Godwin Ojo

Internet Traffic Monitoring

Case Study: The Network of Granlund Oy

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Information Technology

Thesis

2 January, 2013

Helsinki Metropolia University of Applied Sciences

Author Title Number of Pages Date	Godwin Ojo Internet Traffic Monitoring 51 pages 2 January, 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Communication and Data Networks
Instructor(s)	Markku Huupponen, Project Manager Matti Puska, Academic Supervisor

The goal of this final year project was to design, implement, and analyse the results of Internet traffic monitoring processes using the network of Granlund Oy as a case study. The main motivation behind the topic was to provide a concise analysis of the Internet traffic of a medium-sized enterprise network traffic. Furthermore, the project examined the network traffic to find bad traffic logs that negatively affected the network performance and security in general.

In order to get a comprehensive view of how the network resources were being used, both hardware and software-based monitoring devices, Bluecoat ProxySG, Bluecoat ProxyAV, Colasoft Capsa (free), and Snort, were used during the monitoring processes. The monitoring devices were set up to either intercept or mirror the traffic entering and leaving the network.

The results of the project were the prevention of harmful traffic from entering into the network, the discovery of unsolicited traffic being generated in the network by infected hosts and the enforcement of the company's web access policies. These results helped in improving the usage of network resources and improved network performance because the infected hosts were cleaned and the existing bad traffic was removed and prevented from entering the network at gateway level, thereby increasing overall network security.

Keywords	Traffic Monitoring, Network Performance, Network Security

Contents

1 Introduction

Internet Traffic Monitoring (ITM) is a part of network management that has developed interest over the years. There are various sizes of networks everywhere and the need to monitor what enters and leaves the network via the Internet has increased over time due to the insecurity and openness of the Internet. The Internet traffic in general is not regulated but in some regions such as Finland, where it is regulated, the Internet Police (through Internet Service Providers (ISP)) have rights to access the activities of users during criminal investigation [2]. The freedom of internet users raises concern for companies and network administrators who are supposed to be able to control their network resources, especially their bandwidths. Since the Internet freedom cannot be curtailed, hence the need for monitoring the activities of users. At this time, it is enough for network administrators to know more about the traffic passing through their networks. The Internet traffic is usually mirrored or intercepted and grouped into different categories using proxy servers or packet analysers. The former can either mirror or intercept the traffic in order to serve as a proxy, while the latter can only mirror the traffic because they are hardware and software respectively.

The goal of the thesis project was to monitor the Internet traffic of Granlund Oy. Granlund has a medium size network with over 500 workstations (more information about Granlund's network can be found in chapter 2). Each user has free access to the Internet and also administrative rights on their workstations. These administrative rights raise security concerns because it is possible for the network to be infected with network worms and viruses through workstations that got infected while connected to malicious sites. Also, the workstations might be used to download infected files which may make the network vulnerable to enterprise scale attacks. Due to these security concerns, the need for monitoring became a necessity. It is important to note that Internet traffic monitoring and network security are complementary. In order to successfully secure any network, both of them have to be considered. This thesis covers Internet traffic monitoring extensively while also considering the security issues related to it. Network security will not be covered in detail but the part of it that was needed will be covered.

Personally, my motivation for choosing this topic was due to the fact that I had the necessary tools and knowledge to carry out the project. During some networking courses

in school such as CCNA (Cisco Certified Network Associate), I found it intriguing to see network packet being mirrored using Wireshark so I decided to further explore the aspect of traffic monitoring during my thesis project. Also, it was important to carry out a thesis project that was relevant to Granlund Oy since I was a trainee in the company. The thesis project also helped to explore the functionality of Bluecoat ProxySG and ProxyAV, which are devices that were acquired by the company for Internet traffic monitoring purposes.

2 Network Overview of Granlund Oy

The network of Granlund is hierarchical in nature and the network was designed to suit the building. In general, Granlund has over 500 workstations and has over 400 employees in Helsinki and other cities in Finland and Russia. It is a service-based company, so the network is crucial to its everyday operation. Granlund specialises in building services design and also offers consultancy services on energy and environmental matters. Furthermore, it offers SaaS (Software as a Service) and that sector is maintenance management software called RYHTI. There are quite a number of shared network resources and files in Granlund and also a reliable VPN (Virtual Private Network) connection for remote users. The core of the network is in Helsinki and other sites are connected to it via MPLS (MultiProtocol Label Switching). The monitoring in this project was carried out in Helsinki which is the core of the network, so further network details will be restricted to the network of Helsinki.

In Helsinki, the building has three floors and each floor has more than one managed switch (Cisco Catalyst 2960) called Floor Switches (according to the company's naming convention) to which the workstations are connected. The floor switches are connected to a central floor switch (Cisco Catalyst 3560) connected directly to the core level switch called the Core Switch (Cisco Catalyst 3750). The core switch is connected to the ISP router but there is a firewall between them to filter network traffic using policy-based routing. The firewall used is Juniper Cluster SSG 320 while Juniper SSL VPN is used for VPN connection authentication to the network. There is also an external firewall (Juniper SSG 140) which controls access to the extranet. The extranet contains the offered SaaS such as RYHTI. It is worth mentioning that the server LAN is excluded from this report. However, Granlund has over 20 servers, both virtual and physical. Some the servers include DNS (Domain Name Server), DC (Domain Controllers), VMware ESX and DHCP (Dynamic Host Configuration Protocol) server.

Figure 1 shows a concise description of the network of Granlund. Basically, it shows the traffic flow from the user to the Internet. There is a slight variation in the naming of the devices but the idea behind the internal naming convention is the same.

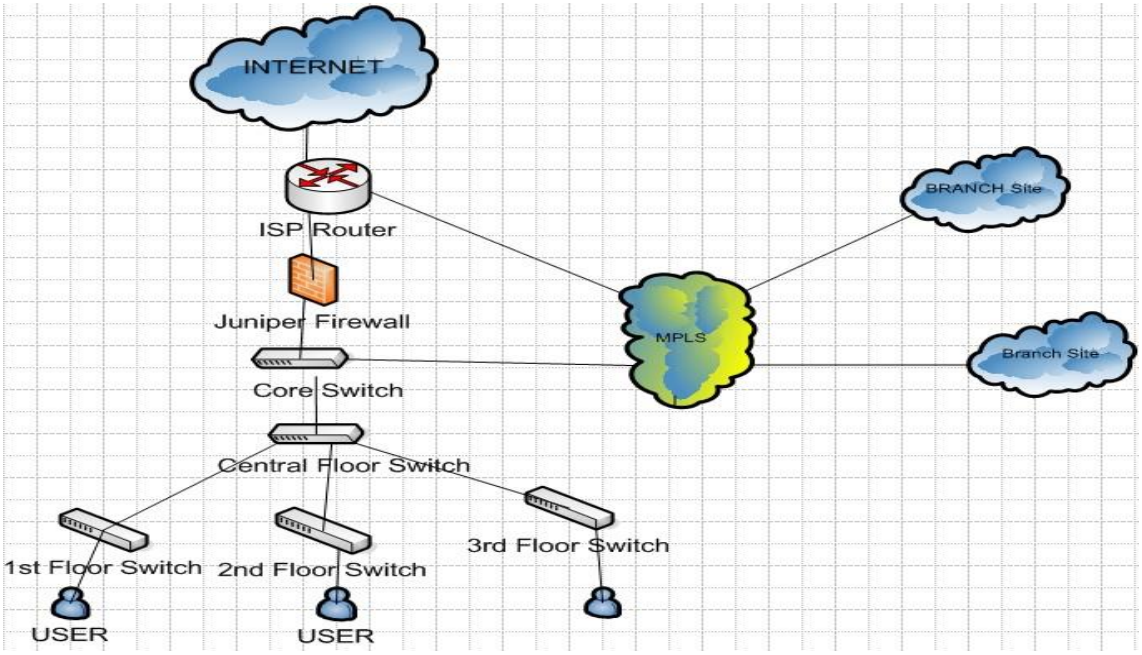


Figure 1: Network overview of Granlund

As shown in figure 1, the MPLS network connects to both the ISP router and the core. These connections enable the branch sites to connect to both the ISP and the core network in Helsinki. The connection between the MPLS and the core switch can be regarded as a WAN (Wide Area Network) link. The WAN connection provides remote connection between the sites but it is important to mention that all the traffic are routed through the core in Helsinki.

3 Internet Traffic Monitoring

Internet Traffic monitoring is a vital part of network management. It provides a means of checking network performance and diagnosing network bottlenecks when they arise. As the network grows, monitoring becomes necessary and at the same time complicated because the monitoring system must be set up in such a way that every branched traffic travelling through the core network should be monitored, so the larger the network, the more complicated it is. The complexity of the network also increases with an increase in the traffic-generated personal devices used by the network users in an enterprise network. Bring-Your-Own-Device (BYOD) is a term used to describe the free-

dom of network users to use their own devices to connect to the network. This increases the need for monitoring because it is impossible to tell which device is clean and which device is infected. If an infected device connects to the network, it is possible for it to infect other devices through network poisoning, which makes network monitoring both crucial and complicated.

Nevertheless, there are ways through which Internet traffic monitoring could be done; it could be done using a hardware or a software approach [1]. The hardware and software approaches differ in quite a few ways. One of their differences is in their respective costs. The hardware approach is much more expensive than the software approach. While the software approach only requires a dedicate network port and a workstation, the hardware requires a dedicate port and a special hardware device used for monitoring, thereby increasing the cost of the later when compared to the former. Another difference is in their mode of monitoring. There are two types of monitoring modes, active monitoring and passive monitoring [12]. In active monitoring, the monitoring device interferes with the network to a measurable extent. Also, the device generates measurable traffic in the network. An example of the traffic generated is the injection of probe packet into the network. In the passive monitoring mode, the tool does not interfere with the network by injecting extra network packet into it [1]. Instead, they act as “sniffers” that only monitor network packet transparently.

Some monitoring hardware could also be used for caching web contents. Due to their caching possibility feature, monitoring hardware are configured to intercept the network traffic, so as to serve as a proxy to other devices in the network, while the software only mirrors the traffic passing through the network without intercepting it. An example of a hardware used for this purpose is Bluecoat ProxySG while both open source and licensed software such as TSTAT and Colasoft Capsa could be used for the software approach. Other aspects of ITM will be discussed subsequently.

3.1 Network Performance Monitoring

Network performance monitoring refers to ways through which a network’s performance can be periodically or constantly measured to see whether it is functioning as expected. It is sometimes referred to as Network Performance Management. Electronic communication privacy laws are in place in Finland and these laws provide guidelines

on network monitoring. The regulatory authority is a body called “Viestintävirasto” (Finnish Communication Regulatory Authority). The main goal of the regulation is to ensure privacy and confidentiality in electronic communication. These regulations govern both users and providers of the networks. Other than for investigative purposes, user data should neither be accessed nor their activity history logged elsewhere (ISPs are allowed to log traffic for a maximum for three weeks) [2].

After setting up a network, it is important to always check its performance to ensure that the desired performance level is being offered. Network alarms can be set so as to inform the network administrator whenever there is a problem in the network or it might alert the network administrator whenever a threshold is reached [13]. For example, the threshold might be a certain level of latency or throughput allowed in the network. There are certain tools that can be used to monitor the performance of a network. Some of those tools include TSTAT, NTOP, and Capsa (Capsa will be covered in section 4.1). There are certain parameters that are measured to determine the performance level of a network. These parameters are called metrics [13]. There are different metrics which should be accounted for in network performance monitoring and they include network performance metrics and system performance metrics [13]. Due to the scope of this thesis, system performance metrics will not be covered but it is important to note that healthy systems or devices perform better than others in a network.

Network performance metrics include channel capacity and utilization, delay and jitter, and packet loss and errors [3]. Channel capacity and utilization refer to the maximum amount of data that can be transmitted through a network channel and the actual fractional capacity of the channel that was utilized during the transmission [13]. This metric depends on the bandwidth and the processing capacity of the medium. It is possible to find out whether the network is being under-utilized or over-utilized using this metric. A delay refers to the amount of time it takes to transmit a packet from source to destination while jitter refers to delay variation (a change in delay) over time [3]. Packet loss and error refer to the failure of a transmitted packet to reach its destination [13]. This metric is often determined by checking the amount of data retransmissions that were recorded by the monitoring tool. Network service performance metrics includes availability, service maintenance, and network usage [12]. Availability refers to the amount of time network resources are available to be used [12]. Service maintenance describes the ability of the network to maintain a certain quality of service. Network usage refers to the amount of network resource that is being used by the users of the network [12].

The network usage statistics enables network administrators to determine its economic value. For example, it is not appropriate for network users to use enterprise bandwidth for entertainment such as audio and video streams because these network requests are bandwidth consuming. Network traffic metrics include bandwidth utilization, network flow, dropped packet and round trip time and jitter [16]. Bandwidth utilization describes how much bandwidth is being consumed and the network requests which they served [16]. Network flow refers to a sequence of packet sent from a source to a destination that the source wishes to describe as a flow, for example, a media stream. Dropped packet refer to those data packet that could not be delivered to their destination due to errors. Round Trip Time (RTT) refers to the time taken by data packet to travel to their destination and back, while jitter refers to variations in the RTT [16].

3.2 Network Security

Network security is one of the most crucial parts of network management. It is not enough to design and set up a well functioning network without making provision for its security. One characteristic of a good network is accessibility. The network must be accessible to its users. In order to ensure that the network is not accessed by unauthorised persons, security measures must be put in place to detect intrusions, and there must be both proactive and reactive plans ready in order to respond to attacks when they occur. The basic goals of network security are prevention, detection and response [17]. Prevention guarantees the right access to the network, detection detects illegal access, while response guarantees a reaction to illegal access [4]. Network security cannot be over-emphasised because it is needed to protect company and personal information on the network from unauthorised access. Also, it is needed as a means of authenticating users accessing the network and to guarantee the availability of network resources when needed. Network security is needed in every network, whether large or small; even home networks should be secured. The focus here will be on securing enterprise networks.

There several ways through which the vulnerability of a network can be exploited but there are also counter-measures to such exploitations. Network security is usually multilayered due to the fact that no one security measure can guarantee 100% security in the network. The multilayered nature of the security measures makes it harder to break into networks because different techniques are needed to attack different security re-

sistance. Some common network attacks are illegal access, which could be prevented by firewalls, network malware and Botnets, which could be detected by intrusion detection, denial of service, which could be prevented by filtering ingress traffic, TCP (Transmission Control Protocol) hijacking, which could be tackled by IPsec (Internet Protocol Security), packet sniffing, which could be prevented by packet encryption, IP (Internet Protocol) spoofing, which could be prevented by using secured network routes, and social errors, such as those from infected social networks, which could be minimised by educating network users. [17]

Furthermore, there are also DNS (Domain Name Server) attacks, Man in the Middle attacks, social engineering which requires network users to reveal their login details to third parties, ICMP (Internet Control Message Protocol) attacks, Smurf attacks, and email spamming. There are many other ways through a network can be breached, hence the need for good security measures to protect the network. Firewalls are set up to prevent unauthorised access to networks. Firewalls can either be software-based or hardware-based. Most enterprises use dedicated hardware for firewalling purposes. Some examples of those are Juniper and Checkpoint firewalls. There are also some firewalls known as next generation firewalls which can also perform application control and intrusion detection; a good example of this is Palo Alto next generation firewall. Firewalls are primarily there to control both incoming and outgoing network traffic. They do so by enforcing certain rules set by the network administrator. For example, an enterprise firewall usually prevents outgoing traffic to remote locations that are not owned by the company.

Firewalls operate by analysing network packet based on preset rules to determine whether or not those packets should be allowed to enter or leave the network. Firewalls define the DMZ (Demilitarized Zone) of enterprise networks. DMZs are usually flanked by firewalls. One firewall is set up between the DMZ and the Internet while the other is set up between the DMZ and the intranet. Different sets of rules exist in both firewalls, with the sole purpose of controlling access to either the Internet or the intranet. Denial-of-Service (DoS) or Distributed-Denial-of-Service (DDoS) attacks are attacks carried out with the aim of making network resources available to intended network users. DDoS is a different kind of DoS attack that targets a large number of users. These kinds of attacks involve sending out irrelevant packet to the network so as to overwhelm it or prevent it from serving its purpose either temporarily or indefinitely. There are various forms of DoS attacks. Some of them are, for example, SYN flood attacks

where a flood of TCP/SYN packet are sent to a host or various hosts in the network to make them wait endlessly for acknowledgement packet from the sender, and Low-rate DoS where TCP retransmission time out (RTO) mechanism is being exploited by making hosts on the network wait endlessly for retransmitted TCP packet. Also in this category is ping-of-death. [17]

Notwithstanding, ingress traffic filtering is a technique of used in preventing DoS attacks. It does so by analysing incoming packet to verify that the sender is legit. If the sender cannot be verified, then the packet is discarded. Furthermore, Smurf attack is a common DoS attack where by the broadcast ping IP is not genuine thereby overloading the target network. TCP hijacking or interception is another common network attack that involves intercepting network packet. It can either be blind TCP hijacking or network session hijacking. In the former, the TCP sequence numbers are forged by either guessing or using brute force and the right port to send SYN or RST flag packet which would disrupt network connections while in the later (also known as Man-In-the-Middle attack), the attacker ceases the session by becoming the middle man in the network thereby making clients to assume that it is the server and making the server to assume that it is the client. Both methods could cause network storms such as ACK and ARP storms in the network. [14]

A workable way of preventing this is by encrypting and authenticating network packet using the IP Security (IPsec) protocol. IPsec is a protocol used in securing IP communication sessions by ensuring mutual authentication between agents before establishing IP session connections. IPsec is one of the main features of IPV6 but it has been optional in IPV4. It helps to protect communication between hosts, networks and gateways. Some other secure systems that are widely used are Secure Sockets Layer (SSL), Transport Layer Security (TSL) and Secure Shell (SSH) [14]. IPsec makes it challenging to sniff network packet. Other forms of attacks are DNS attacks and social engineering. DNS spoofing (sometimes called DNS cache poisoning) is a type of attack that requires malicious data to be injected into the DNS server thereby rerouting traffic to the attacker's computer or site [6]. This attack can also be targeted at the cache. It stores malicious data in the DNS cache, so that when network users make similar queries or requests, the same result will be served. The sites of the attackers often contain executable malicious files that might harm the computers of users or introduce worms into the network. This type of DNS attack can be prevented by authentication through the use of DNS Security Extension (DNSSEC). One example of such a DNS attack

was the DNS attack of 2011 in Brazil. During the attack, the hackers gained access to DSLs (Digital Subscriber Line) handed out by ISPs, thereby redirecting network traffic by changing the primary DNS address to malicious servers thus they were able to steal bank login details (among other confidential information) of some users [20].

Social engineering is a network attack using social tactics or error-of-judgement of the network users by manipulating them into divulging personal or confidential information, usually their login details such as usernames and password [4]. This is a form of attack that is proving increasingly challenging to prevent, due to the fact that network users might not be enlightened enough to spot suspicious requests and the attacker does not have to be physically present to carry out the attack. Also, it is challenging to spot such attacks proactively because the attacker gains access to the network or system illegally using valid credentials. However, the commonest form of social engineering attack is phishing [5]. Phishing involves the attacker (phisher) sending false emails that appear to be from legitimate businesses asking receivers of the emails to send their personal details as replies or to click on malicious verification links in order to get users to type in their login details on the spoofed site [5].

Most enterprises often send anti-spam emails warning their network users about the existence and circulation of such emails known as spams [5]. Educating network or system users and creating awareness has been one of the most effective ways of preventing such attacks and also having a proactive plan in place in case suspicions of such attacks are reported. Furthermore, the organization's security policy should be stressed during orientation. For example, the roles of the technical support team should be well documented so that attackers who make phone calls posing to be technical support can be easily detected when they ask for sensitive information that should not be revealed to any other person. Also, if the user reveals such details, the technical team should be contacted immediately to curtail the attack by maybe disabling that particular account. Considering all these possible attacks and countermeasures, it is necessary to have a system that monitors network activities to detect intrusions. This type of system is called Intrusion Detection and Prevention System.

3.3 Intrusion Detection and Prevention System (IDPS)

Intrusion detection is described as the process of monitoring events in a system or network so as to detect violations to acceptable standards and policies. The IDP (Intru-

sion Detection and Prevention) is usually carried out by a device or software. It detects abnormal activities (whether malicious or not), alerts the security administrator or prevents the activity from execution. These activities could be malware such as worms, spywares, and Trojans, attackers gaining or attempting to gain unauthorised access, and mistakes done by authorised users that raise suspicions for example, trying out remote access to other workstations or servers. [7]

There are two main forms of intrusion detection, Intrusion Detection System (IDS) and Intrusion Prevention Systems. The main difference is that the IPS is able to prevent intrusions when detected, while the IDS cannot prevent intrusion. IDS are designed to monitor intrusions, while IPS focuses on identifying and blocking intrusion threats. In this report, the term IDPS will be used to refer to both because the IPS is simply an extension of the IDS [7]. IDPS are mainly used to identify and prevent unwanted incidents by using certain security thresholds set by the security administrator. IDPS can also be used to ensure compliance to enterprise security policies and to log detected threats and attempts. These functions have made it necessary for organizations to have an IDPS. IDPS technologies perform three main functions: recording and logging observed events, notifying administrators about observed events (this is also known as alerts), and reporting [15]. The reporting might be done by some other reporting software from the logs collected.

Furthermore, the prevention system also stops the attacks detected, changes the content of the attack by filling files with harmless messages and by making changes to the security environment if granted that right. IDPS technologies use at least one of these three detection techniques: signature-based, anomaly-based, and stateful protocol [7]. Multiple techniques are usually used to attain a high level of attack detection because lone techniques might prove insufficient. Signature-based detection is the simplest form of detection and it involves monitoring and comparing network packet with known attack patterns known as signatures. An example of a signature is a suspicious file name that ends with “exe” (freewallpaper.exe). Signature-based detection uses string/character comparison to detect anomaly so any change in the threat name will bypass this technique. For example, if the initial file name (freewallpaper.exe) was renamed to be free1wallpaper.exe, it will not be detected by this technique. Also, this technique lacks the ability to recall previous requests when processing current requests.

The limitations described above necessitated the need for another or additional technique. Anomaly-based detection was developed to complement the weaknesses of the signature-based technique [7]. This detection method detects attacks by detecting network anomalies. The detection system observes normal patterns in the network, such as bandwidth usage, common protocols used, and devices that often connect to each other; so whenever any activity that is unusual occurs in the network, the detection system detects it and alerts the administrator. For an IDPS that uses this technique to function, network profiles describing normal activities in the network should be created, so as to enable the system to compare network activities with predefined network profiles or rules [15]. The challenge with this system is accuracy, frequency, and dynamism of the network profiles. The profiles are usually generated by the system during the training period when the IDS is being tested and is not monitoring intrusion. After generating the profiles, accuracy cannot be guaranteed because a malicious process might be considered safe and the frequency of the profile generation comes into question.

There are two types of profiles generated: static and dynamic profiles [7]. Static profiles are those that remain unchanged when an anomaly is detected, while dynamic profiles are those that are changed when anomaly is detected [7]. Generated profiles should be generated over a period of time ranging from hours to months and they should be able to cope with false positives. False positives are observed events that are categorised as malicious when they are not, while false negatives are observed events that are considered safe when they are malicious. The ratio of false positives should be typically higher than that of false negatives to ensure a high detection rate. A stateful protocol analysis addresses the weaknesses of both signature-based and anomaly-based detection [7]. This analysis involves a comparison of profiles generally accepted protocols against observed activities but it differs from anomaly-based detection. Unlike anomaly-based detection, the network profiles are vendor-provided universal profiles [7]. IDPS which uses this method connects to a vendor-specified server to download profiles which describe how protocols should behave generally in the network. Also, the IDPS is able to understand the state of the network, transport, and application protocols [7].

The stateful protocol analysis method keeps track of the sequence of commands issued in the network, so that whenever a command is issued too many times or a prerequisite command is ignored or not issued, it could classify it as suspicious. For proto-

cols that perform authentication, this method records the authenticator used for each session such as an authenticated FTP (File Transfer Protocol) connection. However, this IDPS analysis method has its flaws as well. Due to the fact that protocol standards made by standard bodies (e.g. IETF (Internet Engineering Task Force) or software vendors can be modified by various vendors, it is almost impossible for the system to understand all variations of various protocols. Furthermore, vendor specific protocols might be challenging for the system because the details of those protocols are not often available freely to the general public. A stateful protocol analysis is also resource-intensive due to its complexity and overheads generated while keeping track of many concurrent sessions. [7]

In the future, considering all the detection techniques, creating a hybrid from them is necessary. The hybrid should combine the strengths of the techniques while minimising the drawbacks that will arise from its application. For any new IDPS to function properly, it should be able to minimise the overheads generated by stateful protocol analysis hence the need for compression and acceleration of known protocols according to the frequencies of their occurrence in the network. Furthermore, there should be new standard bodies formed to manage vendor-specific protocols, so as to create a means of generalisation. When generalisation is in place, the hybrid IDPS can check for certain parameters in order to accurately detect and prevent intrusions as well as to offer reduced false positives. The intrusion detection system that will be covered in this report is Snort. Detailed features and also some configuration procedure will be highlighted.

Generally speaking, even though there are many types of IDPS technologies, they can be categorised into two: host-based and network-based detection [15]. The main difference between them is that host-based IDPS technologies focus on events on a single host, while network-based technologies focus on events on the network [15]. In this report, the focus will be on network-based technologies because the monitoring devices that will be introduced later use network-based technologies. Network-based technologies monitor the network to detect suspicious activities whose source might be a host on the network or an application protocol. They are usually deployed on the borders of the network, for example in between a firewall and router. They can be set to intercept the network traffic, analyse it, and then either forward or drop the packet or they can be set in a promiscuous mode in which they only mirror the traffic passing

through them. Both systems where tested during this thesis project. It is recommended to use the interception method for more satisfactory results.

3.4 Internet Bot and Malware Traffic

Internet bots are also called web robots. They are often used to perform repetitive tasks on the Internet. A good example is automatic bidding on sites such as Ebay. These bids are increased when necessary by bots whose purpose is to keep increasing bids each time a high bid has been offered. When monitoring network traffic, there are always some traffic logs that cannot be explained because they are either being generated by legitimate sites or malicious sites. Such traffic can be bot traffic generated by Facebook Ads and Google's DoubleClick, which are not harmful. However, bots are usually exploited for malicious purposes [14]. Some can be used to gain control of a user's computer when they are run unknowingly and some could be used for harvesting email addresses from forums and online guest lists. These types of bots are called SpamBots.

Bot traffic is usually unnoticed by antivirus or endpoint because it is usually sophisticated and hidden, hence the need for an IDPS system that could detect bot activities in the network. Once any bot traffic has been detected, it is important to sanitise the infect device to prevent the spread of the infection in order for it not grow into a botnet (a group of bot infected devices). Malware traffic refers to those generated by malware, either host-based or network-based malware. They are usually harmful because they cause harm to both their host and the network. A typical traffic generated by malware is a remote connection to an unknown external server. Most malware requires outbound connections to their masters, usually remote servers, to get new instructions and to transfer vital information from their infected hosts to their masters [6]. To limit this possibility it is important to shut down unused ports.

3.5 Application Protocols in Granlund's Network

There are usually many protocols travelling through the network but in this thesis project, some vendor specific protocols will be ignored, while common protocols will be mentioned. However, the characteristics of the individual protocols are beyond the scope of this thesis. Granlund Oy has two functional wired and wireless networks but

the main focus was on the protocols being used in the LAN network. In the physical layer, the protocol used is Fast Ethernet at 100 Mbps, that is, the connection between the workstations of the users and the floor switches, as shown in figure 1. In reality, this speed is not achieved due to latency and host capacity. On an average, 80 Mbps is generally obtainable in the company. Some application protocols as observed in Granlund's network include the following:

- CIFS (Common Internet File System): used for file sharing across networks.
- HTTP/HTTPS (Hypertext Transfer Protocol/ Hypertext Transfer Protocol Secure): used for distributed, collaborative and hypermedia information systems. HTTPS is secured HTTP.
- LDAP (Lightweight Directory Access Protocol): used for maintaining distributed directory information services.
- LDP (Label Distribution Protocol): MPLS label exchange protocol used by MPLS routers.
- Kerberos: provides authentication for applications communicating over an unsecured network.
- MSRDP (Microsoft Remote Desktop Protocol): used for remote connection on windows computers.
- DNS (Domain Name Server): used to name Internet-connected hosts.
- SNMP (Simple Network Management Protocol): used to manage devices in a network.
- SSDP (Simple Service Discovery Protocol): used for host discovery in a network.
- DHCP (Dynamic Host Configuration Protocol): used to dynamically assign address to nodes on the network.
- NTP (Network Time Protocol): used for time synchronisation on the network. Most organization prefers to use NTP to set the time on nodes on the network.
- RSH (Remote Shell): executes shell commands on the network.

However, the traffic of interest is the Internet traffic, so, as expected, IP (Internet Protocol) as well as TCP and UDP protocols were observed in the network. In the network

layer, IP is the protocol used on the Internet and it uses the overlying TCP and UDP protocols to transport IP packet in the transport layer. IPV4 is used but IPV6 traffic can be observed from Internet sources. In the link layer, ARP is used to resolve IP addresses into physical addresses and vice versa. ICMP is commonly used to send error messages and to check the status of the connected hosts on the network.

4 Data Collection Tools

Data collection tools were used to collect network data during the project. Both the hardware and software approach was used during the project. The hardware used was Bluecoat Proxy SG and Proxy AV while the software used was Snort and Colasoft Capsa. Although there are several commercial and open-source monitoring tools available, these tools were chosen based on the needs of the company and the goal of the project. The Bluecoat devices were chosen because of their caching abilities while Colasoft Capsa was chosen because of its user interface, which is relatively easy to use and also its ability to analyse network traffic at packet level and highlight possible network bottlenecks that should be given attention. Snort was chosen based on the fact that it is open-source and provides a means of comparing the performances of high-end commercial monitoring tools such as the Bluecoat devices with a free open-source tool such as Snort. None of the tools is perfect as they all have their unique individual strengths and weaknesses. These tools can be grouped into two: performance and security monitoring tools. Bluecoat Proxy SG has both features when handling HTTP/HTTPS traffic, while the Proxy AV is an antivirus scanner that is capable of scanning files for malicious contents. The first tool that will be considered is Capsa, which is free edition.

4.1 Colasoft CAPSA

Colasoft Capsa (free edition) is network analyzing software that is used to decode packet travelling through a network. It captures network packet, groups them according to their protocols and analyses them in order to find bottlenecks in their operations. An analysis by Capsa enables network administrators to troubleshoot network performance problems. The packet are captured in real-time and analysed accurately, so the network administrator can react proactively to the network performance and security issues detected by Capsa. The monitored port is configured in the promiscuous mode

so as to capture packet effectively. Capsa can be installed on a desktop or laptop and can be used to analyse specific network segments. Capsa is able to save network packet and replay them. Also, it can be configured to perform specific analysis such as using the filters in it for application analysis. Furthermore, just like in Bluecoat Proxy SG, Capsa also monitors HTTP traffic in real-time. It monitors and logs FTP (File Transfer Protocol), email, DNS and chat logs as well. Its GUI is designed like Microsoft's Office 2007, thereby making it easy for administrators to find the information they want, and it is also customizable to suit the needs of whoever is using it. It can be used to plot network statistical graphs and connections between different nodes on the network as well as generate network vital reports when needed.[10]

4.1.1 Features of Capsa

Colasoft Capsa has many features but only those which were beneficial to the execution of the project will be highlighted here. The first of those features is extended network security analysis. Capsa is able to detect multiple common attacks such as ARP attacks, TCP port scanning activities, and network worm activities in real-time by in-depth packet decoding and analysis. Also, Capsa offers advanced network protocol analysis by giving a clear picture of all the protocols travelling through the network, thereby making it less challenging to spot abnormalities. Capsa is also capable of giving automatic expert network diagnosis based on the data it collects. It spots network problems and offers possible solutions to the bottlenecks spotted. Furthermore, it offers conversation analysis by depicting the numerous communications in a matrix showing the source and destination addresses of them all. It is also able to reconstruct packet stream when necessary during investigative monitoring when tracing the source of any alert. Capsa could be used in monitoring multiple network behavior such as HTTP, email, DNS, FTP, and chat applications. In addition, it offers versatile traffic and bandwidth statistics on each host and individual network protocol. [10]

4.1.2 Benefits of Capsa

Colasoft Capsa was beneficial in various ways to both the network of Granlund and the project. The first benefit observed was its in-depth and comprehensive network traffic analysis. Capsa analysed a wide range of traffic including web traffic, LAN traffic, email conversations, and chat application (such as MSN) traffic. This analytic feature gave a

clear picture of the kind of traffic travelling through the network and it made the challenge of concentrating on the goal of the project easier in terms of isolation of network traffic segments. Also, it offered effective network communication monitoring by providing pictorial matrices of the various communications in the network. This enabled me to be able to trace communication paths upon receiving suspicious conversation alerts. It was possible to trace the source and destination as well as the kind of protocol being used during such conversations. [10]

Furthermore, Capsa offered automatic network bottleneck diagnoses (see figure 13 for a list of them). These diagnoses gave an idea of the problem observed by Capsa in the network when analysing copies of captured network packet. Capsa also helped in network security analysis by adding severity tags to observed events. Not all suspicious events are security-related, so by tagging these events, a network administrator can be able to prioritise effectively when handling alerts. Also, those events that were performance-related were also tagged accordingly. During network performance analysis, it was plain to see those communications that encountered performance-related problems such as slow connections. At packet level, Capsa was able to offer network protocol analysis by effectively stripping copies of network packet in order to see the kind of protocols being used in the network for communication whether it was internal or external. [10]

4.1.3 Deployment of Capsa

The position of the Capsa device is essential to the kind of network traffic it sees. It is usually positioned according to the monitoring purpose intended, either a network segment or the whole network. In our case, it was intended to monitor the whole network, so it was deployed to monitor the LAN traffic passing through the core of the network to the Internet. Port mirroring (also called port spanning) when configured on any of the interfaces on a network switch sends a copy of the network packet passing through it or an entire VLAN (Virtual Local Area Network) to a specified destination port [10]. This enables the monitoring device to strip captured copies of network packet and analyse them.

Figure 2 shows the deployment of Capsa in Granlund.

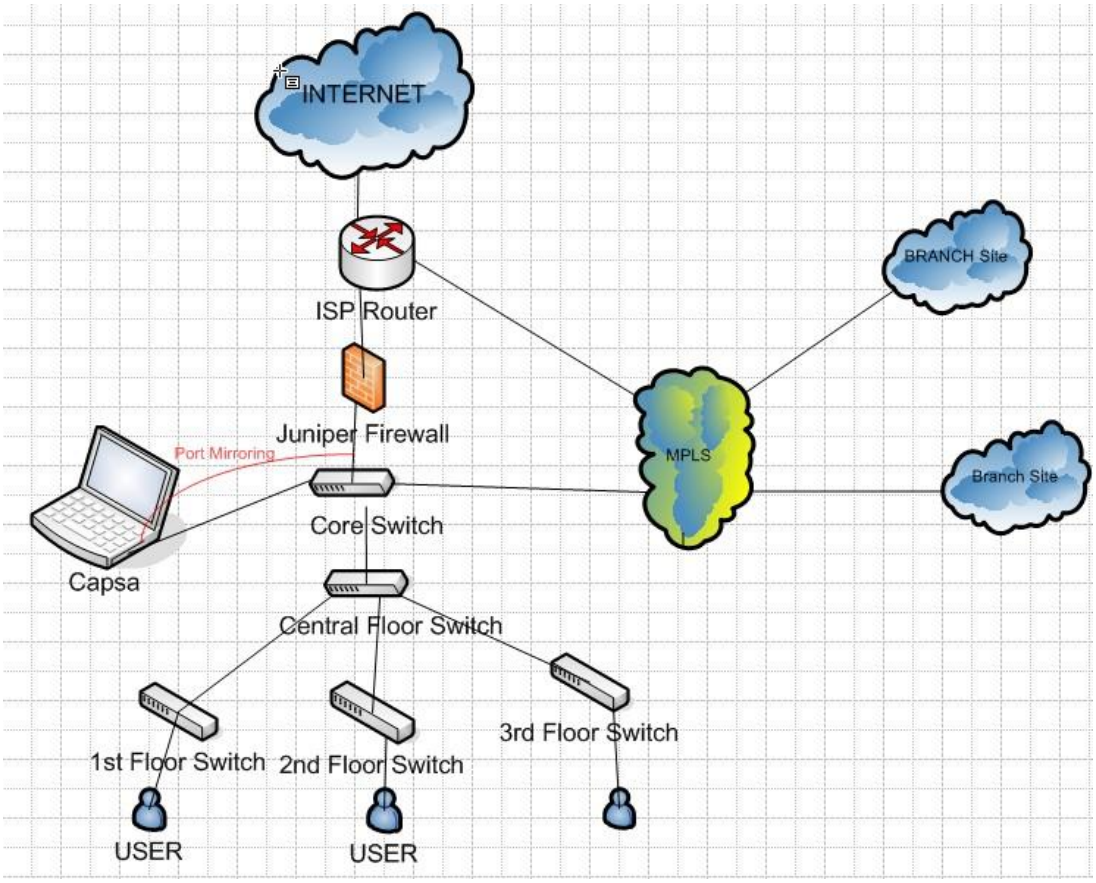


Figure 2: Capsa Deployment in Granlund

As shown in figure 2, the Capsa device is strategically positioned to capture outgoing raw data packet before they reach the firewall while the incoming packet are captured after passing the firewall, which gives a clear picture about the bottleneck in the network. Port monitoring was configured on the switch in such a way that the source ports were specified while the port to which the Capsa device was connected was specified as the destination port. It is important to note that it is generally possible to specify multiple sources and destinations when configuring spanning on a switch. For example, traffic from two different source ports could be sent to a single destination port.

Figure 3 shows an overview of the start page of Capsa.

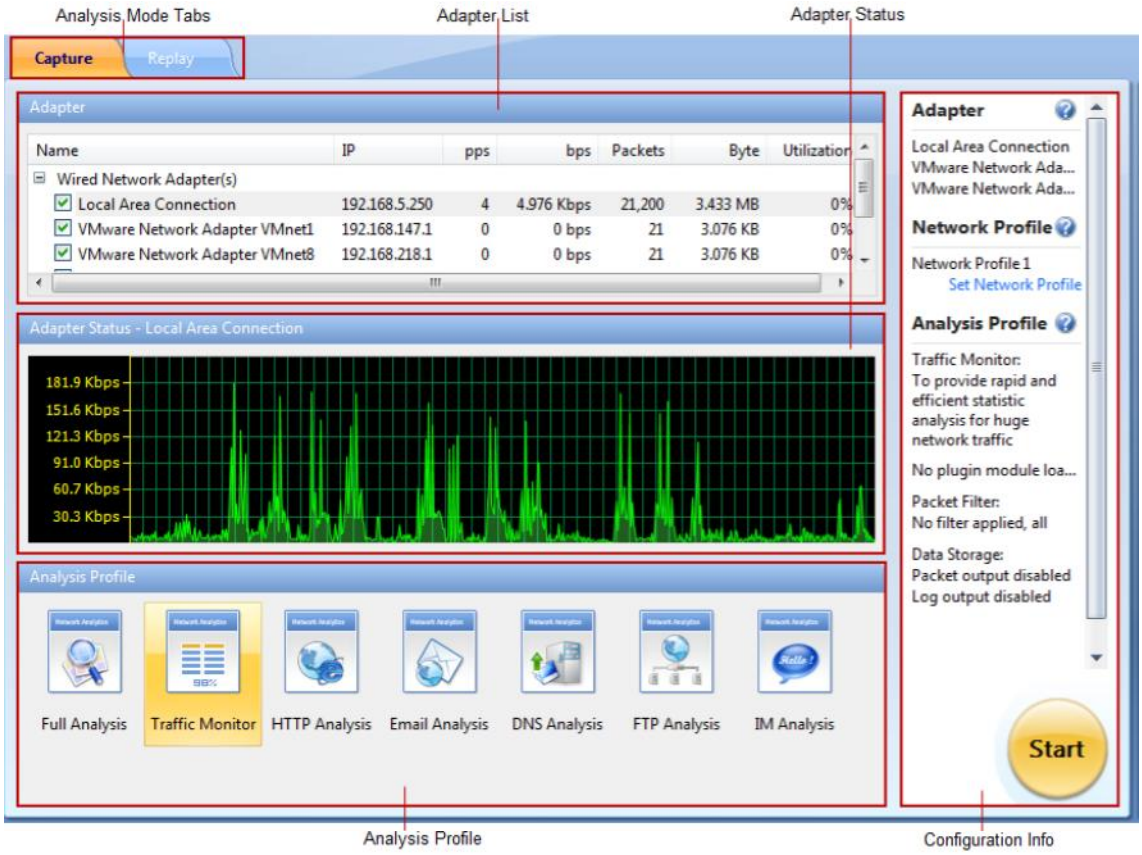


Figure 3: Start page of Capsa (copied from [10])

As shown in figure 3, Capsa can be used to monitor the traffic passing through various network adapters. It can be adapted to monitor certain kinds of traffic including IM (Instant Messenger) traffic. Furthermore, it can be configured to only capture packet that match certain profiles as well as protocols using its embedded filter. The replay section can be used to replay stored packet when necessary. During the project, a full analysis was always used except in certain cases when persistent issues had to be isolated.

Figure 4 shows an overview of main user interface.

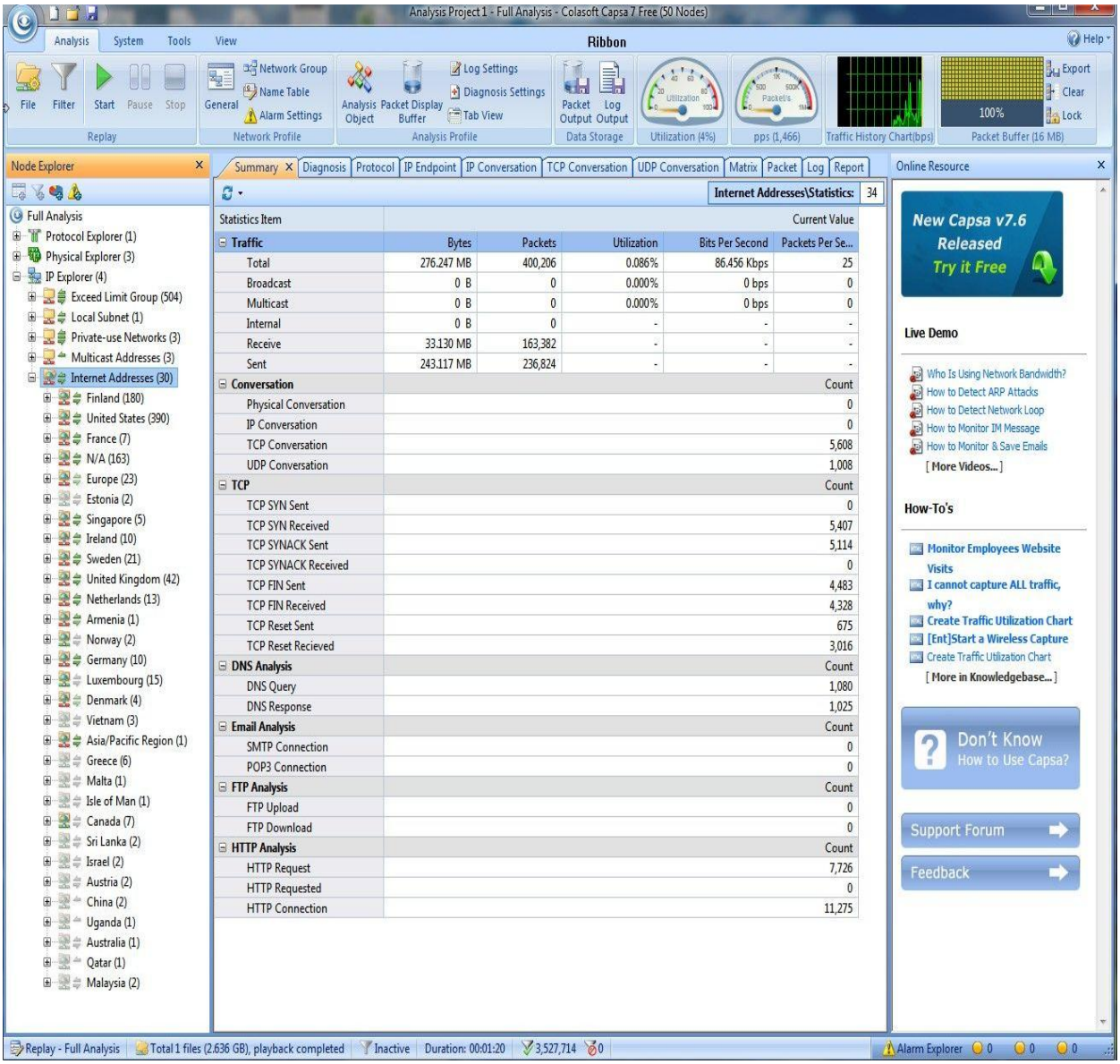


Figure 4: Main user interface of Capsa (copied from [10])

As shown in figure 4, the “Ribbon” mark is pointing to the log output. The logs could be outputted to a specified directory/folder to be replayed later. The node explorer groups the network nodes according to their IP addresses and the WebPages visited are grouped according to the countries hosting them. The mid-section shows the areas of interest during troubleshooting, which are diagnosis and protocol. The diagnosis tab shows the statistics of the errors and bottlenecks found during the packet capture session, while the protocol tab shows the statistics of the protocols observed in the network. Other sections worth mentioning are the report tab and the help tab. The report tab shows pictorial summary of the information collected by Capsa, while the help section contains useful information about the collected information and instructions on how

to use the software. Figure 5 shows an overview of the diagnosis tab. The detailed implications and solutions to the network issues shown in the diagnosis tab will be discussed in chapter 5.

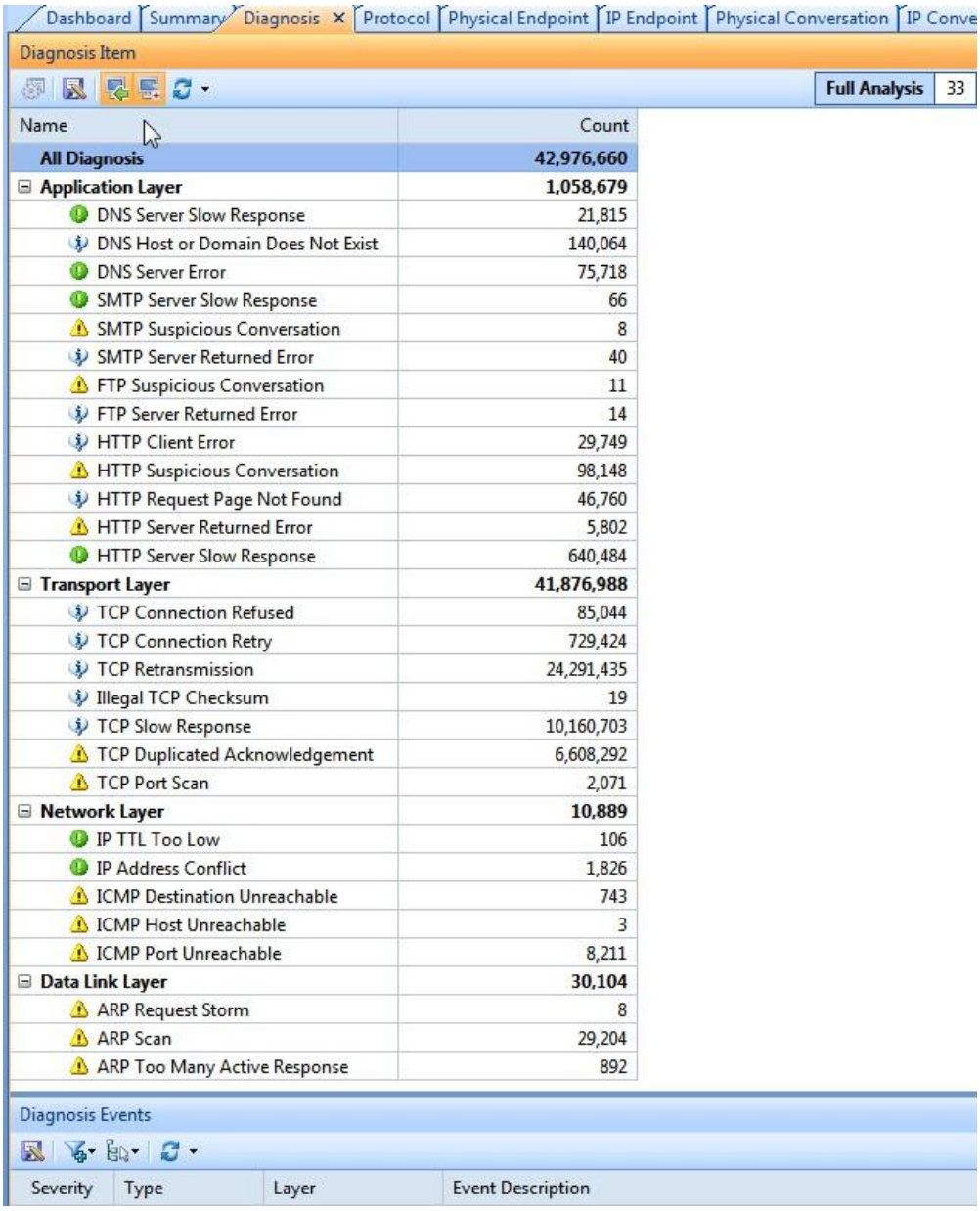
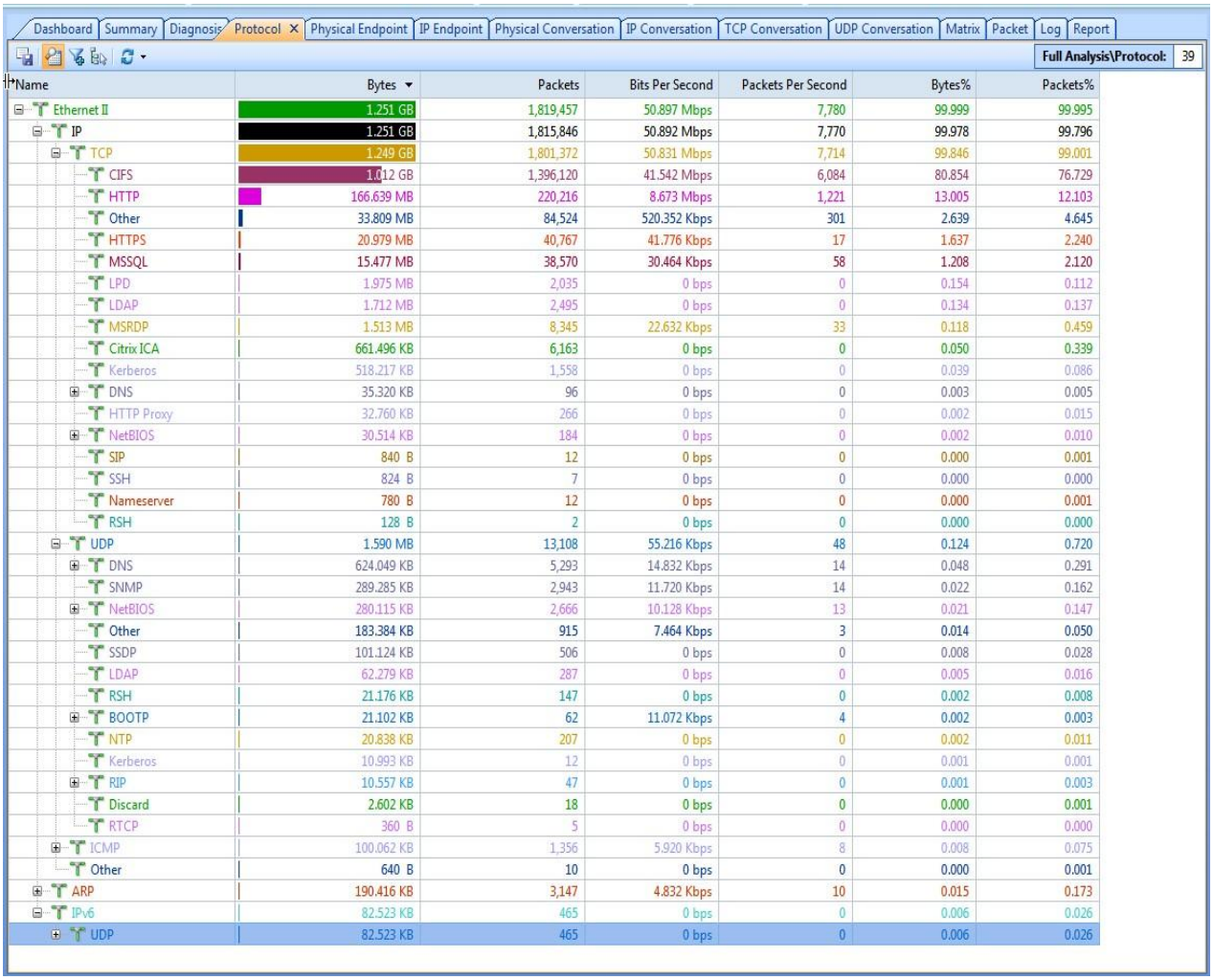


Figure 5: Diagnosis Tab of Capsa.

As shown in figure 5, the network issues are grouped according to the layer of the network in which they occur. The four main layers where network errors occur are application, transport, network, and link layer. The count row shows the number of times the errors occur during the data collection. Figure 6 is an overview of the protocol tab. It shows the type of protocols observed in the network.



Name	Bytes	Packets	Bits Per Second	Packets Per Second	Bytes%	Packets%
Ethernet II	1.251 GB	1,819,457	50.897 Mbps	7,780	99.999	99.995
IP	1.251 GB	1,815,846	50.892 Mbps	7,770	99.978	99.796
TCP	1.249 GB	1,801,372	50.831 Mbps	7,714	99.846	99.001
CIFS	1.012 GB	1,396,120	41.542 Mbps	6,084	80.854	76.729
HTTP	166.639 MB	220,216	8.673 Mbps	1,221	13.005	12.103
Other	33.809 MB	84,524	520.352 Kbps	301	2.639	4.645
HTTPS	20.979 MB	40,767	41.776 Kbps	17	1.637	2.240
MSSQL	15.477 MB	38,570	30.464 Kbps	58	1.208	2.120
LPD	1.975 MB	2,035	0 bps	0	0.154	0.112
LDAP	1.712 MB	2,495	0 bps	0	0.134	0.137
MSRDP	1.513 MB	8,345	22.632 Kbps	33	0.118	0.459
Citrix ICA	661.496 KB	6,163	0 bps	0	0.050	0.339
Kerberos	518.217 KB	1,558	0 bps	0	0.039	0.086
DNS	35.320 KB	96	0 bps	0	0.003	0.005
HTTP Proxy	32.760 KB	266	0 bps	0	0.002	0.015
NetBIOS	30.514 KB	184	0 bps	0	0.002	0.010
SIP	840 B	12	0 bps	0	0.000	0.001
SSH	824 B	7	0 bps	0	0.000	0.000
Nameserver	780 B	12	0 bps	0	0.000	0.001
RSH	128 B	2	0 bps	0	0.000	0.000
UDP	1.590 MB	13,108	55.216 Kbps	48	0.124	0.720
DNS	624.049 KB	5,293	14.832 Kbps	14	0.048	0.291
SNMP	289.285 KB	2,943	11.720 Kbps	14	0.022	0.162
NetBIOS	280.115 KB	2,666	10.128 Kbps	13	0.021	0.147
Other	183.384 KB	915	7.464 Kbps	3	0.014	0.050
SSDP	101.124 KB	506	0 bps	0	0.008	0.028
LDAP	62.279 KB	287	0 bps	0	0.005	0.016
RSH	21.176 KB	147	0 bps	0	0.002	0.008
BOOTP	21.102 KB	62	11.072 Kbps	4	0.002	0.003
NTP	20.838 KB	207	0 bps	0	0.002	0.011
Kerberos	10.993 KB	12	0 bps	0	0.001	0.001
RIP	10.557 KB	47	0 bps	0	0.001	0.003
Discard	2.602 KB	18	0 bps	0	0.000	0.001
RTCP	360 B	5	0 bps	0	0.000	0.000
ICMP	100.062 KB	1,356	5.920 Kbps	8	0.008	0.075
Other	640 B	10	0 bps	0	0.000	0.001
ARP	190.416 KB	3,147	4.832 Kbps	10	0.015	0.173
IPv6	82.523 KB	465	0 bps	0	0.006	0.026
UDP	82.523 KB	465	0 bps	0	0.006	0.026

Figure 6: Protocols detected by Capsa

As shown in figure 6, the protocols are all Ethernet protocols, mostly IP. They are further grouped into TCP, UDP and ICMP. Also, a group was created for ARP and IPV6 observed traffic.

4.2 Snort

Snort was created by Martin Roesch in 1998 but it is now developed by SourceFire (Martin Roesch is the founder of SourceFire) [11]. Snort is an open-source network-based intrusion detection tool with the capacity to capture real-time network traffic and packet logging on an IP network. Snort uses a set of rules to combine some of the benefits of signature, protocol, and anomaly-based inspection methods to achieve intrusion detection. Snort can be used in a network for protocol analysis and content searching/matching [11]. It could also be used to detect probes or attacks such as port

scans, OS (operating system) fingerprinting, buffer overflows, and SMB (Server Message Block) probes. Snort can be configured in three modes: sniffer mode, packet logger mode, and intrusion detection mode [11]. In the sniffer mode, Snort simply promiscuously collects and displays network packet, while in the packet logger mode, it collects the packet and logs them into a database or the hard disk of its host. In the intrusion detection mode, it collects network packet and analyses them using a pre-defined set of rules, in order to decide on how the packet should be handled (whether to drop and log, allow and log, or simply generate an alert).

During the project, Snort was configured in the hybrid mode, that is, in both the packet logger and the intrusion detection mode [11]. The packet logger used was Syslog Watcher, which is a freeware used for logging system data. The rulesets used were downloaded from www.Snort.org. The Snort ruleset can be downloaded only as either a subscriber or a registered user. Registered users get the privilege of getting the rulesets a month before the subscribers but, while the registered subscription is a paid version, the general subscription is free. It is also possible to use self-written rules in Snort. This feature brings flexibility and adaptability to the use of Snort especially in an enterprise environment where too many false positives are obtainable and the need for fine-tuning is ever present. However, Snort does not clean up infected hosts when detected but creates an alert for that particular incident. Most network worms and Trojans operate by scanning the network from infected hosts to find other hosts with exploitable vulnerabilities. These scans usually generate packet in the network which Snort can analyse to detect suspicious activities. This analysis can create false-positives because legitimate scanning from maybe a software license key scanner might be reported as suspicious, hence the need for tailoring self-written or common rules to suit the network environment. One way to achieve this in Snort is to set the appropriate threshold. The configuration file called “threshold.conf” file could be modified for this purpose.

4.2.1 Features of Snort

The first feature of Snort worth mentioning in relation to the project is the fact that it is open source. Its open-source nature and widespread use are some of the reasons why it was chosen for the project. Also, it is an IDPS that could be used in different modes either host-based or network-based. However, it was used as a network-based IDPS during the project. Using syslog or a simple text file, Snort is able to alert/log observed threats in real-time. The alerts are generated with the help of its detection engine which

is able to detect different types of attacks. It matches the observed traffic patterns with the rules in the engine, so as to detect intrusions. Snort could be run on different OS platforms even though its native OS is Linux. The programming language used in Snort's detection engine is a simple language that describes per action tests and actions. It is easy to use and it makes the detection of new threats faster. [19]

4.2.2 Benefits of Snort

Snort is relatively inexpensive because the software is open source and it could be installed on off-the-shelf workstations. Most commercial IDS require expensive dedicated hardware while the use of Snort does not require any. In addition, Snort operations are fast. Snort can be installed on multiple workstations and placed in strategic places in order to effectively monitor network segments. Also, it could be used to monitor multiple locations from one physical location (just like it was set up during the project). Furthermore, Snort rules are easy to write because they do not involve the use of any complex programming language. [19]

4.2.3 Deployment of Snort Sensor

Snort sensor (the workstation on which it was installed) was deployed in a way similar to Capsa as shown in figure 2. However, the hardware and software requirement for Snort is slightly different from that of Capsa due to the fact that Snort can be configured to make decisions based on pre-installed rulesets. This decision-making feature will function optimally when the hardware has moderate specifications (the higher the specification, the faster the packet processing will be). During the project, a dedicated workstation was used for packet capturing. It only received copies of the packet travelling through the network, analysed the packet, and created alerts when suspicious activities were found. It was not configured to make any decision. The Snort sensor had the following specifications:

- Windows 7 64-bit Operating system (OS)
- Intel Core 2 Quad CPU Q9300 @2.50 GHz
- 6 GB RAM.

Snort is open-source so it should be used on an open-source OS such as Ubuntu but there is a way to install it on Windows as well. In order for it to be installed on Windows,

some required software components for its functionality should be installed first. This software includes:

- Antivirus software (Microsoft Security Essential)
- Firewall software (Comodo Firewall)
- Microsoft Baseline Security Analyser
- ActivePerl 5.10.1.1007 (higher versions were not compatible)
- Notepad++
- PDF reader (Adobe Reader)
- Syslog Watcher 4
- 7-Zip
- WinPcap
- Snort 2.9.3.1
- Oinkmaster. [17]

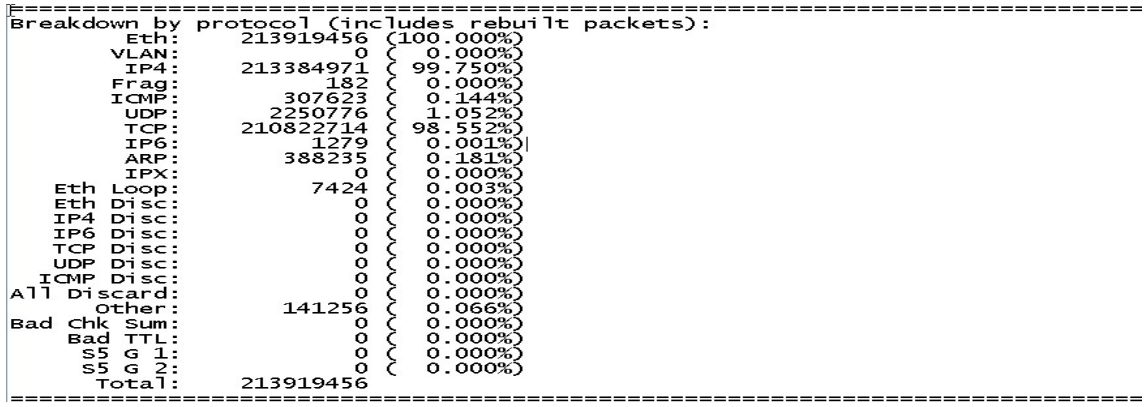
All this software is freeware. The antivirus and firewall listed are different from those used during the project (Symantec Endpoint and Juniper firewall were used). The detailed procedure of the installation is beyond the scope of this report. However, the installation guidelines can be found online [18]. Snort shows certain details on the CLI (command line interface). Figure 7 shows the statistics of the packet analysed by the sensor.

```
===============================================================================
Run time for packet processing was 74327.488000 seconds
Snort processed 213919367 packets.
Snort ran for 0 days 20 hours 38 minutes 47 seconds
    Pkts/hr:     10695968
   Pkts/min:       172794
   Pkts/sec:         2878
===============================================================================
Packet I/O Totals:
   Received:    215202870
   Analyzed:    213919367 ( 99.404%)
    Dropped:      1283600 (  0.593%)
   Filtered:            0 (  0.000%)
Outstanding:      1283503 (  0.596%)
   Injected:            0
-------------------------------------------------------------------------------
```

Figure 7: Snort packet statistics

As shown in figure 7, the time taken for the packet to be analysed was recorded which was approximately 20.5 hours during one test session. Furthermore, the packet flow statistics was also recorded in pkts/hr, pkts/min, or pkts/sec. The sensor succeeded in analyzing the received packet with an efficiency of over 99% due to the fact that only 0.593% was dropped without analysis. Figure 8 shows the protocols observed in the

packet collected. Our monitoring was focused on the LAN so Ethernet protocols were detected as expected.

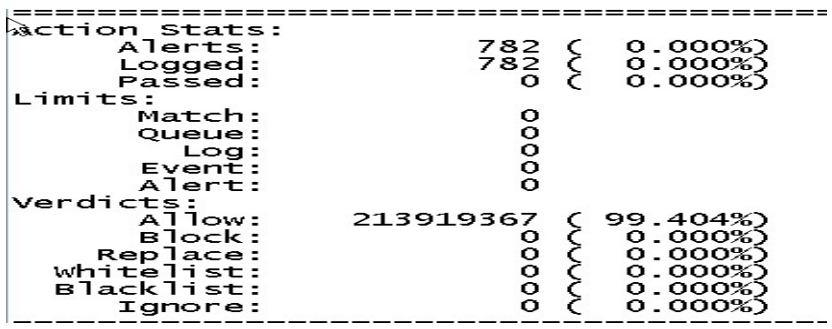


```
===============================================================================
Breakdown by protocol (includes rebuilt packets):
            Eth:    213919456 (100.000%)
           VLAN:            0 (  0.000%)
            IP4:    213384971 ( 99.750%)
           Frag:          182 (  0.000%)
           ICMP:       307623 (  0.144%)
            UDP:      2250776 (  1.052%)
            TCP:    210822714 ( 98.552%)
            IP6:         1279 (  0.001%)
            ARP:       388235 (  0.181%)
            IPX:            0 (  0.000%)
       Eth Loop:         7424 (  0.003%)
       Eth Disc:            0 (  0.000%)
       IP4 Disc:            0 (  0.000%)
       IP6 Disc:            0 (  0.000%)
       TCP Disc:            0 (  0.000%)
       UDP Disc:            0 (  0.000%)
      ICMP Disc:            0 (  0.000%)
All Discard:                0 (  0.000%)
         Other:       141256 (  0.066%)
Bad Chk Sum:                0 (  0.000%)
     Bad TTL:               0 (  0.000%)
      S5 G 1:               0 (  0.000%)
      S5 G 2:               0 (  0.000%)
       Total:    213919456
===============================================================================
```

Figure 8: Protocols observed in the network.

In figure 8, it can be observed that 100% Ethernet traffic was observed as expected. One statistic of interest is Eth Loop (0.003%). These are error broadcast/multicast packet sent as a result of a few possible reasons. It might have resulted from spanning tree errors on a particular switch or excessive multipath-packet-delivery in the network. Further details and solutions will be given in the next chapter. The sum of the packet of other protocols (such as IPV4, ICMP, UDP, TCP, IPV6, and Frag) is equal to the number of Eth packet which is equal to the total number of packet captured.

In figure 9, the statistics of action taken on the analysed packet are shown. The Snort sensor was configured in the sniffer mode (to analysed copies of mirrored packet), so no negative action was expected to be taken on the captured packet.



```
===================================
Action Stats:
      Alerts:             782 (  0.000%)
      Logged:             782 (  0.000%)
      Passed:               0 (  0.000%)
Limits:
       Match:               0
       Queue:               0
         Log:               0
       Event:               0
       Alert:               0
Verdicts:
       Allow:       213919367 ( 99.404%)
       Block:               0 (  0.000%)
     Replace:               0 (  0.000%)
   Whitelist:               0 (  0.000%)
   Blacklist:               0 (  0.000%)
      Ignore:               0 (  0.000%)
-----------------------------------
```

Figure 9: Action statistics of captured packet

As shown in figure 9, the entire packets captured were allowed although it is possible to configure the sensor to take other actions. The alert log created gives a more de-

tailed view of the suspicious packet captured. Usually in modes other than the sniffer mode, the Snort sensor blocks this type of traffic. Snort uses preprocessors through which captured packet are scanned before they are sent to the detection engine. These preprocessors and the detection engine trigger rules according to the protocols they handle. Typically, each protocol has its preprocessor. Some of the widely used preprocessors are HTTP, FTP, GTP (GPRS Tunneling Protocol), SMTP, SIP (Session Initiation Protocol), DCERPC2, Stream5, and Frag3 preprocessors. Preprocessors operate in the same way as plug-ins, so they can either be turned on or off according to the needs of the network administrator.

During the project, HTTP traffic was ignored (until the later stages) in Snort because Bluecoat was the dedicated device monitoring web traffic (details on Bluecoat are given in sections 4.3 and 4.4). The detection engine serves as the decision-making room. It receives data from the preprocessors, checks them through a set of rules, and creates an alert or drops the packet if necessary. It is important to note that attacks can be directed at the Snort sensor so high security practice is advised when setting up the sensor. One of those security measures will be installing regular patches to the system when available. Also, the sensor should have two network interfaces (NIC). One of the interfaces will be assigned an IP address and it will be used as the management interface for remote connection (a private IP address will be most preferred) while the other will be without any IP configuration because it will be used for packet collection.

4.3 Bluecoat ProxySG

Bluecoat ProxySG appliances are part of the family of Bluecoat security solutions. The ProxySG provides a platform for secure web communications and application acceleration. The web security is provided in the form of web filtering while the acceleration is done via its caching ability. ProxySG can be used to implement flexible, granular, policy control over content, users, and applications [8]. The ProxySG editions used during the project were two stacked ProxySG 600-10. Each of them was licensed to handle 500 concurrent users, a 250 GB SATA disk drive, and 4 GB RAM. Furthermore, each device had two 1000Base-T NICs. One of the NICs was used for management while the other was used as a bypass port. A bypass port provides a failover access port for an in-line monitoring device such as the ProxySG [8]. Since two ProxySGs were used, proper load balancing had to be configured. As mentioned in section 2, Granlund has

three floors, so the traffic from the floors was divided between the two ProxySGs. However, the first ProxySG had the highest amount of web traffic directed towards it. [8]

4.3.1 Features of ProxySG

In view of Granlund's network needs, the ProxySG offers some exciting features that made the execution of the project hitch-free. One of those features is complete web protection through web filtering, deep inspection of contents, inspection and validation of SSL traffic, content caching and traffic optimization, bandwidth management, streaming media control, and applications protocols control in order to prevent data loss and to drastically suppress web threats. Web protection is the main feature used in Granlund, so as to ensure the security of the users when accessing web contents. This is due to the autonomy given to each user as regards the use of their workstations. Furthermore, the applications are control or powered by the application policy engine of the device, which made it possible to allow selective web applications access via Granlund's network. This was to ensure web security and in some cases, eliminate online distractions that could reduce work-related productivity of the employees.

The ProxySG offered high performance and reliability due to its multiple processing cores, lightweight customized OS, called SGOS, which rarely needs patching, and the failover feature turned on in the stack to guarantee service delivery in case of a device failure. In case of a device failure, the traffic is automatically routed via the second functioning device. The devices are also equipped with device monitors such as fan rotation and CPU usage monitors which help administrators monitor the health of the devices. Another feature tested during the project was Bluecoat ProxyClient which could be installed on non-stationary devices such as laptops, to ensure that the same web policy used in Granlund's LAN was used whenever the user connected to the Internet using the devices of the company. This feature ensured remote web security and the implementation of uniform security policy for both LAN and remote users. [8]

4.3.2 Benefits of ProxySG

The proxy provided protection to both the hosts and the network from web threats, phishing, and other attacks by filtering the traffic entering the network. This web filtering ability made it possible to enforce Granlund's web access policies in the network by preventing access to certain web categories, such as pornography, that are not allowed

or are not considered to be work-related. Also, its caching ability reduces bandwidth consumption by serving cached copies whenever the same page is requested. This speedy delivery of cached copies can be considered an improvement to general network performance. The proxy does not only support HTTP/HTTPS traffic, but it also supports other protocols such as FTP, RTMP (Real-time Messaging Protocol), RTSP (Real-time Streaming Protocol), DNS, Telnet, and streaming media protocols. [8]

4.3.3 Deployment of ProxySG

The deployment of ProxySG is similar to that of Capsa and it is exactly the same as the deployment of ProxyAV which will be discussed in section 4.4.3. The main highlight of the deployment is the policy-based routing, which means routing network traffic based on firewall policies. The firewall was configured to forward traffic from certain network addresses to the proxy and the proxy checks the traffic before allowing, warning, or blocking it from accessing the network. For example, the firewall does not block traffic to sites with explicit contents but it forwards web traffic to the proxy. The proxy operates based on its own set of rules which it uses to determine what action to take, concerning requests for explicit web contents. The preconfigured action on the proxy for such contents is to block them according to Granlund's web policy, so the access is denied and the user is notified about it. The proxy is deployed right next to the firewall.

Figure 10 shows the deployment of ProxySG in Granlund's network environment.

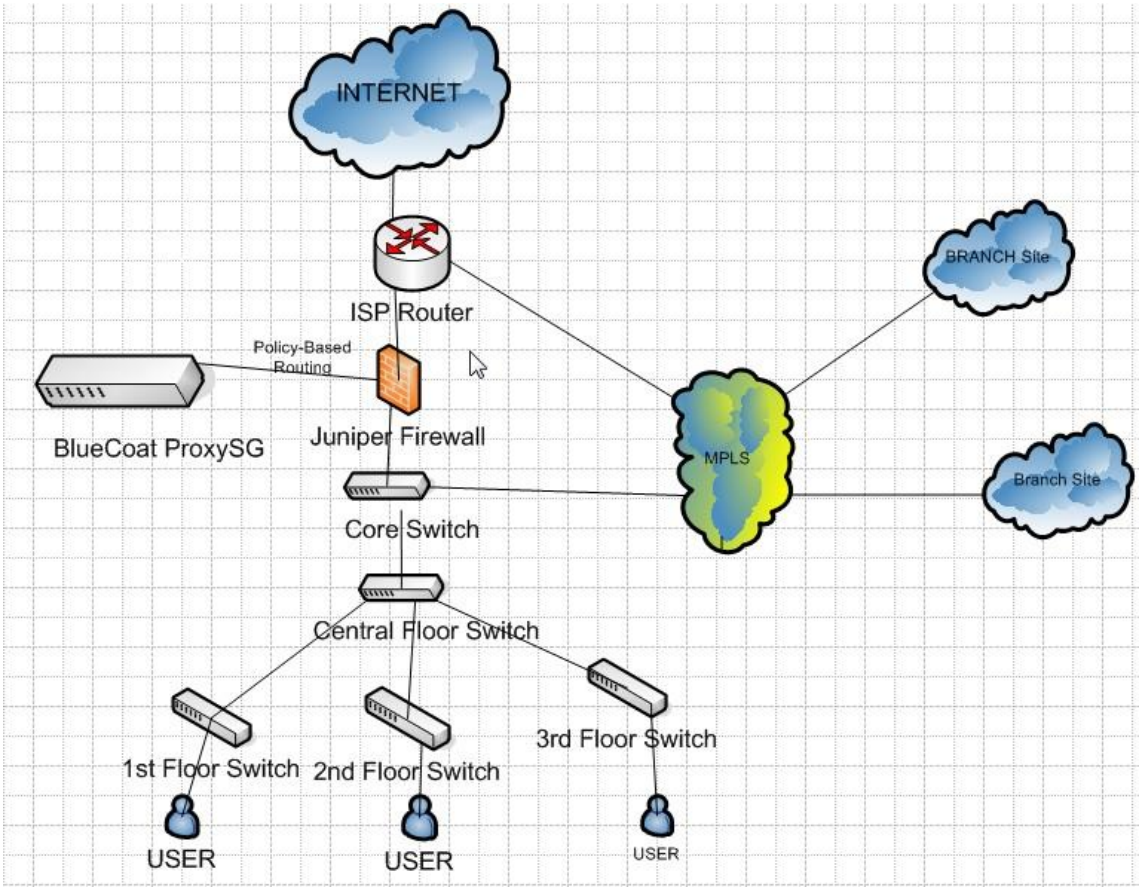


Figure 10: Deployment of Bluecoat ProxySG.

As shown in figure 10, the proxy is positioned next to the firewall and the traffic is directed from the firewall to the proxy via policy-based routing. Exceptions can be created to bypass the proxy when necessary because there is direct route from the firewall to the Internet. Even though it is not depicted in figure 10, the ProxySG is a stack of two proxies, which makes it possible to load-balance the network traffic to ensure fast web responses.

4.3.4 Overview of ProxySG's GUI

Figure 11 shows an overview of the GUI (graphic user interface) of ProxySG's management console.

Figure 11: GUI of ProxySG's Management Console.

As shown in figure 11, ProxySG's GUI is divided into sections. The model and serial number as well as the OS version can be clearly seen while in the far right hand corner, the support and documentation tab can be seen. The documentation tab contains information about each section, which enables users to get an explanation about the specific section in which they are exploring. This feature helps to prevent the study of voluminous manuals. On the left hand side, the possible configurable sections of the proxy can be seen. Under the policy section, the visual policy manager (VPM) manages policy creation and implementation. The ProxyClient section handles the deploy-

ment and configuration of the ProxyClient. The content filtering section controls the web access.

There are options as regards the filter to use during web filtering. Internet Watch Foundation (IWF) and Bluecoat webfilter are commonly used, but during the project, Bluecoat webfilter was the used. The webfilter contains different categories (such as extreme, sports or news) and these categories can be either blocked, allowed, or can warn the user when accessing certain web categories. In the diagnostic section, it is possible to check the categories of any website whose category appears unknown (this is often the case with local sites). It is also possible to submit a website to Bluecoat to be reassessed if it has been wrongly categorized. The external section is configured to connect the ProxySG with the ProxyAV for virus or malware scanning of Internet contents (more details will be given later in section 4.4). The scanning process is referred to as ICAP (Internet Content Adaptation Protocol) scanning. In the maintenance section, the license and health of the proxy device can be monitored. Furthermore, it is also possible to carry out a software upgrade when available. The easiest way to know how healthy the device is, is to check the color of the “OK” mark in the far-right-hand corner of the GUI (green color signifies healthiness while red signifies faultiness). [8]

4.4 Bluecoat ProxyAV

ProxyAV offers advanced malware detection at the gateway level of the network. Traffic routed towards the proxy devices by the firewall are scanned by the ProxyAV before being delivered to the user. The malware scanning feature is configured alongside the ProxySG. In the ProxySG configuration, when ICAP scanning is enabled, the IP address of the ProxyAV is added as the antivirus/antimalware engine to be used for scanning Internet contents before being delivered to the user. For example, when a user tries to download files from the Internet, these files are first scanned by the ProxyAV before sending them to the user via the ProxySG. This type of protection is known as Inline threat protection. Usually, ProxyAV and ProxySG are deployed together in the same network environment for maximum protection and unrivaled performance. These combinations provide Internet security to the user and acts as an added security layer in a case whereby the user uses antivirus software. The ProxyAV hardware is similar to that of the ProxySG. It is equipped with a quad core CPU, 500 GB SATA disk drive, 3 GB RAM, and two 1000 Base-T NICs which serves the same purpose as those in the ProxySG. [9]

4.4.1 Features of ProxyAV

ProxyAV was used in carrying out inline threat analysis using Kaspersky engine as its malware engine. This inline threat analysis prevents web threats from entering into the network. Also, it was used in scanning downloaded Internet files before they are delivered to the user. When an infected file is observed, it is discarded and the download process will be interrupted, so as to ensure the overall security of the network. It also has a high performance processor and it is scalable, that is, it could be stacked as the network grows. Furthermore, it has a deferred scanning mechanism which enables it to intelligently avoid the scanning of media contents such as live streams (both audio and video) and video-on-demand. [9]

4.4.2 Benefits of ProxyAV

When used together with ProxySG, ProxyAV offers both high performance and security effectively. It scans files entering into the network thoroughly in order to detect infected files and block them from being delivered to the user. Infected files are often executable and there are sources which are largely reported as unknown, so whenever a publisher cannot be verified at the proxy level, it is blocked from getting to the user. This blocking effect provides an added security to the network hosts in cases where the antivirus software of the host is either missing or not function properly. Selective scanning can be configured on the proxy. ProxyAV was configured to avoid scanning files larger than 500 MB, so as to avoid unnecessary device resource consumption due to the fact that CAD files, which are usually large, are often sent to and from Granlund's network. Furthermore, a safe list containing websites with large downloadable file often visited in the company was created, to avoid the files from being scanned by the proxy. However, large files leaving the network are not scanned even though the proxy could be used to scan both incoming and outgoing traffic. The proxy's malware definition is updated automatically and frequently, so as to provide zero-day attacks. [9]

4.4.3 Deployment of ProxyAV

The deployment of ProxyAv is the same as that of ProxySG. For curious security enthusiasts, it is worthwhile mentioning that there is a free version of Kaspersky malware scanning engine called K9 Web Protection which can be used to implement web con-

tent filtering and parental control. This web filter is the same as Bluecoat webfilter. It could be installed on computers and mobile phones (including iOS and Android devices).

4.4.4 Overview of ProxyAV's GUI

Figure 12 shows an overview of the ProxyAV's management console GUI.

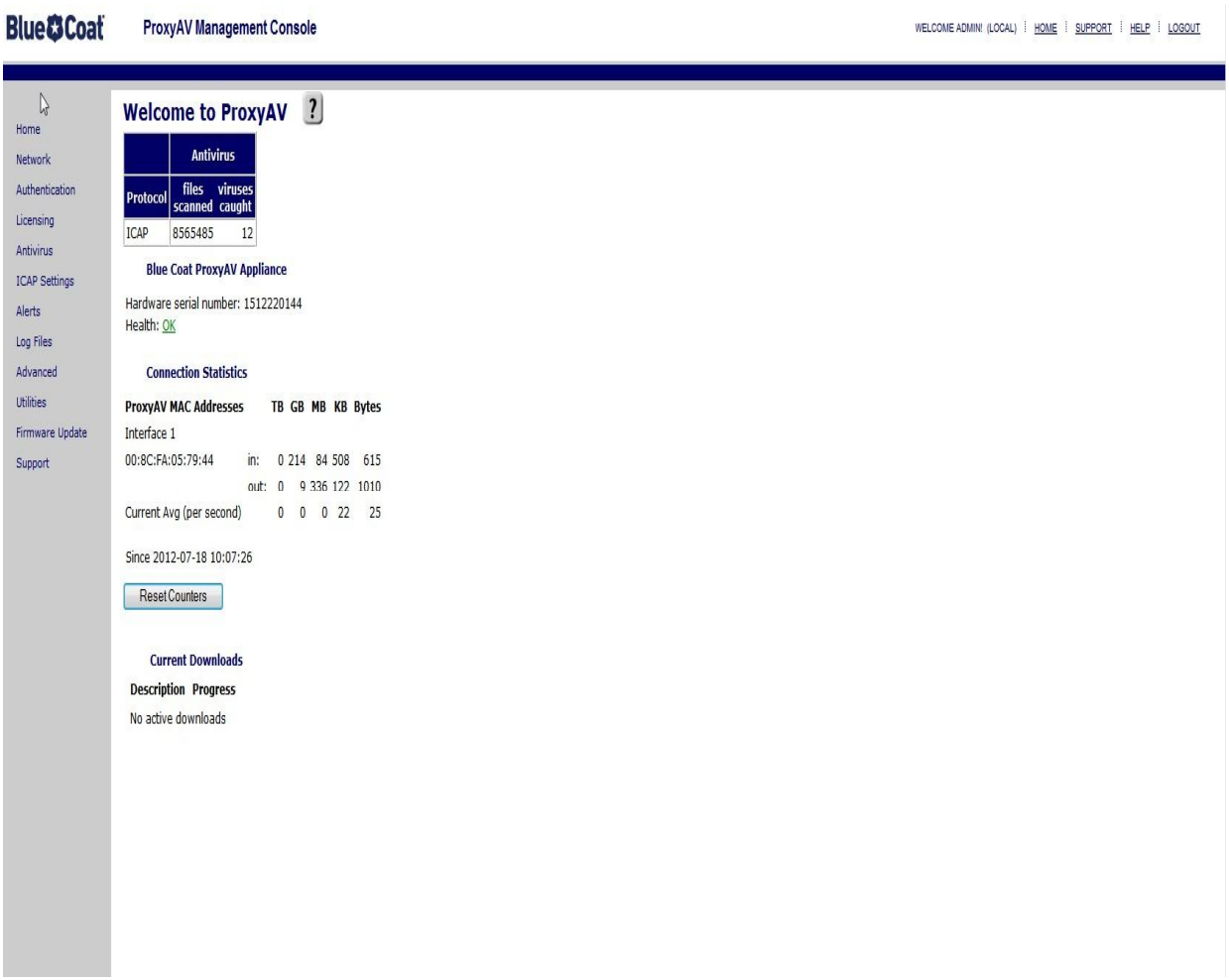


Figure 12: GUI of ProxyAV Management console.

As shown in figure 12, the protocol used is ICAP (Internet Content Adaptation Protocol), which is a lightweight protocol similar to HTTP, used proxy servers in order to implement virus or malware scanning and content filtering in proxy caches. The health status and the help section (named documentation section in ProxySG) are similar to those in the GUI of ProxySG. The number of viruses caught is 12 out of 8565485 files scanned. In the antivirus section, the scanning behavior of the proxy can be configured. Also, the files are logged, to enable network administrators to troubleshoot network

issues when necessary. In order to get well designed graphical reports from Bluecoat proxies, software called Bluecoat Reporter was used. It is log processing software that receives log files via FTP transfer, processes them and displays statistical results graphically.

Figure 13 shows an overview of the GUI of Bluecoat Reporter.

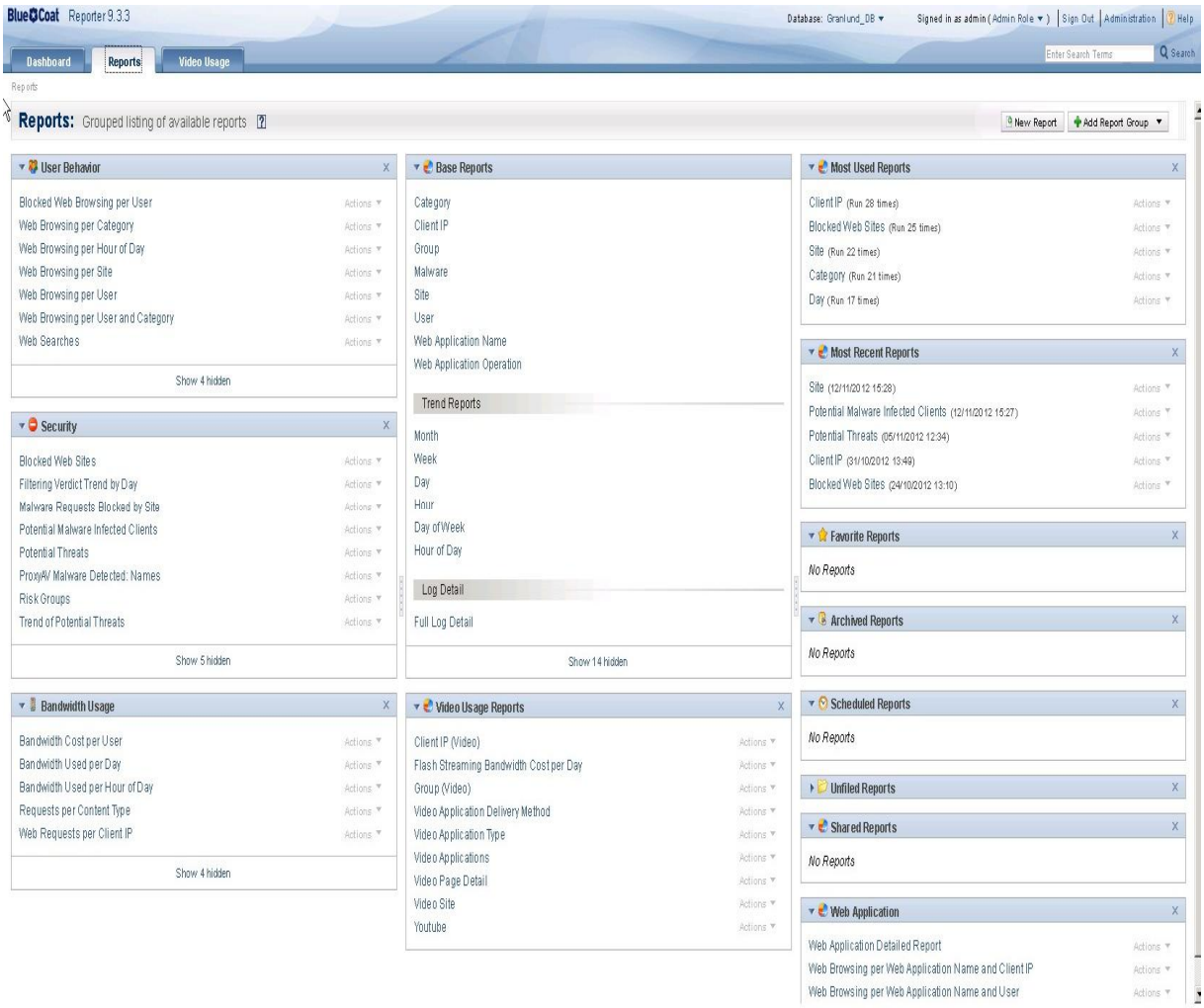


Figure 13: GUI of Bluecoat Reporter.

As shown in figure 13, the possible reports are grouped into categories. The main categories of interest are user behavior and security because it is important to know the general habits of the users and the security threats that follow in order to make policies that will ensure network safety. The dashboard section shows the graphical representation of the listed categories while the video usage section shows the bandwidth usage on media streams in the network. One main concern of most enterprises is the bandwidth usage. It is necessary for acquired bandwidth to be used for work-related purposes, hence the need to monitor its usage periodically. However, according to the

privacy law in Finland, the amount of information gathered on individual browsing habits is limited, so as to give the users the necessary privacy they need when browsing. Detailed drills are only necessary in some cases such as suspected espionage or troubleshooting bottlenecks. This privacy law was adhered to during this project. Chapter 5 explains the results of the project and solutions to network bottlenecks encountered.

5 Data Collected and Analyses of the Results

5.1 Data Collected by Capsa and Analyses of the Results

As mentioned in section 4.1, Colasoft Capsa (free edition) is a network analyzing software that is used to decode packet travelling through a network. Having set up Capsa as described in section 4.1, the main focus was to identify and correct the bottlenecks observed in the network. The bottlenecks were grouped according to the OSI layer in which they occur. Some were in the application layer, transport layer, network layer and data link layer. Figure 13 shows an overview of the number of errors diagnosed during the packet capturing by Capsa. In general, no severe errors were observed. Nevertheless, there were many error packet travelling through the network. The total number of packet captured was approximately 1.1 billion packet.

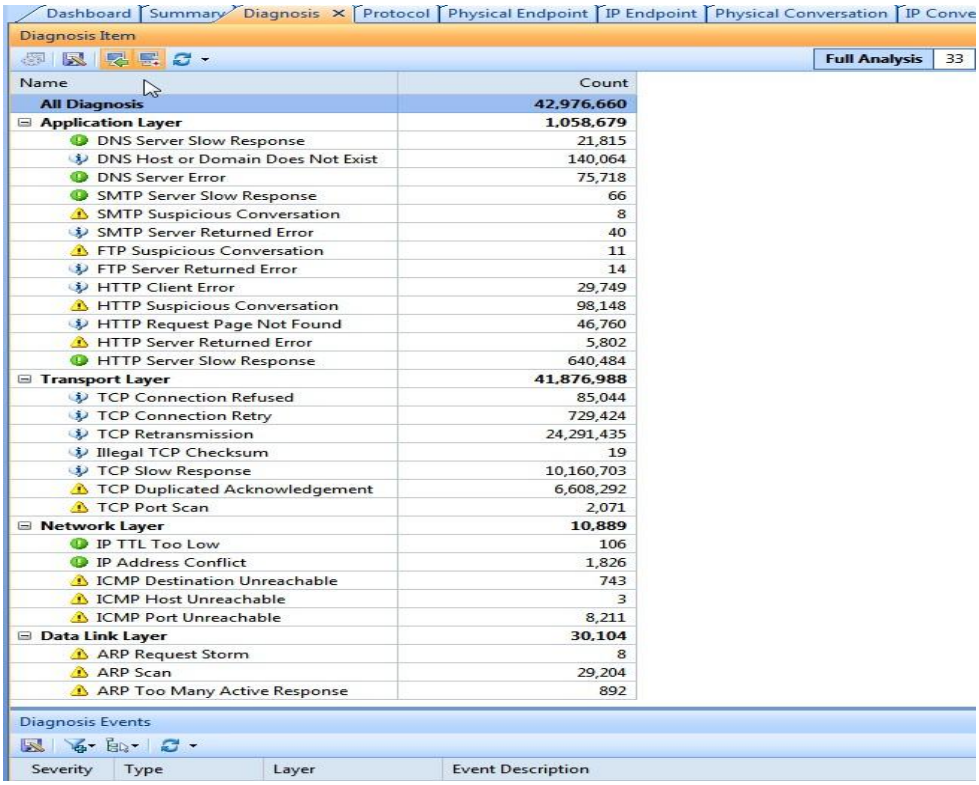


Figure 14: Errors observed by Capsa.

In the application layer, bottlenecks relating to DNS, SMTP (Simple Mail Transfer Protocol), FTP, and HTTP were observed but for the sake fluidity in the report, HTTP security issues will be covered in section 5.2, under Bluecoat while the performance issues will be explained in this section. In the transport layer, TCP retransmission packet account for over 50% of the error packet. TCP is a connection-oriented protocol which guarantees packet delivery during transmission by usually resending damaged packet when acknowledgement (ACK) packet are lost. This unique feature of TCP prompted several retransmissions in the network because it is common for Internet packet to be damaged, lost or delayed during transmission due to differences in the capacity of the sending host and the path through which they are sent.

In the application layer, there were over a million error packet logged. The FTP error-alerts are also false positives because although large numbers of files were sent via FTP to a remote server, the entire files transferred were from the Bluecoat device to the server hosting the Bluecoat reporter software. These log files are often very large and could trigger false alert just as reported in Capsa. The HTTP performance errors are those related with slow server response, server returned error, and the request page not found. These error messages were normal alerts that should exist in a net-

work like Granlund's where there are many simultaneous connections to the Internet. Not all web servers on the Internet are fast and not all of them are geographically close to users. It is normal to get a slow server response when accessing sites that usually experience heavy traffic (such as Facebook and Google) or even local sites that are hosted on slow host servers.

Considering the number of packet in the network, the HTTP alerts were not alarming because quite many sites use HTTP cookies which store users browsing data so as to enable faster information delivery the next time the user visits that particular site. In Granlund, the ProxySG is used as a cache which speeds up the delivery of often visited sites. The ProxySG serves as a proxy server that facilitates access to web contents by serving as an intermediary between users and the web resource they are accessing. Some web contents should not be cached such as news and media streaming sites because they are dynamic (changing often) in nature. Not all Internet pages are accessible, so the requested page not found alert was normal as well.

The inaccessible pages might have been blocked or were offline. Lastly, with regards to the application layer errors, DNS errors were also observed. Some of the DNS errors were performance-related, while others were faults. In Granlund, an internal DNS server behind the firewall is used. Internal DNS servers use private addresses while external DNS servers use public address – Google has a public DNS service that is free (IP addresses 8.8.8.8 and 8.8.4.4). The solution to the slow response DNS server alerts was to check the physical link to make sure it was not overloaded while the other two alerts were insignificant because they were considered normal, considering the total volume of packet travelling through the network and the individual habits of users. Some users might have too many active windows opened on their PC, which carry out automatic updates of their contents without the interaction of the users.

In the transport layer, which is the layer with the most error packet, the total number of error packet captured was around 42.8 million packet. The most prominent error recorded was TCP retransmission. It is important to note that retransmission errors are not security related; instead, they are performance-related. Most of the errors were generated during Symantec Endpoint Protection (SEP) update and DNS response. SEP is hosted on a dedicated server which distributes security policy updates to its remote clients so it is not unusual to find out that during packet transmission, retransmission was required considering the behavior of TCP which guarantees packet delivery. Some

workstations also reported many TCP retransmissions, and when checked, it was discovered that some of the workstations had too many active programs (such as Skype and iTunes update service) that were in frequent connection with remote servers, which occupied the network making it slow.

A slow network path is also the major cause of other TCP errors such as duplicate acknowledgement, connection retry, and connection refused. Slow network path signifies congestion on the network or the transmission port being used by the sending and the receiving hosts or server. It is easier to troubleshoot a slow workstation or server than a whole network, so before checking the network devices for bottlenecks, the end-hosts were checked first. Workstations could be sped up by reducing the number of concurrent services and programs running in them, thereby freeing CPU resources for a quick response when necessary. It is necessary to point out that the workstations in Granlund's network are equipped with Gigabit Ethernet NICs, so it is difficult for the interface to be overwhelmed by the network traffic. However, the network link between the hosts and the destination might be affected.

The illegal TCP checksum errors were not many, only 19, and most of them were from a particular server, so it was checked to see if it was performing hardware TCP offload checksum. Hardware could be used to automatically compute the checksum of network packet in the network adapter before transmission to the network or reception from the network for validation. This checksum offload can generally increase network performance by saving CPU cycles from calculating the checksum. This hardware checksum offload is often reported as illegal by most packet analysers. TCP port scans were carried out by domain controllers and servers, such as Exchange and Spiceworks servers, but some workstations were also reported to have done this. The TCP port scan is a way of checking which ports are open on a host in order to initiate a TCP three-way handshake. This should not be done by unauthorized workstations unless they are acting as servers for some software services. The workstations reported were thoroughly scanned for malicious software applications. TCP scanning often triggers alerts in monitoring and IDS devices.

In the network layer, the network error messages are centered on IP and ICMP. High error packet in this layer usually signifies wrong network configuration and poor network health but it fortunately had the least number of errors. The IP TTL (Time To Live) too low error was a false positive when checked, because it is different for different

operating systems. The affected systems were pinged and the TTL values returned were as expected. TTL is the value used to make sure a packet does not travel through the network in an endless loop, thereby creating unwanted traffic in the network. The TTL value of Windows, Linux, and Cisco operating systems are 128, 64, and 256 respectively. Capsa recorded errors because it was designed for Windows systems and treated all other TTL values as errors. An IP address conflict was reported as one of the errors but the report was false because Capsa reported Ethernet broadcast MAC addresses (FF:FF:FF:FF:FF:FF) as conflicting addresses. An Ethernet broadcast is used by ARP to translate IP addresses to MAC addresses so in a DHCP-enabled network. It is normal to get quite a few of them periodically.

ICMP errors account for the errors got when one host tries unsuccessfully to ping another host. In Granlund, due to the use of the firewall, not all devices can be pinged. Most of the errors reported were those unsuccessful ping trials targeted at the Bluecoat ProxySG. This was functioning as it should because only the workstations of the administrators were configured to access that network. Also, some servers did not accept ICMP requests due to the firewall setting. This practice helps to prevent ICMP attacks if ever they should occur, so it is necessary to put a certain server and proxy devices on a separate network beyond the reach of other standard hosts in the network. ICMP host unreachable error messages were also generated when the target host was maybe offline.

In the data link layer, the protocol in use is ARP. The error messages generated were false because the ARP requests logged were Ethernet broadcast messages which are not harmful to the network but necessary to successfully map MAC addresses to their respective IP addresses. To ensure that there was no illegal ARP scan request in the network, the source had to be verified, and to avoid storms in the network, storm-control was enabled on the switches. Storm-control is a command in Cisco IOS that could be used to control the amount of ARP request/received on a particular interface. The gateways of the network handle quite many ARP requests, so it is important to use network equipment with high capacity ports so as not to introduce unwanted lags in the network.

To summarise, Capsa was able to highlight and group the problems and bottlenecks discovered according to the various layers in which they occurred from top to bottom. As observed in the diagnosis section of Capsa, the application layer reported DNS,

SMTP, FTP, and HTTP network problems while the transport layer contained TCP-related problems. Furthermore, the network layer highlighted problems related to both IP and ICMP, while at the data link layer reported ARP-related issues.

5.2 Data Collected by Bluecoat and Analyses of the Results

In this section, the results analysed will not be separated into ProxySG and ProxyAV because both devices worked together to provide maximum web security and their logs were analysed by the reporter software. HTTP traffic was directed by the firewall to the Bluecoat devices and the ProxySG was configured to carry out ICAP scanning on the traffic it received using the ProxyAV engine. The inline scanning enables Bluecoat to provide adequate security against web threats even when they are hidden in non-suspicious Internet traffic. ProxySG operates by either blocking or allowing certain rules according to the network configuration via the VPM (Visual Policy Manager). These rules were grouped into different user-defined layers. Figure 14 shows an overview of the various layers configured in ProxySG.

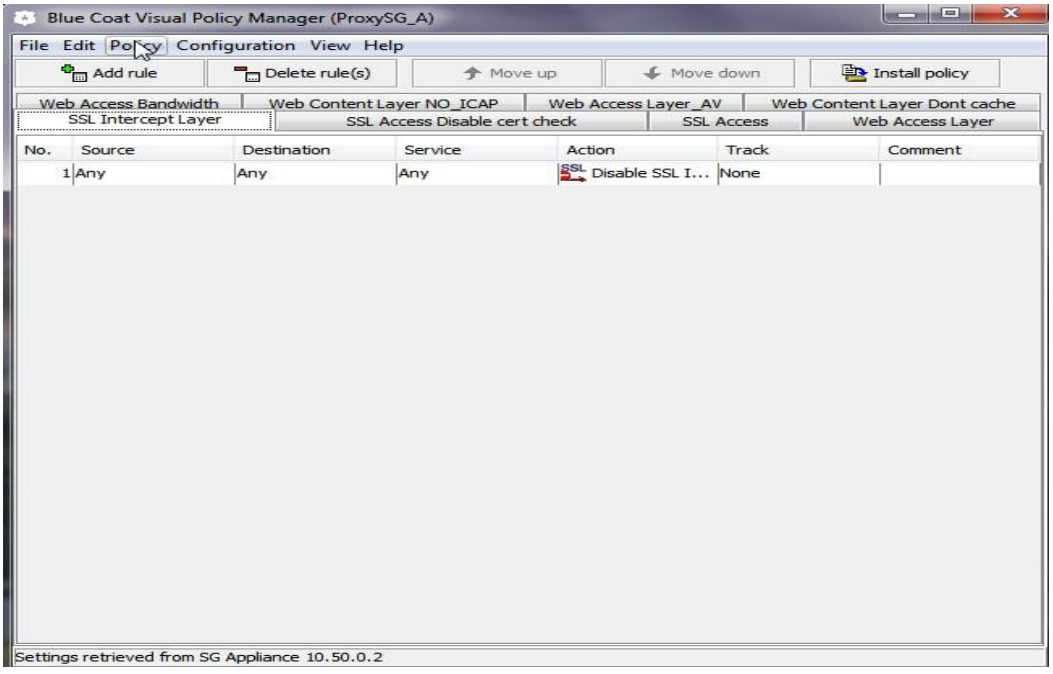


Figure 15: ProxySG's VPM GUI.

As shown in figure 14, there were eight defined layers but all controlling users' web access. The sections that are often edited are the destination and action section. The destination specifies where the traffic is heading, while the action specifies what verdict

to give to such traffic. The SSL rules are those that specify how to treat secure web connections while the “web content layer no-ICAP” rule fine tunes the “web access layer” rule in such a way that it allows some web requests from certain sites (such as Adobe and Microsoft) to bypass the ProxyAV in order not to scan their contents. This feature saves proxy resources because those requests are not harmful and do not require scanning. The “web access bandwidth” rule allows administrators to limit the bandwidth consumptions of certain sites (such as YouTube). These websites consume a lot of bandwidth and are often visited by network users The “web content layer don’t cache” instructs the proxy to not cache sites (such as news/media) that require frequent updates. The “web access layer” contained web categories that could be blocked or allowed and these categories were defined in the web filter. Bluecoat web filter categorizes billions of web pages into 85 categories in over 50 languages.

The webfilter is powered by over 75 million users in the WebPulse collaborative defense. Uncategorized, new, or unknown sites could be sent online to WebPulse for analysis and categorization. Also, wrongly categorized sites (especially local sites) could be sent for re-evaluation. The webfilter database was updated frequently and automatically. From the reporter, detailed statistics of the network usage could be seen. The reports show the bandwidth usage, most visited websites, browsing time of certain websites, web applications used, potential threats, and the protocols requested. The reports were grouped into large groups: user behavior, security, and bandwidth usage, and video usage reports among others. Some reports that directly relate to this thesis will be discussed here. The potential threats report shows the number of malicious threats that were intercepted by the proxies. The threats detected by the ProxyAV were fewer compared to those detected by the ProxySG, showing that harmful web attacks are more sophisticated and smaller in size when compared to a few years ago.

The ProxyAV detected threats in files that were downloaded from the Internet and were about 500 MB in size. However, larger files were not scanned because these files were downloaded from trusted sources. The ProxyAV interrupts downloads of large files and asks the user to contact the administrator. The administrator checks the legitimacy of the site before adding it to the “web content layer no_ICAP” group, so as to bypass the proxy during subsequent downloads. From the “trend of potential threats” report, it could be seen that the days with the highest number of web threats are Mondays and Fridays. Also, the number of threats prevented increased after blocking certain catego-

ries such as web advertisement, non-viewable content, and tv/video streams. These categories are the ones that are normally found when viewing some legitimate sites. They establish their own connections without the consent of users, thereby consuming network resources.

ProxyAV detected some malware and the list of those malware can be found in the “ProxyAV malware detected names” section of the reporter. Only three instances of malware activities was detected, and they are Trojans: HEUR:Trojan.Script.iframer and HEUR:Trojan.Script.Generic. Although these threats were prevented, they were logged and the affected client’s IP could be seen when assessing the log details. The two clients involved were scanned with SEP to make sure that the infection attempt was unsuccessful as reported. Web advertisements are a common way of infecting hosts connected to the Internet. In terms of user behavior reports, the bandwidth consumption was the main interest. Slow network connections can arise from inappropriate use of the network bandwidth. Over a period of two weeks, Youtube was the most bandwidth intensive web application used for video streams. Not all streams are unproductive because some are used for training purposes, which adds value to the company.

Slow network connections can arise from too many active video streams, hence the need to prioritize network protocol request. The speed of video streams was limited to 10 Mb/s, so as to avoid congestion and HTTP requests were given the most priority so as not to disrupt other activities in the network. In this era of social media, there were typically more requests to social sites than there were to streaming sites but in terms of the network resources consumed, the streaming sites were (as expected) more resource consuming (over 1500 times more than that required for social media requests). Network bandwidth loss was also limited by blocking access to web advertisements, non-viewable contents, large software downloads, and malicious sources.

5.3 Data Collected by Snort and Analyses of the Results

Snort is the IDPS setup during the project to monitor network packet for suspicious intrusive behavior. Snort was discovered to be full of false positives. After analysing the packet, the alerts were discovered to be false positives due to a few explainable reasons. The main reason was the position of the Snort sensor in the network. Snort sensors should be placed in the DMZ of the network so as to enable it to detect intrusion

attempts and bad network traffic. Another reason was that the traffic from the proxy (Bluecoat) and the virtual servers were not regular Windows traffic. Vendor-specific traffic triggered false alerts which Snort reported. Traffic between the proxies is usually encrypted just like those between servers. However, it is good that this traffic anomaly was noticed by Snort. In order to reduce this false alert, a network segment exception could be added to the Snort configuration but this is not advisable because some events might be missed. False alerts were also created when SNMP traffic was detected.

Examples of these alerts can be found below (bold markings are safe network addresses):

- Proxy traffic false positive:

[**] [120:1:1] (http_inspect) ANOMALOUS HTTP SERVER ON UNDEFINED HTTP PORT [**]
[Classification: Unknown Traffic] [Priority: 3]
12/05-15:00:20.880621 00:D0:83:07:1C:CD -> 00:8C:FA:05:79:44 type:0x800 len:0x172
10.50.0.4:61637 -> 10.50.0.3:1344 TCP TTL:64 TOS:0x0 ID:7359 IpLen:20 DgmLen:356
AP Seq: 0x700E1760 Ack: 0x1019F8F0 Win: 0xFFFF TcpLen: 32
TCP Options (3) => NOP NOP TS: 3143564216 79741442

- SNMP false positive:

[**] [1:1411:16] SNMP public access udp [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/05-15:00:27.993660 00:27:0C:98:B0:46 -> 00:26:73:0A:1B:1C type:0x800 len:0x79
172.20.0.45:1042 -> 172.20.3.244:161 UDP TTL:127 TOS:0x0 ID:265 IpLen:20 DgmLen:107 Len: 79
[Xref => http://cve.mitre.org/data/downloads/allitems.html2002-0013][Xref => http://cve.mitre.org/data/downloads/allitems.html2002-0012][Xref => http://cve.mitre.org/data/downloads/allitems.html1999-0517][Xref =>

http://www.securityfocus.com/bid/4089][Xref => http://www.securityfocus.com/bid/4088][Xref => http://www.securityfocus.com/bid/2112

SNMP traffic was generated by requests directed at the network printers and was not harmful.

6 Conclusion

In conclusion, the project was successful. The intended traffic monitoring process was carried out successfully, albeit it was more challenging than expected, due to the large volumes of traffic logs that were analysed, and the acquisition, setup and configuration of the tools used during the project. The goal of the project was to carry out Internet traffic monitoring using the network of Granlund Oy as a case study. The main task was to monitor the traffic logs generated to find anomalies, locate its source, and to verify the veracity of any alert by checking the affected device. In most cases, a general malware scan (using SEP) was enough to remove any malware found. In cases where slow connections were reported, the host was checked to see which process was using up its resources and then the network connection (host – switch) was checked to make sure that the speed and duplex of the connection was at its maximum.

During the project, the process of monitoring the Internet traffic of an enterprise network was analysed by first choosing the right tools to use from the pool of available tools and then proceeding to set them up and to configure them to suit the intended purpose. The most challenging part of the setup process was the Snort sensor because it involved tuning the software made originally for open-source operating systems to suit a network with primarily Microsoft Windows devices. It involved the installation of some prerequisite software, which would enable the desired software function properly. Also, checking each rule and adding new rules (where necessary) was found to be technically challenging but worth trying out. At the network packet analysis level, Capsa was effective in capturing most packets and analyzing them to find out where bottlenecks might exist in the network during transmissions. However, Capsa would be more effective when configured to monitor specific network segments as opposed to the whole network due to its ability to generate false alerts.

Future studies could be carried out on using Snort in the host-based monitoring mode and to check its monitoring efficiency when placed in various segments of the same network. Furthermore, more alternative tools such as Nessus could be tried out in future projects. Other possible uses of the Bluecoat devices mentioned in the thesis could also be studied.

References

1 Finamore A, Melia M, Meo M, Munafo M, Rossi D. Experiences of Traffic Monitoring with Tstat [Online] March/April 2011; 25(3):8-14.
URL: http://www.tlc-networks.polito.it/oldsite/mellia/papers/tstat-IEEENET.pdf. Accessed 10 September 2012.

2 Finnish Communications Regulatory Authority. Act on the Protection of Privacy in Electronic Communications [Online] 516/2004.
URL: http://www.finlex.fi/en/laki/kaannokset/2004/en20040516.pdf. Accessed 22 November 2012.

3 Wei D, Ansari N. IP Traffic Monitoring: An Overview and Future Considerations [Online]. Advanced Networking laboratory, New Jersey. 2004.
URL: http: //web.njit.edu/anl/papers/01PCM.pdf. Accessed 10 September 2012.

4 Cisco Systems Inc. A Beginner's Guide to Network Security [online]. 2001.
URL: http://www.cisco.com/warp/public/cc/so/neso/sqso/beggu_pl.pdf. Accessed 10 September 2012.

5 Ross Anderson. Security Engineering – A Guide to Building Dependable Systems [Online Book]. Wiley. 2001.
URL: http://www.cl.cam.ac.uk/~rja14/musicfiles/manuscripts/SEv1.pdf. Accessed 10 September 2012.

6 Guang Yang. Introduction to TCP/IP Network Attacks [Online]. Secure Systems Lab. November 1997.
URL: http://seclab.cs.sunysb.edu/sekar/papers/netattacks.pdf. Accessed 19 September 2012.

7 Scarfone K, Mell P. Guide to Intrusion and Prevention Systems (IDPS). National Institute of Standards and Technology. Special Publication 800-94. February 2007.
URL: http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf. Accessed 10 September 2012.

8 Blue Coat Systems Inc. Blue Coat Full Proxy Edition - ProxySG 300/600. 2011.
URL:http://www.Bluecoat.com/sites/default/files/documents/files/bcs_ds_fullproxy_300-600_v2h.pdf. Accessed 24 October 2012.

9 Blue Coat Systems Inc. Blue Coat Full Proxy Edition - ProxyAV 1200/1400/2400. 2011.
URL:http://www.Bluecoat.com/sites/default/files/documents/files/Blue_Coat_ProxyAV_1200-1400-2400.9%20%281%29.pdf. Accessed 24 October 2012.

10 Colasoft LLC. Capsa – Real-time Network Analyser (professional edition). 2012.
URL: http://www.colasoft.com/download/capsa-pro-usermanual.pdf. Accessed 7 November 2012.

11 Tony Howlett. Open Source Security Tools – A Practical Guide to Security Applications. New Jersey: Prentice Hall; 2005.

12 Alisha Cecil. A Summary of Network Traffic Monitoring and Analysis Techniques [online]. Department of Computer Science and Engineering, Washington University, St. Louis.
URL: http://www.cse.wustl.edu/~jain/cse567-06/ftp/net_monitoring.pdf. Accessed 10 September 2012.

13 Thomas M. Chen, Lucia Hu. Internet Performance Monitoring [online]. Lyle School of Engineering, Southern Methodist University, Dallas Texas. August 2002.
URL: http://www3.engr.smu.edu/~tchen/papers/ProcIEEE-Aug2002.pdf. Accessed 12 September 2012.

14 Cisco Systems Inc. Network Security [online]. Internet Society Conference, Japan 2000.
URL: http://www.potaroo.net/t4/pdf/security.pdf. Accessed 10 September 2012.

15 Yan Chen. Intrusion Detection/Prevention Systems [online]. Electrical Engineering and Computer Science Department, NorthWestern University, Evanston, Illinois. Modified December 2012.
URL: http://www.cs.northwestern.edu/~ychen/classes/msit458-f12/ids.ppt. Accessed 6 November 2012.

16 Miao Luo, Wei Jiang. Network Monitoring and Measurement and its Application in Security Field [online]. Department of Computer Science and Engineering, The Ohio State University, Columbus Ohio.
URL:http://www.cse.ohiostate.edu/~luom/788/Network%20monitoring%20and%20measurement.ppt. Acessed 12 September 2012.

17 Justin Weisz. Network Security [online slides]. School of Computer Science, Carnegie Mellon University, Pittsburgh Pennsylvania. Fall 2002.
URL: http://www.cs.cmu.edu/~srini/15-441/F02/lectures/lec21-security.ppt. Accessed 14 September 2012.

18 Kasey Efaw. Installing Snort 2.8.6.1 on Windows 7 [online]. Snort. 2010.
URL:http://www.Snort.org/assets/151/Installing_Snort_2.8.6.1_on_Windows_7.pdf. Accessed 5 September 2012.

19 Martin Roesch. Snort – Lightweight Intrusion Detection for Networks [online]. The USENIX Association. 1999.
URL: http://static.usenix.org/event/lisa99/full_papers/roesch/roesch.pdf. Accessed 4 October 2012.

20 Fabio Assolini. Massive DNS poisoning attacks in Brazil [online]. Kaspersky Lab. 2011.
URL:http://www.securelist.com/en/blog/208193214/Massive_DNS_poisoning_attacks_in_Brazil. Accessed 19 September 2012.