Key factors in building a Secure Web Gateway

A thesis

submitted in partial fulfillment

of the requirements for the degree

of

Master of Cyber Security

at

The University of Waikato

by

Jeffrey Yeh

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2017

Abstract

A Secure Web Gateway, according to Gartner's definition, is a solution that provides URL filtering, malicious code detection and blocking, and application controls for cloud applications to filter out objectionable software/malware in outbound Internet traffic generated by end-user devices and has the capability to enforce corporate policy and regulatory compliance [1]. Its predecessor - Web proxy - has been around since the beginning of the Web and evolved to meet the needs of a fast-changing Web ecosystem. Traditionally, Web proxy servers have been used to fulfil the following requirements: 1) Enable several machines to share a single Internet connection; 2) Improve Web performance and save bandwidth by caching repeatedly-accessed content locally; 3) Provide a basic URL filtering capability. However, these capabilities are no longer sufficient to meet the requirements of today's Web ecosystem. Firstly, with the development of Network Address Translation in the late 1990s, the needs to use proxy servers to share an Internet connection has been superseded. Secondly, caching does not improve performance much for mobile clients, and mobile traffic volume has already exceeded that of desktop's [2, 3]. Thirdly, a Web content filter based on using a URL database cannot keep up with the growth of Internet traffic [4]. In addition, it has become difficult to detect and stop threats such as Botnet and Advanced Persistent Threat [5] because of: 1) The polymorphic characteristics of the threats; 2) The increasing use of encryption on the Web; 3) The increase in threats targeting end-users - the weakest link; 4) The increasing need to use a variety of end-user devices from multiple locations such as the BRING YOUR OWN DEVICE (BYOD) policy requirement. Hence, there is an imminent need to evolve from the current Web proxy solution to a Secure Web Gateway solution. This research provides a categorisation of the key factors in building a Secure Web Gateway, proposes a reference design and architecture, a practical implementation for a home vDSL connection and finally, a testing framework that can be used to evaluate the effectiveness of a Secure Web Gateway deployment.

Acknowledgements

I would like to thank my wife, Ally, for her support during my study over the last two years. As a father of three and full-time worker, without her sacrifices, I would not have been able to complete my Master's study. I would also like to thank my supervisor, Dr Ryan Ko, for his encouragement and guidance. He taught me about research methodology from ground up and how to apply it to solve a real-world problem. This journey has been very rewarding and has created a whole new vision for my life. Thanks also to my managers at work, Chris and Thomas, for allowing me to take time off for study and thanks also to my company for sponsoring my study. Finally, my thanks go to my mum and dad for their love and support throughout the study.

Contents

List of Figures

List of Tables

Glossary

Advanced Persistent Threat According to US National Institute of Standards and Technology (NIST), APTs have the following characteristics: 1) Attackers usually hold sophisticated levels of expertise and significant resources; 2) Attackers achieve their objectives by using various attack vectors such as physical, deception and cyber; 3) The objectives of attacks are to impede critical aspects of an organisation or exfiltrate information or prepare themselves to execute these objectives in the future; 4) APT attacks usually last a long period accompanied with repetitive attempts; 5) APT attacks will adapt to defenders' countermeasures; 6) APT attacks usually maintain the level of access required to carry out their objectives [6].

Botnet A network of computers compromised by infestation of malicious software and controlled by malicious users without their owners' knowledge.

Bring Your Own Device BYOD policy is a policy that enables employees to use their personally-owned devices such as tablets, smartphones or laptops to access privileged information and applications in the workplace.

Command and Control (C2) This term is commonly used in computer security and the context of cyber warfare. It refers to the ability the attacker has to influence a group of compromised computers under his/her control.

Cross-site Scripting XSS is a technique used by attackers to inject malicious client-side script into a trusted website. Hence, the malicious code is run

by unsuspecting users. This usually happens when a vulnerable web application does not validate or encode the input it takes from users and use that input directly within the output it sends to users.

Data Loss Prevention DLP is a business strategy to prevent users from transferring sensitive or critical information outside the corporate network either inadvertently or deliberately.

Domain Generation Algorithm DGAs is commonly used by malware to generate a large number of domain names that can be used by infected computers to access their command and control servers. Malware that depends on a pre-defined list of domain names or IP addresses can be blocked quickly so that DGA is used to circumvent the detection.

Fast-Flux Fast-Flux is a technique to circumvent security controls by assigning multiple (hundreds or even thousands) IP addresses to a domain name. These IP addresses are changing with an extreme frequency, using a combination of round-robin IP addresses and a very short Time-To-Live (TTL) for DNS records. The goals are not only to hide the websites that used to download malware but also to ensure that compromised systems controlled by attackers have the best possible bandwidth and service availability [7].

Intel Software Guard Extensions SGX is a set of new CPU instructions that allows user-level code to be put into a private region of memory, called enclave that is protected from other processes running at higher privilege levels such as processes run by the computer administrators or malware.

Internet Content Adaptation Protocol ICAP as specified in RFC 3507 is a lightweight HTTP-like protocol that has been used to extend the capabilities of transparent proxy servers. The proxy server first accepts and holds the connection to be inspected by another solution such as DLP or Virus Scanning. Then the proxy server uses ICAP to pass the request to the other solution for inspection, and the other solution returns its

response via ICAP. Depending on the response the proxy server receives, it will then either forward or reject the request [8] .

Internet of Things IoT are smart devices equipped with computer software and network connectivity that allows these devices to exchange and collect data.

Man In The Middle MITM refers to an attack where the attacker covertly relays and possibly alters the communication between two parties without their knowledge in the context of computer security and cryptography.

Open Source Host-based Intrusion Detection System OSSEC is a free host-based intrusion detection (HIDS) system that runs on Windows, MacOS, Solaris, FreeBSD, Linux, and OpenBSD. OSSEC offers the following functionalities: 1) correlation and analysis engine, 2) log analysis, 3) file integrity checking, 4) Windows registry monitoring, 5) centralised policy enforcement, 6) rootkit detection, and 7) real-time alerting and active response.

Open-source intelligence OSINT is an intelligence derived legally and ethically from publicly available information such as magazines, television, radio, computer-based information, and newspapers. Nowadays, OSINT involves collecting information about a subject from either free or paid Internet sources.

Point-to-Point Protocol over Ethernet PPPoE is a network protocol allows tunnelling network traffic to the ISP's IP network by encapsulating PPP frames inside Ethernet frames. PPP facilities authenticate users with a username and password through the use of PAP or CHAP protocol.

Remote Access Tool RAT can be used by system administrators to legitimately control or access a computer remotely. Malicious users can also use it to control the system without users' knowledge, and when it is used

with malicious intents, it is known as Remote Access Trojan. A RAT can typically perform the following operations remotely: 1) screen and camera capture, 2) file access, 3) registry management, 4) code execution, 5) password sniffing, and 6) key logging.

Security Information and Event Management Server SIEM is a software system that collects event data generated by network infrastructure, security devices, applications, and systems to correlate and analyse the events with contextual data about assets, users, vulnerabilities, and threats [9].

Server Name Indication SNI is a TLS extension that enables clients to indicate which hostname it is trying to connect to at the beginning of the SSL handshaking process.

Social Engineering Social-Engineer.org defines social engineering as that "any act that influences a person to take an action that may or may not be in their best interest" [10]. In the context of cyber security, it usually refers to getting the target to execute malware or obtaining sensitive information.

Spear Phishing is a type of scam email that only targets a small group of selected people and trick users to either click on a link to a malicious website or download an attachment that contains a malware but disguised itself as a benign file.

SQL Injection SQL Injection describes a web-based attack in which attackers insert an SQL query into the input data to the web application. Through this kind of attack, attackers can read, write, modify, and execute administrative operations on the database.

Tor Anonymity Network TOR is a free software that connects a group of volunteer-operated servers together and allows its users to connect through a series of virtual tunnels rather than making a direct connection to destinations. By doing so, users can reach otherwise blocked content

or destinations and circumvent the Internet censorship. TOR is used by normal people on the Internet to improve the privacy and security of their Internet browsing, but also used by criminals to hide their traces.

Watering Hole Attack A Watering Hole Attack starts by infecting websites that are trusted by the target and frequently visited with malware. The delivery of the malware is accomplished when the target visits the infected websites.

Web Cache Communication Protocol WCCP is a proprietary protocol developed by Cisco that allows some routers and switches to transparently redirect various traffic such as HTTP, TCP, UDP. WCCPv1 only supports the redirection of HTTP traffic, and WCCPv2 supports for other traffic.

Zero-day Exploits are undisclosed software vulnerabilities that hackers can exploit to adversely affect software programmes, data or networks. The term zero-day is used because it is unknown to the vendor, leaving the vendor with zero to create patches or to advise workarounds to mitigate the impact.

Chapter 1

Introduction

1.1 Background

A Web proxy is a dedicated appliance or computer software system acting as an intermediary between an end-user device and the Web. In the early 1990s, when most Internet connections were still on dial-up connections, Web proxy was mainly used for sharing one Internet connection with multiple machines on the same network. However, with the advent of broadband connections and the use of Network Address Translation (NAT) on broadband routers, this requirement was superseded. In late 1990s and early 2000s, due to the exponential growth of the Web, and the speed of Internet connection being unable to keep up with demands, the focus of the Web proxy was shifted to improve Internet performance and save bandwidth. Much research was conducted on the use of a Web proxy cache to improve performance in the following three ways: 1) Increasing the Web proxy cache hit ratio to reduce the user-perceived latency of accessing the Web; 2) Reducing the network load as the content could be served locally from the Web proxy cache; 3) Reducing the Web server load as the content was served by the Web proxy cache that would otherwise be served by the Web server [11, 12]. In recent years, there have been ongoing research interests in improving the performance of the Web, such as Google's Flywheel project [13], but it is more specifically for mobile devices due to the high costs of mobile data, and throttling of Internet connections for mobile devices. However, although data usage on average was reduced by 58%,

the Page Load Time (PLT) increased by 6%. In [2], Vesuna et al. concluded that caching does not improve PLT significantly for mobile devices because of the slow CPU speeds and difficulty in caching items on the critical path.

Today, due to increasing Internet-borne threats, the growing use of cloud-based applications, Internet policy compliance requirements, and the trend toward Bring Your Own Device policy; there is an enormous paradigm shift from a performance-oriented Web proxy solution to a Secure Web Gateway solution [14, 15]. In [14], it forecasts that the market for the Secure Web Gateway will grow at a Compound Annual Growth Rate (CAGR) of 20.5% from US$2.20 billion in 2015 to US$5.60 billion by 2020.

1.1.1 The current Web Ecosystem

The current Web Ecosystem has changed dramatically from a simple, static, text-based Web to the current dynamic, multimedia Web with over 25% of the world's population connected to it [16]. Unfortunately, the growth and dynamic of the Web has been accompanied by the evolution of web-based crime. This has not only resulted in the development of new types of crime, but also new techniques of committing old crimes. In the ENISA threat landscape report, the top 15 threats present on the Web are [17]:

1. Malicious Code: Worms/Trojans
2. Web-based Attacks
3. Web Application Attacks/Injection Attacks
4. Botnets
5. Denial of Service
6. Spam
7. Phishing
8. Exploit Kits
9. Data Breaches

10. Physical Damage/Theft/Loss
11. Insider Threat
12. Information Leakage
13. Identity Theft/Fraud
14. Cyber Espionage
15. Ransomware/Rogueware/Scareware

These threats are increasingly targeted at the users - the weakest link and make it difficult to detect and stop. On top of this, there is also much content on the Web that may be considered objectionable because of policy compliance or regulatory requirements.

1.1.2 Terminology

Many terms are used to describe the functions of the Secure Web Gateway or the market segment for particular products. Some terms are interchangeable with others. In additional to Secure Web Gateway, the following terms are often used: content-filtering software, web filtering software, content-control software, content-censoring software, content-blocking software, accountability software, and parental control software. "Content-censoring software" is a term often used by those critical of such software. Parental-control software is specifically designed for parents to monitor and control Internet access by their children. Accountability software is mainly used for Internet usage reporting rather than filtering or blocking, although some can be configured to do so. Other terms used in this thesis are described in the Glossary.

1.1.3 Secure Web Gateway vs. Web Proxy vs. Next-Generation Firewall

Based on author's empirical evidence as an IT infrastructure engineer with more than ten years of industry experiences, a Secure Web Gateway is a solution built upon Web Proxy with an emphasis on providing security func-

tions. Web Proxy, on the other hand, is predominantly used to provide content caching functionality and improve user-perceived Internet performance. The Next-Generation Firewall (NGFW), as the name suggests, is the successor to the traditional network firewall. In addition to standard firewall capability, the NGFW proactive in blocking new threats, such as Botnet and targeted attacks; providing wire-speed scanning that performs deep traffic analysis to identify applications and enforce network security policies at the application layer [18]. A Secure Web Gateway is more user-centric focusing on enforcing outbound user access control and inbound malware prevention through integrated URL filtering and threat detection, whereas an NGFW provides more network-based monitoring. NGFW does not terminate and intercept traffic and it is a stream-based technology, monitoring traffic as it passes by. It is more effective in detecting and stopping threats utilising non-web-based protocols, but is generally lack of ability to perform SSL scanning and in-depth URL analysis and categorisation. However, in [19], Musich suggests that in the long term, the line between a Secure Web Gateway and an NGFW will become blurred as vendors continue to integrate and create overarching security platforms.

1.1.4 Top Market Drivers for Secure Web Gateway

The market drivers continue to be strong for Secure Web Gateways. In [20], Joshua Mittler forecasts the following top drivers in the Secure Web Gateway market, which will have a medium to high market impact until 2020.

Productivity Management

A Secure Web Gateway provides fine-grained access control over some or all functions of Web applications based on users' roles. It can also provide control over the use of social media and web browsing not related to work efforts during work hours, thus increasing productivity.

Regulatory compliance requirements

Children's Internet Protection Act (CIPA) expects libraries, schools and other public institutions that accept certain federal funding to provide filtered Internet use to protect minors. A Secure Web Gateway can achieve compliance by blocking objectionable materials through the use of URL filtering, management of web categories by content type, restraint of keyword searches, and limits on application and file sharing. A Secure Web Gateway can also protect end-users against web-based attacks such as SQL Injection or Cross-site Scripting. Companies that wish to pursue Payment Card Industry Data Security Standard (PCI-DSS) compliant can benefit from this protection.

Complex malware delivery through web browsing

Web browsers are popular vehicles through which hackers can deliver malware to users. A Secure Web Gateway offers a layer of security between users and the Internet and reduces the chance of users encountering malware by restricting access to social media and networking sites, wikis, blogs, online file-sharing applications and other high-risk websites/applications.

Mobile and cloud application access controls

It has become commonplace for organisations to allow their users to bring their own technology or devices to the workplace and use them to connect to corporate networks and access corporate data. This improves users' productivity but presents yet another mean by which malware can be delivered especially through the wireless and mobile phone networks. The other category in this driver is the increasing deployment of cloud-based service. This is the primary driver in the growing adoption of cloud-based Secure Web Gateway deployments [15].

Web/Internet usage reporting

A Secure Web Gateway provides insight into both real-time and historical Web traffic usage. This is an important factor in meeting compliance and regulatory requirements, establishing network activity baselines, offering data

loss prevention capability and enforcing acceptable corporate usage guidelines. These metrics, collected over time, become important data for detecting and blocking complex malware such as Advanced Persistent Threat [21].

1.2 Motivation

Web proxy has been around since the beginning of the Web and evolved to meet the needs of a fast-changing Web ecosystem. Traditionally, Web proxy was used to fulfil these needs by: 1) Enabling several machines to share a single Internet connection; 2) Improving Web performance and save bandwidth by caching repeatedly-accessed content locally; 3) Providing basic URL filtering. However, this is no longer sufficient to meet the requirements of the present Web ecosystem. Firstly, with the development of Network Address Translation in the late 1990s, the need for proxy servers to share an Internet connection was superseded. Secondly, caching does not significantly improve performance for mobile clients, and the mobile traffic volume has already exceeded that of desktops [2, 3]. Thirdly, a Web content filter based on URL databases cannot keep up with the growth in Internet traffic [4]. In addition, it has become difficult to detect and stop threats such as Botnets and Advanced Persistent Threats [5] because of: 1) The polymorphic characteristics of the threats; 2) The increasing use of encryption on the Web; 3) Increased threats targeting end-users - the weakest link; 4) The growing need to use a variety of end-user devices in multiple locations such as the BRING YOUR OWN DEVICE (BYOD) requirements. Hence, there is an imminent need to move from the previous Web proxy solution to a Secure Web Gateway solution. Currently, there are many expensive solutions on the market [22, 23, 24]. However, there is a scarcity of literature and research on the key factors required to build an effective solution to mitigate the risks mentioned above, particularly in the Home and small business environment.

1.3 Research Goal

1.3.1 Objective

The objective of this research was to categorise the key factors required to build an effective Secure Web Gateway to protect a network of any size and provide a reference architecture for implementing a Secure Web Gateway that can be either on-site or in the cloud. The ultimate goal is to improve the security of the Web and help organisations to take back the control of the Internet.

1.3.2 Scope

The scope of the research is limited to categorising the following key factors with a specific emphasis on protecting **outbound** traffic from end user devices.

- Key factors required to effectively filter traffic by controlling the end-points, network services and rendezvous services involved in providing access to Internet content and the key factors required to evaluate the effectiveness of different methods based on the following criteria: scope, granularity, efficacy, and security.

- Key factors required to implement Data Loss Prevention (DLP) and use the following seven major analysis techniques to detect data leakage: 1) Database Fingerprinting; 2) Partial Document Matching; 3) Rule-Base/Regular Expression; 4) Exact File Matching; 5) Statistical Analysis; 6) Categories; 7) Conceptual/Lexicon.

- Key factors required to detect and block Advanced Persistent Threat (APT) by implementing controls to break the lifecycle of an APT: 1) Reconnaissance and weaponization phase; 2) Delivery phase; 3)Initial intrusion phase; 4) Command and control phase; 5) Lateral movement phase, and 6) Data exfiltration phase.

- Key factors required to inspect SSL encrypted traffic by using Man In The Middle (MITM), an Adaptive Security Model, and Server Name

Indication (SNI) filtering. It also discusses two approaches, BlindBox HTTPs and Privacy Preserving Inspection, for inspecting HTTPs traffic without compromising the security and privacy of the communication.

- Key factors required in implementing privacy-preserving monitoring. Privacy is an important factor in the successful implementation of a Secure Web Gateway in the workplace. It is a dilemma as employers need to monitor employees' Internet usage to protect business interests, while employees demand the right to privacy in the workplace. Also, covert and excessive monitoring has a negative impact on productivity. This research reviews literature in the relevant areas and provides guidelines for balancing the interests of both sides.

1.3.3 Key Contributions

This research not only provides a categorisation of key factors in building a Secure Web Gateway, but also a practical implementation model. In addition, it provides a framework for evaluating the effectiveness of an existing or new Secure Web Gateway deployment. It also discusses the issues and challenges that may be encountered in deploying a Secure Web Gateway.

1.4 Outline

This research began with the goal of identifying key factors in building a Secure Web Gateway and recognised that the investment in Cyber Security is long overdue despite the continued rise of cyber-crime [25, 26]. This is partly due to a lack of knowledge, but also to the high cost of an enterprise solution. A Secure Web Gateway is the first defence for an organisation as it protects the outbound traffic generated by humans which are often the weakest link in Cyber Security. Chapter 2 discusses current research on the factors related to building a Secure Web Gateway. These include content filtering, data loss prevention, advanced persistent threat, SSL filtering and privacy factor. Chapter 3 discusses the process that was undertaken to solve the aforementioned problems. This includes a reference architecture based

on layered defence and the key factors identified in building a Secure Web Gateway. Chapter 4 discusses the results of URL Filter Testing, False-Positive Testing, Exploits Testing and DLP Testing that were used to validate the effectiveness of the proposed design and architecture. Chapter 5 provides a summary of the gaps and challenges identified after implementing and testing the Secure Web Gateway using a residential vDSL connection. Chapter 6 summarises key outcomes and values delivered by this research and suggests potential future lines of research.

Chapter 2

Literature Review

This chapter discusses the existing literature on the following factors relating to building a Secure Web Gateway:

1. Content filtering: This is the ability to classify the different resources on the Web accurately;

2. Data loss prevention: This is the ability to prevent the leakage of critical business data;

3. APT detection and prevention: This is the ability to detect and defend against advanced persistent threats;

4. SSL filtering: This is the ability to inspect SSL encrypted data without breaking the trust and integrity of the Internet; and

5. Privacy preservation: This is a non-technical factor but a key to avoid unnecessary risk of legal liability and regulatory compliance when implementing a Secure Web Gateway in the workplace.

2.1 Content Filtering Factors

Generally speaking, content filtering is achieved by controlling the components involved in providing access to Internet content, services or endpoints. In [27], Barnes et al. consider the following elements are commonly involved in content filtering:

1. Server and Client Endpoints: These are applications running on computers systems participating in the communication. For example, a web browser running on a computer is an endpoint.

2. Network Services: These are services enabling communication between endpoints. For example, the network routing protocol is required to exchange packets between the endpoints

3. Rendezvous Services: Server or client endpoints use rendezvous services to identify other endpoints on a network. For instance, a domain name system is used to resolve a "human-friendly" name to an IP address, and an SIP proxy is used to identify an IP phone.

2.1.1 Evaluation Factors

To evaluate the technical implications of different filtering methods, they are compared based on the following four criteria: scope, granularity, efficacy, and security. Scope refers to the extent of the impact as a result of blocking. A system is perceived as less objectionable by users if the extent of the impact is as narrow as possible while still being effective. Granularity refers to the specificity of the filtering. A system is perceived as less objectionable if it is highly granular and does not cause any collateral damage. Efficacy refers the effectiveness in preventing users from circumventing the filtering imposed by the policy setting entity. Security refers to the ability to preserve the integrity and trust of Internet protocols while still being able to provide an effective filtering capability.

2.1.2 Network-Based Filtering

Network-Based Filtering inspects traffic as it travels through the network. Based on the characteristics or the content of a communication, the system decides whether it should be blocked or allowed to pass through. For example, Web filtering devices can compare the requested URL to a blacklist or whitelist database to decide whether to allow the request to go through or not. Cloud

Security Alliance suggests the following features should exist in Web (URL) filtering. [5]: 1) Known blacklist database maintained by industry or vendor; 2) user-configurable addition to the blacklist database; 3) Categorisation of websites; 4) Automatic background updates that do not require user intervention; 5) Comprehensive and accurate categorisation; 6) User-defined bypass URLs (whitelist); 7) Domain names are used for rating in addition to URL and IP addresses; 8) Provide multi-language support for international companies; 9) Provide the ability to carry out dynamic categorisation that does not rely on pre-determined user classification.

A key factor in successful Network-Based filtering is an effective traffic classification. Traffic classification provides the ability to automatically identify an application from a given stream of packets collected using either direct or passive observation of traffic coming into the network. In [28], Biersack, Callegari, and Matijasevic attempted to classify different techniques based on the following characteristics:

- **Granularity:** A coarse-grained algorithm can only distinguish the difference between large families of protocols such as Streaming vs. HTTP. In contrast, a fine-grained classifier is possible to identify a specific protocol such as eDonkey vs. BitTorrent file-sharing, or even a specific application such as SopCast vs. PPlive live streaming.

- **Timeliness:** This is an attempt to characterise the speed of classification using three types: first packet, after a few packets, and after flow termination. The post-mortem classification that analyses traffic after flow termination is usually for monitoring tasks, such as charging.

- **Computational Cost:** This field indicates the CPU power required to make the classification decision and inspect the traffic. Packet memory access requires the most processing power, followed by regular expression matching.

Table 2.1 shows the characteristics of each traffic classification approach.

A **Port-Based** approach identifies applications by extracting the port number from the transport header and then searching for it in a table con-

Table 2.1: Taxonomy of traffic classification techniques [28]

Approach	Properties exploited	Granularity	Timeliness	Comput. Cost
Port-based	Transport-layer port	Fine grained	First Packet	Lightweight
Deep Packet Inspection	Signatures in payload	Fine grained	First Payload	Moderate, access to packet payload
Stochastic Packet Inspection	Statistical properties of payload	Fine grained	After a few packets	High, eventual access to payload of many packets
Statistical	Flow-level properties	Coarse grained	After flow termination	Lightweight
	Host-level properties	Fine grained	After few packets	Lightweight
Behavioural	Host-level properties	Coarse grained	After flow termination	Lightweight
	Endpoint rate	Fine grained	After a few seconds	Lightweight

taining port-application associations. A **Deep Packet Inspection** approach matches the payload of the packet against a list of known patterns. **Stochastic Packet Inspection** tries to identify an application by examining the statistical properties of the payload for common string patterns, using the values of the first payload bytes as features for machine learning algorithms, assessing the randomness of the first payload bytes, and finally calculating the entropy of the first payload bytes. The L7-filter is an example of implementation of DPI in Linux kernel and the website of [29] contains classifiers that identify a comprehensive list of applications such as HTTP, Citrix, Kazaa, Jabber, BitTorrent, Gnucleus, eDonkey2000, and FTP, regardless of which port number being used. **Statistical** approaches try to classify traffic by applying statistical tools to the flow-based or host-based measurements. **Behavioural** approaches use similar data mining methods to the statistical approach, but analyse patterns generated by a host or endpoint (IP:port) that are further up in the network stack. The fundamental idea is that different applications are likely to exhibit different behaviours in terms of transport layer protocols

used, numbers of hosts contacted, numbers of distinct ports contacted, and connection graphs and patterns between endpoints.

2.1.3 Rendezvous-Based Filtering

Rendezvous-Based Filtering is a technique that controls common rendezvous services such as certificate authorities (CA), search engines, Domain Name Systems, Internet Route registries, and WHOIS databases which are often required for the proper operation of an Internet application. **DNS Filtering** is one of the easiest ways to conduct content filtering. It works by stopping users from being able to resolve the name of a website or domain, thus blocking access to these Web resources. Many paid or free services such as OpenDNS allowing users to block different categories of websites (e.g. Adult materials), instead of defining individual names. DNS blocking was proposed to be mandated by the Stop Online Privacy Act (SOPA) and the Protect IP Act (PIPA) in late 2011 in the US to address large-scale online copyright and trademark infringement because of the simplicity of implementation [30]. Another example is to advertise bogus routes through BGP so that users are unable to reach the real destination. A famous incident occurred in 2008, in which Pakistan Telecom advertised a bogus route, claiming to be the legitimate destination for the YouTube IP address as an attempt to block access to YouTube [31].

2.1.4 Endpoint-Based Filtering

Endpoint-Based Filtering is where the filtering decision is made by either a client or a server endpoint. An example is the "Safe Browsing" service offered by Google, which lets client applications such as the browser check URLs against Google's constantly updated list of unsafe web resources. Other products such as Norton Safe Web and McAfee SiteAdvisor provide a similar service. On the other side of end-to-end communication - Server Endpoints - access to contents can be controlled by using a whitelisting method such as IP ACLs or password authentication.

2.2 Data Loss Prevention (DLP) Factors

Data Loss Prevention (DLP) is more than just a myriad of solutions and services available in the market; it is also a business strategy to avoid users from transferring sensitive or critical information outside a corporate network either inadvertently or deliberately. DLP is the most popular term. However, the following terms are also used interchangeably:

- Data Leak Prevention
- Information Loss/Leak Prevention
- Extrusion Prevention
- Content Monitoring and Filtering/Protection

Vontu, now acquired by Symantec, in figure 2.1 shows the risk of Data Loss in an average organisation [32]. The main driver for the adoption of a DLP strategy is increasing privacy requirements and the risk of insider threat, which is one of the top cyber-threats according to the ENISA Threat Landscape 2014 report [17]. The DLP market is expected to grow from US$0.96 billion in 2015 to US$2.64 billion by 2020, at a CAGR of 22.3% from 2015 to 2020 [33].

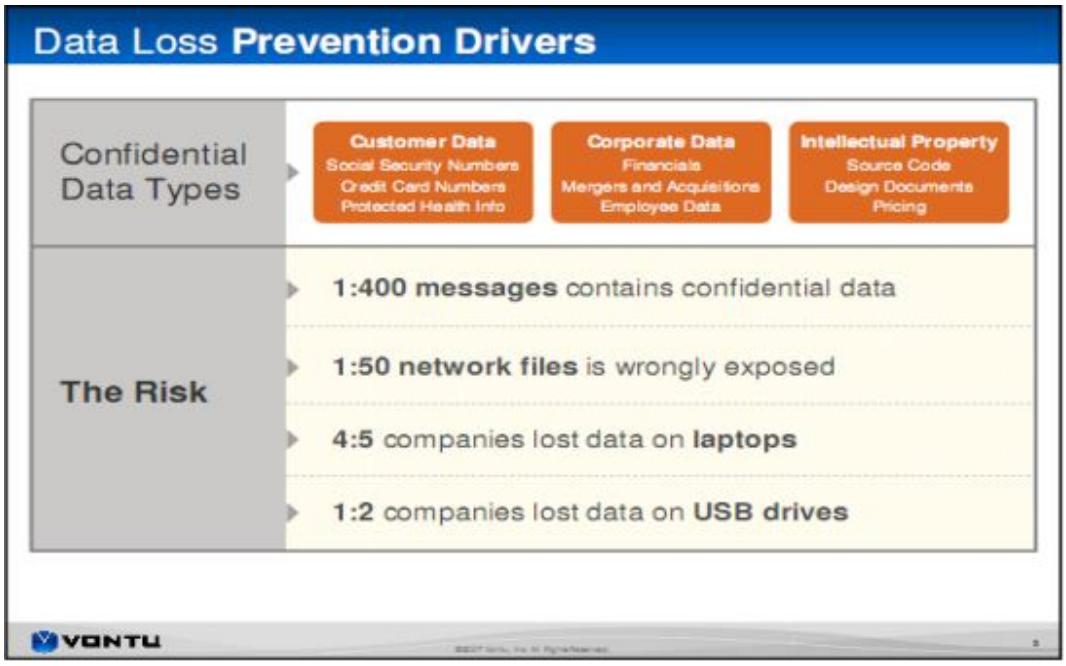


Figure 2.1: Data Loss Prevention Drivers

2.2.1 The DLP Lifecycle

In [34], Mogull defines three major areas which to protect to protect content throughout its lifecycle:

- **Data At Rest** The aim is to protect data stored either on-site or in the cloud by scanning and identifying critical and sensitive information. For example, DLP software can scan network shares, identify documents with credit card numbers, and then perform any appropriate actions to protect them.

- **Data In Motion** This is to protect data in transit by sniffing the network traffic passively or in-line via a proxy to identify the content sent across communication channels such as HTTP, FTP, IM, P2P and SMTP.

- **Data In Use** To protect data in this state typically requires monitoring the data at the endpoint where the user interacts with the data. For example, DLP software can monitor any data leaving via removable devices, such as CDs, and USBs. This usually requires an agent deployed at the end-user workstation.

According to Gartner, DLP technology is divided into two categories [35]:

- **Enterprise DLP** This usually requires holistic functionalities such as a centralised management console, support for advanced policy definition, and event management workflow to solve the business and technical problems of preventing data loss and leakage.

- **Integrated DLP** This only has limited DLP features integrated with other security products such as secure web gateways, secure email gateways, email encryption products, enterprise content management platforms, data classification tools, data discovery tools, and cloud access security brokers.

2.2.2 DLP techniques

A comprehensive discussion of DLP techniques is beyond the scope of this study, which focuses only on the integrated DLP features that provide protection to Data In Motion.

An effective DLP solution must be able to perform both deep content inspection and contextual analysis of data using a variety of techniques. Content, as the name suggests, is the actual information, whereas context includes everything else such as source, destination, size, recipients, sender, header information, metadata, time, and format. As suggested in [34], the following seven major analysis techniques can be used in content analysis:

1. **Rule-Based/Regular Expressions:** This is the most common analysis technique for quickly identifying structured information like healthcare codes/records, credit card numbers, and social security numbers.

2. **Database Fingerprinting:** This technique looks for exact matches by taking either a database dump or live data (via ODBC connection) from a database.

3. **Exact File Matching:** This works by taking a hash of a file and monitoring any files that match that exact fingerprint. This technique does not analyse the contents.

4. **Partial Document Matching:** This technique usually takes a cyclical hash of a complete or partial content and looks for a match. Cyclical hashing works by taking a hash of a portion of the content, offset by a predetermined number of characters, and then taking another hash of the content. The process is repeated until the entire content is parsed. This creates a series of overlapping hash values.

5. **Statistical Analysis:** This uses a machine learning algorithm such as Bayesian analysis or other statistical analysis of content to find policy violations in content that resembles the protected content.

6. **Conceptual/Lexicon:** This uses a combination of sources like dictionaries, rules and other analyses to protect ambiguous content that resem-

bles an "idea". For example, it could use key phrases, word counts, and positions to find traffic that resembles insider trading, sexual harassment, or job hunting.

7. **Categories:** This technique uses pre-compiled categories that look for common types of sensitive data such as PCI DSS protection, and HIPAA.

In Table 2.2, it shows an overview of the strengths and weaknesses of the seven techniques mentioned above.

Table 2.2: Strength and Weakness of DLP Techniques

Technique	Strength	Weakness
Rule-Based/Regular Expressions	Fast and easy to use. The technology is well understood.	High false positives and offer little protection for unstructured content
Database Fingerprinting	low false positives. Allow protecting some data while ignoring other.	can have a performance impact on large databases. It will not contain transaction data since the last extract.
Exact File Matching	works on any file type and low false positives	can be circumvented easily and unable match for content that is edited.
Partial Document Matching	can protect unstructured data and low false positives. It can identify partial violation within a document.	performance limitations, common phrases may trigger false positives, easy to circumvent via like ROT 1 encryption.
Statistical Analysis	This can work on nebulous content where exact match is not possible	susceptible to false positives and false negatives, requires large corpus of source content
Conceptual/Lexicon	works best on unstructured ideas that cannot be described using specific examples.	very prone to false positives and false negatives and can only be built by the DLP vendor (cost more).
Categories	simple to configure, save time in building the policy	The generic policy may not fit for every business.

DLP is a cyclical process that consists of the following steps: 1) Detect: paint a picture of the data and environment the DLP needs to deal with; 2) Mark/tag: Add a meta data classification tag; 3) Monitor: monitor the leak-

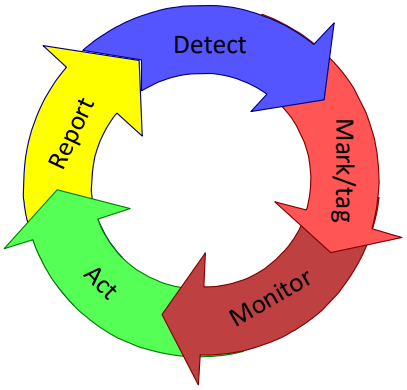


Figure 2.2: Data Loss Prevention Process

age of the classified data; 4) Act: if the data is identified, enforce policy-based actions; 5) Report: report the status of the identified data based on the defined rule set. Fig 2.2 provides an overview of the DLP process.

2.3 Advanced Persistent Threat Factors

Advanced Persistent Threat (APT) has been a catchphrase for many years in the security industry, and almost all security vendors claim their products can give some level of protection against APT. This is mostly due to the high profile nature of several large-scale security breaches in the past, such as Operation Aurora [36], intrusions to SK Communications [37], RSA Breach [38], Operation Ke3chang [39] and Operation SnowMan [40] etc. To understand how to protect an organisation from APT, one needs to understand the characteristics and the attack model of the APT. In [41], Bejtlich provides a good comparison between a traditional attack and an APT attack and also an anatomy of the attack model. In Table 2.3 shows the differences between a traditional threat and an APT. As shown in the table, APT attack follows a similar model to

a traditional attack, but the techniques used in each stage are different. An APT attack usually consists of the following phases, which are analogous to the concept of the intrusion kill chain introduced in [42] and the attack lifecycle introduced in [43].

Table 2.3: Comparison of traditional and APT attacks [41]

	Traditional Attacks	APT Attacks
Attacker	Mostly single person	Highly organised, sophisticated, determined and well-resourced group
Target	Unspecified, mostly individual systems	Specific organisations, governmental institutions, commercial enterprises
Purpose	Financial benefits, demonstrating abilities	Competitive advantages, strategic benefits
Approach	Single-run, "smash and grab", short period	Repeated attempts, stays low and slow, adapts to resist defences, long term

2.3.1 APT Lifecycle

1. **Reconnaissance and Weaponization Phase:** Attackers attempt to collect information about the target. The more information attackers collect, the more likely they are to succeed in the later attack phases. Open-source intelligence (OSINT) or Social Engineering are popular techniques for collecting the information. Attackers can use this information to prepare the necessary tools and construct an attack plan.

2. **Delivery** Attackers deliver their malware to targets either directly through different Social Engineering techniques such as Spear Phishing or indirectly via a compromised third-party website trusted by the targets, such as the Watering Hole Attack technique.

3. **Initial Intrusion Phase:** After successfully executing the malware on the target, attackers then try to establish a foothold in the environment by controlling computers within the target organisation remotely. This is done by exploiting vulnerabilities in popular software such as Adobe Flash, Internet Explorer, Microsoft Office and Adobe PDF. Some APTs

leverage zero-day exploits while others may employ old exploits that target unpatched applications.

4. **Command and Control Phase:** Command and Control (C2) is a technique for taking control of a group of compromised computers. C2 enables further exploitation of the network. This is usually done stealthily, leveraging different legitimate services and publicly available tools such as Tor Anonymity Network, Remote Access Tool (RAT) and social networking websites.

5. **Lateral Movement Phase:** This is an iterative cycle involving the following activities: 1) perform internal reconnaissance to obtain more information about the target environment; 2) harvest credentials and gain elevated privileges by compromising more computer systems; 3) identify and collect valuable data. This phase often lasts a long time as it is designed to run low and slow to make it difficult to detect.

6. **Data Exfiltration Phase:** Stealing data such as internal memoranda or policy papers, business contracts or negotiations, and intellectual property is the primary goal of APTs. The data is often compressed and encrypted before transferring it to an external location under the attacker's control via TLS or Tor Anonymity Network.

2.3.2 Countermeasures

Due to the stealthiness and complexity of APTs, defence usually requires a combination of different security countermeasures. This study discusses the security countermeasures that can be implemented in a Secure Web Gateway by inspecting the outbound traffic to uncover the APT. These countermeasures are useful in detecting and blocking APTs in the following phases of the APT attack lifecycle.

Delivery Phase: In addition to the traditional defence mechanisms such as URL filtering, malware protection, and Adware/Spyware protection, a Secure Web Gateway can incorporate the Sandboxing technique for analysing the

behaviour of an executable, which allows defenders to detect Zero-day Exploits [44].

Command and Control Phase: As an APT depends on remote access and control of infected computers inside a network, APTs can be contained and disrupted by analysing the network activities associated with the remote control. In [45], Binde, McRee, and O'Connor proposes the following four approaches to detect APTs through the analysis of outbound network traffic.

- **Rule Sets:** This is a signature-based detection method, matching network traffic to known malicious patterns of system and network behaviours. Examples of these include identifying phishing campaigns and; recognising and blocking malicious traffic such as that associated with the Poison-Ivy Remote Access Toolkit.

- **Statistical and Correlation Methods:** This approach involves studying normal behaviour and searching for anomalous activities. For example, attackers can use a method called fast-flux to make tracking of data exfiltration difficult. The defender can analyse the output of generated fast-flux network traffic to detect statistical variation from expected norms. This requires analysing massive amounts of data, potentially through machine learning. Some research has been done using big data analytics for APT detection [46, 30]

- **Manual Approaches:** This uses digital forensic techniques to manually detect anomalous behaviour via logging and monitoring. For example, a SQL statement that is abnormally larger shows a sign for further investigation. Other examples include that unusual outbound traffic initiated from the target organisation, DNS logs, and abnormal traffic as compared to known good NetFlow baselines.

- **Detecting and Blocking Data Exfiltration:** The following lists the methods to detect and block data exfiltration by analysing the characteristics of outgoing traffic: 1) detect and block RAR files; 2) undertake an Open Source Host-based Intrusion Detection System (OSSEC) Active

Response; 3) limit outbound access; 4) monitor for precursor attacks. RAR compression has been used as a way to obscure data. RAR files should be blocked if an enterprise determines that it should never egress from a given network. OSSEC Active Response runs preventative commands or responses on the client or server side when some events are triggered. It is a good protection against port scans, brute forces and other information-gathering attacks. It is also useful to allow only a Secure Web Gateway to initiate outbound traffic to the Internet and block all other egress traffic. Even if the device is exploited and malware is installed, it may not be able to exfiltrate the data to an external host. Monitoring for precursor attacks can prevent more users being exploited by a known attack. This can be automated by integrating with blacklists or scripts developed by [47] to remove JavaScript inside the Portable Document Formats (PDF) files as they are transferred through the Secure Web Gateway.

2.4 SSL filtering Factors

2.4.1 Background to the SSL protocol

Transport Layer Security (TLS) protocol is the successor of the Security Socket Layer (SSL) protocol [48]. These two terms are used interchangeably unless reference is being made to a specific version of the protocol. SSL is the most widely used term for the encryption protocol on the Internet. HTTPs is the secure application protocol running on top of the SSL and is commonly used to secure communications between browsers and the Web. A research that had monitored 25,000 residential ADSL customers between 2012 and 2014 showed that HTTPs flows are more than doubled in two years [49]. Google, FaceBook, YouTube and other major content providers are changing to HTTPs. However, this also brings a new challenge to the Secure Web Gateway as now it becomes difficult to filter out unwanted content within the encrypted traffic. SSL has not only been used in benign communications to protect sensitive and confidential information but has also been used in malicious traffic to get

around traditional security controls.

Three main pillars underpin the operation of the SSL protocol - privacy, integrity and authenticity. Privacy prohibits others from intercepting the traffic; integrity ensures the data has not been altered in transit; and authenticity aims at validating the communicating parties. SSL provides privacy through the use of symmetric encryption, integrity via cryptographic message digests, and authenticity by using the X.509 public key infrastructure. Figure 2.3 shows a basic handshake to establish a SSL session. The process starts with a SSL client sending a hello message to a SSL server. The message includes a supported protocol version, a list of supported cipher suites, and a client-generated random number. Then, the SSL server responds it with a Server Hello message. The message contains a server certificate, the server's preferred cipher suite, and a server-generated random number. The server certificate contains the SSL server's public key and hostname, digitally signed by a CA. In the next step, the SSL client sends the ClientKeyExchange message that contains a pre-master secret encrypted by using the SSL server's public key. The SSL client and server can acquire the same session key from the pre-master secret and random numbers. In the last step, the SSL client and server exchange ChangeCipherSpec message to inform each other that subsequent data will be encrypted using the session key derived from the previous step.

During the SSL handshake process, the ssl client is responsible for validating the certificate presented by the ssl server by following the X.509 specification [50]. In [51], Jarmoc and Unit provides a summary of the RFC5280 validation process:

- Verify the certificate's digital signature.
- Determine the CA and all intermediate certificates by following the certificate chain.
- Check if the client browser trusts the root CA.
- Check issuance and expiration dates of a certificate to confirm the certificate's temporal validity.

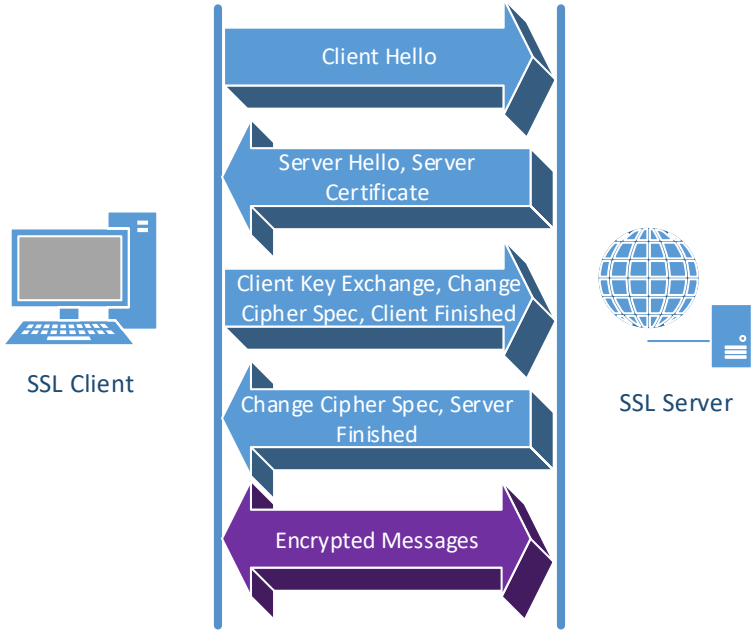


Figure 2.3: A basic handshake to establish an SSL session

- Compare the subject in the certificate with the expectation in the higher level protocol such as application layer to determine if they are the same.
- Use the Online Certificate Status Protocol (OCSP) and check the certificate revocation list to determine whether the certificate has been revoked [52].

2.4.2 SSL Interception Proxy

To intercept and inspect SSL-encrypted traffic, the SSL session needs to be terminated at the proxy server, and the proxy server has the private key to decrypt the traffic. Figure 2.4 shows that the SSL proxy substitutes the certificate with its own that is issued by a CA trusted by the SSL client. The SSL proxy then generates another SSL session with the destination. This method is called Man In The Middle (MITM). The SSL proxy can sign such certificates using one of the following two methods: 1) The most frequently used method is using a private CA to sign the certificate, and the private CA's certificate is imported into the client's trusted root certificate store, or 2) Create a SubCA

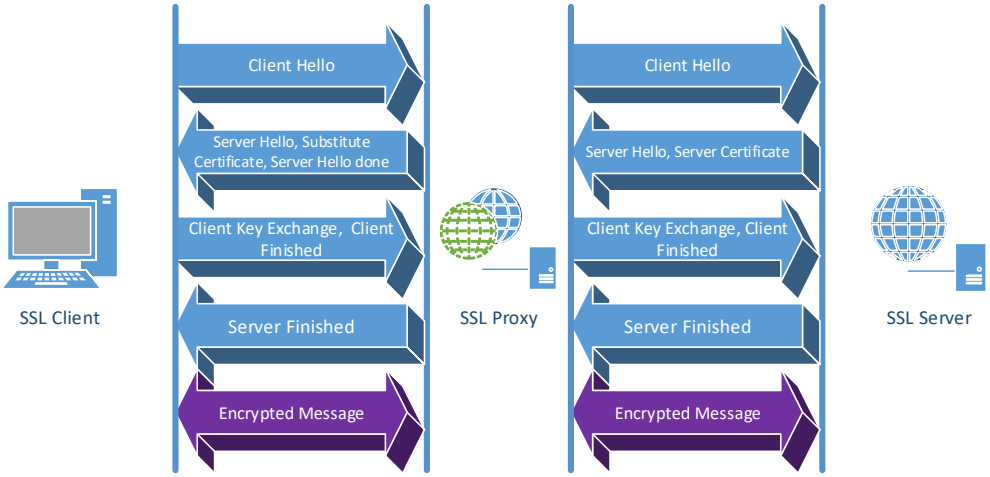


Figure 2.4: SSL session establishment with involving an SSL proxy

that acts as an intermediate signing authority authorised by a public trust root CA.

2.4.3 Adaptive Security Model

In [53], Jawi, Ali, and Zulkipli proposed an adaptive security model. Figure 2.5 illustrates the components of the proposed solution. The proposed solution consists of the following components:

- **Monitoring component** collects environment factors from client devices and certificate information from SSL servers.
- **Analysis component** is responsible for analysing the information passed through from the monitoring component
- **Response component** is in charge of making the decision on permitting or rejecting the connection

An adaptive proxy operates in two modes based on the current security threat level. When the threat level is low, the pass-through mode is used, and it only exposes the server certificate and the URL information to the proxy server. When the threat level is high, the MITM method as described above is used to intercept and inspect the full payload of the traffic.

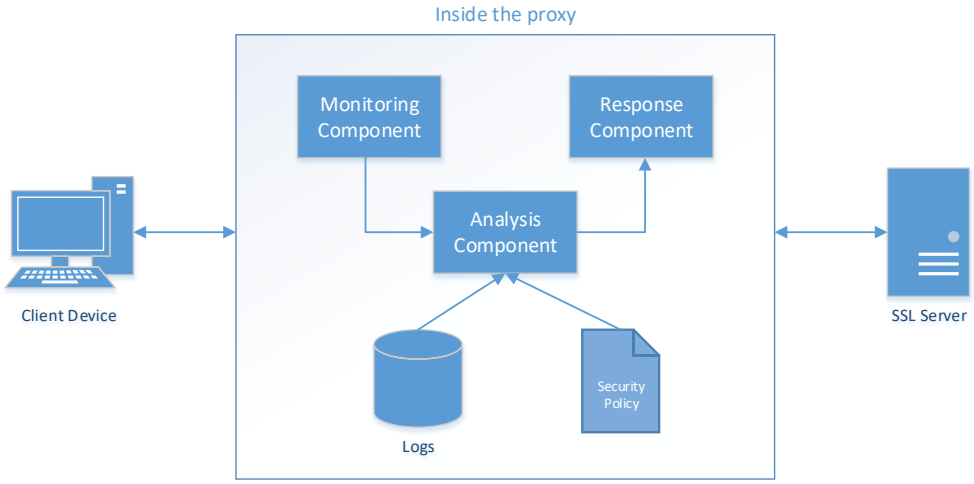


Figure 2.5: Adaptive proxy components [53]

2.4.4 Server Name Indication (SNI)-based filtering

The SNI extension allows clients to indicate which hostname they are trying to connect to at the start of the SSL handshaking process [54]. This extension is created to allow multiple secure websites with different certificates to be served from the same IP addresses. This technique inspects the "server_name" value inside the SNI extension to determine whether the traffic should be allowed or blocked. As shown in Fig. 2.6, the gateway can either allow the Client Hello message to be sent to the destination server to complete the handshake or reset the connection. However, SNI-based filtering can be easily circumvented as suggested by Shbair et al. in [55].

2.4.5 BlindBox HTTPS

In [56], Sherry et al. presents a novel approach - BlindBox HTTPS - that maintains the privacy of SSL encryption and at the same time provides the ability to carry out deep packet inspection directly on the encrypted traffic. The system is built upon a new searchable encryption scheme called DPIEnc that supports inspecting the encrypted payload by using exact string matching, regular expression or scripting. Other encryption schemes such as fully homomorphic encryption or functional encryption can achieve the same result but are prohibitively slow for use in a production environment. BlindBox

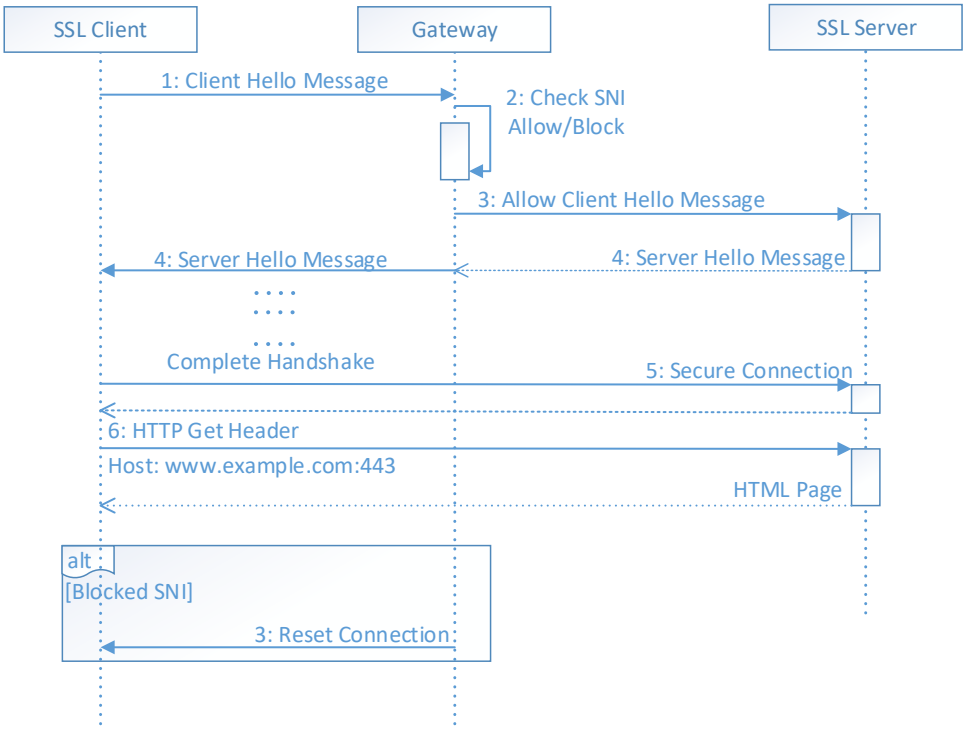


Figure 2.6: SNI Filtering Sequence Diagram [55]

claims that it can inspect packets at a rate of up to 186Mbps per core, which is comparable to most standard IPS/IDS implementations. Fig. 2.7 shows the system architecture of the BlindBox. BlindBox basically contains four parties: 1) Sender (S), 2) Receiver (R), 3) middlebox (MB), 4), and rule generator (RG). At a high level, it functions as follows:

1. **Initialisation:** The RG role is normally fulfilled by organisations like Emerging Threats [57], McAfee, or Symantec. RG generates a set of rules signed with its private key before sharing them with MB. S and R obtain RG's public key by installing a BlindBox HTTPS configuration. Beyond this, RG does nothing else.

2. **Connection Setup:** In this phase, S and R perform a normal SSL handshake to agree on a key K_0. Then S and R use K_0 to derive three other keys: 1) K_{SSL}: a regular SSL key to encrypt the traffic, 2) K, which is used in the detection protocol, and 3) K_{rand} is used as a seed for randomness so that both S and R will generate the same randomness. The rules from RG are encrypted with key K and MB does not learn

the value of K. The process is done in such way that R and S do not learn what the rules are. The whole process is described as obfuscated rule encryption. The only downside of this process is that it removes the transparency of the MB as S and R need to communicate with the MB.

3. **Sending Traffic:** Before sending the traffic, the sender prepares the traffic in the following ways: 1) encrypting the traffic using regular SSL; 2) splitting the traffic into substrings (tokenize) taken from various offsets, and 3) using the DPIEnc encryption scheme to encrypt the resulting tokens.

4. **Detection:** When the BlindBox receives the encrypted tokens and the encrypted traffic, it uses a function called BlindBox Detect to search for matching between the encrypted rules and the encrypted tokens. Based on the detection result and the policy defined, BlindBox can then decide what actions to take against the traffic (drop, stop, or notify). At the end of detection, MB forwards both encrypted traffic and encrypted tokens to the receiver.

5. **Receiving Traffic:** There are two actions performed by the receiver. First, the receiver performs the regular SSL decryption and authentication. Second, the receiver verifies the integrity of the token by making sure it has been generated properly by the other endpoint and the other endpoint is not trying to circumvent the checks at the MB by generating only a subset of the tokens.

With this architecture, BlindBox supports two classes of privacy model: exact match privacy and probable cause privacy. Under the exact match privacy model, the MB only learns at which positions in a flow, matched keywords occur. For the probable cause privacy model, the MB can only see a decrypted packet, or flow if the flow contains a matched keyword. Both privacy models are much stronger than the MITM approach discussed earlier in this review.

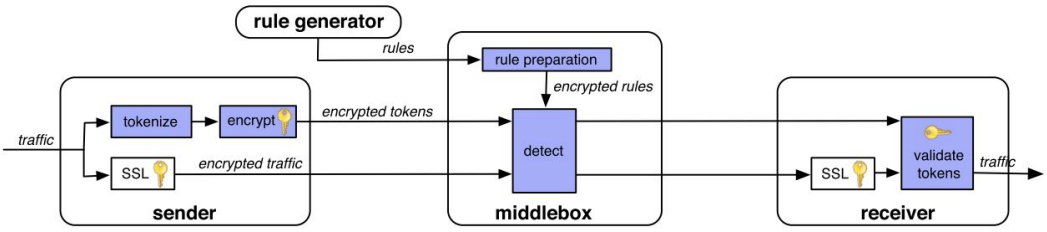


Figure 2.7: BlindBox System Architecture [56]

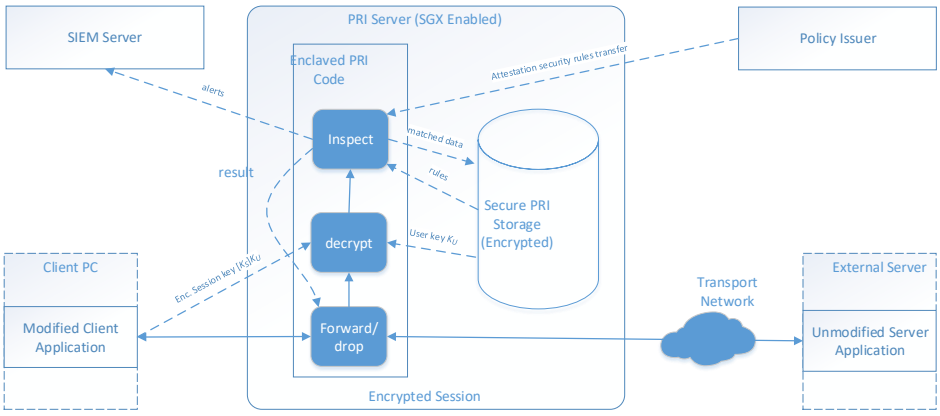


Figure 2.8: A PRI System used for prevention [58]

2.4.6 Privacy Preserving Inspection (PRI)

Privacy Preserving Inspection (PRI) is another attempt to solve the privacy issue of inspecting encrypted network traffic. In [58], Schiff and Schmid proposed a novel approach that leverages Intel Software Guard Extensions, a type of hardware architecture that provides data privacy. The solution decouples the different roles required for network traffic inspection such as users, rules, configuration provider and administrators. With the support of the right hardware, the implementation of PRI is simple and requires only a minor software update to the client. At a higher level, here is how the inspection is carried out in a PRI system as shown in Fig. 2.8:

1. If the solution is run the detection mode, the client will establish an encrypted session with the server. If the solution is run in prevention mode, the client will check with the Enclaved PRI code to determine whether to allow the traffic to move through the PRI system.

2. The client sends the session key encrypted by the shared user key to the

PRI server: $[K_S]K_U$.

3. PRI duplicates and decrypts the session traffic using the session key. The decrypted traffic is then processed by the PRI.

4. PRI inspects the session data using the rules supplied by the policy issuer.

5. The packet is stored in the Secure PRI Storage when the data matches a rule, Then, an alert is sent to a Security Information and Event Management Server for the purpose of analysis and reporting.

2.5 Privacy Factors

Employee monitoring is a common practice in the workplace, including Internet usage monitoring using a Secure Web Gateway. However, incorrect implementation of employee monitoring or excessive monitoring can result in loss of trust, high staff turnover, lower productivity and even personal grievance action from employees. Thus, a key factor in the successful implementation of Secure Web Gateways in the workplace is to be able to balance employers' interests against the privacy concerns of employees.

2.5.1 Privacy Definition

Louis Brandeis and Samuel Warren in 1890 described "Privacy" as "the right to be let alone" [59]. Tipping J extended this definition to "the right to have people leave you alone if you do not want some aspect of your personal life to become public property" in the Hosking v. Runting case [60]. Universally, privacy is often seen as a fundamental human right. However, it has never been an absolute right; in the employment context, employers' interests often trump employees' rights to privacy.

2.5.2 Employees' Rights to Privacy

A book published by the privacy commissioner of New Zealand states that it is reasonable for employers to apply some form of control over how employer-provided resources such as the Internet and email can be used. However, the

act of monitoring must be done in a fair and reasonable manner [61]. Employment agreements determine employees' rights to privacy and employees cannot insist on their privacy rights upon entering an employment relationship if the contract waived those privacy rights [62]. Some argue that monitoring employees' Internet usage not constitute the definition of "collect" in the Privacy Act because the information is unsolicited and is already in the system when the traffic moves through the network Roth. Excessive intrusion of privacy can have adverse impact on a person's dignity, emotional well-being and autonomy, but employees often have no choice but to give up their rights due to inequality bargaining power between employers and employees; Oliver believes that privacy rights should not be able to be contracted out to employers if these rights are seen as fundamental human rights [64]. In some legal cases about employees' dismissal due to an inappropriate use of the Internet in the workplace, the judge's decision was based on whether employees had a reasonable expectation of privacy and whether the policy had provided a clear boundary [63]. In New Zealand, privacy principles one to four deal with the collection of personal information, which has some level of restraint on Internet usage monitoring.

2.5.3 Legal Framework

In UK, the following legislation can have implications for employee monitoring [64]:

- **The Human Rights Act 1998** is based on the Article 8 of the European Convention on Human Rights (ECHR). ECHR proscribes monitoring in areas where employees have a reasonable expectation of privacy. However, this expectation is often controlled by company policies.
- **The Regulation of Investigatory Powers Act 2000:** This implements Directive 97/66/EC concerning the privacy of telecommunications. Under this legislation, employers can only carry out interception if employers have obtained consent from either the sender or the recipient.
- **The Data Protection Act 1998:** This law regulates the processing

of personal data by data controllers as set out in Directive 97/66/EC, concerning the privacy of telecommunication. Under this legislation, it is unlawful to perform the monitoring if the process is unwarranted or employers have no legitimate interests as per Paragraph 6(1) of Schedule 2.

In [62], Britton lists the following legal frameworks in New Zealand that could be applied to privacy rights in the workplace:

- **Employment Relations Act 2000** deals with unequal bargaining power between employers and employees, which is an issue in employee monitoring as discussed in the previous section. Employers must make sure they carrying out monitoring in a fair and reasonable manner to reduce the risk of employees' personal grievance claims.

- **Human rights Act 1993** proscribes discrimination against an individual on any of these prohibited grounds: "sexual orientation, sex, family status, marital status, religious belief, employment status, ethical belief, political opinion, colour, age, race, disability, national origins or ethnic", using the information collected from a monitoring device.

- **Privacy Act 1993** consists of twelve information privacy principles that govern how an agent (employer) can collect, use, disclose, store and give access to personal information.

- **Health and Safety in Employment Act 1992** limits employees' rights to claim privacy in the workplace. Employers are compelled to create a safe work environment and proactively manage hazards in the workplace. Hence, for security purposes, employers may be able to justify privacy-invasive monitoring techniques.

- **Section 216B of Crimes Act 1961** disallows purposely intercepting any private communication using a monitoring device except for circumstances in which any party in the communication reasonably expects that some other people may intercept the communication. To avoid violat-

ing this law, employers often create work policies to control employees' expectations of privacy.

In the United States, the following federal law imposes various restrictions on how employers monitor their employees' electronic communications [65]:

- **The Federal Wiretap Act** FWA prohibits purposely using any electronic or mechanical device to intercept any wire, oral, or electronic communication, excluding any communication in which the person does not have a reasonable expectation of privacy.

- **The Stored Communications Act** SCA prohibits purposely access a wire or electronic communication without authorisation while it is in the electronic storage of an electronic communication system. This excludes authorised to access and access by the providers of the electronic communications service. For example, employers are excluded from the SCA as they are probably the provider of internal Internet and email systems.

- **The Pen Register Act** PRA prohibits recording dialled telephone numbers or the telephone numbers of incoming calls. However, the PRA excludes the providers of the electronic communication system, hence employers may be excluded. However, a court case held that the PRA does not apply to IP address as they are more like the material of a telephone call rather than the telephone numbers [66].

- **The Electronic Communications Privacy Act** ECPA allows the following three exceptions which are related to employee monitoring: 1) Employees have consented the monitoring in either an express form where employees sign a notice or an inferred form, where employees have only been notified about the monitoring; 2) Employers are allowed to monitor on the ground of "ordinary course of business", and 3) Providers of the system are exempted from this.

- **The Computer Fraud and Abuse Act** CFAA prohibits unauthorised access, access exceeding current authorisation, access causing damages,

and accessing causing loss to a "protected computer". CFAA is usually seen as limiting the privacy rights in the workplace.

- **The National Labor Relations Act** Under NLRA, it is prohibited for employers to take action against employees for conducting lawful union business or union organising activities using the company-provided resources; moreover, employers cannot even have a policy to discourage such activities.

- **Protections of Whistleblower** Federal statutory schemes such as Dodd-Frank Wall Street Reform, Consumer Protection Act of 2010, and Exchange Act of 1934 protect employees' rights to report employers' certain unlawful conducts. For example, employers cannot take adverse actions against their employees for reporting their employers' misconducts.

- **The Bankruptcy Act** This prohibits employers from using the information gathered from monitoring employees' electronic activities to discriminate against employees who have taken advantage of bankruptcy protection.

In the United States, the most relevant legislation to privacy in the electronic communication is ECPA which proscribes the unconsented interception, storage and disclosure of electronic communications, with the following exceptions [67]:

- **Business Exception:** Employers can intercept communications using a qualified device for the ordinary course of business purposes.

- **Consent:** Employers can intercept a communication if they have obtained consent from one of the parties involved in the communication.

- **Service Provider:** Employers who are the providers of electronic communication services can retrieve information maintained on their systems. Employees' rights to privacy must yield to employers' property rights.

The Cybersecurity Act of 2015 that was signed into law by President Obama in 2015 provides employers with authority to monitor a company's information systems, use defensive measures on a company's information system, and share both incoming and outgoing information with others for "cybersecurity purposes" Kerr. Also, employers can outsource these functions to a third-party company. However, it is unclear what constitutes a "cybersecurity purpose", and the new legislation arguably gives a broader scope than existing provider exceptions.

2.5.4 Privacy Preserving Monitoring

To balance employers' business interests with employees' rights to privacy, and to create a productive work environment and reduce the risk of legal liability, companies should consult their legal department, human resources department, employees, and union representatives, if applicable; these provide input to create an acceptable computer-use policy [67]. Cox, Goette, and Young provides the following guidelines for implementing an acceptable computer-use policy in the workplace [67]:

1. Employers should include a written policy in employee manuals and literature.

2. Employers should ensure employees have the opportunity to read and accept the policy in writing or electronically.

3. Employers should remind their employees about the existence and content of the policy regularly.

4. Employers should send out a notification to their employees about the monitoring of Internet usage on a regular basis. One example is to configure the computer to display a warning message the first time employees log-on to the PC.

5. Employers should advise employees that password protection or HTTPs encryption are not immune to inspection.

6. Employers should advise their employees that violations of the policy may result in disciplinary actions up to and including termination of employment.

7. Employers should give examples of what counts as appropriate usage and what does not, including excessive personal use, should personal use be allowed by policy.

Table 2.4 lists the key components recommended by the US Government Accounting Office that should be included in an acceptable computer-use policy [69].

Table 2.4: Important components of an Acceptable Computer-Use Policy [69]

Police Component	**Component Description**
Monitoring use of company provided resources	This component should include statements that company provided resources are for business use and any information accessed, produced, or stored using these resources are company's property and subject to audit and monitoring.
Set no expectation of privacy	This component should include statements that clearly articulate the extent or limitations of privacy protections for using the company provided resources such as Internet or email.
Inappropriate use	This component should include some examples of inappropriate use including extensive personal use during working hours, access of any offensive material (e.g., pornographic, racist, malicious, obscene, embarrassing, sexist, fraudulent, derogatory, intimidating, terrorist or unlawful), and use of harassment language in the email and Internet.
Appropriate Use	This component should provide examples of the allowable use of company provided resources, including whether personal use is authorised or not.
Protection of sensitive information	The policy should provide instructions on how to handle critical and sensitive company information.
Disciplinary action	This component should include statements about penalties and disciplinary actions such as termination of employment.
Policy acknowledgement	This component should include a statement requiring that employees to accept their responsibility to adhere to the policy and acknowledge they have read and understood the policy.

Chapter 3

Methodology

This chapter discusses the steps that have been undertaken to identify and categorise the key factors, develop a practical design and architecture for an implementation and finally a testing framework to evaluate the effectiveness of a deployment. This chapter concludes with the reasoning behind this methodology.

3.1 Approach

A Secure Web Gateway is essentially a Web Proxy with various incorporated security countermeasures to protect end users from web-borne threats. Although many commercial products in the market claim they can perform the functions of a Secure Web Gateway, there has not been much academic research in this area. Most existing research focuses either on the performance of the web proxy [70, 11, 71, 13] or on individual security countermeasures [45]. Thus, this thesis approaches this problem by first identifying and categorising the key factors in building a Secure Web Gateway, followed by selecting suitable hardware and software for running a Secure Web Gateway, developing a design and architecture for a home or small office implementation, developing a testing framework to evaluate the effectiveness of the implementation, and finally testing the implementation using a residential vDSL connection. The following outlines the stages undertaken in this research:

1. Research - Chapter 2

- Identifying market drivers in transforming the traditional web proxy to a Secure Web Gateway
- Analysing related works and existing security countermeasures

2. Planning - Chapter 3

   - Selecting the key factors in building a Secure Web Gateway
   - Creating a reference design and architecture to deliver the required capability
   - Selecting suitable hardware for running a Secure Web Gateway

3. Development - Appendix A, B

   - Installing and configuring the hardware and software according to the defined configuration
   - Defining a test plan
   - Creating a testing framework to automate the tests defined in the test plan

4. Testing and evaluation - Chapter 4, 5

   - Testing the Secure Web Gateway implementation and evaluating its effectiveness according to the findings in the research stage
   - Comparing the test results with another commercial implementation
   - Identifying the gaps and challenges in building a Secure Web Gateway

3.2 Reasoning

People need to understand the reasoning behind investing in a Secure Web Gateway, and therefore it is important to understand the market drivers for it. To transform a traditional web proxy to a Secure Web Gateway, one needs

to understand the different security countermeasures required to thwart web-borne threats. After analysing the related work in the research stage, a planning stage is required to categorise the key factors in building a Secure Web Gateway. The categorisation helps to identify the optimal network and software configuration and the required hardware that can be used to validate the key factors. Gaps and challenges identified in the planning stage are used as an input for developing the design and architecture in the subsequent stage. In the development stage, it is necessary to demonstrate the ability to setting up a Secure Web Gateway, and devise a test plan and testing framework for testing the effectiveness of the implementation. It is hard to measure the effectiveness of a Secure Web Gateway without something to compare it against and this is why this research chose to compare the test results with another commercial implementation.

3.3 Key Factors

The review of the related literature led to the development of the Content-filtering, DLP, APT, SSL-filtering, and Privacy factors for building the Secure Web Gateway. Fig 3.1 provides a view on these factors and their associated sub-factors. A Secure Web Gateway needs to be able to classify Internet resources based on their content through network-based, rendezvous-based or endpoint-based filtering. SSL-filtering is a vital capability as more and more Internet resources are delivered through SSL encryption. MITM is currently the most widely used method, but presents a security and privacy risk. Blindbox and PRI offer a novel approach to maintaining the balance between privacy and security, but they are not yet available for a production implementation. An Adaptive Security model using SNI-based filtering is the most practical method that allows administrators to disable SSL inspection on privacy-sensitive Internet resources selectively. APT is becoming more prevalent and is used by organised criminals targeting enterprises. A Secure Web Gateway can break the APT lifecycle by controlling the communication between end-user devices and C2 servers, and the delivery of malware through compromised Internet

resources. DLP is often integrated with a Secure Web Gateway and offered as an optional feature. A Secure Web Gateway can inspect data as it traverses the network and uses one of the DLP techniques to match and block exfiltration of classified information. Privacy is a non-technical factor that one should consider when implementing a Secure Web Gateway. A company should work with its legal counsel to develop and publish a computer use policy before implementation, to avoid unnecessary risk of legal liability and regulatory compliance.

Figure 3.1: Secure Web Gateway Key Factors

3.4 Reference Design and Architecture

This section discusses the design and architecture of a Secure Web Gateway implementation. Based on the factors discussed in the previous section, the

design and architecture need to satisfy the following requirements:

R1 **Effectiveness** The Secure Web Gateway needs to be able to detect and block web-based attacks effectively with low false positive and false negative rates.

R2 **Granularity** The Secure Web Gateway needs to be able to work seamlessly with all end user devices such as desktop, laptop, tablet or mobile devices. It should allow access to any benign Internet services such as video streaming, websites, and web-based API while maintaining an effective control over access to malicious resources.

R3 **Security** The Secure Web Gateway needs to be able to provide Internet protection without breaking security properties of Internet protocols such as Transport Layer Security (TLS) and IPsec, which are designed to ensure communication between endpoints is secured.

R4 **Affordability** This research aims to find a solution that is practical and affordable for small business or ordinary home use. There are already enterprise solutions on the market such as ZScaler, Blue Coat, Cisco, Forcepoint (formerly Websense), and Intel McAfee, which target large enterprise customers.

3.4.1 Network Architecture

Three possible network architectures were considered: 1) Explicit Web Proxy architecture that requires clients to configure their browsers or applications to use the proxy server; 2) Transparent Web Proxy architecture in which the proxy server is the intermediary that sits in between the client and server on the Internet; 3) Leverage the Web Cache Communication Protocol (WCCP) if the routers or switches in the environment support this protocol. The first option is prone to circumvention and does not work for applications or devices that are not proxy-aware, and the third option requires the support of CISCO hardware. Option two was chosen to meet the requirements R2 and R4. The diagram in Fig. 3.2 depicts the high-level network architecture of an implementation

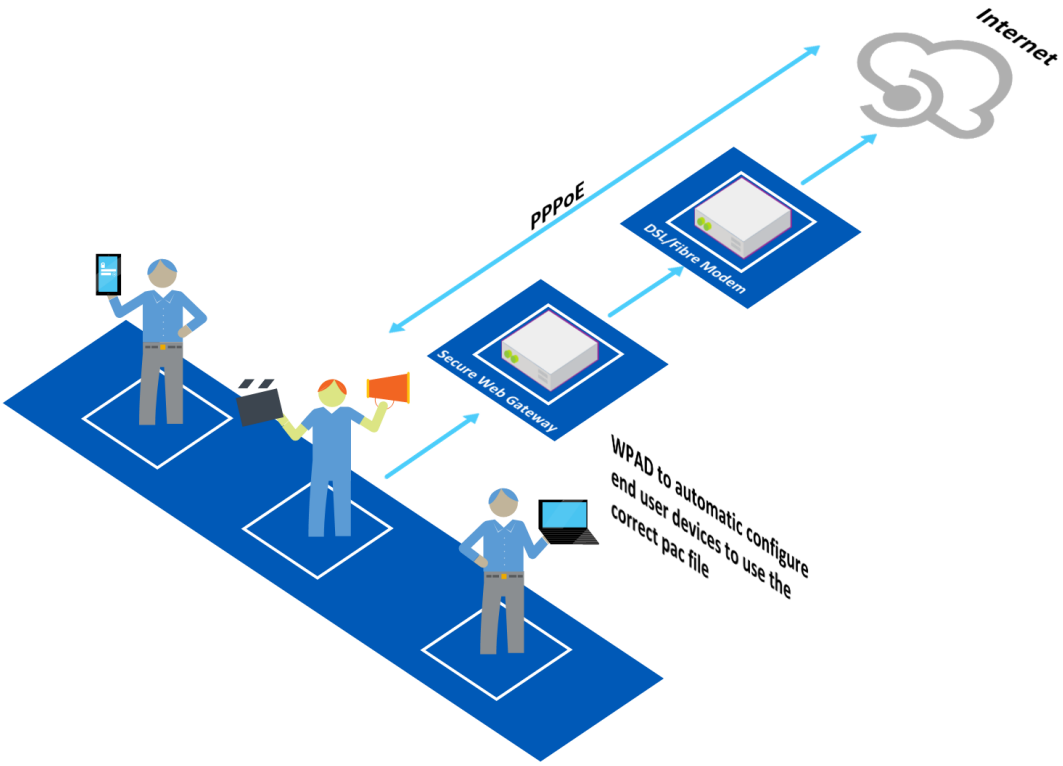


Figure 3.2: High Level Network Architecture of the Secure Web Gateway Implementation

on a vDSL connection that supports Point-to-Point Protocol over Ethernet (PPPoE).

3.4.2 System Architecture

A layered defence is needed to meet requirement R1. Fig. 3.3 shows the sequence of filtering done by the Secure Web Gateway. The function of each defence layer is described below:

IPS/IDS

IPS/IDS fulfils the role of network-based filtering and can classify the traffic as it traverses through the Secure Web Gateway. Traffic classification provides the ability to map a given stream of packets to a known pattern of malicious traffic.

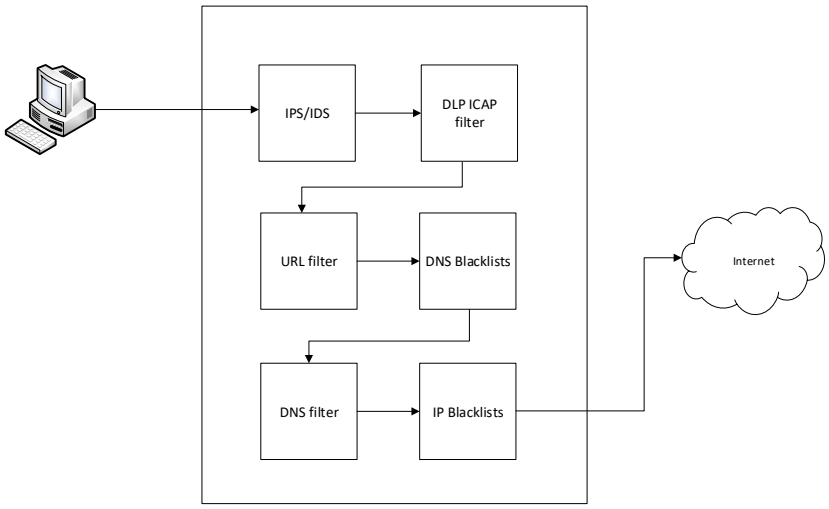


Figure 3.3: The Secure Web Gateway Block sequence

DLP ICAP filter

There are two ways to prvent the leakage of data in motion by using the Secure Web Gateway. The first option is to use IPS/IDS to identify certain types of data from the network traffic. The downside of using IPS/IDS is that it will block all traffic to the same destination IP that triggered the rule. DLP rules within IPS/IDS are usually set to alert only. The second option is to integrate a third-party DLP solution with the Web Proxy through the use of ICAP protocol.

URL filter

URL filtering is the core function of every Secure Web Gateway. The purpose is to allow a network administrator to design a filtering policy based on the website categories. A URL filter uses a pre-existing blacklist that is a collection of URL lists grouped into different categories.

DNS Blacklist

DNS is a critical infrastructure that a network administrator needs to have control of. DNS can be used by attackers to deliver malware [72, 73] or attackers may use it to control compromised PCs [74, 75]. DNS Blacklist is a domain blacklist maintained by some organisations on the Internet.

DNS filter

As discussed in the previous chapter, DNS filters achieve blocking by preventing clients from resolving malicious domain names into IP addresses. DNS is the most common rendezvous-based filtering type.

IP Blacklist

IP Blacklist, like DNS Blacklist is maintained by some organisations on the Internet, and it contains a list of IP address that have been detected as generating malicious traffic. Some of the malicious traffic is communicated over IP addresses instead of domain names, so IP Blacklist is also required.

3.4.3 Hardware Design

Because of the chosen network architecture above, the Secure Web Gateway needs to be built upon a network firewall/router that is capable of routing network traffic. Hence, the hardware needs to support the firewall solutions currently available on the market and must also be able to be customised as a Secure Web Gateway solution. The following two types of hardware were considered: 1) Renting a VM from a cloud provider such as Google, Amazon or Microsoft; and 2) A physical Intel-based hardware architecture that most firewall/router software can run on. The first option offers the most re-usability as a single implementation can be re-used easily across multiple premises, but the downside is the performance impact of routing all the traffic from the on-premise network through the cloud network. Table 3.1 provides a comparison between the Cloud VM and Physical Device. Both options can satisfy the requirements above. This implementation was done using a physical device,

but the design can be easily ported to a cloud VM for other usage scenarios.

Appendix B documents the hardware and software components along with the configurations that were used in an implementation on a residential vDSL connection.

Table 3.1: Comparison of cloud VMs and physical device

	Cloud VM	Physical Device
Re-Usability	High	Low
Performance Impact	High	Low
Cost	High operational cost but low initial cost	High Initial cost of buying the hardware + small ongoing operational cost such as electricity and hardware maintenance
Scalability	Highly scalable	Limited to what physical hardware can offer
Usage Scenarios	Business that has multiple offices and require single enforcement point	Home or office that has single premise

Chapter 4

Findings

In this chapter, a Secure Web Gateway implementation is tested using a framework based on a subset of the testing methodology developed by the NSS Labs [76]. Refer to Appendix A for the technical details of the testing framework. The challenge of the testing is to find reputable known-good or known-bad sites that the test result can be compared against, and there is simply no single source of truth to determine whether a website is malicious or benign. After research, Google Safe Browsing and the Web of Trust were chosen to build the control group for the following reasons: 1) Both offer API that allows the use of a script to check the reputation of a website automatically; and 2) Both data sets are compiled from a substantial amount of data. Thus, the assumption of this experiment is that both sources provide close-to-accurate information about the reputation of a website. The final result is further compared with testing results against a Sophos UTM implementation, which is a commercial Secure Web Gateway freely available for home or personal use [77]. Sophos UTM is enabled for Intrusion Prevention, Web Filtering and Advanced Threat Protection. Web filtering is specifically configured for blocking malicious categories only. Sophos UTM is tested using the same framework as the one used for testing our own implementation. Only a sample of data is included; the full data set is available on the Cyber Security Researchers of Waikato (CROW) website: https://crow.org.nz/people/jeffrey.

1. **Google Safe Browsing** is a blacklist service run by Google that lists the web resources that contain malware, phishing, or unwanted software

that is either deceptive or hard to uninstall or sites that have been compromised [78]. Google Safe Browsing is used by more than 1 billion people all over the world and discovers over 50,000 malware sites and 90,000 phishing sites every month.

2. **Web of Trust (MyWOT)** was founded in 2006 and by 2013, it had over 100 million downloads. MyWOT provides a website reputation and review service that gives users the ability to provide a rating to a website through a browser add-in. Its data is based on a combination of crowdsourced reviews and data from other sources [79].

4.1 URL Filter Testing

The aim of this test is to determine the effectiveness of the URL filtering capability. The script attempts to access the Alexa Top one million sites through the Secure Web Gateway [80] and records the result of the access based on the response it gets. Table 4.1 shows the first 20 sites of the one million list that are being tested. The script starts with resolving the DNS of a site to an IP address. If the Secure Web Gateway blocks the site, the IP address should be resolved into an IP address assigned by the Secure Web Gateway as shown in Table 4.2. Next, the script attempts to access the site using the script and then inspects the response and the returned status code. The result is then classified according to the returned status code and the response body of the request. The result of the access should equal one of these three states: allow, denied or unknown. The final result is cross-referenced to the test results using both Google Safe Browsing API and Web of Trust API.

4.1.1 Google Safe Browsing Comparison

Table 4.3 shows the first thirty sites in the Alexa top one million that were classified as unsafe by Google Safe Browsing and Table 4.4 shows the number of sites in each threat type. When comparing this result against the test results using the Secure Web Gateway, as shown in Fig. 4.1, 30% of unsafe websites blacklisted by Google Safe Browsing API were also blocked by the Secure Web

Table 4.1: First twenty sites of Alexa one million sites

Key	Domain
1	google.com
2	youtube.com
3	facebook.com
4	baidu.com
5	yahoo.com
6	wikipedia.org
7	google.co.in
8	tmall.com
9	qq.com
10	amazon.com
11	sohu.com
12	google.co.jp
13	taobao.com
14	live.com
15	vk.com
16	twitter.com
17	linkedin.com
18	360.cn
19	instagram.com
20	yahoo.co.jp

Gateway, 60% were allowed to pass through the Secure Web Gateway, and 10% were unknown, which means they were either unreachable or inaccessible through the Secure Web Gateway at the time of testing. When this result is further compared with the result of access through the Sophos UTM, both results are similar. As shown in Fig. 4.2, 798 unsafe websites were not blocked by Sophos UTM, and 864 websites were not blocked by the Secure Web Gateway. The Secure Web Gateway had a slightly higher miss rate. If one extends the blocking categories to include questionable categories like Parked Domain, Illegal Software, Suspicious URLs, Potentially Unwanted Programmes, and hacking/computer crimes, one can see a decrease in the percentage of miss-blocked websites to about 40%, down from the original 60%. Tables 4.5 and 4.6 show the access results of the first thirty unsafe websites through Sophos UTM and our Secure Web Gateway. swg_filter_type column depicts the type of filtering engine that was triggered. The result column shows that access is either allowed or blocked by the filtering engine.

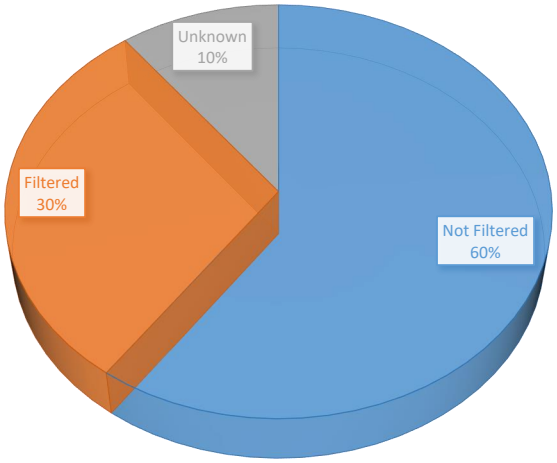


Figure 4.1: Unsafe Websites blocked by the Secure Web Gateway

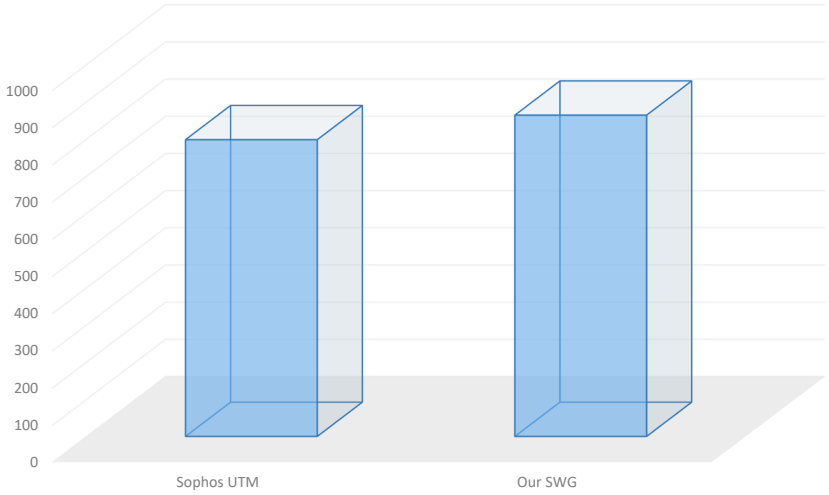


Figure 4.2: Unsafe Websites not detected by Sophos UTM vs. our Secure Web Gateway

Table 4.2: OpenDNS & pfBlockerNG Block Page IP

Name	IP Address
pfBlockerNG block page	10.10.10.1
Domain List Block Page	146.112.61.104
Botnet Block Page	146.112.61.105
Content Category Block Page	146.112.61.106
Malware Block Page	146.112.61.107
Phishing Block Page	146.112.61.108
Suspicious Response Block Page	146.112.61.109
Security Integrations Block Page	146.112.61.110

4.2 False-Positive Testing

This test is designed to test the Secure Web Gateway's ability to identify and allow legitimate traffic while maintaining protection against malicious traffic. The test methodology is similar to URL filtering, but the result is cross-referenced to the website reputation provided by the Web of Trust. Web of Trust classifies websites into different categories, and for each assigned category, the Web of Trust also computes a confidence value for it. The higher the confidence value, the more reliable the category assignment is. Out of the one million websites tested, only 347677 websites had a category assigned. Amongst the websites that have a category assigned, 6793 websites were blocked by the Secure Web Gateway. However, it is surprising to learn that out of the websites blocked by the Secure Web Gateway, 5059 sites have the category ID 501 (Good site) which equates to more than 70% false-positives. However, after analysing the cumulative distribution of the confidence value of each assigned category as shown in Fig. 4.3, more than 95% of the assigned 501 category sites had a confidence value less than 60. According to the Web of Trust, a confidence value greater 60 is considered to be a reliable rating. Based on this calculation, the false-positive rate approximately 3.9%. The result is also compared with the test results of access through the Sophos UTM; as shown in Fig. 4.4, both display a similar false-positive rate. If one extends the blocking categories to include questionable categories, as shown in Fig. 4.5, the false-positive rate is increased to about 15%, more than three times higher than without questionable categories. Table 4.7 and 4.8

Table 4.3: First thirty websites classified as unsafe by Google Safe Browsing

target	threat_type
askcom.me	SOCIAL_ENGINEERING
xossip.com	SOCIAL_ENGINEERING
cloudscar.com	MALWARE
yoo9ier.top	SOCIAL_ENGINEERING
whenvideoupsafesystem4unow.space	SOCIAL_ENGINEERING
mycelebritydaily.com	MALWARE
gossiplab.com	MALWARE
hihable.com	SOCIAL_ENGINEERING
netcentrum.cz	MALWARE
topeasysofttoigetalwaysfree.website	SOCIAL_ENGINEERING
holdlaky.com	SOCIAL_ENGINEERING
preparevideosafesystem4unow.space	SOCIAL_ENGINEERING
preparevideosafesystem4unow.site	SOCIAL_ENGINEERING
aiohow.tv	SOCIAL_ENGINEERING
rapidvideo.org	SOCIAL_ENGINEERING
lazymor.com	SOCIAL_ENGINEERING
prepare2upvideosafesystem4setnow.online	SOCIAL_ENGINEERING
prepare2upvideosafesystem4setnow.site	SOCIAL_ENGINEERING
bestsoftsforyourmachinetoday.website	SOCIAL_ENGINEERING
yeabests.cc	SOCIAL_ENGINEERING
prepare2upvideosafesystem4setnow.pw	SOCIAL_ENGINEERING
upnow2appsafesystemset4now.online	SOCIAL_ENGINEERING
yunweiwei.com	MALWARE
westbats.com	SOCIAL_ENGINEERING
westbeds.com	SOCIAL_ENGINEERING
76mi.com	MALWARE
zatan.com	MALWARE
naturalbd.com	SOCIAL_ENGINEERING
keyupgrade45678safesystems.website	SOCIAL_ENGINEERING
pleaseupdatesafesystemset4now.host	SOCIAL_ENGINEERING

Table 4.4: Number of websites classified as unsafe by Google Safe Browsing

Threat Type	Count
Malware	800
Social_Engineering	643
Total	1443

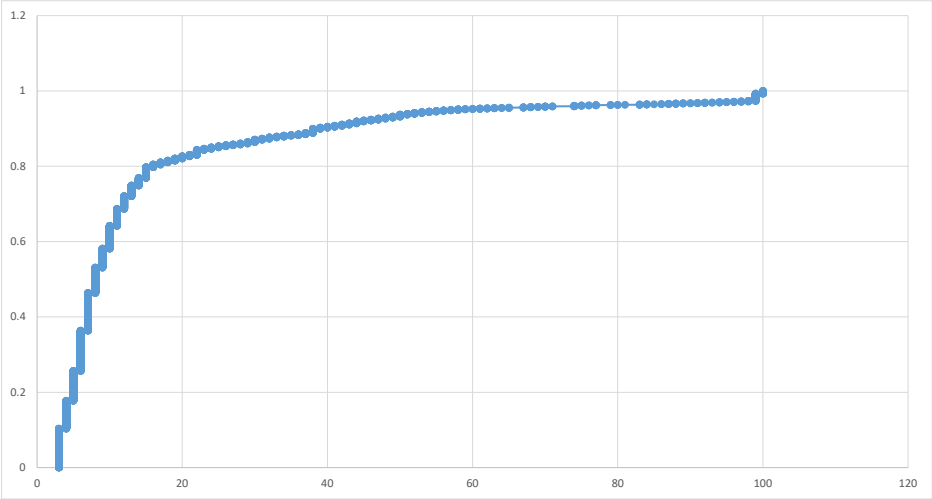


Figure 4.3: CDF Confidence Value of benign websites blocked by SWG

show websites blocked by our Secure Web Gateway and Sophos UTM with WOT Category ID 501 and confidence value greater than 60.

4.3 Exploit Testing

This test focuses on client-side initiated attacks because the main function of the Secure Web Gateway is to protect the outbound traffic generated by the end-user device. This test leverages the Metasploit framework, a tool for developing and executing exploit code against a remote target machine [81]. This includes exploits such as reverse shell, a bind shell that allows an attacker to execute arbitrary commands, install a malicious payload and render the system unresponsive. In this testing, a Windows 7 VM was setup to run behind the Secure Web Gateway, and the Metasploit framework was running

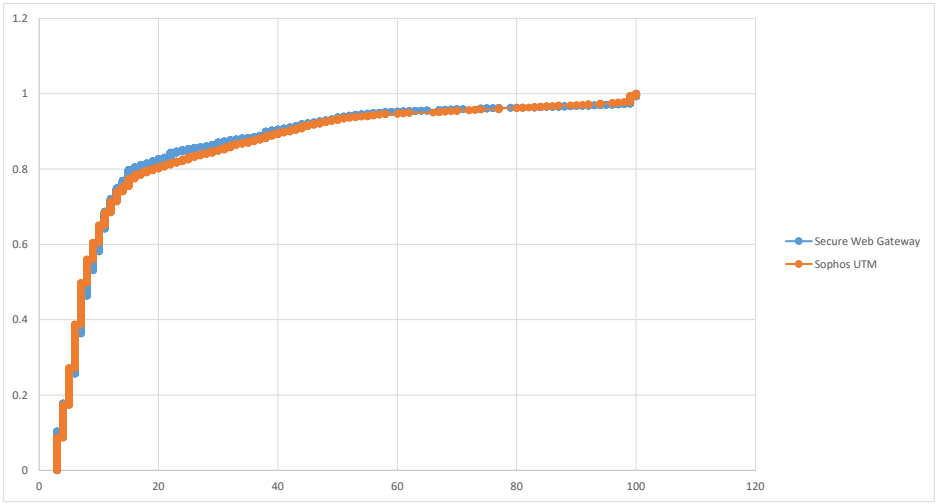


Figure 4.4: CDF Confidence Value of benign websites blocked by Sophos VS. SWG

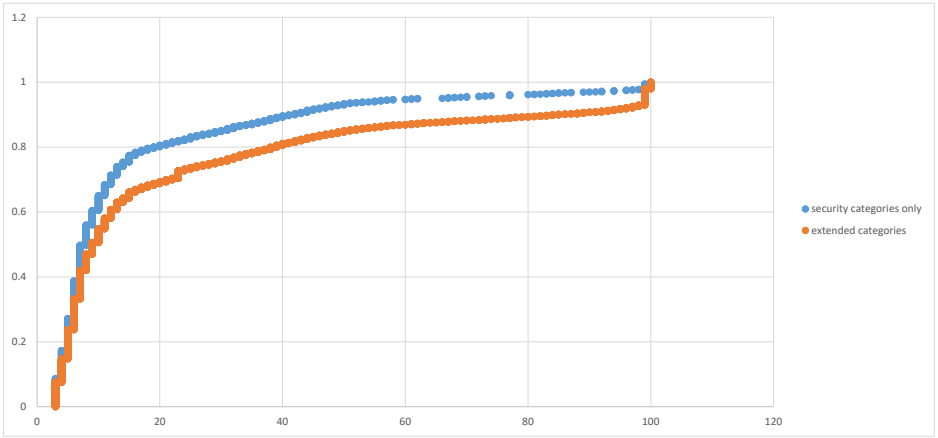


Figure 4.5: CDF Confidence Value of benign websites blocked by using security-only categories vs extended categories

on a virtual server hosted in the cloud. A reverse_tcp payload was created using the Metasploit framework and configured to connect back to a custom port of the virtual server in the cloud. The payload (EXE) file was delivered via a manual transfer method. The payload file was first compressed with a password to evade AV detection during the transfer. The payload was then manually run on the local Windows 7 machine, and the traffic between the local Windows 7 machine and the Metasploit server in the cloud was monitored on the Secure Web Gateway. With the initial setup, both My Secure Web Gateway and Sophos UTM failed to detect and block the traffic. After a few exploitations, it was discovered that setting IPS Policy to Maximum Detection would allow the IPS to include the required signatures to detect this malicious traffic. Maximum Detection policy contains vulnerabilities with a CVSS score of at least 7.5, published after 2005 or later, and also critical malware and exploit kit rules. In this testing, no performance degradation was observed as a result of choosing this setting. However, changing the setting to "Maximum Detection" also led to a much higher false-positive rate. The false-positive rate is reduced by using the suppression list and the SID Management Configuration files to disable rules that triggered false-positive alerts. These settings need to be tuned according to the traffic pattern of the network and cannot be set-and-forget.

4.4 DLP Testing

This provides a summary of the steps that had been followed to test the effectiveness of the MyDLP solution.

1. Created a policy in MyDLP to detect Social Security Numbers and Credit Card Numbers

2. Created a spreadsheet or word document populated with names, social security numbers and credit card numbers. Sample data was downloaded from the DLPTest website [82].

3. Used the HTTP Post function on the DLPTest website to upload the

document prepared in step 1 [82].

4. Checked the Logs in the MyDLP admin interface to determine whether the traffic had been detected.

The testing was completed successfully with traffic violating the policy correctly detected by MyDLP. However, sometimes, MyDLP only reported the detection of one information type even when the sample data contained more than one type.

Table 4.5: First twenty unsafe websites checked by Sophos UTM

Threat Type	SWG Filter Type	SWG Test Date	SWG System Name	Target	Result
SOCIAL_ENGINEERING	Malicious Sites	4/03/2017 16:26	sophos	askcom.me	Block
SOCIAL_ENGINEERING	url	26/02/2017 18:29	sophos	xossip.com	Allow
MALWARE	url	26/02/2017 19:04	sophos	cloudscar.com	Block
SOCIAL_ENGINEERING	403 Forbidden	4/03/2017 16:30	sophos	yoo9ier.top	Block
SOCIAL_ENGINEERING	url	4/03/2017 11:16	sophos	whenvideoupsafesystem4unow.space	Block
MALWARE	port:Troj/JSRedir-RX	4/03/2017 20:23	sophos	mycelebritydaily.com	Block
MALWARE	port:Troj/JSRedir-RX	4/03/2017 20:23	sophos	gossiplab.com	Block
SOCIAL_ENGINEERING	Phishing	4/03/2017 16:30	sophos	hihable.com	Block
MALWARE	url	26/02/2017 19:54	sophos	netcentrum.cz	Block
SOCIAL_ENGINEERING	url	4/03/2017 11:17	sophos	topeasysofttoigetalwaysfree.website	Block
SOCIAL_ENGINEERING	url	26/02/2017 19:56	sophos	holdlaky.com	Allow
SOCIAL_ENGINEERING	url	4/03/2017 11:17	sophos	preparevideosafesystem4unow.space	Block
SOCIAL_ENGINEERING	url	4/03/2017 11:17	sophos	preparevideosafesystem4unow.site	Block
SOCIAL_ENGINEERING	url	26/02/2017 20:32	sophos	aiohow.tv	Allow
SOCIAL_ENGINEERING	url	26/02/2017 20:37	sophos	rapidvideo.org	Allow
SOCIAL_ENGINEERING	Parked Domain	4/03/2017 16:33	sophos	lazymor.com	Allow
SOCIAL_ENGINEERING	url	4/03/2017 11:19	sophos	prepare2upvideosafesystem4setnow.online	Block
SOCIAL_ENGINEERING	url	4/03/2017 11:19	sophos	prepare2upvideosafesystem4setnow.site	Block
SOCIAL_ENGINEERING	url	4/03/2017 11:19	sophos	bestsoftsforyourmachinetoday.website	Block
SOCIAL_ENGINEERING	Malicious Sites	4/03/2017 16:35	sophos	yeabests.cc	Block

Table 4.6: First twenty unsafe websites checked by our Secure Web Gateway

Threat Type	SWG Filter Type	SWG Test Date	SWG System Name	Target	Result
SOCIAL_ENGINEERING	url	28/03/2017 17:45	pfSense	askcom.me	Allow
SOCIAL_ENGINEERING	url	28/03/2017 18:15	pfSense	xossip.com	Allow
MALWARE	url	3/04/2017 19:20	pfSense	cloudscar.com	Block
SOCIAL_ENGINEERING	url	4/04/2017 20:53	pfSense	yoo9ier.top	Block
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:54	pfSense	whenvideoupsafesystem4unow.space	Block
MALWARE	url	28/03/2017 19:58	pfSense	mycelebritydaily.com	Allow
MALWARE	url	28/03/2017 19:58	pfSense	gossiplab.com	Allow
SOCIAL_ENGINEERING	url	4/04/2017 20:54	pfSense	hihable.com	Block
MALWARE	url	28/03/2017 20:30	pfSense	netcentrum.cz	Allow
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:55	pfSense	topeasysofttoigetalwaysfree.website	Block
SOCIAL_ENGINEERING	url	28/03/2017 20:34	pfSense	holdlaky.com	Allow
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:55	pfSense	preparevideosafesystem4unow.space	Block
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:55	pfSense	preparevideosafesystem4unow.site	Block
SOCIAL_ENGINEERING	url	28/03/2017 21:31	pfSense	aiohow.tv	Allow
SOCIAL_ENGINEERING	url	28/03/2017 21:39	pfSense	rapidvideo.org	Allow
SOCIAL_ENGINEERING	url	28/03/2017 21:44	pfSense	lazymor.com	Allow
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:57	pfSense	prepare2upvideosafesystem4setnow.online	Block
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:57	pfSense	prepare2upvideosafesystem4setnow.site	Block
SOCIAL_ENGINEERING	DNS Not Exist	4/04/2017 20:57	pfSense	bestsoftsforyourmachinetoday.website	Block
SOCIAL_ENGINEERING	url	28/03/2017 22:36	pfSense	yeabests.cc	Allow

Table 4.7: First twenty websites blocked by our Secure Web Gateway with WOT Category ID = 501 & Confidence Value >60

Category Identifier	Category Confidence	SWG Filter Type	SWG Test Date	SWG System Name	Target
501	70	blk_BL_spyware	4/04/2017 20:51	pfSense	nudevista.com
501	100	blk_BL_spyware	4/04/2017 20:51	pfSense	ebates.com
501	100	IPBL	28/03/2017 18:06	pfSense	economist.com
501	98	IPBL	28/03/2017 18:06	pfSense	tvn24.pl
501	100	IPBL	28/03/2017 18:12	pfSense	sci-hub.cc
501	88	IPBL	28/03/2017 18:15	pfSense	oriflame.com
501	100	IPBL	28/03/2017 18:16	pfSense	ouest-france.fr
501	100	IPBL	28/03/2017 18:16	pfSense	laredoute.fr
501	100	IPBL	28/03/2017 18:21	pfSense	leroymerlin.fr
501	100	IPBL	28/03/2017 18:22	pfSense	staseraintv.com
501	100	IPBL	28/03/2017 18:22	pfSense	gametop.com
501	99	IPBL	28/03/2017 18:22	pfSense	bankersadda.com
501	100	IPBL	28/03/2017 18:24	pfSense	evite.com
501	100	OpenDNS-Content Category Block	28/03/2017 18:24	pfSense	igg-games.com
501	99	timed out	10/04/2017 9:20	pfSense	fritz.box
501	95	OpenDNS-Content Category Block	28/03/2017 18:30	pfSense	skidrow-games.com
501	68	IPBL	28/03/2017 18:35	pfSense	indgovtjobs.in
501	97	IPBL	28/03/2017 18:36	pfSense	animeid.tv
501	100	IPBL	28/03/2017 18:37	pfSense	virustotal.com
501	100	IPBL	28/03/2017 18:46	pfSense	surrenderat20.net

Table 4.8: First twenty websites blocked by Sophos UTM with WOT Category ID = 501 & Confidence Value >60

Category Identifier	Category Confidence	SWG Filter Type	SWG Test Date	SWG System Name	Target
501	100	Spam URLs	4/03/2017 16:26	sophos	rutor.info
501	99	Malicious Sites	4/03/2017 16:26	sophos	bc.vc
501	99	Malicious Sites	4/03/2017 16:27	sophos	way2sms.com
501	99	Malicious Sites	4/03/2017 16:28	sophos	pornbb.org
501	94	Malicious Sites	4/03/2017 16:28	sophos	yify.tv
501	99	Malicious Sites	4/03/2017 16:29	sophos	icefilms.info
501	81	Malicious Sites	4/03/2017 16:30	sophos	rgho.st
501	89	Malicious Sites	4/03/2017 16:31	sophos	intporn.com
501	92	Spam URLs	4/03/2017 16:32	sophos	profitcentr.com
501	96	Malicious Sites	4/03/2017 16:32	sophos	the-cinema.ru
501	92	Malicious Sites	4/03/2017 16:32	sophos	monosnap.com
501	99	Spam URLs	4/03/2017 16:32	sophos	fast-bit.org
501	99	Malicious Sites	4/03/2017 16:32	sophos	serials.ws
501	99	Spam URLs	4/03/2017 16:36	sophos	ru-free-tor.org
501	99	Spam URLs	4/03/2017 16:36	sophos	open-tor.org
501	70	Spam URLs	4/03/2017 16:36	sophos	porn720.net
501	91	Spam URLs	4/03/2017 16:37	sophos	leporno.org
501	99	Spam URLs	4/03/2017 16:38	sophos	ru-tor.net
501	94	Malicious Sites	4/03/2017 16:39	sophos	shuame.com
501	99	Spam URLs	4/03/2017 16:39	sophos	free-tor.org

Chapter 5

Discussion

This section discusses the gaps and challenges identified in this research. The main challenge in content-filtering is the balance between false-positive and false-negative rate. As the testing showed in Section 4.2, increasing the number of blocking categories increased the detection rate, but at the same time, it also increased the false-positive rate. There is no silver bullet for dealing with this issue due to the organic and volatile nature of Web resources. In Sections 5.2 and 5.3, the author concludes that a Secure Web Gateway cannot be used alone to defend against DLP and APT. A Secure Web Gateway needs to cooperate with other countermeasures to form multiple layers of defence; this is the best way to thwart these problems.

5.1 Gaps in the content filtering factors

5.1.1 Network-based filtering

There are several challenges presented in the network-based filtering.

- URL filtering is useful for detecting known threats but is ineffective against advanced threats and zero-day attacks. It is also becoming difficult to manage because of the exponential growth rate of new Internet sites.

- Port-based and signature-based malware detection are very cost-effective and accurate methods to detect known and static malware but are very

inefficient against advanced malware that leverages zero-day vulnerabilities. Other behaviourally and statistically based detection techniques can compensate for the shortfall in detection capability but have low accuracy and high false-positives.

5.1.2 Rendezvous-Based Filtering Type

The rendezvous-based filtering method is one of the easiest implementations of Internet filtering, but it is often accompanied by some collateral damage. For example, DNS injection is a very popular mechanism for filtering Internet traffic, but the enforcing party is often unaware of the collateral damage of such filtering may potentially affect users outside of its network. This is solely because the DNS traffic is routed through the censored network. A famous example happened in 2010 in which queries from Chile were routed through a Chinese root server [83].

5.1.3 Endpoint-Based Filtering Type

The main challenge of Endpoint-Based filtering is that it requires the cooperation of the endpoints. The problem is exacerbated when there is a diverse set of endpoints such as mobile, laptop, desktop, and tablet. It is a challenge to deploy the required software update to all endpoints, and it is subject to circumvention if users possess administrative privileges on the endpoint.

A Secure Web Gateway is primarily operated in proxy mode, which provides full control over the TCP connection. However, a proxy-based Secure Web Gateway can cause user experience degradation issues for the following reasons: 1) Not all applications and devices can operate in proxy mode especially modern devices like Internet of Things (IoT), smartphones and tablets; and 2) It can introduce unwanted latency in web application performance, which is especially noticeable in latency-sensitive applications such as web conferencing and voice over IP (VoIP) software. The latency issue is exacerbated by the increasing use of encryption in web traffic and the need to decrypt and intercept encrypted traffic. A Secure Web Gateway needs to be transformed into an overarching security platform by integrating with different types of

filtering and security controls such as a network-based firewall.

5.2 Gaps in the DLP factors

DLP cannot be just "set and forget" and must involve business stakeholders to develop a strategy for how an organisation should address data leakage [84]. DLP techniques used within the Secure Web Gateway are primarily based on rules and dictionaries that are suitable for structured data. More advanced DLP detection such as statistical and conceptual analysis often needs to be offloaded to other dedicated DLP solution via a protocol like Internet Content Adaptation Protocol (ICAP). However, most detection techniques for unstructured data have a high false-positive rate.

5.3 Gaps in the APT factors

Defending against APT requires improvement of the overarching security of an organisation. Although a Secure Web Gateway can inspect, filter and monitor inbound content and outbound Internet Web communications, it is still operated in a silo and does not exchange information with other networks, edges, endpoints and data security systems. This can reduce an organisation's ability to prevent, detect and respond to an APT [85]. For example, an organisation can integrate its Secure Web Gateway with the SIEM to improve contextual awareness and provide a higher-level alert management capability.

5.4 Gaps in the SSL filtering factors

SSL Interception Proxy breaks the privacy of encryption and the issue is exacerbated when the public subCA is used [86]. The following risks are introduced by an interception proxy:

- **Legal Exposure** An organisation that implements an interception proxy may face increased legal exposure as employees may expect privacy in communication with confidential websites such as banking or health websites.

- **Increased threat surface** SSL Interception Proxy becomes a single point where all encrypted sessions can be viewed in plain text. Attackers can compromise the interception proxy to inspect and potentially modify plain-text contents of any encrypted sessions.

- **Decreased cipher strength** As the cipher suites of client and server SSL sessions are negotiated independently, the strength of the cipher of the SSL session is determined by the strongest cipher supported by the interception proxy. There is a possibility that the interception proxy supports a weaker cipher than the client endpoint.

- **Transitive Trust** SSL Interception Proxy introduces a phenomenon called "transitive trust" in which if the SSL client trusts the SSL proxy and the SSL proxy trusts the SSL server, then the SSL client trusts the SSL server. In [51], Jarmoc and Unit suggest that "transitive trust" can expose several flaws in the operation of the SSL protocol.

Direct Validation of Certificates (DVCert) [87], SSL/TLS Session-Aware User Authentication [88, 89], the proposed TLS-SRP protocol [90], and Google's proposal of Certificate Pinning [91] are all security countermeasures to combat MITM attacks. These may stop the legitimate SSL proxy server from accessing encrypted websites as the communication may be detected as an MITM attack.

5.4.1 BlindBox

BlindBox offers a potential solution to the aforementioned issues, but the proposed architecture requires cooperation from both client and server endpoints and additional computational overhead for the client endpoint for computing hashes of traffic segments, which make it difficult for the current browser and web server to adopt this method.

5.4.2 PRI

PRI offers a simpler solution with lesser overheads than BlindBox. However, the downside is that it depends on the support of a specific type of hardware

- Intel Software Guard Extensions. SGX was introduced in 2015 with Intel Core microprocessors based on the Skylake microarchitecture. At the time of writing, it is still difficult to find suitable hardware that supports the SGX feature and also meets other criteria. Also, in [92], Costan and Devadas raised a concern with a control feature in SGX that requires software developers to enter a business agreement with Intel. For the author of the software to take advantage of the SGX's protection, the software must obtain a SGX attestation key from Intel [93].

Chapter 6

Concluding Remarks

This chapter discusses the key outcomes and value of this research and concludes by suggesting potential future work in the field of intercepting SSL/TLS encrypted traffic.

6.1 Summary

This project achieved the majority of its goals, with some gaps and challenges as discussed in the previous chapter. The experiment showed that the prototype can achieve similar effectiveness to the other commercial alternatives. This is done by leveraging open-source and publicly available information such as DNS/IP blacklists, IPS/IDS, integrated DLP solution, URL filters, and DNS filters. The prototype created in this research had only a 4% higher miss rate than the commercial alternative. However, the conundrum is that reducing the miss rate by blocking questionable websites would also increase the false positive rate, thus blocking legitimate sites. This study found that the DLP function in a Secure Web Gateway is often delivered through Integrated DLP as opposed to Enterprise DLP. A Secure Web Gateway monitors network traffic as data traverses through it and applies DLP techniques to identify any data leakage. This implementation integrated with a third-party DLP solution via ICAP protocol and successfully identified leakage of items such as credit card numbers and social security numbers. Also, it found the DLP can sometimes lead to a high false positive rate when certain techniques like statistical and

contextual/conceptual analysis are used. A Secure Web Gateway is just part of the solution to thwart APT. A Secure Web Gateway can break an APT lifecycle by controlling the delivery of malware via the Web and communication with Command and Control servers. This research demonstrated the ability of the prototype to stop the execution of a Metasploit reverse shell by successfully identifying and blocking the communication from the reverse shell to the control server. This was achieved by installing the IPS/IDS on the Secure Web Gateway with correct signature to identify a malicious traffic pattern. To date, the most effective way to inspect SSL/TLS encrypted traffic is by leveraging SNI filtering and adaptive security models. SNI filtering can stop the establishment of an SSL tunnel with known malicious websites but cannot detect malicious content delivered through benign websites. The adaptive security model is a way to achieve a balance between privacy and security by only decrypting privacy-insensitive sites using the MITM technique. This implementation only partially achieves this goal by using an SSL proxy with an access control list and is unable to implement a full adaptive security model. Lastly, a guideline on how to implement a Secure Web Gateway in a workplace to avoid unnecessary risk of legal liability and regulatory compliance is provided, and along with a summary of key elements that should be included in a company's computer use policy. A well-implemented computer use policy is the key to the successful implementation of a Secure Web Gateway.

This thesis has also established the possibility of using commodity hardware and open source technologies at a relatively low cost, yet still achieving the same effectiveness as other commercial alternatives. This enables anyone who has some IT background to set one up for home or small business use. The system architecture developed by this research can also be easily transported to a cloud platform, making it possible to share the same Secure Web Gateway between multiple businesses or households. Another advantage of having the Secure Web Gateway in the cloud is the ability to protect mobile traffic over 3G/4G or public WIFI networks. The full implementation details are described in Appendix. B. Finally, the testing framework developed by this research can also be used to verify the effectiveness of other Secure Web

Gateway implementations. The system architecture of the test environment and the description of each artefact is included in Appendix A.

6.2 Future Work

The biggest challenge is SSL-filtering, which remains an open issue as discussed in Section 5.4. As the industry is pushing harder and harder for encrypting all web resources with SSL/TLS encryption, this can potentially render the Secure Web Gateway useless. During the writing of this thesis, the specification for TLS 1.3 was released, and MITM is no longer possible with TLS 1.3 [94]. Major browsers like Chrome and Firefox are displaying a warning about insecure login pages [95]. Many reports showed that HTTPs adoption has doubled in the last 12 months [96, 97]. Although both the BlindBox proposed by Sherry et al. in [56] and PRI proposed by Schiff and Schmid in [58] offer some promise in solving this conundrum, they all have some limitations and further work is required to make these solutions ready for production use.

References

[1] Gartner. *Secure Web Gateway*. URL: http://www.gartner.com/it-glossary/secure-web-gateway/ (visited on 10/21/2016).

[2] Jamshed Vesuna et al. "Caching Doesn't Improve Mobile Web Performance (Much)". In: *2016 USENIX Annual Technical Conference (USENIX ATC 16)*. USENIX Association. 2016.

[3] James O'Toole. "Mobile apps overtake PC Internet usage in US". In: *Retrieved on July* 11 (2014), p. 2014.

[4] BLOXX Ltd. *Protecting Your Network Against Risky SSL Traffic*. 2015. URL: https://www.bloxx.com/media/1360/bloxx_whitepaper_protectnetwork_fromssl_us.pdf (visited on 10/14/2016).

[5] CS Alliance. *SecaaS Implementation Guidance, Category 3: Web Security*. Sept. 2012. URL: https://cloudsecurityalliance.org/download/secaas-category-3-web-security-implementation-guidance/.

[6] C Furlani. *Managing Information Security Risk: Organization, Mission, and Information System View*. 2011.

[7] Jamie Riden. *HOW FAST-FLUX SERVICE NETWORKS WORK*. 2008. URL: http://www.honeynet.org/node/132 (visited on 12/17/2016).

[8] Data Losss Prevention Experts (DLPX). *PREVENTING DATA LOSS = DLP + ICAP PROXY*. 2014. URL: http://dlpexperts.com/2014/09/preventing-data-loss-dlp-icap-proxy-2/ (visited on 12/24/2016).

[9] Kelly Kavanagh, Oliver Rochford, and Toby Bussa. *2016 Magic Quadrant for SIEM*. 2016. URL: http://www.gartner.com/it-glossary/security-information-and-event-management-siem/ (visited on 02/23/2017).

[10] Social Engineering. URL: http://www.social-engineer.org/ (visited on 12/15/2016).

[11] Ramon Caceres et al. "Web proxy caching: The devil is in the details". In: *ACM SIGMETRICS Performance Evaluation Review* 26.3 (1998), pp. 11–15.

[12] Martin Arlitt et al. "Evaluating content management techniques for web proxy caches". In: *ACM SIGMETRICS Performance Evaluation Review* 27.4 (2000), pp. 3–11.

[13] Victor Agababov et al. "Flywheel: Google's data compression proxy for the mobile web". In: *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. 2015, pp. 367–380.

[14] Newstex. *CompaniesandMarkets.com: Secure web gateway market expected to grow by a CAGR of 20% until 2020*. May 2015. URL: http://search.proquest.com.ezproxy.waikato.ac.nz/docview/1682224415/fulltext/6A27059AA3ED4978PQ/1?accountid=17287 (visited on 10/21/2016).

[15] Google. *Magic Quadrant for Secure Web Gateways*. 2016. URL: http://www.gartner.com/home.

[16] United Nations Interregional Crime and Justice Research Institute. *Cyber Threats*. 2016. URL: http://www.unicri.it/special_topics/securing_cyberspace/cyber_threats/ (visited on 10/14/2016).

[17] Louis Marinos. *ENISA Threat Landscape*. 2014. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2014 (visited on 10/14/2016).

[18] John Pescatore and Greg Young. "Defining the next-generation firewall". In: *Gartner RAS Core Research Note, from http://www. ga1tner. com* (2009).

[19] Paula Musich. *Is the Secure Web Gateway Market Disappearing?* July 28, 2016. URL: https://www.nsslabs.com/blog/is-the-secure-web-gateway-market-disappearing/ (visited on 10/21/2016).

[20] Paula Musich Joshua Mittler. "CY2015 – Secure Web Gateway". In: (2015). URL: https://www.nsslabs.com/research-advisory/library/content-security/secure-web-gateways/secure-web-gateway-cy2015/.

[21] Jisang Kim et al. "Detection of advanced persistent threat by analyzing the big data log". In: *Advanced Science and Technology Letters* 29 (2013), pp. 30–36.

[22] FORCEPOINT. Ed. by FORCEPOINT Web Security. 2017. URL: https://www.forcepoint.com/product/cloud-security/forcepoint-web-security-cloud (visited on 05/01/2017).

[23] Symantec. *Symantec Secure Web Gateway: PrProxy & ASG*. 2017. URL: https://www.symantec.com/products/web-and-cloud-security/secure-web-gateway-proxy-sg-and-asg (visited on 05/01/2017).

[24] zScaler. *zScaler Internet Access*. 2017. URL: https://www.zscaler.com/products/zscaler-internet-access (visited on 05/01/2017).

[25] Chloe Winter. *Cyber crime continues to rise*. May 18, 2015. URL: http://www.stuff.co.nz/technology/digital-living/68636916/cyber-crime-continues-to-rise (visited on 04/16/2017).

[26] NCSC. *Cyber Security Incidents*. 2017. URL: https://www.ncsc.govt.nz/ (visited on 04/16/2017).

[27] Richard Barnes et al. *Technical Considerations for Internet Service Blocking and Filtering*. Tech. rep. 2016.

[28] Ernst Biersack, Christian Callegari, Maja Matijasevic, et al. *Data Traffic Monitoring and Analysis*. Springer, 2013.

[29] l7filter. *Application Layer Packet Classifier for Linux*. Accessed: 2016-11-04. URL: http://l7-filter.clearos.com/.

[30] Mark A Lemley, David S Levine, and David G Post. "Don't break the internet". In: *Stanford Law Review Online* 64 (2011), p. 34.

[31] Sharon Goldberg. "Why is it taking so long to secure internet routing?" In: *Communications of the ACM* 57.10 (2014), pp. 56–63.

[32] Prathaben Kanagasingham. *Data Loss Prevention*. 2008. URL: https://www.sans.org/reading-room/whitepapers/dlp/data-loss-prevention-32883 (visited on 11/05/2016).

[33] Information Technology Newsweekly. *Data Loss Prevention Market Global Forecast*. May 15, 2016. (Visited on 11/05/2016).

[34] Rich Mogull. *Understanding and Selecting a Data Loss Prevention Solution*. Ed. by Chris Pepper. 2010. URL: https://securosis.com/assets/library/reports/DLP-Whitepaper.pdf (visited on 11/05/2016).

[35] Gartner. *Magic Quadrant for Enterprise Data Loss Prevention 2016*. 2016. URL: https://www.gartner.com/doc/reprints?id=1-2X96R6A&ct=160128&st=sb (visited on 11/05/2016).

[36] McAfee. *Protecting Your Critical Assets: Lessions Learned from "Operation Aurora"*. 2010. URL: https://www.wired.com/images_blogs/threatlevel/2010/03/operationaurora_wp_0310_fnl.pdf (visited on 12/10/2016).

[37] Command Five. *SK Hack by an Advanced Persistent Threat*.

[38] Uri Rivner. *Anatomy of an Attack*. 2011. URL: https://blogs.rsa.com/anatomy-of-an-attack/ (visited on 12/11/2016).

[39] Nart Villeneuve et al. *Operation" Ke3chang: Targeted Attacks Against Ministries of Foreign Affairs*. 2013.

[40] Darien Kindlund et al. *Operation snowman: Deputydog actor compromises US Veterans of Foreign Wars website*. 2014.

[41] Richard Bejtlich. *What Is APT and What Does It Want?* 2010. URL: https://taosecurity.blogspot.co.nz/2010/01/what-is-apt-and-what-does-it-want.html (visited on 12/11/2016).

[42] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains". In: *Leading Issues in Information Warfare & Security Research* 1 (2011), p. 80.

[43] APT Mandiant. "Exposing One of China's Cyber Espionage Units". In: *available from intelreport. mandiant. com/Mandiant_APT1_Report. pdf* (2013).

[44] M Zubair Rafique et al. "Evolutionary algorithms for classification of malware families through different network behaviors". In: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2014, pp. 1167–1174.

[45] Beth Binde, Russ McRee, and Terrence J O'Connor. "Assessing outbound traffic to uncover advanced persistent threat". In: *SANS Institute. Whitepaper* (2011).

[46] Shun-Te Liu, Yi-Ming Chen, and Shiou-Jing Lin. "A novel search engine to uncover potential victims for apt investigations". In: *IFIP International Conference on Network and Parallel Computing*. Springer. 2013, pp. 405–416.

[47] Terrence Oconnor. *Animal Farm: Protection From Client-side Attacks by Rendering Content With Python and Squid*. 2011. URL: `https://www.sans.org/reading-room/whitepapers/intrusion/animal-farm-protection-client-side-attacks-rendering-content-python-squid-33614` (visited on 12/18/2016).

[48] Symantec. *What is an SSL Certificate?* 2017. URL: `https://www.symantec.com/page.jsp?id=ssl-information-center` (visited on 06/18/2017).

[49] David Naylor et al. "The cost of the s in https". In: *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM. 2014, pp. 133–140.

[50] RFC5280. "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile". In: (). URL: http://www.ietf.org/rfc/rfc5280.txt.

[51] Jeff Jarmoc and DSCT Unit. "SSL/TLS interception proxies and transitive trust". In: *Black Hat Europe* (2012).

[52] RFC5019. *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*. URL: http://tools.ietf.org/html/rfc5019.

[53] Suhairi Mohd Jawi, Fakariah Hani Mohd Ali, and Nurul Huda Nik Zulkipli. "a proxy-based adaptive security model for secure socket layer (ssl)". In: (2013). URL: http://www.cybersecurity.my/en/knowledge_banks/journal_conference/main/detail/2387/index.html.

[54] D Eastlake. "RFC 6066-Transport Layer Security (TLS) extensions: Extension definitions". In: (2011).

[55] Wazen M Shbair et al. "Efficiently bypassing SNI-based HTTPS filtering". In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE. 2015, pp. 990–995.

[56] Justine Sherry et al. "Blindbox: Deep packet inspection over encrypted traffic". In: *ACM SIGCOMM Computer Communication Review*. Vol. 45. 4. ACM. 2015, pp. 213–226.

[57] Emerging Threats. *Open Source Signatures*. URL: https://rules.emergingthreats.net/open/snort-2.9.0/rules/ (visited on 11/17/2016).

[58] Liron Schiff and Stefan Schmid. "PRI: Privacy Preserving Inspection of Encrypted Network Traffic". In: *Security and Privacy Workshops (SPW), 2016 IEEE*. IEEE. 2016, pp. 296–303.

[59] Samuel D Warren and Louis D Brandeis. "The right to privacy". In: *Harvard law review* (1890), pp. 193–220.

[60] David Harvey. *Collisions in the Digital Paradigm: Law and Rule Making in the Internet Age*. Bloomsbury Publishing, 2017.

[61] *Privacy At Work*. The Office of the Privacy Commissioner, 2008. URL: https://www.privacy.org.nz/assets/Files/Brochures-and-pamphlets-and-pubs/Privacy-at-Work-2008.pdf.

[62] Rebecca Britton. "An Employer's Right to Pry: A Study of Workplace Privacy in New Zealand". In: *Canterbury L. Rev.* 12 (2006), p. 65.

[63] Paul Roth. "Privacy in the Workplace". In: *Labour, Employment and Work in New Zealand* (1994).

[64] Hazel Oliver. "Email and Internet Monitoring in the Workplace: Information Privacy and Contracting-Out." In: *Industrial Law Journal* 31.4 (2002).

[65] C David Morrison and Robert L Bailey. "Employee Privacy Rights: Employer Monitoring and Investigating Employees' Electronic Activities and Communications". In: *Energy & Mineral Law Institute* 32 (2011), pp. 66–173.

[66] *Capitol Records, Inc. v. THOMAS-RASSET*. 2009.

[67] Scott Cox, Tanya Goette, and Dale Young. "Workplace Surveillance and Employee Privacy: Implementing an Effective Computer Use Policy". In: *Communications of the IIMA* 5.2 (2015), p. 6.

[68] Orin Kerr. *How does the Cybersecurity Act of 2015 change the Internet surveillance laws?* Dec. 24, 2015. URL: http://wapo.st/1J6NdFy?tid=ss_tw&utm_term=.3de8bdc98ac0 (visited on 05/15/2017).

[69] U.S. GAO. *Employee Privacy: Computer-Use Monitoring Practices and Policies of Selected Companies*. Tech. rep. Committee on Education and the Workforce, House of Representatives, 2002.

[70] Alec Wolman et al. "On the scale and performance of cooperative web proxy caching". In: *ACM SIGOPS Operating Systems Review* 33.5 (1999), pp. 16–31.

[71] Anja Feldmann et al. “Performance of web proxy caching in heterogeneous bandwidth environments”. In: *INFOCOM’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. IEEE. 1999, pp. 107–116.

[72] Edmund Brumaghin and Colin Grady. *Covert Channels and Poor Decisions: The Tale of DNSMessenger*. Mar. 2, 2017. URL: http://blog.talosintelligence.com/2017/03/dnsmessenger.html (visited on 03/14/2017).

[73] Steve McKenzie. *VPN over DNS*. May 12, 2016. URL: https://www.shellntel.com/blog/2016/3/30/vpn-over-dns-1 (visited on 03/15/2017).

[74] Josh Grunzweig, Mike Scott, and Bryan Lee. *New Wekby Attacks Use DNS Requests As Command and Control Mechanism*. May 24, 2016. URL: http://researchcenter.paloaltonetworks.com/2016/05/unit42-new-wekby-attacks-use-dns-requests-as-command-and-control-mechanism/ (visited on 03/14/2017).

[75] Raphael Mudge. *Hacking through a Straw (Pivoting over DNS)*. July 9, 2016. URL: https://blog.cobaltstrike.com/2013/07/09/hacking-through-a-straw-pivoting-over-dns/ (visited on 03/15/2017).

[76] NSS Labs Inc. *Test Methodology Secure Web Gateway (SWG)*. 2015. URL: https://www.nsslabs.com/linkservid/A7C6C1D9-5056-9046-93D7AF14BE569E5B/ (visited on 03/27/2017).

[77] Sophos. *Sophos UTM Home Edition*. 2017. URL: https://www.sophos.com/en-us/products/free-tools/sophos-utm-home-edition.aspx (visited on 04/10/2017).

[78] Panayiotis Mavrommatis. *Protecting people across the web with Google Safe Browsing*. Mar. 12, 2015. URL: https://googleblog.blogspot.co.nz/2015/03/protecting-people-across-web-with.html (visited on 03/28/2017).

[79] MyWOT. *Web of Trust (WOT) – Crowdsourced web safety*. 2017. URL: https://www.mywot.com/en/aboutus (visited on 03/28/2017).

[80] Alexa. *Alexa Top Sites*. 2017. URL: https://aws.amazon.com/alexa-top-sites/ (visited on 03/29/2017).

[81] Rapid7. *Metasploit*. 2017. URL: https://www.rapid7.com/products/metasploit/ (visited on 03/28/2017).

[82] DLPTEST. *DLP TEST*. 2017. URL: https://dlptest.com/ (visited on 04/15/2017).

[83] Mauricio Vergara Ereche. *Odd behaviour on one node in I root-server*. 2010. URL: https://lists.dns-oarc.net/pipermail/dns-operations/2010-March/005260.html (visited on 12/22/2016).

[84] Rob McMillan and Eric Quellet. *Best Practices for Data Loss Prevention: A Process, Not a Technology*. Tech. rep. Gartner, 2012.

[85] Lawrence Pingree and Peter Firstbrook Neil MacDonald. *Best Practices for Detecting and Mitigating Advanced Persistent Threats*. Tech. rep. Gartner, 2015.

[86] Bugzilla@Mozilla. *Remove Trustwave Certificate(s) from trusted root certificates*. 2012. URL: https://bugzilla.mozilla.org/show_bug.cgi?id=724929 (visited on 12/24/2016).

[87] Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. "Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties". In: *European Symposium on Research in Computer Security*. Springer. 2012, pp. 199–216.

[88] Rolf Oppliger, Ralf Hauser, and David Basin. "SSL/TLS session-aware user authentication–Or how to effectively thwart the man-in-the-middle". In: *Computer Communications* 29.12 (2006), pp. 2238–2246.

[89] Rolf Oppliger, Ralf Hauser, and P David BASIN. "SSL/TLS session-aware user authentication". In: *Computer* 41.3 (2008), pp. 59–65.

[90] David Taylor et al. *Using the Secure Remote Password (SRP) protocol for TLS authentication*. Tech. rep. 2007.

[91] Chris Evans and Chris Palmer. "Certificate pinning extension for HSTS". In: (2011). URL: https://tools.ietf.org/html/draft-evans-palmer-hsts-pinning-00 (visited on 12/11/2016).

[92] Victor Costan and Srinivas Devadas. "Intel SGX Explained." In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 86.

[93] Internet-Security.ca. *MIT accuses Intel of breaking its SGX security model!* 2016. URL: http://www.internet-security.ca/internet-security-news-archives-051/mit-accuses-intel-of-breaking-sgx-security-model.html (visited on 02/24/2017).

[94] Filippo Valsorda. *An overover of TLS 1.3 and Q&A*. Sept. 23, 2016. URL: https://blog.cloudflare.com/tls-1-3-overview-and-q-and-a/ (visited on 04/16/2017).

[95] Vincent Lynch. *Firefox and Chrome Now Warning About Insecure Login Pages*. Jan. 27, 2017. URL: https://www.thesslstore.com/blog/firefox-chrome-warning-about-insecure-login-pages/ (visited on 04/16/2017).

[96] Builtwith. *SSL by Default Usage Statistics*. 2017. URL: https://trends.builtwith.com/ssl/SSL-by-Default (visited on 04/16/2017).

[97] HTTPArchive. *HTTPArchive Interesting Stats*. 2017. URL: http://httparchive.org/interesting.php (visited on 04/16/2017).

[98] SQLiteStudio. *SQLite Studio*. 2017. URL: https://sqlitestudio.pl/index.rvt (visited on 04/16/2017).

[99] SQliteDatabaseBrowser. *DB Browser for SQLite*. 2017. URL: http://sqlitebrowser.org/ (visited on 04/16/2017).

[100] Qotom. *QOTOM-Q190G4 4 LAN Mini PC*. URL: http://www.qotom.net/goods-129-QOTOM-Q190G4+4+LAN+Mini+PC.html (visited on 02/27/2017).

[101] pfSense. *pfSense - Open Source Security*. URL: https://pfsense.org/ (visited on 02/27/2017).

[102] Cisco. *OpenDNS*. 2017. URL: https://www.opendns.com/ (visited on 03/15/2017).

[103] pfBlockerNG. *pfBlockerNG*. URL: https://forum.pfsense.org/index.php?topic=86212.0 (visited on 02/27/2017).

[104] *The SPAMHAUS Project*. URL: https://www.spamhaus.org/ (visited on 03/15/2017).

[105] *Internet Storm Center*. URL: https://www.dshield.org/ (visited on 03/15/2017).

[106] *Cisco Talo Intelligence*. URL: http://www.talosintelligence.com/ (visited on 03/16/2017).

[107] Dellas Haselhorst. *USING PFBLOCKERNG (AND BLOCK LISTS) ON PFSENSE*. Feb. 8, 2017. URL: https://www.linuxincluded.com/using-firewall-block-lists/ (visited on 03/15/2017).

[108] Martin Roesch. *Cisco Announces OpenAppID – the Next Open Source 'Game Changer' in Cybersecurity*. Feb. 25, 2014. URL: http://blogs.cisco.com/security/cisco-announces-openappid-the-next-open-source-game-changer-in-cybersecurity (visited on 03/16/2017).

[109] bmeeks. *Suricata true inline IPS mode coming with pfSense 2.3 – here is a preview*. Mar. 8, 2016. URL: https://forum.pfsense.org/index.php?topic=108010.0 (visited on 03/17/2017).

[110] Squid. Ed. by What is Squid? URL: http://www.squid-cache.org/Intro/ (visited on 02/27/2017).

[111] squidGuard. Ed. by squidGuard. URL: http://www.squidguard.org/ (visited on 02/27/2017).

[112] squid-cache.org. *Feature: SslBump Peek and Splice*. Feb. 1, 2017. URL: http://wiki.squid-cache.org/Features/SslPeekAndSplice (visited on 03/21/2017).

[113] Shalla Secure Services. *Shalla's Blacklists*. 2017. URL: http://www.shallalist.de/ (visited on 03/23/2017).

[114] Université Toulouse. *Squidguard*. URL: http://dsi.ut-capitole.fr/documentations/cache/squidguard_en.html#contrib (visited on 03/23/2017).

[115] COMODO. *MYDLP*. 2017. URL: http://www.mydlp.com/ (visited on 04/15/2017).

Appendices

Appendix A

Testing Framework

The test environment consists of the following components: 1) An SQLite database for storing the test result data; 2) PowerShell scripts for executing various test cases. The test scripts need to be run on a machine behind the Secure Web Gateway with all traffic passing through it. The following components are required to be installed on the test machine.

- **Windows Management Framework** This is the required component for running a PowerShell script. It is recommended to upgrade it to the latest version available for the OS.
- **Precompiled SQLite Binaries for .NET** This is downloadable from the SQLite website and is needed for the PowerShell script to work with the SQLite database.
- **Database Managers** SQLite Studio was used for managing most of DB tasks [98]. SQLite Database Browser was used for the export function due to a bug in the SQLite Studio [99].
- **Precompiled SQLite Binaries for Windows** This is the CLI tool for managing the SQLite database. The import function in the CLI is much faster than importing the CSV file using the GUI tool. The testing was executed on two different machines and hence the need of using the import function to consolidate records in two databases.

Figure A.1 shows the system architecture of the test environment.

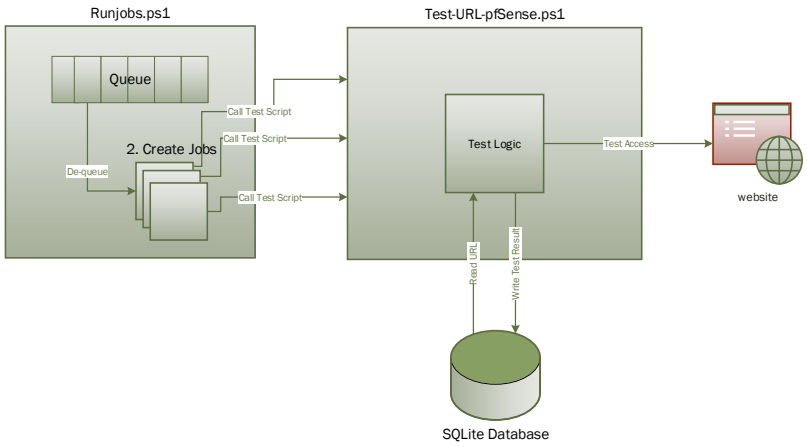


Figure A.1: Test Environment System Architecture

1. Runjob.ps1 populates the queue with ID of each URL being tested.

2. Runjob.ps1 creates multiple jobs. The $maxConcurrentJobs variable controls the maximum number of jobs.

3. All jobs run concurrently retrieving and removing ID from the queue. Each job will run continuously until no more ID left in the queue.

4. Each job calls the test-url-pfsense.ps1 script and passes the ID being tested to the script.

5. Test-url-pfSense.ps1 retrieves the actual URL from the database based on the ID it receives from the Runjob.ps1 script.

6. Test-url-pfsense.ps1 tests the URL using the testing logic defined in the script.

7. Test-url-pfsense.ps1 writes the test result into the database.

A.1 SQL Tables

main table for storing the test result data.

swg_filter_result

```
PRAGMA foreign_keys = off;
BEGIN TRANSACTION;

DROP TABLE IF EXISTS swg_filter_result;

CREATE TABLE swg_filter_result (
    swg_filter_result_id INTEGER  PRIMARY KEY AUTOINCREMENT
                                  UNIQUE
                                  NOT NULL,
    url_key              INTEGER  NOT NULL,
    swg_filter_type      TEXT     NOT NULL,
    swg_filter_result    INTEGER  NOT NULL,
    swg_test_date        DATETIME NOT NULL,
    http_status_code     VARCHAR,
    swg_system_name      VARCHAR
);

DROP INDEX IF EXISTS swg_filter_result_index;

CREATE INDEX swg_filter_result_index ON swg_filter_result (
    url_key ASC
);

COMMIT TRANSACTION;
PRAGMA foreign_keys = on;
```

table for storing the domains or URLs of the websites to be tested

url

```
PRAGMA foreign_keys = off;
BEGIN TRANSACTION;

-- Table: url
DROP TABLE IF EXISTS url;

CREATE TABLE url (
    url_key INTEGER NOT NULL
                    PRIMARY KEY AUTOINCREMENT
                    UNIQUE,
    target  TEXT    NOT NULL,
    url     VARCHAR
);


COMMIT TRANSACTION;
PRAGMA foreign_keys = on;
```

table for storing the domains or URLs of the websites that are blacklisted by the Google Safe Browsing

url_safebrowsing

```
PRAGMA foreign_keys = off;
BEGIN TRANSACTION;

-- Table: url_safebrowsing
DROP TABLE IF EXISTS url_safebrowsing;

CREATE TABLE url_safebrowsing (
    url_safebrowsing_id INTEGER  PRIMARY KEY
                                 NOT NULL,
    url_key             INTEGER  NOT NULL,
    threat_type         TEXT     NOT NULL
                                 DEFAULT (NULL),
    check_date          DATETIME
);


-- Index: url_safebrowsing_index
DROP INDEX IF EXISTS url_safebrowsing_index;

CREATE INDEX url_safebrowsing_index ON url_safebrowsing (
    url_key ASC
);


COMMIT TRANSACTION;
PRAGMA foreign_keys = on;
```

table for storing the data of URL to category and confidence value assigned by the Web of Trust.

url_category

```
PRAGMA foreign_keys = off;
BEGIN TRANSACTION;

-- Table: url_category
DROP TABLE IF EXISTS url_category;

CREATE TABLE url_category (
    url_category_id     INTEGER  NOT NULL
                                 PRIMARY KEY AUTOINCREMENT
                                 UNIQUE,
    url_key             INTEGER  NOT NULL,
    category_identifier INTEGER  NOT NULL,
    category_confidence INTEGER  NOT NULL,
    check_date          DATETIME
);
```

```
16
17
18  -- Index: url_category_index
19  DROP INDEX IF EXISTS url_category_index;
20
21  CREATE INDEX url_category_index ON url_category (
22      "url_key" ASC
23  );
24
25
26  COMMIT TRANSACTION;
27  PRAGMA foreign_keys = on;
```

A.2 PowerShell Scripts

This is a wrapper script that utilises the queue and job to achieve multi-threading and running the testing on multiple websites concurrently. This helps to increase the speed of the testing.

runjobs.ps1

```
1   $maxConcurrentJobs = 10;
2
3
4   $queue = [System.Collections.Queue]::Synchronized( (New-Object System.Collections.Queue) )
5   for($i=469502;$i -le 500000; $i++)
6   {
7       $queue.Enqueue($i)
8   }
9
10  Function RunJobFromQueue
11  {
12      Get-Job -State Completed | Remove-Job
13      if( $queue.Count -gt 0)
14      {
15          #Start-Sleep (Get-Random -Minimum 1 -Maximum 5)
16          $j = Start-Job -Filepath C:\scripts\test-url-pfsense.ps1 -ArgumentList
     ↪    $queue.Dequeue()
17          Register-ObjectEvent -InputObject $j -EventName StateChanged -Action {
     ↪    RunJobFromQueue; Unregister-Event $eventsubscriber.SourceIdentifier; Remove-Job
     ↪    $eventsubscriber.SourceIdentifier } | Out-Null
18      }
19
20  }
21
22  for( $i = 0; $i -lt $maxConcurrentJobs; $i++ )
23  {
```

```
    RunJobFromQueue
}
```

This is the actual test script that is responsible for executing different test cases against a website and record the test result in the SQLite database.

test-url-pfsense.ps1

```
param(
    $url_key
)

$AllProtocols = [System.Net.SecurityProtocolType]'Ssl3,Tls,Tls11,Tls12'
[System.Net.ServicePointManager]::SecurityProtocol = $AllProtocols

if (-not
 ↪   ([System.Management.Automation.PSTypeName]'ServerCertificateValidationCallback').Type)
{
$certCallback=@"
    using System;
    using System.Net;
    using System.Net.Security;
    using System.Security.Cryptography.X509Certificates;
    public class ServerCertificateValidationCallback
    {
        public static void Ignore()
        {
            if(ServicePointManager.ServerCertificateValidationCallback ==null)
            {
                ServicePointManager.ServerCertificateValidationCallback +=
                    delegate
                    (
                        Object obj,
                        X509Certificate certificate,
                        X509Chain chain,
                        SslPolicyErrors errors
                    )
                    {
                        return true;
                    };
            }
        }
    }
"@
    Add-Type $certCallback
 }
[ServerCertificateValidationCallback]::Ignore();

function test-uri2
{
```

```
42      param( $new_uri )
43      $test_date = (Get-Date -Format "yyyy-MM-dd HH:mm:ss")
44
45     try {
46         $dnsresponse = Resolve-DnsName $new_uri
47         if($dnsresponse.IP4Address.Contains("10.10.10.1")) {
48             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'DNSBL', 1, '" + $test_date + "','0', 'pfSense');"
49             $insert_output = $sql.ExecuteNonQuery()
50         }
51         elseif($dnsresponse.IP4Address.Contains("146.112.61.104")) {
52             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Domain List Block', 1, '" + $test_date + "','0', 'pfSense');"
53             $insert_output = $sql.ExecuteNonQuery()
54         }
55         elseif($dnsresponse.IP4Address.Contains("146.112.61.105")) {
56             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Botnet Block', 1, '" + $test_date + "','0', 'pfSense');"
57             $insert_output = $sql.ExecuteNonQuery()
58         }
59         elseif($dnsresponse.IP4Address.Contains("146.112.61.106")) {
60             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Content Category Block', 1, '" + $test_date + "','0',
    ↪   'pfSense');"
61             $insert_output = $sql.ExecuteNonQuery()
62         }
63         elseif($dnsresponse.IP4Address.Contains("146.112.61.107")) {
64             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Malware Block', 1, '" + $test_date + "','0', 'pfSense');"
65             $insert_output = $sql.ExecuteNonQuery()
66         }
67         elseif($dnsresponse.IP4Address.Contains("146.112.61.108")) {
68             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Phishing Block', 1, '" + $test_date + "','0', 'pfSense');"
69             $insert_output = $sql.ExecuteNonQuery()
70         }
71         elseif($dnsresponse.IP4Address.Contains("146.112.61.109")) {
72             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Suspicious Response Block', 1, '" + $test_date + "','0',
    ↪   'pfSense');"
73             $insert_output = $sql.ExecuteNonQuery()
74         }
```

```
75        elseif($dnsresponse.IP4Address.Contains("146.112.61.110")) {
76            $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪ swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪ $url_key + "', 'OpenDNS-Security Integrations Block', 1, '" + $test_date + "','0',
    ↪ 'pfSense');"
77            $insert_output = $sql.ExecuteNonQuery()
78        }
79        else {
80            try {
81                $test_date = (Get-Date -Format "yyyy-MM-dd HH:mm:ss")
82                $response = Invoke-WebRequest -Uri $new_uri -TimeoutSec 20
83
84
85                $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪ swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪ $url_key + "', 'url', 0, '" + $test_date + "','" + $response.StatusCode + "',
    ↪ 'pfSense');"
86                $insert_output = $sql.ExecuteNonQuery()
87                #echo "£(£uri):0"
88
89            }
90            catch {
91                    if ($_.Exception.Response.StatusCode.Value__ -eq 403) {
92                        $test_date = (Get-Date -Format "yyyy-MM-dd HH:mm:ss")
93                        $_.Exception.Response.GetResponseStream()
94                        $reader = New-Object System.IO.StreamReader($result)
95                        $reader.BaseStream.Position = 0
96                        $reader.DiscardBufferedData()
97                        $responseBody = $reader.ReadToEnd();
98                        if ($responseBody -like "*pfSense*") {
99                            if ($responseBody -like "*Target group*") {
100                               $start_pos = $responseBody.IndexOf("<b> Target group:   </b>
    ↪ ") + 25
101                               $length = ($responseBody.IndexOf("<b> URL:") -11) -
    ↪ $start_pos
102                           }
103                       $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪ swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪ VALUES ('" + $url_key + "', '" + $responseBody.Substring($start_pos, $length) + "', 1,
    ↪ '" + $test_date + "','" + $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
104                       $insert_output = $sql.ExecuteNonQuery()
105                       }
106                       else {
107                           $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪ swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪ VALUES ('" + $url_key + "', 'url', 0, '" + $test_date + "','" +
    ↪ $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
108                           $insert_output = $sql.ExecuteNonQuery()
109
```

```
110                         }
111
112                     }
113                     elseif($_.Exception.Response.StatusCode.Value__ -eq 503) {
114                         if ($_.ErrorDetails -like "*(13) Permission denied*") {
115                             $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'IPBL', 1, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
116                             $insert_output = $sql.ExecuteNonQuery()
117                         }
118                         else {
119                             $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'url', 0, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
120                             $insert_output = $sql.ExecuteNonQuery()
121                         }
122
123                     }
124                     else {
125                         if($_.Exception.Message -eq "The operation has timed out.") {
126                             $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'timed out', 0, '" + $test_date + "','0', 'pfSense');"
127                         }
128                         else {
129                             $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'url', 0, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
130                         }
131                         $insert_output = $sql.ExecuteNonQuery()
132                     }
133             }
134
135         }
136
137     }
138     catch {
139         $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'DNS Not Exist', 2, '" + $test_date + "','0', 'pfSense');"
140         $insert_output = $sql.ExecuteNonQuery()
141     }
142
143 }
144
145 Add-Type -Path "C:\Program Files\System.Data.SQLite\2015\bin\System.Data.SQLite.dll"
```

```
$con = New-Object -TypeName System.Data.SQLite.SQLiteConnection
$con.ConnectionString = "Data Source=C:\temp\websites.sqlite"
$con.Open()
$offset = 0
$sql = $con.CreateCommand()

#echo $sql.CommandText

$sql.CommandText = "SELECT url.target from url where url_key = $($url_key);"
#echo $sql.CommandText

$adapter = New-Object -TypeName System.Data.SQLite.SQLiteDataAdapter $sql
#we create the dataset
$data = New-Object System.Data.DataSet
#and then fill the dataset
[void]$adapter.Fill($data)

$target = $data.Tables[0].Rows[0].target

$uri = $target

$test_date = (Get-Date -Format "yyyy-MM-dd HH:mm:ss")

try {
    $dnsresponse = Resolve-DnsName $uri
    if($dnsresponse.IP4Address.Contains("10.10.10.1")) {
        $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
 ↪  swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
 ↪  $url_key + "', 'DNSBL', 1, '" + $test_date + "','0', 'pfSense');"
        $insert_output = $sql.ExecuteNonQuery()
    }
    elseif($dnsresponse.IP4Address.Contains("146.112.61.104")) {
        $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
 ↪  swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
 ↪  $url_key + "', 'OpenDNS-Domain List Block', 1, '" + $test_date + "','0', 'pfSense');"
        $insert_output = $sql.ExecuteNonQuery()
    }
    elseif($dnsresponse.IP4Address.Contains("146.112.61.105")) {
        $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
 ↪  swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
 ↪  $url_key + "', 'OpenDNS-Botnet Block', 1, '" + $test_date + "','0', 'pfSense');"
        $insert_output = $sql.ExecuteNonQuery()
    }
    elseif($dnsresponse.IP4Address.Contains("146.112.61.106")) {
        $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
 ↪  swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
 ↪  $url_key + "', 'OpenDNS-Content Category Block', 1, '" + $test_date + "','0',
 ↪  'pfSense');"
        $insert_output = $sql.ExecuteNonQuery()
```

```
186     }
187     elseif($dnsresponse.IP4Address.Contains("146.112.61.107")) {
188         $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Malware Block', 1, '" + $test_date + "','0', 'pfSense');"
189         $insert_output = $sql.ExecuteNonQuery()
190     }
191     elseif($dnsresponse.IP4Address.Contains("146.112.61.108")) {
192         $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Phishing Block', 1, '" + $test_date + "','0', 'pfSense');"
193         $insert_output = $sql.ExecuteNonQuery()
194     }
195     elseif($dnsresponse.IP4Address.Contains("146.112.61.109")) {
196         $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Suspicious Response Block', 1, '" + $test_date + "','0',
    ↪   'pfSense');"
197         $insert_output = $sql.ExecuteNonQuery()
198     }
199     elseif($dnsresponse.IP4Address.Contains("146.112.61.110")) {
200         $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'OpenDNS-Security Integrations Block', 1, '" + $test_date + "','0',
    ↪   'pfSense');"
201         $insert_output = $sql.ExecuteNonQuery()
202     }
203     else {
204         try {
205             $test_date = (Get-Date -Format "yyyy-MM-dd HH:mm:ss")
206             $response = Invoke-WebRequest -Uri $uri -TimeoutSec 20
207
208
209             $sql.CommandText = "Insert into swg_filter_result (url_key, swg_filter_type,
    ↪   swg_filter_result, swg_test_date, http_status_code, swg_system_name) VALUES ('" +
    ↪   $url_key + "', 'url', 0, '" + $test_date + "','" + $response.StatusCode + "',
    ↪   'pfSense');"
210             $insert_output = $sql.ExecuteNonQuery()
211             #echo "£(£uri):0"
212
213         }
214         catch {
215              if ($_.Exception.Response.StatusCode.Value__ -eq 403) {
216                     $test_date = (Get-Date -Format "yyyy-MM-dd HH:mm:ss")
217                     $_.Exception.Response.GetResponseStream()
218                     $reader = New-Object System.IO.StreamReader($result)
219                     $reader.BaseStream.Position = 0
220                     $reader.DiscardBufferedData()
221                     $responseBody = $reader.ReadToEnd();
```

```
222                     if ($responseBody -like "*pfSense*") {
223                         if ($responseBody -like "*Target group*") {
224                             $start_pos = $responseBody.IndexOf("<b> Target group:   </b> ")
    ↪   + 25
225                             $length = ($responseBody.IndexOf("<b> URL:") -11) - $start_pos
226                         }
227                     $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', '" + $responseBody.Substring($start_pos, $length) + "', 1,
    ↪   '" + $test_date + "','" + $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
228                     $insert_output = $sql.ExecuteNonQuery()
229                     }
230                     else {
231                             $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'url', 0, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
232                             $insert_output = $sql.ExecuteNonQuery()
233
234                     }
235
236             }
237             elseif($_.Exception.Response.StatusCode.Value__ -eq 503) {
238                if ($_.ErrorDetails -like "*(13) Permission denied*") {
239                     $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'IPBL', 1, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
240                     $insert_output = $sql.ExecuteNonQuery()
241                 }
242                 else {
243                     $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'url', 0, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
244                     $insert_output = $sql.ExecuteNonQuery()
245                 }
246
247             }
248             else {
249                if($_.Exception.Message -eq "The operation has timed out.") {
250                     test-uri2 -new_uri "www.$($uri)"
251                 }
252                 else {
253                     $sql.CommandText = "Insert into swg_filter_result (url_key,
    ↪   swg_filter_type, swg_filter_result, swg_test_date, http_status_code, swg_system_name)
    ↪   VALUES ('" + $url_key + "', 'url', 0, '" + $test_date + "','" +
    ↪   $_.Exception.Response.StatusCode.Value__ + "', 'pfSense');"
254                     $insert_output = $sql.ExecuteNonQuery()
```

```
                }
            }
        }

    }

}
catch {
    test-uri2 -new_uri "www.$($uri)"
}


```

Appendix B

Secure Web Gateway Implementation

Based on the aforementioned findings, this section shows an implementation of Secure Web Gateway.

To cater for home or small business usage scenario, a low power, low noise (fanless), small form factor with multiple network ports PC was chosen for this implementation. At the time of writing, there is no suitable hardware in the market supports Intel Software Guard Extensions and also meet other requirements. Fig. B.1 shows the pictures of a Qotom Mini PC used in this implementation [100]. The hardware comes with Intel Celeron quad core 2GHz CPU and supports memory up to 8GB which provides a reasonable amount of resources to run the required functionalities. The motherboard is based on the Intel chipset, and 4 x integrated Intel Ethernet controller. Intel based hardware is compatible with most operating systems.

(a) Motherboard View (b) Front & Rear View

Figure B.1: A Mini PC with views from inside and outside

B.1 Software Components

Because of the reasons above, most of the software components selected in this implementation are either open source or affordable commercial add-ons that can be easily integrated into the Secure Web Gateway.

B.1.1 pfSense

pfSense project started in 2004 as a fork of the m0n0wall project and now has come with many features out-of-box that are essential for a Secure Web Gateway implementation [101]. As a result, pfSense was chosen as the firewall for this project. The first step is to configure the firewall to operate in the bridge mode. In the bridge mode, the public IP address is assigned to the interface on the firewall and the firewall will have the complete control over the traffic coming in and going out of the network without other forms of Network Address Translation. Fig. B.2 shows the configuration of a DSL router. As the switch used in this implementation does not support VLAN tagging, VLAN 1 is used to disable the VLAN tagging. Next, create a VLAN interface bounded to the external interface as shown in the Fig. B.3. VLAN ID is provided by the ISP, and in this case, VLAN-10 is used by Spark NZ the ISP that provides the Internet connection for this implementation. Finally, create a new PPP interface by selecting PPPoE as the Link Type, VLAN interface set up in the previous step as the Link Interface and entering the username and password supplied by the ISP (Fig. B.4). Then add a new virtual interface to this newly created PPPOE interface. Now, the firewall should try to authenticate with the ISP and obtain a public IP address.

In this implementation, OpenDNS was chosen to extend the DNS protection capability by adding the Phishing and Malware/Botnet Protection features [102]. Also, a firewall rule is added to block any outbound DNS traffic from the Internal network. As shown in the Fig. B.5, add the OpenDNS servers IP to the DNS Server Settings and then enable forwarding mode in the DNS Resolver. It is also advisable to create a firewall rule to block all outbound connection to DNS port (TCP/UDP 53) to prevent applications to

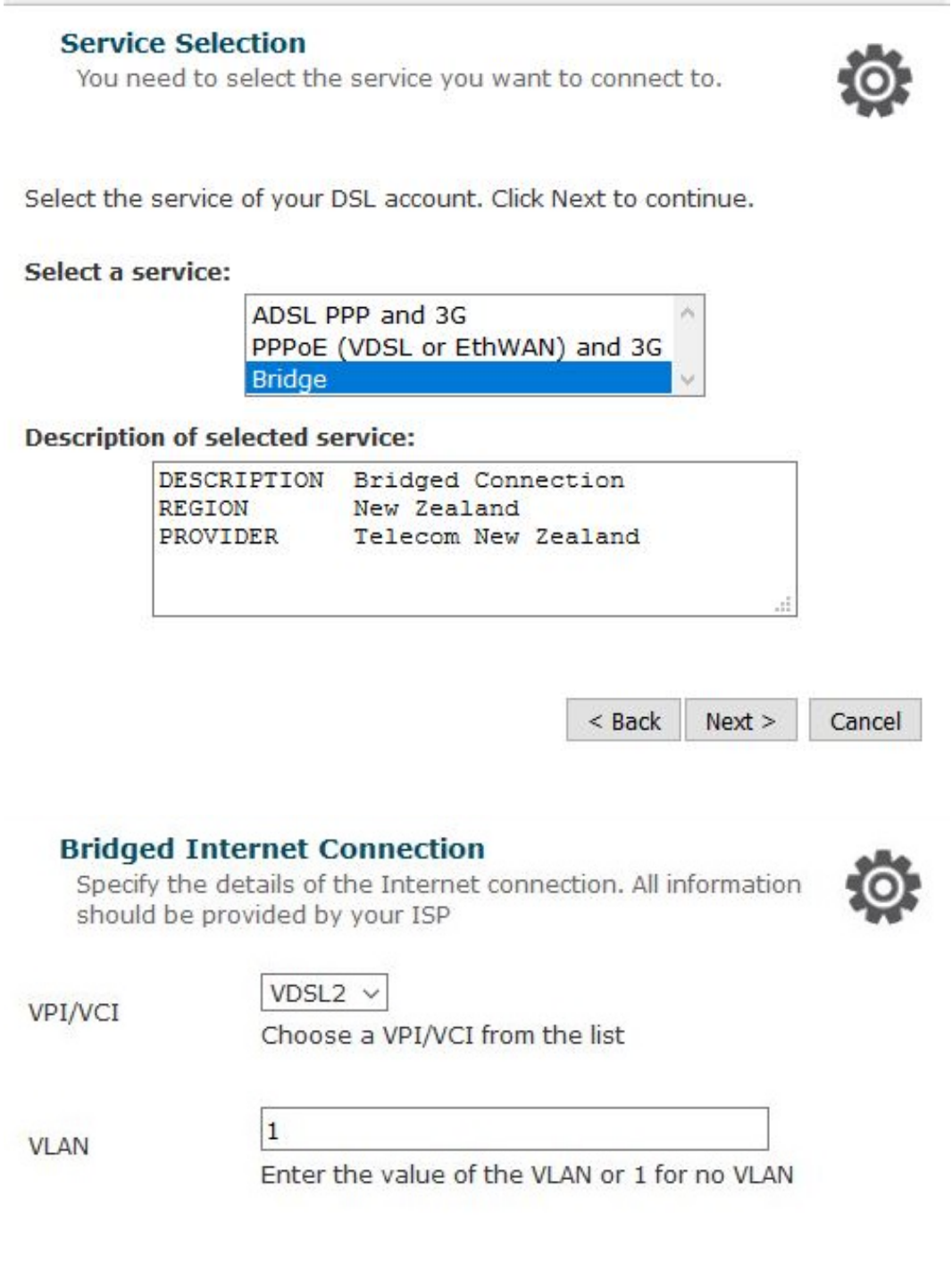


Figure B.2: Spark VDSL router configuration

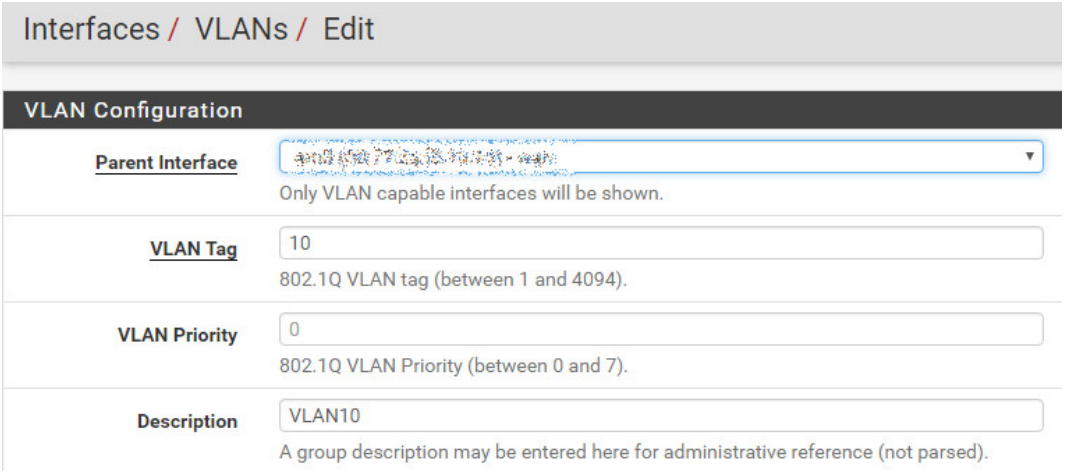


Figure B.3: pfSense VLAN Configuration

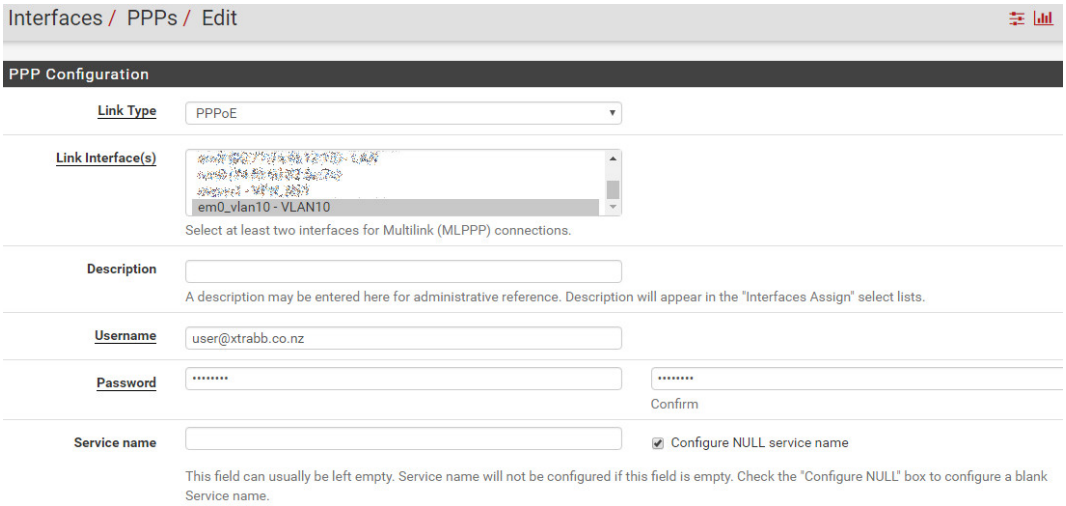
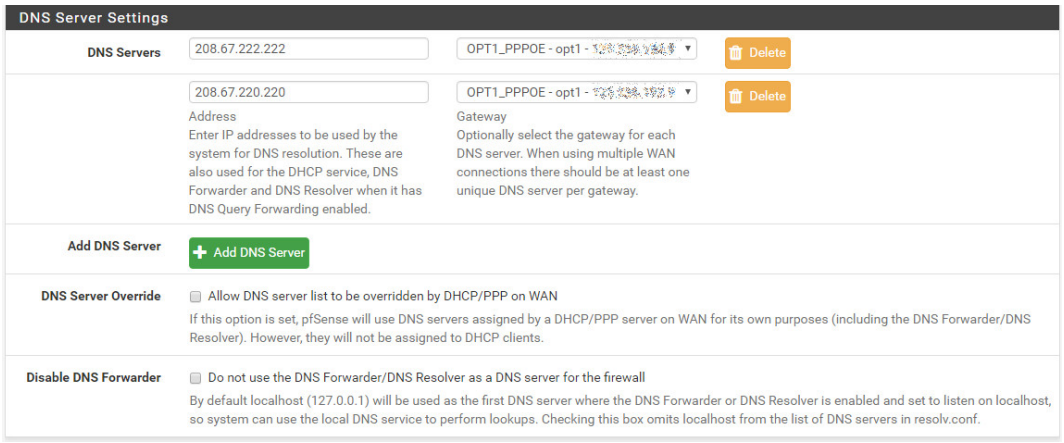
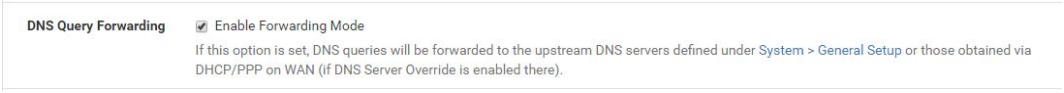


Figure B.4: pfSense PPP Interface Configuration



(a) DNS Server Setup in General Setup



(b) DNS Resolver Setting

Figure B.5: pfSense DNS Server Settings

bypass the local DNS server.

B.1.2 pfBlockerNG

pfBlockerNG is a package run on pfSense and is an IP and Domain Names download manager. It can collect IPs and Domain Names from multiple sources and then automatically create firewall rules to Deny, Permit or Match the traffic [103]. When Squid proxy server operates in the transparent mode, web traffic is intercepted before it is passed to the firewall and this will stop the pfBlockerNG from blocking the IPs in the blacklists. To block traffic to the blacklists IPs, perform the following steps: 1) Select both LAN and WAN interface for Outbound Firewall Rules; 2) Enable floating rules so the firewall can block traffic generated from the Squid proxy server.

The following IP Black lists were used:

1. **Emerging Threats Compromised Hosts** The list contains hosts that are known to be compromised by bots, phishing sites, or spewing hostile traffic.
   https://rules.emergingthreats.net/blockrules/compromised-ips.txt

2. **Emerging Threats Blocked IPs** This list contains Spam nets identi-

fied by [104] and top attackers listed by DShield [105].
https://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt

3. **Spamhaus DROP lists** These lists consist of netblocks that are hijacked or leased by professional spam or cybercrime operations and are used for dissemination of malware, trojan downloaders, or botnet controllers
https://www.spamhaus.org/drop/drop.txt https://www.spamhaus.org/drop/edrop.txt

4. **Talos Intelligence Blacklist** Talos is an organisation that is the primary member of Cisco's Collective Security Intelligence (CSI) ecosystem and is dedicated in providing threat intelligences [106]. The list provides a list of known malicious network threats.
http://talosintel.com/feeds/ip-filter.blf

5. **DShield Recommended Block List** This list summarises the top 20 attacking class C (/24) subnets over the last three days.
https://isc.sans.edu/block.txt

6. **ISC Top 1000 Attack Source** This list provides the top 1000 attacks IPs in the last 30 days
https://isc.sans.edu/api/sources/attacks/1000/

7. **The CI Army List** The list sourced from information gathered from the CINS system and contains IP address meet two criteria: 1) The IP's recent Rouge Packet score factor is very poor, and 2) It hasn't yet been identified as malicious by the InfoSec community. The list serves as supplement and enhance to other lists.
http://cinsscore.com/list/ci-badguys.txt

8. **Bambenek Consulting C2 List** This list contains a master feed of known, active and non-sinkholed Command and Control (C2) IP addresses.
https://osint.bambenekconsulting.com/feeds/c2-ipmasterlist.txt

9. **ZeuS IP blocklist** This blocklist contains all IPv4 addresses associated with ZeuS Command and Control (C2) which are currently being tracked

by Zeus Tracker. This list may contain some false-positives.
https://zeustracker.abuse.ch/blocklist.php?download=badips

10. **SSL IP Blacklist** This list contains all IPs that SSLBL has seen in the past 30 days being associated with a malicious SSL certificate.
https://sslbl.abuse.ch/blacklist/sslipblacklist.csv

11. **SSL IP Blacklist Dyre botnet** This list contains all IPs that SSLBL has seen in the past 30 days being associated with a malicious SSL certificate used by Dyre botnet.
https://sslbl.abuse.ch/blacklist/dyre_sslipblacklist.csv

12. **Ransomware Tracker IP Blocklist** This list contains IP addresses that have been associated with Ransomware in the past 30 days.
https://ransomwaretracker.abuse.ch/downloads/RW_IPBL.txt

The following DNS Blacklists were used:

1. **DShield Suspicious Domains List** These are lists provided by the Internet Storm Centre. The lists contain suspicious domains with different level of sensitivity. To reduce false-positives, only high level sensitivity list is used.
https://isc.sans.edu/feeds/suspiciousdomains_High.txt

2. **Bambenek Consulting C2 List** This is a master feed of known, active and non-sinkholed Command and Control (C2) domain names
http://osint.bambenekconsulting.com/feeds/c2-dommasterlist.txt

3. **Bambenek Consulting DGA List** This list contains all known Domain Generation Algorithm generated domains used by malware for domains 2 days prior to 3days after the current data.
http://osint.bambenekconsulting.com/feeds/dga-feed.gz

These blacklists provide a good layer of defence. However, it can potentially introduce some false-positives or false-negatives. It is advisable to choose the list carefully and not solely rely on the blacklist as the only defence. This guide

provides a good starting point for configuring blacklist in the pfBlockerNG [107].

B.1.3 Suricata

In this project, the author had evaluated two most popular open source intrusion detection and prevention system - Snort and Suricata. Snort has been around since 1998 and has the largest community of users and copious amount of documentation on the Internet. The main developer group behind the Snort is Sourcefire, which is acquired by Cisco in 2013. In 2014, Cisco released OpenAppID to the open source community for Snort. OpenAppID allows Snort users to easily write a rule to detect, monitor and manage usage of thousands of different applications in their networks [108]. The downside of Snort is that it is single-threaded and does not support running the IPS in the Inline mode. The true inline mode IPS sits in between the network interface card and the OS kernel. All traffic can be evaluated, alerted or dropped in real-time. Without in-line mode, IPS uses the PCAP engine to generate copies of packets for inspection as they traverse through the interface and hence some leakage of packets can occur before IPS can determine if the traffic matches a rule. As a result, there are some performance limitations within the current Snort architecture. The advantage of Suricata over Snort is its ability to operate in the in-line mode and also the support of multi-threading, it can deliver a much higher throughput than Snort on the hardware of the same capacity. Because this implementation is designed to run in a home or small-office environment with limited hardware resources, Suricata was chosen for this implementation. Table B.1 provides a comparison between Snort and Suricata.

Suricata like Snort is also a rule based IPS/IDS and is compatible with Emerging Threats rules and Snort VRT Rules. As shown in Fig. B.6, it requires Oinkmaster Code to download Snort VRT rules. The code can be obtained after registering an account at the Snort website. The paid subscription can receive the rules immediately upon release, whereas the free version will only receive the rules that are older than 30 days.

As the goal of this implementation is to protect the outbound traffic, a

Table B.1: Snort vs Suricata

	Snort	Suricata
Developer	Sourcefire, Inc.	Open Information Security Foundation (OISF)
Availability	Since 1998	Since 2009
Operating System	Cross-platform	Cross-platform
Threads	Single-threaded	Multi-threaded
In-line mode	No	Yes
OpenAppID	Yes	No
Snort (VRT) Rules Support	Yes	Yes
Emerging Threats Rules Support	Yes	Yes

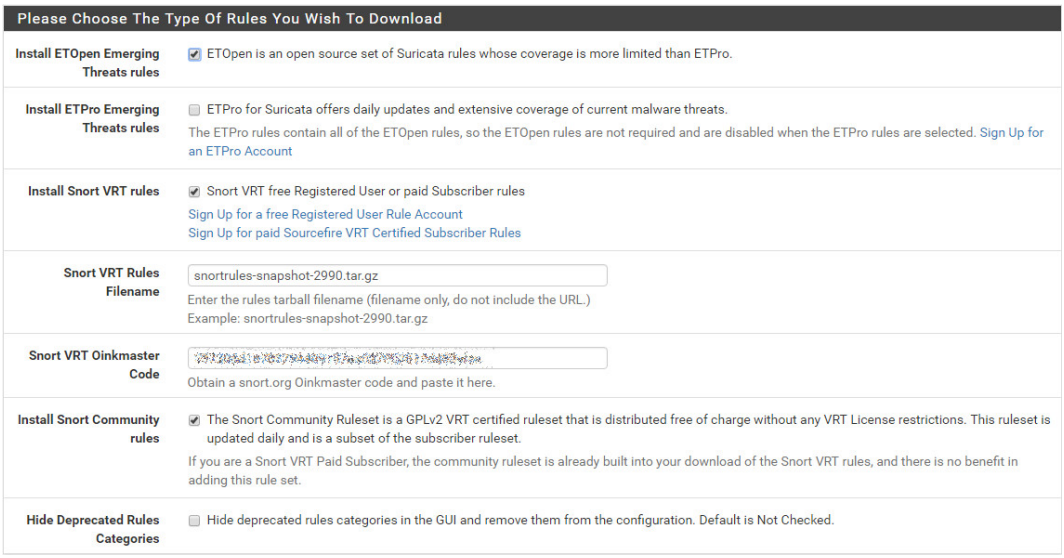


Figure B.6: Suricata Global Settings

Suricata interface is created on the LAN interface to have the visibility of internal source IP address. On the interface setting, there are two block modes available - Legacy and Inline mode as shown in the Fig. B.7. The Inline mode requires the network cards that support Netmap. Netmap is an extremely fast and efficient packet I/O framework for both userspace and kernel clients. The network cards in the hardware used in this implementation do support Netmap. In [109], it details the instruction on how to setup Inline mode for Suricata.

Suricata provides a powerful capability to detect malicious network traffic. However, it is also not something can be set-and-forget. It is recommended

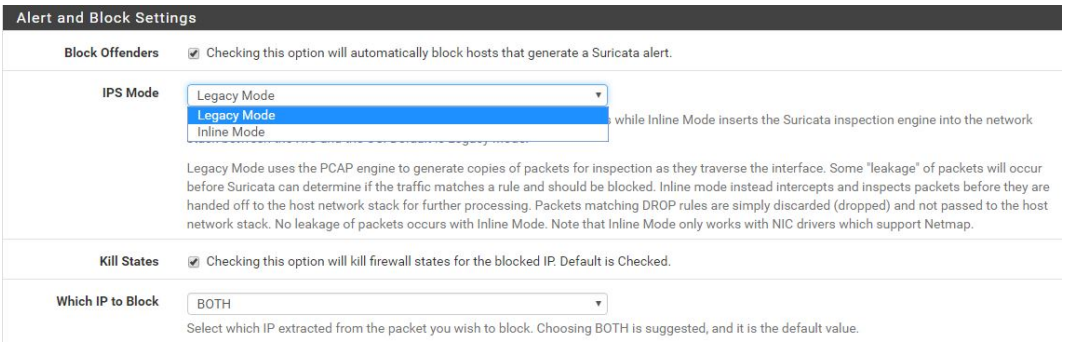


Figure B.7: Suricata Block Mode

to start with alerting mode only and gradually tune the setting to suppress rules that generate false-positive alerts. The following factors were considered when choosing the different rulesets: 1) There are some emerging threats and snort rulesets that are created based on the blacklists that have already been used in pfBlockerNG. These rulesets were excluded to avoid double handling and unnecessary performance overhead; 2) Avoid selecting rulesets that are designed for scanning inbound traffic to servers such as IMAP or POP3 that are not in use in the environment; 3)The events rulesets such as dns-events or tls-event are designed to detect non-compliant traffic, and as result, they may cause high numbers of false-positives. Disable these rulesets or suppress individual rules within the ruleset that caused false positives. Table B.2 below provides the summary of the configuration of rulesets and rules for this implementation.

B.1.4 Squid

Squid is one of the oldest open source projects on the planet and has been around since the early 1990's. It is a fully-featured proxy offering rich access control, authorization, and logging environment [110]. In the beginning, it was often used for improving the Web performance, but lately combining with the plug-in squidGuard, an URL redirector software, it can be used for categorising websites and determine the action (allow, deny or whitelist) based on the category [111]. The proxy server adds another layer of security and provides the ability to filter the traffic between the client and server. The first step is to enable transparent proxying on the Squid server. Many mobile applications

Table B.2: Suricata Rules Configuration

ET Open Rules	Snort Text	Disable Rules
emerging-activex.rules	snort_browser-chrome.rules	decoder-events.rules
emerging-attack_response.rules	snort_browser-firefox.rules	dns-eventshttp-events.rules
emerging-botcc.portgrouped.rules	snort_browser-ie.rles	smtp-events.rules
emerging-current_events.rules	snort_browser-other.rules	stream-events.rules
emerging-dns.rules	snort_browser-plugins.rules	tls-events.rules
emerging-dos.rules	snort_browser-webkit.rules	
emerging-exploit.rules	snort_exploit-kit.rules	
emerging-malware.rules	snort_file-executable.rules	
emerging-misc.rules	snort_file-flash.rules	
emerging-mobile_malware.rules	snort_file-image.rules	
emerging-netbios.rules	snort_file-java.rules	
emerging-scan.rules	snort_file-multimedia.rules	
emerging-shellcode.rules	snort_file-office.rules	
emerging-trojan.rules	snort_file-other.rules	
emerging-user_agents.rules	snort_file-pdf.rules	
emerging-web_client.rules	snort_indicator-compromise.rules	
emerging-web_specific_apps.rules	snort_indicator-obfuscation.rules	
emerging-worm.rules	snort_indicator-scan.rules	
	snort_indicator-shellcode.rules	
	snort_malware-backdoor.rules	
	snort_malware-cnc.rules	
	snort_malware-other.rules	
	snort_malware-tools.rules	
	snort_netbios.rules	
	snort_os-linux.rules	
	snort_os-mobile.rules	
	snort_os-other.rules	
	snort_os-windows.rules	
	snort_protocol-dns.rules	
	snort_protocol-ftp.rules	
	snort_protocol-icmp.rules	
	snort_protocol-imap.rules	
	snort_protocol-pop.rules	

do not support traditional proxy connection, and very few clients support SSL proxy connections. Transparent proxying also improves the usability of the proxy server considerably. As shown in the Fig. B.8, enable transparent mode and also it may be required to bypass proxy for servers that are incompatible with transparent proxying. It is likely to encounter some SSL sites that don't work with this configuration. It is also advisable to disable outgoing traffic to TCP port 443 and 80 to make sure all network traffic are going through the proxy server.

To enable SSL interception, create an internal certificate authority under System → Certificate Manager → CAs. In the Squid on pfSense, there are two SSL/MITM Mode: 1) Splice Whitelist, Bump Otherwise, and 2) Splice All. Bump establishes a TLS connection to the server and then establishes a TLS connection with the client using a mimicked server certificate. Splice

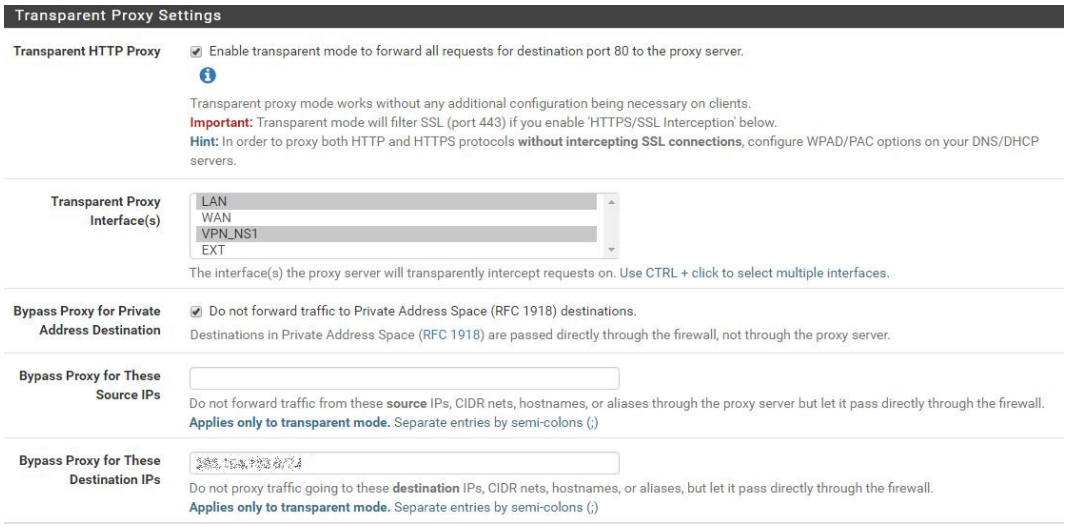


Figure B.8: Squid Transparent Proxy Settings

creates a TCP tunnel without decoding the connection that allows the client and server to exchange the data as if there is no proxy in between [112]. The bump can be problematic and may not be able to access all SSL sites if the sites are setup to detect and reject MITM SSL connection. Bump also requires the CA certificate to be imported onto all devices. Splice was chosen for this implementation as it provides basic site filtering using SNI field and at the same time provides maximum compatibility for accessing SSL sites. Please note content filtering feature such as Antivirus is not available in Splice mode. The following configurations were used:

- **SSL/MITM Mode:** Splice All
- **SSL Proxy Compatibility Mode:** Intermediate. This modes provides the maximum coverage including sites running TLS v1.0.
- **DHParams Key Size:** 2048 (default)

pfSense provides an interface to enable ClamAV Anti-Virus Integration via C-ICAP. This can be enabled on the Antivirus tab. To test if the AV function is functioning correctly, try downloading the test virus files from eicar website here - http://www.eicar.org/85-0-Download.html.

squidGuard provides URL filtering capability, which can be used for control of websites users can access. The blacklists are the heart of every URL filter. There are a few free blacklists to choose from 1) Shalla's Blacklist [113], and

2) Université Toulouse blacklist collection [114]; and some commercial options as well: 1) URLBlacklist.com; and 2) SQUIDBLACKLIST.ORG. The Shalla's Blacklist was chosen for this implementation.

B.2 MyDLP

In this implementation, we have chosen a DLP product called MyDLP [115]. MyDLP is an open source DLP solution which not only capable of monitoring data in motion but also monitoring data at rest. MyDLP community Edition can only be used to log the matched traffic and the paid/enterprise edition waive this restriction. For the Secure Web Gateway, only the web rule type is used to monitor and control traffic over HTTP or HTTPs. Other types of rules such as mail rules, removable storage rules, removable storage inbound rules, removable storage encryption rules, printer rules, screenshot rules and API rules are outside of the scope of this implementation. In MyDLP, it can apply a different action to the traffic when the traffic matches to a policy rule. The table B.3 depicts the different actions available for the web rule.

Table B.3: MyDLP Rule Actions for Web Rule

Rule Action	Description
PASS	As the name suggests, it allows the information to pass through without generation of any log entries.
LOG	It allows information to pass through, but generates event log.
ARCHIVE	In addition to the log action, it archives a copy of the information. Administrator can download the files from the log interface.
BLOCK	This action prevents information to pass through and generates an event log.
QUARANTINE	In addition to the Block action, it archives a copy of the information.

There are many pre-existing information types available in MyDLP, or one can create a custom information type by leveraging the built-in matcher function such as the function to match the source code expression or to match document hashes. The installation of MyDLP can be done by either using the MyDLP CD Image to install it on a physical or virtual machine or installing

MyDLP on an Ubuntu server manually. This implementation has chosen the option of using the CD image. Enter the configuration in Table. B.4 in the Custom Options (Before Auth) of the advanced features of the SQUID configuration in pfSense.

Table B.4: SQUID Configuration for MyDLP

icap_enable on
icap_io_timeout 30 minutes
icap_preview_enable off
adaptation_send_client_ip on
icap_service service_req reqmod_precache bypass=on icap://MyDLP Server IP or Hostname/dlp
adaptation_access service_req allow all