, the deployment will be made from a verified ISO image download hosted by GitHub. The installation media itself can be prepared to be either a bootable optical media or a bootable flash storage like an USB memory stick. The chosen hardware for this case study has both optical tray and USB ports but creating a re-usable USB media is easier so this case study will use tools to make a bootable USB stick based on the ISO image. (Security Onion Solutions, 2018, p. Installation)

Security Onion can be deployed either in evaluation mode or in production mode, evaluation is for deployments meant for familiarizing and evaluating the capabilities of the solution whereas production deployment will be full deployment and configuration to be a part of production infrastructure. This case study will make use of the production deployment and the prepared hardware for this is a basic 1U-size server with RAID 0 storage due to hardware restraints from serial attached storages and limited amount of disk bays. The vendor's server hardware only allows RAID 0 or 1, with an array of RAID 1 only allowing single array on the server that consists of just 2 disks, meaning single disk of local storage and this kind of configuration cannot be used for data retention.

The deployment was done following the guide from developers and adheres to the recommendations laid out in their documentation, the file system is encrypted so that the data cannot be read by plugging out the hard disks nor can it be booted without the password for it. User account and host naming follows the procedures of KajakDC infrastructure and general best practices for encryption and password strengths measured by complexity (e.g. length, lower/upper case, and special characters). (Security Onion Solutions, 2018, p. Production Deployment)

5.3.1 Suite Configuration

After basic deployment the master server has been run through the basic configuration by running “sudo soup” from terminal which aids the deploying organization go through the setup with the best practices from the developers regards to networking settings and deployment configurations (e.g. default administrative credentials, CPU allocations, chosen IDS engine, and rules). After the initial so-setup process, the enterprise firewall and routers need configuration so that the standalone deployment has proper data source to monitor, this work is done outside of the master server and with the help by hosting organization’s laboratory staff.

Post-installation customization checklist for Security Onion and the standalone master server looks something like this in this case study and are modelled after Security Onion Solutions recommendations and future considerations for the staff of laboratory staff (Security Onion Solutions, 2018, p. Post Installation):

- ☐ Check running services from terminal with “sudo service nsm status” and restart if needed with “sudo service nsm start”.
- ☐ With monitoring source active, check that the sensor is properly coping with packet load from terminal with “sudo sostat | less”.
- ☐ Configure data retention period by setting DAYSTOKEEP variable in “/etc/nsm/securityonion.conf” and the storage auto-purge/alert variables.
- ☐ Create needed analyst user accounts from terminal with “sudo nsm_server_user-add” and allow analyst connections from chosen IP addresses with “sudo so-allow”.
- ☐ Test the deployment by generating an IDS alert from terminal with “curl http://testmyids.com”

Optional configurations that this practical portion will not be using are NTP configurations, version control of “/etc” path on master server and remote desktop. Remote access will be limited to analyst endpoints only to provide isolation.

Security Onion distribution comes with a setup script which will allow the master server to automatically configure an email service for the deployment. This setup script can be invoked from terminal with “sudo so-email” and will guide the user through the setup. The automated email setup has been deployed for this case study with the added cronjob of emailing the output of “sudo sostat” to the inbox of laboratory administrators, which will provide visibility without logging into the analyst endpoints that the services are running and operating normally and that the sniffing interface does not suffer from packet loss. Alternatively, Sguil, OSSEC, Bro, and ELSA all support sending emails and can be further configured for deployments with multiple analysts and use cases. (Security Onion Solutions, 2018, p. Email)

The default firewall in Security Onion distribution will by default only allow traffic through the port 22 and SSH protocol and does this by utilizing ufw service. When deployment adds new sensor installation for example on critical hosts, the installed sensors automatically add their own firewall rules to the master server. Security Onion distribution also comes with a script that allows the administrators to configure the firewall solution for traffic through ports 22 for SSH, 4505/4506 for Salt and 7736 for Sguil. This script can be invoked from terminal with “sudo so-allow” which will then ask for input on if the new rule addition is analyst, syslog device, ossec agent or Security Onion server. If non-standard firewall configurations are needed then administrators can do their own rule additions manually by following the original documentation of ufw. (Security Onion Solutions, 2018, p. Firewall)

Security Onion for this case study will rely on the standalone master server and has been configured with all the sensor processes active since the standalone needs all the sensors running to provide efficient network security monitoring. Additional sensor installations can benefit from less resources being used by disabling different agents and processes as needed. As this case study will also see further use by the laboratory administration, there will be an installation of analyst endpoint done as part of the deployment. This also adds into security as the master server will be only accessible from analyst endpoint or physically with KVM switch. (Security Onion Solutions, 2018, p. Post Installation)

5.3.2 Data Gathering & Tuning

With the hardware requirements dictated by local storage in mind, this Security Onion's deployment was configured to retain 7 days of full packet captures to facilitate the laboratory staff the option to pivot into packet transcripts and full packet capture inspection. If the 7–day retention is not possible due to increase in monitored traffic, the master server will start to purge the local storage on master server from the oldest entries to keep ten percent of the file storage free to avoid operational disruptions from server running out of storage. The Sguil database can be purged if needed by setting the DAYSTOKEEP variable to sufficiently low value and running "sudo squil-db-purge" command from master server's terminal. (Security Onion Solutions, 2018, p. Post Installation)

As soon as Security Onion has been deployed and the sniffing interface is being fed with traffic, typically from a SPAN or TAP port from another network or security device, the master server's sensors will start capturing all the packets and the events associated with those packets. Initially as there is no way for the developers to classify what constitutes as harmless events or benign activity the event queue will start to fill with uncategorized events. This is where the monitoring part of NSM solutions becomes important. Squert, Sguil and ELSA come with auto-categorization, but the initial tuning is best done by analyst(s) that know the network topology and who can decide based on the details such as source/destination as shown in Figure 4 if it is benign or suspicious activity. (Security Onion Solutions, 2018, p. Managing Alerts)

Figure 4. Signature details of an IDS alert viewed in Squert, location/destination address retracted.

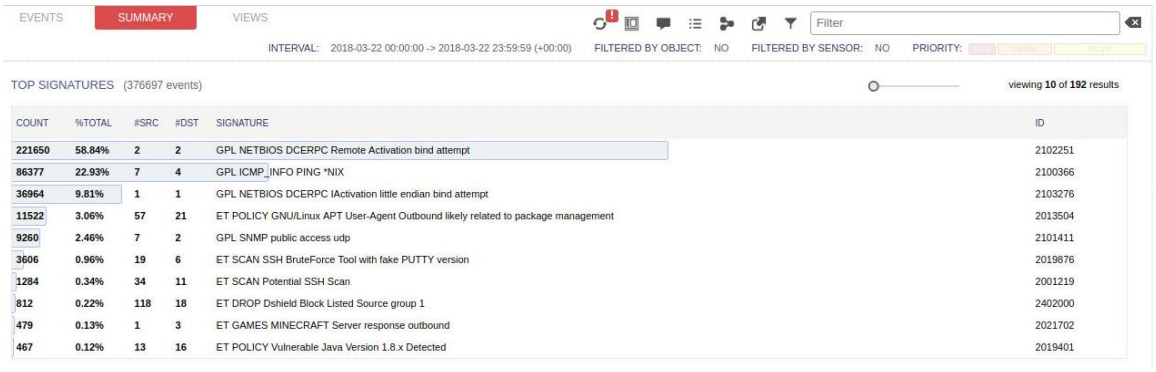

Figure 5. Top signatures of single day before any tuning

The figures of this chapter are captures from Squert interface on the master server, which rely on the Sguil database for IDS alerts. The above figure is day's statistics before any kind of tuning or categorization has been done on the Security Onion deployment. As the figure above shows, after a full day's worth of packet captures and monitoring, the uncategorized event counts reach hundreds of thousand events, even when overall traffic amount monitored is below 50 Mbps. Security Onion offers multiple ways to manage these overtly active signatures and alert counts with methods like auto-categorization, disabling signature IDs or categories, rewriting signatures themselves, and lastly signature suppression. The difference between the top ten signatures detailed in Figure 5 and Figure 6 is signature suppression that allows analysts to suppress signals based on rules such as destination or source addresses. Analyst(s) that know the organization network can suppress benign activity like network services broadcasting between gateways and domain-critical servers. With completely disabling signatures like "SMNP public access udp" as shown in Figure 4 comes the risk of lowering security when a benign protocol can also be used for malicious activity, thus it is wiser to suppress the events. (Security Onion Solutions, 2018, p. Managing Alerts)

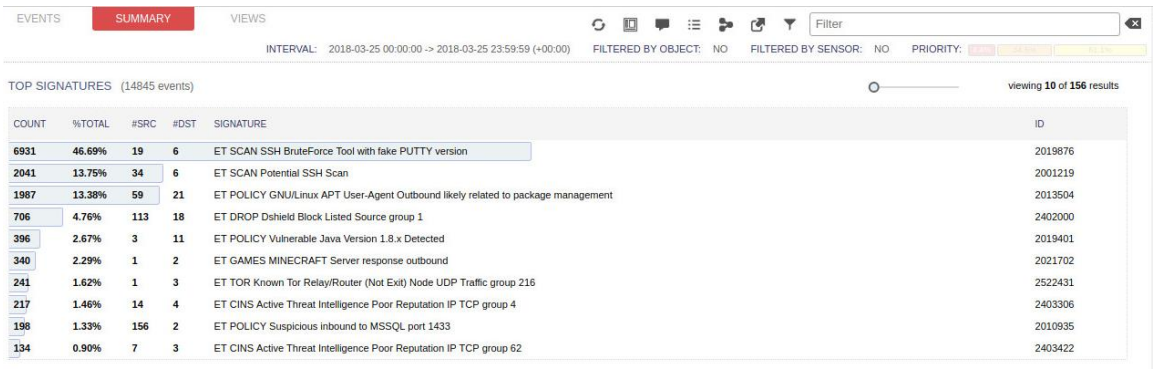

Figure 6. Top signatures of single day after initial tuning

5.3.3 Security Rules & MySQL Tuning

The rulesets for the chosen IDS engine during deployment are automatic with the deployment setup and are specified in “/etc/nsm/pulledpork/pulledpork.conf”. Both Snort and Suricata have multiple ruleset providers with options available and within the deployment setup the chosen IDS engine will be configured to use a free ruleset. Changing the rulesets is trivial configuration change and then reloading the rules. By default, Security Onion will try to download new rules once per day.

One of these providers is Emerging Threats, which provides the free ruleset “ET Open” and Proofpoint which provides “ET Pro” ruleset that has a license fee per sensor where applied. Both rulesets are optimized for Suricata but are also available for Snort. Snort has its own optimized rulesets, Snort Community, Snort Registered and Snort Subscriber (Talos). Both Community and Registered rulesets are free, where community ruleset is open community work and registered version is the same as Snort Subscriber ruleset provided by Cisco’s Talos workgroup but come with thirty-day delay. Snort Subscriber as with ET Pro comes with license fee per sensor where applied and rules are available as soon as they’re released. (Security Onion Solutions, 2018, p. Rules)

As Security Onion comes with MySQL to provide the database functionalities the database operation can be tuned for better performance to consider the monitoring needs of Security Onion. The optimization is not installed with Security Onion but can be installed from terminal with “sudo apt update & sudo apt install mysqltuner” and then run with “sudo mysqltuner”. By default, the tuner comes with initial recommendations and according to Security Onion Solutions the most common variables needing tune are open-files-limit, table_cache, key_buffer, and max_connections. For bigger Sguil databases where data retention length affects the size and load times the developers recommend adjusting check_for_crashed_tables variable and if data retention period is longer than thirty days then table_definition_cache should be raised from the default value of 400. (Security Onion Solutions, 2018, p. MySQL Tuning)

5.3.4 Analyst VM

For analyst duties and tasks, the case study has created an analyst virtual machine for the laboratory staff, complete with separate user credentials and a way to connect to the master server's Squert, Sguil and ELSA from outside, without requiring physical access. This of course requires enterprise firewall configuration and opening the ufw firewall at master server for analyst connection from a static IP address.

If further analyst virtual machines are needed, then they can be created by laboratory staff on top of any Ubuntu 14.04 LTS distribution image. Before installing Security Onion packages, users need to configure MySQL to not prompt for root password with "echo "debconf debconf/frontend select noninteractive" | sudo debconf-set-selections", clean apt list repositories with "sudo rm -rf /var/lib/apt/lists/*" and "sudo apt-get update". After this users need to add the Security Onion stable repository with "sudo apt-get -y install software-properties-common", "sudo add-apt-repository -y ppa:securityonion/stable", "sudo apt-get update" and install the securityonion-all metapackage with "sudo apt-get -y install securityonion-all syslog-ng-core" and run "sudo sosetup". (Security Onion Solutions, 2018, p. Installing on Ubuntu)

Analysts can connect to the master server's Squil database by launching the analyst virtual machine and then opening any of the three web interfaces, Squert, Squild, or ELSA and after the chromium browser is launched, replace localhost in the URL address with the master server's IP address. Connection to the master server will then the user to supply the user credentials for analysts. After credential verification the analyst can view the master server's databases and frontends in real-time. (Security Onion Solutions, 2018, p. Connecting to Squil)

The analyst virtual machine when installed on top of preferred Ubuntu 14.04 LTS will offer analysts local copies of Wireshark, NetworkMiner and customized Squil client. If more comprehensive forensics and reverse-engineering is needed, then the developers of Security Onion recommend the "SIFT Workstation VM" by SANS and the toolkit is available both as virtual machine appliance and as installation on top of Ubuntu 16.04 LTS. (Security Onion Solutions, 2018, p. Analyst VM)

6 CONCLUSIONS & FUTURE CONSIDERATIONS

For the conclusions of this case study, the answers for the research questions set in Chapter 1.2 are as follows: the key functions required of an IDS as described by NIST in Chapter 2.1 are fulfilled by Security Onion as it stores full packet capture from the monitored interface and events attached or generated from the monitored traffic. Security Onion as an IDS can notify administrators of important observed events and performance statistics via automated daily emails and reports. The optimal method has been described in Chapter 5 for the datacentre lab and the hardware chosen and the choice was done considering the time and resource constraints with 7 days of data retention for analysis. Security Onion as the chosen solution for the case study does improve the security posture of the datacentre laboratory on top of the already existing security processes but also requires an administrator to review the events either daily or weekly to improve the security according to the principles of network security monitoring.

Regards to the planning of this thesis the original table of contents contained more chapters but during the writing process the topics originally considered to be important like checking the datacentre laboratory for best practices in virtualization security and the re-adjustment or complete rework of virtualized production network of the laboratory turned out to go beyond the scope of this thesis and unrelated to the case study and deployment of Security Onion. Another consideration and adjustment done during the thesis process was the selection of choosing to use either physical hardware or virtualized production environment.

Future considerations for the information security aspect of the datacentre laboratory that can be undertaken by the following datacentre students include the topics of expanding the initial deployment of Security Onion to cover production environment critical hosts such as database and domain servers, virtualization security hardening, penetration testing, or even upgrading the Security Onion tool distribution to the upcoming Elastic stack and architecture after it has been finalized by Security Onion Solutions.

7 REFERENCES

Berman, A. E., & Dorrier, J. (2016, March 22). *Technology Feels Like It's Accelerating — Because It Actually Is*. Retrieved from Singularity Hub: https://singularityhub.com/2016/03/22/technology-feels-like-its-accelerating-because-it-actually-is/

Gartner. (2017, May 3). *Gartner Says Organizations Are Unprepared for the 2018 European Data Protection Regulation*. Retrieved from Gartner Newsroom: https://www.gartner.com/newsroom/id/3701117

Kini, R. B. (2013, October 6). *How Technology Knowledge Has Become A Basic Requirement*. Retrieved from NWI Times: http://www.nwitimes.com/niche/inbusiness/how-technology-knowledge-has-become-a-basic-requirement/article_f348a84e-5c82-5b77-9c98-e3744d63d9c7.html

Krutz, R. L., & Vines, R. D. (2010). *Cloud Security: A Comprehensive Guide to Secure Cloud Computing.* Wiley Publishing Inc.

Lyon, G. (2016, November 10). *Nmap Network Scanning: Legal Issues*. Retrieved from Nmap: The Network Mapper: https://nmap.org/book/legal-issues.html

Perlin, M. (2012, September 17). *Downtime, Outages and Failures - Understanding Their True Costs*. Retrieved from Evolven: https://www.evolven.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html

Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS).* National Institute of Standards and Technology.

Security Onion Solutions. (2018, March 15). *Analyst VM.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Analyst-VM

Security Onion Solutions. (2018, March 15). *Architecture.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Architecture

Security Onion Solutions. (2018, March 15). *Autoruns.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Autoruns

Security Onion Solutions. (2018, March 15). *Bro.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Bro

Security Onion Solutions. (2018, March 15). *CapMe.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/CapMe

Security Onion Solutions. (2018, March 15). *Connecting to Squil.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/ConnectingtoSguil

Security Onion Solutions. (2018, March 15). *Elastic.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Elastic

Security Onion Solutions. (2018, March 15). *ELSA.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/ELSA

Security Onion Solutions. (2018, March 15). *Email.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Email

Security Onion Solutions. (2018, March 15). *Firewall.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Firewall

Security Onion Solutions. (2018, March 15). *Hardware.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Hardware
Security Onion Solutions. (2018, March 15). *Installation.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Installation
Security Onion Solutions. (2018, March 15). *Installing on Ubuntu.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/InstallingOnUbuntu
Security Onion Solutions. (2018, March 15). *Introduction To Security Onion.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/IntroductionToSecurityOnion
Security Onion Solutions. (2018, March 15). *Managing Alerts.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/ManagingAlerts
Security Onion Solutions. (2018, March 15). *MySQL Tuning.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/MySQLTuning
Security Onion Solutions. (2018, March 15). *OSSEC.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/OSSEC
Security Onion Solutions. (2018, March 15). *Post-Installation.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/PostInstallation
Security Onion Solutions. (2018, March 15). *Production Deployment.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/ProductionDeployment
Security Onion Solutions. (2018, March 15). *Rules.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Rules
Security Onion Solutions. (2018, March 15). *Snort.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Snort
Security Onion Solutions. (2018, March 15). *Squert.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Squert
Security Onion Solutions. (2018, March 15). *Squil.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Sguil
Security Onion Solutions. (2018, March 15). *Syslog.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Syslog
Security Onion Solutions. (2018, March 15). *Sysmon.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/Sysmon
Security Onion Solutions. (2018, March 15). *Third-Party Integration.* Retrieved from Github: https://github.com/Security-Onion-Solutions/security-onion/wiki/ThirdPartyIntegration
Shackleford, D. (2013). *Virtualization Security: Protecting Virtualized Environments (1).* Sybex.
Taylor, A., Alexander, D., Finch, A., & Sutton, D. (2013). *Information Security Management Principles.* BCS Learning and Development Ltd.
TechUK. (2016, June). *Policy Issues for European Data Centres.* Retrieved from TechUK.org: https://www.techuk.org/images/Policy_Priorities_and_Actions_1606.pdf