An Exploratory Framework for Extrusion Detection

Etienne Stalmans[1] and Barry Irwin[2]
Security and Networks Research Group
Department of Computer Science
Rhodes University
Grahamstown,South Africa
E-mail: g07s0924@campus.ru.ac.za[1]; b.irwin@ru.ac.za[2]

***Abstract*—Modern network architecture allows multiple connectivity options, increasing the number of possible attack vectors. With the number of internet enabled devices constantly increasing, along with employees using these devices to access internal corporate networks, the attack surface has become too large to monitor from a single end-point. Traditional security measures have focused on securing a small number of network end-points, by monitoring inbound connections and are thus blind to attack vectors such as mobile internet connections and removable devices. Once an attacker has gained access to a network they are able to operate undetected on the internal network and exfiltrate data without hindrance. This paper proposes a framework for extrusion detection, where internal network traffic and outbound connections are monitored to detect malicious activity. The proposed framework has a tiered architecture consisting of prevention, detection, reaction and reporting. Each tier of the framework feeds into the subsequent tier with reporting providing a feedback mechanism to improve each tier based on the outcome of previous incidents.**

***Index Terms*—Security, Extrusion Detection, Security model, BYOD**

I. Introduction

THE network security threat model has changed from the 1990s and early 2000s, when attackers primarily focused on gaining unauthorised access through server-side attacks. Modern network architecture provides many possible attack vectors, such as mobile internet connections, removable devices and social engineering attacks [1]. The growth of high speed networks and the constant introduction of new devices capable of network connectivity has lead to a situation where it is not possible to secure every device that connects to the network. Each new device connecting the network creates another avenue of attack. These new attack vectors allow attackers to bypass traditional network security systems such as Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). Current security systems such as IDS and IPS are focused on incoming traffic, allowing attackers free reign on the internal network once bypassed. Mikko Hypponen, the chief of research at F-Secure noted that all of the Fortune 500 companies had been breached by attackers [2], dispelling the myth that networks are secure. In a 2012 article, Richard Bejtlich was quoted as saying that the average cyberespionage attack goes on for 458 days [3]. Current detection and mitigation techniques are clearly failing. With the estimated cost of network crime exceeding $388 billion in 2011 [4], the cost of not defending the network far outweighs the cost of implementing new defensive technology.

A framework for detecting and defending against network attacks is proposed, focusing on monitoring the internal network as opposed to only monitoring the network end-points. The proposed framework consists of a tiered architecture. The first tier proposed by the framework consists of prevention through intelligent design. A well designed network assists in preventing the propagation of attacks through the network and increases the ease of monitoring the network. The increased monitor-ability of the network is incorporated into the second tier, detection, where multiple detection techniques are proposed to identify network incidents as early as possible. The third tier proposes multiple incidence response actions that can be taken. The final tier, reporting, outlines how reporting should be performed and a minimum set of information that should be included with each report. Reporting is structured to feedback into the three tiers above it in the framework, leading to constant improvement to the framework as the threat model evolves.

The paper discusses related work in Section II. Prevention through intelligent network design is discussed in Section III. Techniques for detecting malicious network activity are discussed in Section IV. Section V discusses reactive measures, which may be applied once suspicious traffic has been detected on the network. Reporting is discussed in Section VI, with concluding remarks in Section VII and future work is proposed in Section VIII.

II. Related Work

Previous research into extrusion detection is limited, with Bejtlich [1] providing the most comprehensive body of work on the subject. His work focused on preventing, detecting and mitigating security breaches of the internal network. The results of Bejtlich's work provides a baseline for the development of an extrusion detection framework.

Research focusing on outbound connection monitoring has largely revolved around the advance of data-loss prevention (DLP) systems. Despite this paper not being another DLP system, techniques employed in recent DLP research do have a bearing. The work by Borders, Lake and Arbor [5] used session data for the detection of information leakage. This

The authors would like to acknowledge the financial support of Telkom, Comverse, Stortech, Tellabs, Amatole Telecom Services, Bright Ideas 39, and THRIP through the Telkom Centre of Excellence in the Department of Computer Science at Rhodes University.

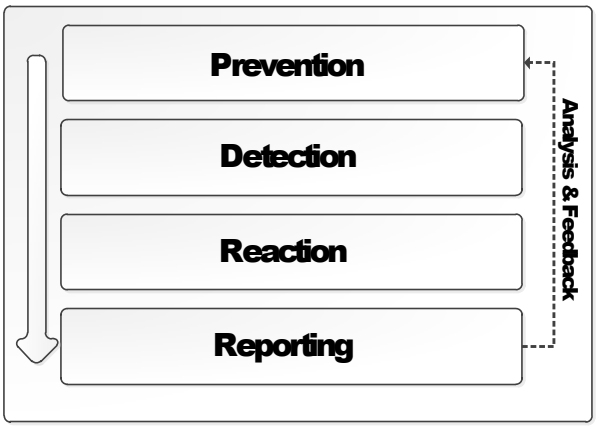

Figure 1. High Level View of Proposed Extrusion Detection Framework

work highlighted the benefits of session data analysis in environments where full content data is not always possible or viable due to encryption or obscure network protocols.

Malware detection has been a major area of extrusion based research, with researchers attempting to identify means of detecting malware infection at both the network edge and on the infected host. A host based solution was investigated by Xiong, Prateek, Deian, Wu and Yao [6] for the detection of outbound connections initiated by malware. Their research shows that host based network monitoring allows for the detection of malware that may be missed by standard anti-virus solutions. The malware referred to in their study was rootkit based and used alternative data communication channels, effectively allowing the malware to remain invisible to traditional detection technologies. This observation is useful for the development of an extrusion detection system as host-based detection systems may be used to detect intruders attempting to use alternate data channels to hide their activities and to ex-filtrate information.

Network based detection systems have largely been used in the detection of botnets and other malicious software. Reiter [7] propose a system for malware detection through network traffic aggregation. He observed that stealthy malware, attempting to avoid detection, was detectable through the aggregation of network flows to the same external network. The principle of traffic aggregation can be applied to extrusion detection. Intruder activity can be detected as they move through the internal network and different internal hosts report back to the intruder.

Wool described how direction based firewall filtering could be used to prevent the traffic of malicious activities originating on the internal network from leaving the network and causing further damage [8]. It was noted that the lack of standardisation between different firewall providers and the difficulty in configuring firewalls has lead to outbound traffic not being effectively firewalled, with administrators focusing on inbound configuration.

III. Prevention

A well designed network lends itself to minimising damage caused by an intruder, allowing for efficient traffic flow while preventing the spread of malware and the reach of intruders. Furthermore, well designed networks lend themselves to pervasive network awareness, ensuring administrators have a complete picture of what is occurring on the network [9]. Current prevention strategies rely on Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) and host based anti-virus solutions. These systems fail once an intruder has gained access to the network as they are designed to be outward facing as opposed to monitoring internal network activity. Administrators should be aware of services that require external connectivity and allow these through the system firewall, all other services should be blocked. This more aggressive approach to traffic filtering helps prevent attacks, such as zero-day exploits, from bypassing filters that look for known attack patterns.

Dividing the network into multiple subnets based on the functionality and connectivity required by hosts allows for better internal network monitoring. Monitoring traffic within each subnet along with attempted communication between subnets allow for the detection of network scanning activity by intruders and malware. A subnet should only be given access to other subnets that are critical for correct functionality of both subnets. This strategy prevents intruders from moving between subnets freely once a host has been compromised on a single subnet. Furthermore, the spreading of malware through network vulnerabilities are minimised as hosts on a subnet can only infect hosts on the same subnet. This limits the number of infected hosts from a single incident and assists in post incident handling and cleanup.

IV. Detection

Detection forms the second tier of the framework however, detection also forms a vital part of tier one, prevention. Multiple detection strategies are proposed allowing for detection of all network threats. These range from heavyweight, full content data inspection to lightweight, statistical and session data analysis. The output from detection feeds into the reaction tier where responses to network threats occur.

A. Full Content Data

Monitoring traffic at the network edge allows for access to full content data. Full content data refers to the header and application level information contained in packets. Analysis of full content data requires applications capable of level 7 (application layer) analysis, such as firewalls and proxy systems. Full content data analysis is currently used in systems monitoring incoming traffic such as email spam filtering and intrusion prevention systems. By employing systems monitoring outgoing traffic, data leakage and detection of other malicious activity is possible [5] .

E-mail analysis is the first area that comes to mind when discussing full content data and has been covered extensively by existing DLP solutions. By examining outgoing e-mail text for specific keywords such as *private*, *confidential* and *annual report*, the leakage of sensitive information by staff may be detected. Full content data analysis also includes e-mail attachments. E-mail attachments can be examined to determine if they contain sensitive files. File names of e-mail attachments,

along with file meta-data may be compared to a blacklist of sensitive internal documents. When an e-mail containing a sensitive attachment is detected, it may be held for manual verification before being allowed to egress from the network. Creating an MD-5 or SHA-1 hash of documents before they are distributed to employees can aid in data leakage detection and prevention. During full content data analysis, all files leaving the network can be reassembled and hashed. The hashes of these files are then compared with the hash list of sensitive internal documents, allowing for accurate identification of data leakage, even when employees have changed file names and extensions to avoid e-mail filters. The efficiency of this system can be enhanced by using a hashing algorithm such as Tiger-Tree-Hashing (TTH), where only part of the file needs to be reconstructed and hashed to accurately detect sensitive data. The same principles applied to email attachment filtering discussed above may be applied to other areas of file transfer including FTP and instant messaging.

B. Session Data

Full content data analysis is blind to application layer data that has been encrypted. Session data analysis is immune to obfuscation of data through encryption, as it does not rely on the contents of the data being transmitted. Session data represents a summary of data communication between two systems. The basic elements recorded by session data include;

- Source IP address
- Source Port
- Destination IP address
- Destination Port
- Protocol
- Timestamp
- Amount of data exchanged

Session data may be used in multiple ways to detect malicious activity. Examining the source and destination addresses may provide indicators of malicious activity. The destination address is already used in security solutions such as blacklists and spam filtering. Updated blacklists are provided by numerous companies, unfortunately these blacklists are not always effective as malicious traffic may still be allowed to egress from the network if the destination address does not yet appear in a blacklist. Source address filtering allows this limitation to be overcome. Network administrators should be aware of all host addresses internal to the network and record which hosts require outbound connections as well as common external addresses accessed. All hosts that do not require external connections should be added to a block list, preventing any traffic originating from these hosts from leaving the network. Logging of connection attempts by hosts on the block list provides administrators with a good indication of which hosts may contain malicious or unconfigured software. Furthermore, when the system detects an IP address not internal to the network attempting to establish a connection, a spoofing attack [10], [11] may be in progress. A non-standard IP address may also indicate a host that has been manually reconfigured by an attacker or a user attempting to bypass network filters.

Systems trying to directly communicate with external addresses by bypassing Domain Name System (DNS) may indicate malware activity. The average user and application use DNS to resolve addresses. While malware usually comes with addresses hard-coded [12], [13], [14].

C. Statistical Data

Statistical analysis of network traffic allows for standard network behaviour to be observed and recorded. Once a baseline for network behaviour has been established traffic that falls outside of the baseline may be deemed as suspicious. Levels of severity may be created where traffic can be classified according to the amount of deviation from the norm. Depending on the level of severity different actions may be performed. These actions, as discussed in Section V, may consist of simply throttling the network speed until manual inspection can be performed. At the most severe level, network traffic is completely blocked until manual inspection can clear legitimate traffic and rules added to allow this deviation from the baseline.

Statistical analysis may be performed on multiple facets of network traffic. Time of day statistics allow for easy to implement and adjust rules, while providing a higher degree of protection from malicious network activity. Network traffic occurring during standard work hours will be allowed as normal, while network activity outside of work hours may indicate illicit network usage either by malware or an attacker. Statistics can be adopted on a fine grain level, where network users are allowed access to the network based on their work patterns.

Detailed traffic analysis can lead to statistic based rules pertaining to the structure of network packets. As discussed by Bykova and Ostermann [15], IP and TCP headers of packets can be analysed to identify violations of existing network standards. Packets may also be classified as belonging to the network or not based on these IP and TCP headers. Header analysis allows for a lightweight detection model as opposed to full content analysis, which is more resource intensive and may affect network performance on high activity networks.

D. Alert Data

Alert data refers to traffic and data generated by specialised detection engines. These alerts are created when observed traffic matches signatures of known malicious network traffic. Alert data is primarily used in Intrusion Detection Systems (IDS) such as Snort . Data created by these detection systems should be used to actively defend the network, and not just as a reporting engine for post incident analysis. Numerous off-the-shelf security products provide adequate detection of known attack patterns and malware signatures. This allows for the rapid deployment of security solutions, ensuring a degree of protection while more comprehensive security mechanisms are put in place.

E. Network Based

Through intelligent network design, as discussed in Section III, effective network based detection of threats is achievable. Network based sensors may be deployed within each subnet or at a minimum between each subnet. These sensors provide a holistic view of activity on the network. One of the first

actions performed by an attacker once they gain access to a network is to map all hosts on the network [1]. Existing Intrusion Detection Systems (IDS) are ideal as network based sensors. These systems can be re-purposed and moved to the internal network and not be confined to the network entry points as is traditional. IDS signatures that detect network scanning activity may be adjusted to detect scanning of the internal network, producing alert data. It has also been shown that IDS sensors can be used to detect network malware activity based on malware signatures [16].

F. Host Based

Host based monitoring is proposed to detect internal security threats that may not be detected by intrusion detection systems and network based sensors. Once an attacker has gained access to an internal network, they might perform reconnaissance to determine which systems are available on the network and to map out an attack pattern [1]. Furthermore, many network worms will attempt to scan the internal network to determine if other hosts on the network are vulnerable to infection [17]. Host based detection systems allow for the detection of this scanning behaviour. Host based systems may be configured to detect abnormal Internet Control Message Protocol (ICMP) traffic, indicative of network scanning. Under normal conditions, a host would not send an ICMP ping request to every host on the network. This behaviour is flagged as suspicious and reported to the system administrator. The host based system may further be configured to block all ICMP traffic after a set number of consecutive network pings. This would limit the number of hosts a scanning network worm or attacker could detect.

The internal network could be configured to contain a so called honeypot subnet or honeynet [18]. These subnets would function in a manner similar to network telescopes. The honeynet address would not be broadcast on the network and any connection attempts to this subnet would indicate possible malicious traffic. Host based systems could be configured to report all attempts to contact the honeynet, along with the application or user attempting to initiate the connection. This information would be useful in attack analysis and identifying malicious software that installed on the host machine.

V. Reaction

The response to incidents on the network falls into two categories, these being automatic and manual. Automatic reaction relies on predefined rules, which trigger specific responses based on the type and severity of incident taking place. Automated reaction systems provide the capability to rapidly respond to threats, before escalation can occur. On the downside, automated reaction systems fail when previously unseen network incidents occur. Furthermore, an attacker that is aware of automated reaction systems, may structure their activities to fall outside the trigger points usually associated with automated response systems. To counter these attackers manual reaction is required. Manual inspection of network activity logs and alert data can identify network activity that has not been dealt with by automated systems, thus forming a critical part of the reaction phase.

Four strategies for dealing with unwanted network traffic are outlined by the framework. Depending on the severity of the incident and the type of incident occurring, a different reactionary tactic may occur. The totalitarian and most effective strategy is to block all traffic, this option has the downside of disrupting legitimate network activity, introducing network downtime while incidents are investigated and dealt with. Throttling of network traffic may allow for investigation into network incidents, without network downtime. Proxying and modifying network traffic allow for constant *reaction*, where undetected incidents may be slowed or prevented. The four reactionary tactics are discussed in more detail in the following subsections.

A. Blocking Network Traffic

Network traffic may be blocked, preventing it from leaving the network. The majority of security threats may be blocked by only allowing essential network traffic to egress from the network. Administrators should be aware of the services present on the network along with which of these services require outbound connectivity. Once a list of essential services has been defined, the network can be configured to allow these services outbound connectivity, while everything else is blocked.

Traditionally blocking has been focused on layer 3 (protocol) and layer 4 (port) blocking. The application level has been used in more recent blocking strategies, or so called level 7 firewalls and intrusion prevention systems. Application level analysis allows traffic tunnelled through allowed services, such as HTTP, to be examined and blocked as necessary. In the case of malware using HTTP as a communication channel, layer 7 analysis can be used to detect and block this behaviour. Layer 7 analysis may be further enhanced through the use of deep packet inspection (DPI). In DPI the payload of each packet or packet stream is analysed and is used to determine whether a packet should be blocked or not. A disadvantage of DPI is the extra processing overhead incurred. Strict blocking rules will mitigate the risk posed by malware and zero-day attacks, which bypass traditional detection systems, but are not expecting egress filtering.

B. Proxying Network Traffic

The proxy system is placed between the client and server and all traffic between the two systems must pass through the proxy. This position allows the proxy to be used to monitor and log all outbound traffic. Furthermore, the proxy may be used to block traffic depending on the traffic content. Traffic filtering can be applied through the use of blacklists and traffic type filtering. A more effective and harder to bypass solution sees the use of whitelists along with blacklists. Whitelists consist of allowed applications that should be allowed to traverse the proxy, while all other traffic should be blocked. This solution allows for fewer rules that need to be administered as well as providing cover for zero-day exploits. Zero-day exploits rely on previously unknown vulnerabilities and allow attackers to bypass current security solutions. By blocking all traffic other than legitimate traffic, previously unseen attacks are blocked as they are unlikely to be in the whitelist of allowed connections.

Proxy authentication is traditionally used for user access management, but offers a useful means for blocking malware. When a host has been infected with malware, the malware attempts to communicate back to the attacker [6], [19]. If the malware has not been configured to expect a proxy or to authenticate with a proxy, it will not be able to communicate back to the attacker. Even when malware has been configured to expect a proxy, it would need to depend on the user to authenticated with the proxy, further frustrating the attacker.

The logging capabilities offered by proxy systems provide valuable information when trying to determine if a breach has occurred. Log analysis may also be used in incident response to determine how an attacker was able to gain access to the system. As well as indicating what information the attacker was able to extract from the system. Proxying is a critical component of manual reaction and incident investigation and reporting. Logging of network traffic allows for analysis of the full duration of incidents.

C. Throttling Network Traffic

Traffic throttling has already been employed in numerous networks in the form of guaranteed Quality of Service (QoS). This strategy is usually seen as a mechanism of punishing high bandwidth consumers who are degrading the service for low bandwidth consumers [20]. QoS rules may be adjusted to deal with network threats, retarding the spread of malware and the leaking of sensitive company data. These rules are implemented dynamically based on data received from the detection tier, where suspicious network traffic is throttled based on automatic rules, allowing time for manual inspection to occur. This strategy offers the benefit of increasing the time allowed to respond an incident, while minimising damage caused by an incident. Furthermore, as traffic throttling is not as aggressive as traffic blocking, the effect of false positives are reduced as legitimate users may continue using the network, though at a greater inconvenience, usually not noticeable to the average, low bandwidth user.

D. Modifying Network Traffic

When network traffic passes through a proxy system it may be modified according to predetermined rules. Modifying network traffic ensures that it appears to the end user as if communication is occurring as normal, while malicious content is silently filtered out. Traffic modification can occur at different levels of the IP stack, examples include modifying destination address, destination port and modifying the data contents of a packet.

Traffic that has been detected as being possibly malicious may be redirected to areas of the network known as sink-holes. These sink-holes receive connections as normal but all traffic is dropped as it enters the sink-hole. This technique may fool malicious users and applications into believing that their communication is proceeding as requested to the designated end-point. However, from a network perspective, these connections are ending at the sink-hole, preventing the exfiltration of data and the creation of backdoors by attackers on the network. Sink-holes may be modified to consist of honeypot machines, a honeypot provides an operating environment for attackers to interact with. By allowing attackers to interact with a system that is controlled and monitored, it is possible to learn how attackers interact with the system and what systems the attackers are interacting with [18]. This solution allows administrators to monitor the activity of attackers and malware on the network and gain valuable insight into what has been compromised on the network, as well as giving an indication of what an attackers end goal might be.

VI. Reporting

Effective reporting constitutes the most important step in incident handling [1]. Reporting needs to encompass all tiers, from prevention through to incident response. Through effective reporting, future incidents can be prevented as system weaknesses are identified and appropriate responses are formulated. The results obtained from reporting should be used as a feedback mechanism into the framework, and used to improve each tier of the framework. Incident reporting needs to be completed as soon as possible after an incident occurring, with the use of both automated reporting tools and detailed manually compiled reports.

Each report should contain a basic set of information that is standardised for each incident. This information includes;

- Type of incident: The incident needs to be categorised according to type. Types of incidents include insider attack, malware, external intrusion and phishing attacks. The type of attacks that can occur should be clearly defined and understood by all parties involved.
- Detection technique: How the incident was detected, this will describe which detection techniques detected the threat and the signature that triggered the detection engine. This section of the report should include automated reports from the relevant detection engines.
- What was affected: It is essential that all infrastructure affected by the incident is recorded. This will help in determining which services may still be at risk as well as helping in identifying how the compromise occurred.
- Cost estimate: Both the cost in terms of monetary value and time lost should be calculated for each incident. Cost estimates help in justifying greater expenditure on defensive technology and helps raise awareness of security issues affecting the network.
- Response: The manner of incidence response should be documented, this will assist in determining how future incidents should be responded to. Furthermore, analysing the effectiveness of incident response helps identify weaknesses in the system and assists in improving the overall defensive structure of the network.

Reporting should be performed on a consistent basis, even when no incident has occurred. Continual monitoring of the network and reporting on the network states helps in maintaining an understanding of which areas of the network are vulnerable. Furthermore, continual reporting helps identify what attacks are taking place against the network, all attacks against the network should be recorded to provide a better understanding who is attacking the network and what techniques are being used. Knowledge of what attacks are taking

place provides a baseline from which abnormal activity can be detected. If the number of detected and blocked attacks diminish, it could be an indication that the network has been breached and attackers are present within the network.

The occurrence of unique or previously unseen attacks against the network should be reported to external parties. This step helps increase the overall understanding of the modern threat model and assists in the creation of new techniques for combating emerging threats.

VII. Conclusion

The modern threat landscape has changed and network defence needs to be adjusted accordingly to meet new threats and challenges head on. Traditional network security focused on monitoring the network entry points and attempting to prevent intruders from breaching the network defences. This strategy has failed as the number of network entry points have increased and the monitoring of all these entry points have become infeasible. A framework is proposed for implementing network security, with a core focus of monitoring the internal network and traffic leaving the network. This framework does not propose that traditional security measures should be replaced, but rather that they should be augmented with new defensive techniques. Prevention through intelligent network design is proposed as the first line of defence, where attackers are limited in what they are able to exploit if a single point in the network is breached. Through monitoring internal network communication, network administrators are able to gain a clear picture of what is occurring, and thus detect any malicious activity that has been missed by traditional intrusion detection systems. Five detection strategies are proposed, allowing for greater redundancy and improved detection. Data generated by network monitoring feeds into the reaction tier, where defenders are able to both automatically and manually respond to network intrusions. Four responses to a network intrusion are proposed depending on the severity and type of alert generated during the detection phase. Finally efficient reporting of all incidents and network activity is used to gain a greater understanding of existing and future network threats. Reporting is used as a feedback mechanism to improve all tiers of the framework, as more knowledge about the network and threats are gained.

VIII. Future Work

Implementation of the framework on a small test network will be used to identify the framework's feasibility along with any shortcomings present in the framework. Once the framework has been tested and modified, it will be implemented on a large scale network to evaluate the effectiveness of the proposed tiered architecture.

References

[1] R. Bejtlich, *Extrusion Detection: security monitoring for internal intrusions*. Addison-Wesley, 2005.

[2] Milken Institute. (2012) Cybersecurity, when hackers attack. [Online]. Available: http://www.milkeninstitute.org/newsroom/newsroom.taf?function=currencyOfIdeas&blogID=462

[3] Wired Threat Level. (2012) Everyone has been hacked. now what? [Online]. Available: http://www.wired.com/threatlevel/2012/05/everyone-hacked/all/1

[4] Symantec. (2012) Norton cybercrime report. [Online]. Available: https://www.symantec.com/content/en/us/home_homeoffice/html/ncr/

[5] K. Borders, W. Lake, and A. Arbor, "Towards quantification of network-based information leaks via http atul prakash university of michigan."

[6] H. Xiong, P. Malhotra, D. Stefan, C. Wu, and D. Yao, "User-assisted host-based detection of outbound malware traffic," *University Computing*.

[7] M. K. Reiter, "Traffic aggregation for malware detection," *Work*, 2007.

[8] A. Wool, "The use and usability of direction-based filtering in firewalls," *Computers Security*, vol. 23, no. 6, pp. 459–468, 2004. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167404804000951

[9] Microsoft Corperation, "Microsoft security intelligence report," White paper, May 2012. [Online]. Available: http://download.microsoft.com/download/C/9/A/C9A544AD-4150-43D3-80F7-4F1641EF910A/Microsoft_Security_Intelligence_Report_-Volume_12_English.pdf

[10] F. S. K. Silva and P. Barber, "Anatomy of recent DNS reflector attacks from the victim and reflector point of view," *VeriSign*, pp. 1–16, 2006.

[11] C. C. Attribution-sharealike, "DNS amplification attacks preliminary release randal vaughn and gadi evron," *Analysis*, 2006.

[12] A. V. Barsamian, "Network characterization for botnet detection using statistical-behaviour methods," Ph.D. dissertation, 2009.

[13] H. Binsalleeh and A. Youssef, *An implementation for a worm detection and mitigation system*. IEEE, Jun. 2008.

[14] P. Čeleda, J. Vykopal, T. Plesník, M. Trunečka, and V. Krmíček, "Malware detection from the network perspective using netflow data," *Perspective*, 2010.

[15] M. Bykova and S. Ostermann, "Statistical analysis of malformed packets and their origins in the modern internet." in *Internet Measurement Workshop*. ACM, 2002, pp. 83–88.

[16] G. Gu, P. Porras, V. Yegneswaran, M. Fong, W. Lee, and M. Park, "Bothunter: Detecting malware infection through ids-driven dialog correlation," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*. USENIX Association, 2007, pp. 12:1 – 12:16.

[17] D. Whyte and E. Kranakis, "DNS-based detection of scanning worms in an enterprise network," *Network*, 2004.

[18] The Honeynet Project. (2012) The honeynet project. [Online]. Available: http://www.honeynet.org/

[19] J. A. Morales, "Analyzing and exploiting network behaviors of malware," in *SecureComm'10*, 2010, pp. 1–17.

[20] G. Goth, "Isp traffic management: Will innovation or regulation ensure fairness?" *IEEE Distributed Systems Online*, vol. 9, no. 9, 2008.

Mr Etienne Stalmans completed his Honours in Computer Science in 2010 under the guidance of Dr Karen Bradshaw in the Department of Computer Science at Rhodes University. Etienne is currently working towards his MSc in Computer Science.

Dr Barry Irwin is a senior researcher and lecturer in the Department of Computer Science at Rhodes University. Dr Irwin is in charge of the Security and Networks Research Group.