Intrusion Detection and Security Assessment in a University Network

Darren Fitzpatrick

A Dissertation Submitted to the School of Computing

Faculty of Engineering and Computing

Dublin City University

For The Degree of Master of Science

Supervisor: Dr. Darragh O'Brien

December 2009

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Science is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

ID No.: _____

Date: _____

Acknowledgements

I would like to express my gratitude to Dr. Darragh O'Brien for his support, patience and knowledge sharing through the course of these past two years. The level of my MSc degree can be attributed largely to his encouragement and immense effort, without which, completion of this thesis would not have been possible.

I am indebted to my colleague Hai Ying Luan who worked closely with me throughout the project and could not have been more helpful, enthusiastic and hard working. His drive and determination to reach such a high level of education in a non-native tongue impresses me immensely.

I thank Enterprise Ireland for funding the development of our prototype Intrusion Detection system and DCU for providing the environment in which to take this on.

Finally I would like to thank my family whose tremendous support over the years has been an underlying force in reaching my goals.

Abstract

This thesis first explores how intrusion detection (ID) techniques can be used to provide an extra security layer for today's typically security-unaware Internet user. A review of the ever-growing network security threat is presented along with an analysis of the suitability of existing ID systems (IDS) for protecting users of varying security expertise. In light of the impracticality of many IDS for today's users, a web-enabled, agent-based, hybrid IDS is proposed. The motivations for the system are presented along with details of its design and implementation. As a test case, the system is deployed on the DCU network and results analysed. One of the aims of an IDS is to uncover security-related issues in its host network. The issues revealed by our IDS demonstrate that a full DCU network security assessment is warranted. This thesis describes how such an assessment should be carried out and presents corresponding results. A set of security-enhancing recommendations for the DCU network are presented.

Table of Contents

List of Figures

List of Tables

1. Introduction

Today 1.7 billion people are using the Internet [1]. The Internet is now engrained in modern society and culture with 77% of Americans using the Internet and 72% online daily [2]. In Ireland, as of 2009, 63% of households have Internet access [3]. In addition to its expanding cultural role, the Internet has also become a key business-enabler. In Ireland, 92% of Small to Medium Enterprises (SMEs) have an Internet connection [3] and nearly two thirds of enterprises in the EU have a website [4]. In 2008, e-commerce sales were worth 106 billion euros to the EU economy [5]. This figure is projected to reach 323 billion euros by 2011. The European Future Internet ultimately foresees more than 4 billion Internet users [6].

Criminals too have noticed this explosion in Internet use as they increasingly target Internet users for financial gain through malware and social engineering attacks. In response, a range of security products has been developed by industry to keep Internet users safe. Yet Internet crime continues to thrive and according to Kaspersky Labs there was an 800% increase in the number of new malicious programs between 2001 and 2006 [7]. Computer crime often results in financial loss. 49% of Irish companies reported theft of sensitive IT assets and 30% reported denial of service (DoS) attacks in 2007 [8]. Cybercrime costs vary, however Irish organisations have been the victims of single incidents costing over 250K euro to repair and 14% of organisations are spending over ten working weeks responding to single issues [8].

The remainder of this chapter is structured as follows. In section 1.1 we consider the factors that contribute to the Internet security problem and make it difficult to solve. In section 1.2 we present the research questions this thesis seeks to answer. The chapter concludes with an outline in section 1.3 of the structure of the remainder of this thesis.

1.1 The Internet Security Problem

The three pillars of information security are confidentiality, integrity and availability [9] and their maintenance in a computer network is the realm of network security. Ensuring network security requires a layered approach, vigilance and regular updating in order to protect against the latest threats. Threats target vulnerabilities and vulnerabilities have been appearing in increasing numbers as illustrated by Figure 1.1 produced from numbers made available by the Computer Emergency Response Team (CERT) [10].

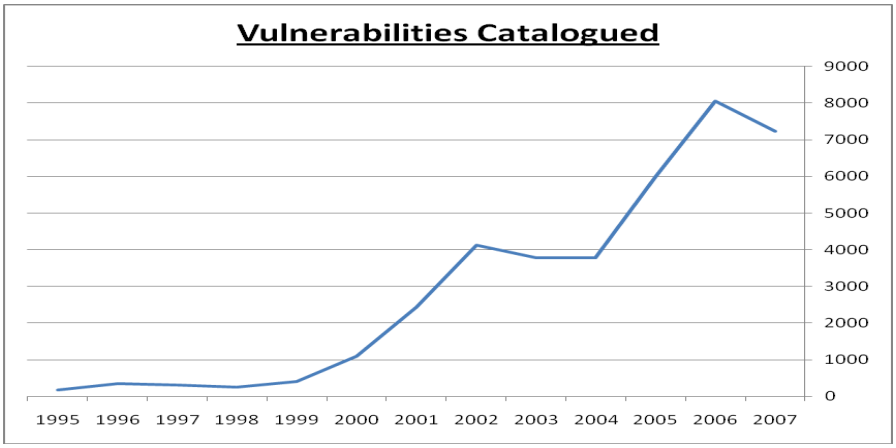

Figure 1.1 (from [10]) – Vulnerabilities catalogued yearly by CERT

The widespread presence of vulnerabilities and the availability to hackers of tools such as nmap [11] and frameworks such as Metasploit [12] that automate the hacking process, all combine to make attacks relatively simple to mount. Phishing toolkits, distributed denial of service (DDoS) kits, virus writing guides, trojans and botnet management consoles are all readily available. Figure 1.2 (from [13]) illustrates this rising attack sophistication vs. diminishing skill trend.

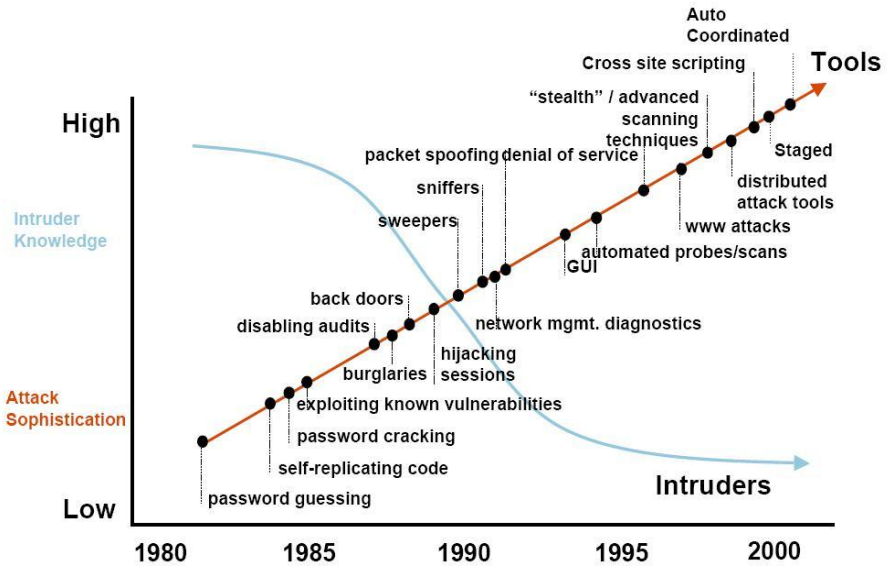

Figure 1.2 (from [13]) - Attack Sophistication Vs. Intruder Knowledge

After a successful attack, sophisticated malware may be installed to retain a permanent presence on an exploited host and it too can be freely downloaded [14]. The popularity of the Internet means that malware, when released, spreads rapidly. The SQL Slammer worm, whose global coverage after half an hour is depicted in Figure 1.3 (from [15]), at which point 74,855 hosts were infected, was the fastest spreading computer worm in history. Although the worm was released in 2003, the underlying vulnerability (a buffer

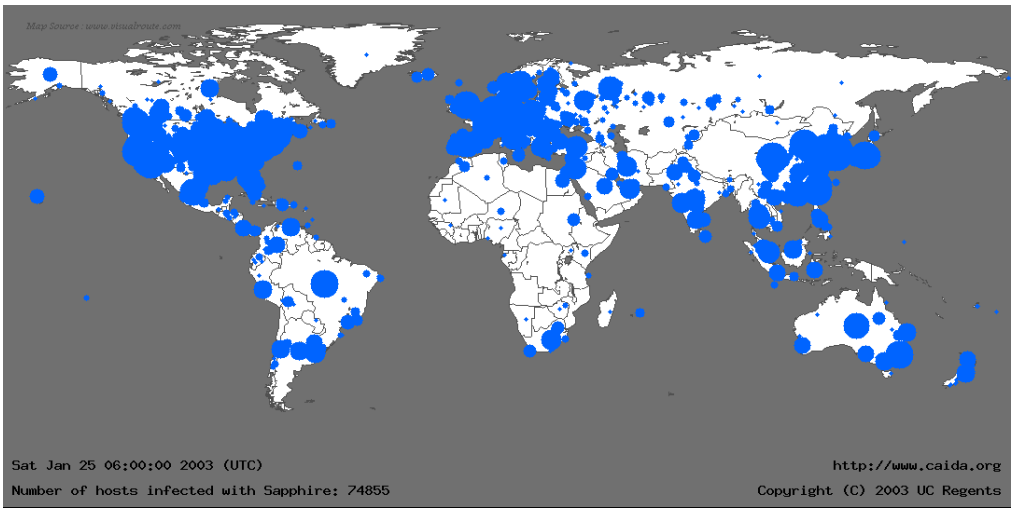

Figure 1.3 (from [15]) – SQL Slammer Worm Global Coverage after Half an Hour

overflow) had been discovered in 2002. However, a large proportion of Microsoft SQL Server administrators failed to apply the released patch.

Also contributing to the Internet security problem is how users are connecting to it. Recent years have seen a surge in the availability of wireless-network-ready devices and wireless networks. Smart phones, PDAs and laptops come WiFi-ready as standard. Wireless networks represent a relatively cheap and simple means for businesses to attract and meet the needs of Internet-hungry customers with the result that they are increasingly common in cafés, hotels, airports etc. Most users of such networks are unaware of the implicit trust they place in its administrators and in their fellow users. The administrators of such networks are often not security conscious. A recent Irish study [16] discovered, during a war-driving exercise, that as seen in Figure 1.4 (from [16]), of 3143 access points encountered in Dublin city, 60% utilised WEP (Wired Equivalent Privacy), 14% applied WPA (WiFi Protected Access) while 25% were open and did not encrypt traffic. WEP is notoriously insecure and can be broken in a matter of minutes using online utilities such as Aircrack [17].

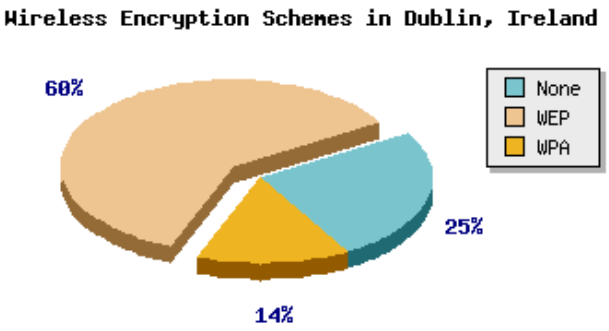

Figure 1.4 (from [16]) – WEP Prevalence

The profile of the typical Internet user is also changing. A growing number of "security unconscious" users are attracted by the World Wide Web (WWW) and a new generation of Web 2.0 applications promises to attract even more. The social networking site Facebook has over 300 million active users (50% of whom log in on any given day [18]). Such users install Trojan software, fall for phishing scams and fail to heed browser warnings [19]. In 2005 Australian researchers found that only one in seven computers in Australia used a firewall and only about one in three used up-to-date

antivirus software [20]. In the same report, increasing the security awareness of such Internet users was identified as a key objective in fighting the spread of malware. For the purpose of making users aware of the Internet threat and demonstrating to them that they are targets, an Intrusion Detection System (IDS) is ideal. However, given the challenges posed to typical users in running relatively simple software such as firewalls and antivirus packages, installing, configuring, maintaining and interpreting the output of a full IDS such as Snort [21] is infeasible.

1.2 Research Question

Intrusion Detection, as part of an overall layered security architecture has the potential to greatly decrease the Internet threat for both home users and organisations alike. IDS can make users aware of the threats they face and bring to light insecurities in an organisation's network. However, often the security expertise required to install, configure and maintain an effective IDS (as well as decipher and react to attacks) is unavailable.

The research question this thesis sets out to answer is whether effective IDS can be offered over the Internet as a web service. Can we develop a system to deliver to a new generation of non-security aware users the extra layer of protection an IDS provides? The system should be administered over the Web by a skilled third party thus solving the complexity issue. Given the mobility of modern users, monitoring and feedback should also be provided where that feedback arrives via a medium with which the vast majority of users are familiar, their web browser.

This thesis describes the motivations for and design and implementation of an IDS aimed at answering the above question. As a test case the system is deployed on the DCU network and results are analysed. One of the goals of an IDS is to identify problems in its host network in order that they can be rectified. The results obtained by deploying our IDS on the DCU network, and the security issues they reveal, motivate

the second question this thesis seeks to answer: how secure is the DCU network? The thesis proceeds to describe the results of a network security assessment of the DCU network. Some recommendations for improving overall network security are made.

1.3 Thesis Stucture

Chapter 2 presents the Internet security threat. To understand the threat, some context is required and the chapter includes a review of network models and protocols. An attack taxonomy is described and applied to a real world security incident. Evidence, made available by honeypots, of the prevalence of attacks on the Internet is presented in order to illustrate the hostility of the environment in which the modern Internet user operates.

Chapter 3 reviews the defences available to fend off Internet attacks. Firewalls and IDS are covered. This chapter also includes a description of our web-enabled, hybrid IDS designed to meet the modern Internet user's needs. Design and implementation issues are covered, as are results obtained through deploying the system on the DCU network.

Issues identified through IDS deployment warrant a full security assessment of the DCU network. The procedures by which such an assessment is typically carried out are presented in Chapter 4.

Chapter 5 presents the results of our network security assessment.

Chapter 6 concludes this thesis. It reviews the thesis's contribution to IDS, makes suggestions for future work and presents a set of security-enhancing recommendations for the DCU network.

2. The Network Security Threat

As we have seen in Chapter 1, there is a clear need for protecting network connected devices against attack. The aim of this chapter is to present and explain a selection of the threats with which network and Internet users are faced today.

This chapter is structured as follows. To understand network threats, some context is required and in sections 2.1 and 2.2, an overview of Internet standards including the TCP/IP model [22] and some of the protocols that implement it are presented. In section 2.3 we discuss an example network attack taxonomy which defines what constitutes an attack and provides a standard means for the classification and documentation of individual incidents. We apply the taxonomy to analyse an example real world attack. Finally, in section 2.4, we provide some information regarding the current network threat climate to highlight the frequency of attacks.

2.1 TCP/IP Model

The TCP/IP model is a four layer descriptive framework for network protocols. It emerged from work carried out by the United States Department of Defence (DoD) military technology department, the Defence Advanced Research Projects Agency (DARPA) [23] as part of the ARPANET project [24]. The layers of the model need not be strictly adhered to when developing network protocols, rather they provide a general guideline. TCP/IP model layers are presented below.

2.1.1 Application Layer

This topmost layer or application provides protocols used by applications to communicate over a network. The user interacts with an application which in turn passes data to the operating system to be encapsulated by the layer below (transport

layer). The application developer works with application layer protocols and uses operating system routines to pass application layer data to the transport layer.

Examples of application layer protocols include DNS (Domain Name System) [25], FTP (File Transfer Protocol) [26], HTTP (Hypertext Transfer Protocol) [27], RPC (Remote Procedure Call) [28], and SNMP (Simple Network Management Protocol) [29]. DNS is an Internet-based naming service which converts between binary, machine-readable IP addresses and textual, human-readable domain names. DNS servers around the world co-operate within a distributed, hierarchical structure to support the Internet naming service. FTP is an efficient and reliable means of transferring files between clients across a network. HTTP is used to request and transfer hypertext documents on the World Wide Web over the Internet. RPC allows a client to call routines located on an RPC server across a network connection which then execute at that remote location. SNMP is used to monitor and configure networked devices across a TCP/IP connection.

2.1.2 Transport Layer

Application data is subsequently encapsulated within a transport layer data unit, and a transport layer header added. Transport layer technologies may be connection-oriented (e.g. TCP) or connectionless (e.g. UDP) and add session management to the communication. The transport layer header contains information related to status of the connection. Port numbers provide application addressing and segmentation, flow control, congestion control and error control are also handled. Abstracting this layer from the underlying network layer allows session management to be implemented without regard to the supporting network technologies. The TCP and UDP protocols are described in more detail in section 2.2.

2.1.3 Internet Layer

The Internet layer deals with routing, or transferring data from one network to another. Sending of data across networks entails machine addressing and identification in order to pass data between routers along the route from source to destination address. The primary protocol at this level is the Internet Protocol (IP) [30] which is used to transfer data across routers, spanning packet-switched networks. Host-addressing within IP is represented by a four byte data type, often written in dotted decimal format, e.g. 136.206.18.12. Routing is carried out by a combination of protocols, including Border Gateway Protocol (BGP) [31], the main routing protocol of the Internet. Routing support protocols such as the Internet Control Message Protocol (ICMP) [32] (mainly used to pass error messages between operating systems) also exist at this layer. Transport layer data is encapsulated within the IP header and network hardware needs only inspect this header and not within to route the packet.

2.1.4 Link Layer

Also known as the Network Interface Layer, in the link layer we find protocols used to send data along physical links from device to device. 48 bit Media Access Control (MAC) physical addressing is used at this layer by network hardware for sending packets between physical nodes. The link layer thus handles next hop, short-haul routing whereas the Internet layer provides long-haul routing across multiple networks. Here addresses are re-written along the journey from source to destination based on the two physical connections currently passing data between each other. An important protocol at this level is the Address Resolution Protocol (ARP) [33] which translates IP addresses to physical Ethernet MAC addresses so that the destination interface can be located and communicated with over Ethernet.

2.1.5 Comparisons with the OSI Model

A number of alternate network models exist. Besides TCP/IP, the most influential of these is the OSI model. The Open Systems Interconnection (OSI) Reference Model (ISO 7498 [34]) similarly applies a layered approach for guiding development of networking protocols. The OSI model consists of seven layers however which loosely correspond to the four layers of the TCP/IP model as shown in Figure 2.1.

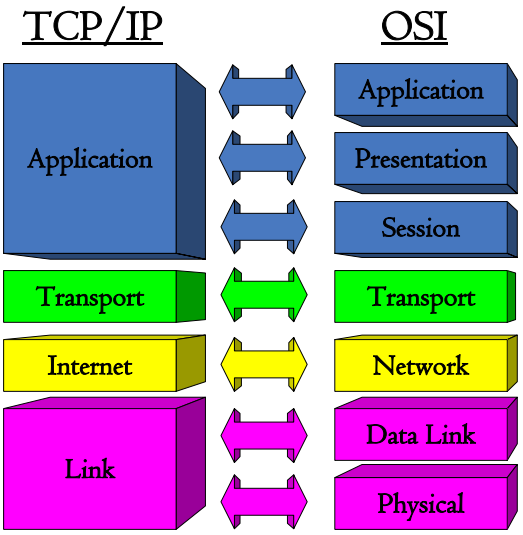

Figure 2.1 – TCP/IP Model & Equivalent Layers in OSI Model

Today, Internet protocols are inspired primarily by the TCP/IP model with the OSI model providing a useful tool for network teaching. We concentrate on the TCP/IP model here since we deal with real world security issues.

2.1.6 Encapsulation

The layered approach used in the TCP/IP and OSI models means that a protocol handles only the intricacies relevant to a specific function. It encapsulates this data within a header and passes it on to be dealt with at another layer. For example, to follow Figure 2.2 (adapted from [35]), an application developer writes an application that constructs a message using some application protocol (A) and hands it off to the OS. The OS implements and supports transport layer protocols such as UDP or TCP. The OS places a transport layer header (B) on this application header. Because the OS handles transport layer details, applications can use this facility without regard to its internal operation. Next the packet is given an Internet layer header (C) for sending to the Internet. This header will be inspected by routers which do not need to look any deeper

into the packet before making decisions. The final addition to the packet is a link layer header and footer (D) for transferral of data on the local physical link. This data and its controlling protocols are relevant only to each interconnection of nodes residing on either end of a physical link.

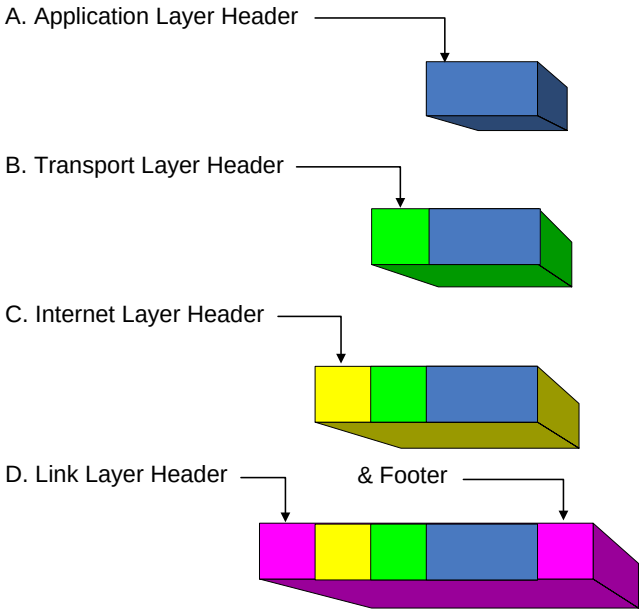

Figure 2.2 (adapted from [35]) – TCP/IP Model Encapsulation

Example

Consider a user requesting a web page through their Internet browser. At the application layer, the HTTP request is created by the browser. Also at the application layer a DNS request is made to translate the server name into an IP address. Next, at the transport layer, a TCP header is added by the OS, enclosing the previous HTTP request's application layer data. An Internet layer header is added by the OS which specifies source and destination addresses allowing network devices to route the packet between client and server. Lastly, the link layer header and footer are added and specify information required for passing the packet out on the local network interface. The packet is now ready for passing to the network.

2.2 Protocols

We look in detail at a selection of TCP/IP protocols here (we do so to provide context for the network threat and to better understand scanning techniques presented in Chapter 4). The main transport layer protocols of the Internet are UDP and TCP. We look

firstly at the simpler of the two, UDP before moving on to TCP. Finally we look at the Internet layer protocol, ICMP.

2.2.1 User Datagram Protocol (UDP)

UDP, RFC 768 [36] is a lightweight, stateless, connectionless, unreliable, unordered protocol whose simple transmission service is used to provide a quick and simple transfer of packets where error checking and correction are either not necessary or are provided at the application level. The efficiency gained by neglecting connection management functions within the protocol is sometimes desirable. UDP traffic is increasing on the Internet recently due to the rising popularity of peer-to-peer (P2P) technologies and streaming media [37] applications. Errors and ordering issues are corrected by the application.

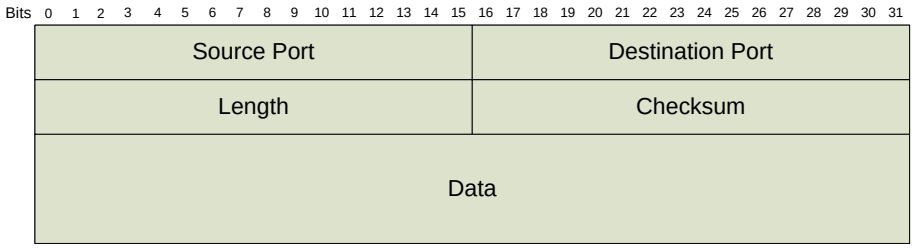

Figure 2.3 (adapted from [36]) – UDP Header

As can be seen in Figure 2.3 (adapted from [36]), the UDP header is accordingly very simple, having only parameters for addressing, datagram length and a checksum for basic error-checking.

The Port Abstraction

A ‘port’ abstraction provides for application addressing. Services bind themselves to ports so that applications can locate them, e.g. DNS may be running on 192.168.1.1 at UDP port 53. Source and destination ports are allocated two bytes of space in the UDP header so there are 65,526 possibilities. Some port numbers are reserved for a particular

purpose by The Internet Assigned Number Authority (IANA) [38]. Port numbers are subdivided by function as follows:

Start	End	Description
1	1023	These 'well known' port numbers are reserved for common services and should be used only by them. Only applications running with elevated privileges may listen on these ports in many environments including Unix-based OSes for security reasons.
1024	49,151	The 'registered ports' are designated for a specific purpose as a convenience to Internet programmers. Applications on these ports do not require elevated privileges.
49,152	65,535	These high numbered ports may be utilised by any application with any privilege level and are not each associated with a particular purpose. They are often used temporarily during communications with a server, e.g. a web server on port 80 may communicate with a client awaiting a response on port 56,789.

2.2.2 Transmission Control Protocol (TCP)

TCP (RFC 793) [39] is a connection-oriented protocol which provides session management functions such as requesting re-sending of lost packets for reliable communications between client and server. IP alone is unreliable and requires TCP at the transport layer to ensure arrival and reassembly of all packet transmissions. Communication is optimised for reliability rather than speed. Like UDP, TCP uses port numbering for local addressing, but the operation of TCP is more complex given its extra responsibilities. As depicted in Figure 2.4 (adapted from [39]), the protocol header contains significantly more parameters than that of UDP.

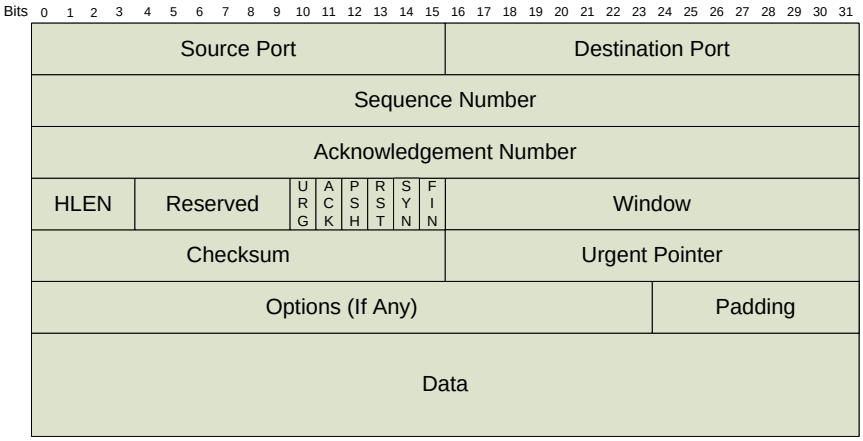

Figure 2.4 (adapted from [39]) – TCP Header

Connection Establishment

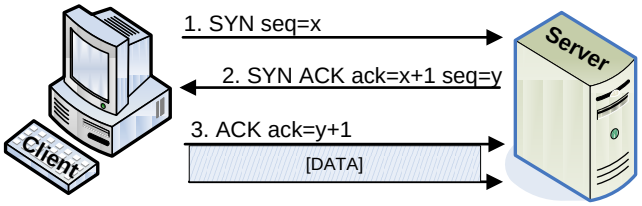

Figure 2.5 (adapted from [39]) – TCP Connection Establishment Handshake

Connection establishment is a key feature of the TCP protocol. A three-way handshake takes place as depicted in Figure 2.5 (adapted from [39]), to initiate the connection and synchronise the machines before data transmission may commence. The client sends a packet with the SYN flag set and a 32 bit sequence number x to the server to start the process. The server responds with a packet which has the SYN and ACK flags set and acknowledges receipt of the client's SYN packet by placing x+1 in the acknowledgement number field. The server also defines its own sequence number y in the sequence number field. The client responds with a packet which has the ACK flag set and acknowledges receipt of the server's sequence number by placing y+1 in the acknowledgement number field. At this point (even within this ACK packet) data transmission may commence. Sequence numbers keep track of ordering of received packets. After sending of data is complete, connection termination takes place.

Connection Termination

Graceful TCP session termination is specified by RFC 793 [39]. As can be seen in Figure 2.6 (adapted from [39]), the teardown process begins when a packet is sent by the client with the FIN flag set and the current sequence number x. The server acknowledges this sequence number in a packet with the ACK flag set and the acknowledgement number field set to x+1. This packet is followed by another from the server, this time with the FIN and ACK flags set, acknowledgement number again set to x+1 and the sequence number field populated with the server's current sequence number, y. The client acknowledges the server sequence number by placing y+1 in the acknowledgement number field of a packet with the ACK bit set.

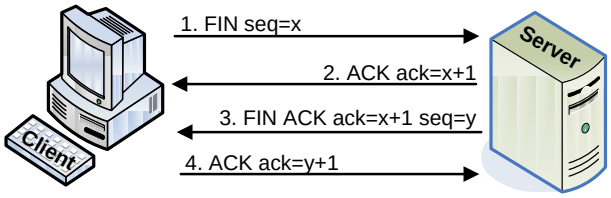

Figure 2.6 (adapted from [39]) – TCP Connection Termination

Connection termination may also automatically occur after a timeout to remove dormant connections. Incorrectly programmed applications or unforseen events such as a power failure can cause such situations to arise. RST packets sent by either side of the communication alert the other that no more data should be sent and the connection is thereby abruptly terminated.

Packet Sequencing

Sequence and acknowledgement number fields are used to reorder, track and detect lost TCP segments. The sequence number is incremented for each segment sent to record the current byte number. Each device keeps track of the current sequence number of the other and acknowledges receipt of packets by populating the acknowledgement number field with the sequence number of the byte which it expects to see next. A lost segment

will be detected as the sender receives no corresponding acknowledgement. See Figure 2.7 (From [40]) for an example of packet sequencing in action.

Flow Control

Flow control aims to optimise the rate of data transmission without flooding the target host or network. The window field of the TCP header is used for this purpose in a technique known as "sliding windows". A window is the maximum number of bytes which a receiver agrees to accept in the current transmission sequence. Capacity can fluctuate over the duration of a communication session and therefore is known as a sliding window. Although the 16 bit size of this data field limits window size to 65,535 bytes, TCP options can increase this size further. Starting window size will usually be specified by the operating system or application and defaults to 536 bytes. The sender must buffer both outgoing and already sent data until the receiver has acknowledged its receipt (in case retransmission is required). If the client's buffer fills, as in Figure 2.7 (from [40]), the receive window size is set to zero and the window is said to be frozen, meaning that the sender must wait until the receiver sends a packet with the window size set to a non-zero figure for transmission to recommence.

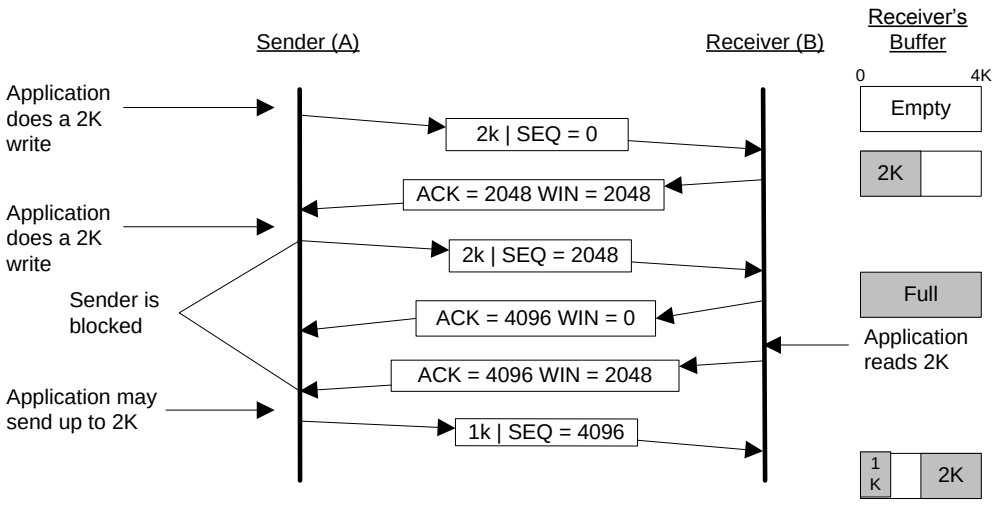

Figure 2.7 (from [40]) – Sequencing and Flow Control in TCP

2.2.3 Internet Control Message Protocol (ICMP)

ICMP (RFC 792) is one of the communication protocols at the Internet layer of the TCP/IP model. It is used primarily by OSes to send error messages notifying other systems of unreachable destinations, gateways with insufficient buffering capacity etc. There are also information requests defined within the protocol however and these may be used for troubleshooting tasks such as to discover if systems are alive and responding. ICMP does not employ the port abstraction associated with the TCP and UDP transport layer protocols and therefore messages are addressed towards the target machine rather than towards a particular application as illustrated in Figure 2.8 (adapted from [32]).

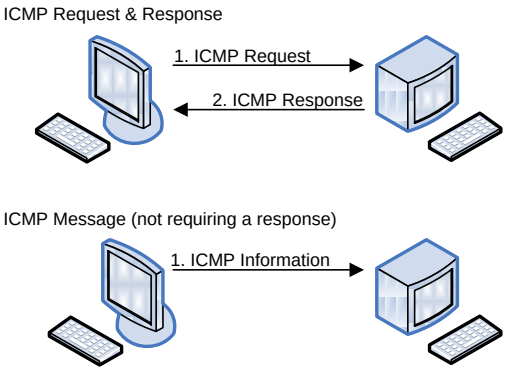

Figure 2.8 (adapted from [32]) – ICMP Protocol

Type	Description	Type	Description
8	Echo Request	0	Echo Reply
13	Timestamp Request	14	Timestamp Reply
15	Information Request	16	Information Reply
17	Address Mask Request	18	Address Mask Reply
10	Router Solicitation	9	Router Advertisement
3	Destination Unreachable		
4	Source Quench		
5	Redirect		
11	Time Exceeded		
12	Parameter Problem		
30	Traceroute		

Table 2-1 - ICMP Messages and Their Relationships (if any)

From Table 2.1, echo, timestamp, information, address mask and router solicitation requests may invoke a response from a machine. All other message types supply information without request.

2.3 Attack Taxonomies

Having looked at the network context of attacks in the last section we look here at the attacks themselves. Several computer security attack taxonomies have been published [41, 42] which attempt to bring structure and categorisation to the topic of attack analysis. We take the example of Howard's taxonomy [43], as it is specific to network and computer attacks and allows depiction of the entire attack process.

According to [44] attack classifications are generally based on the attributes of the attack, the underlying vulnerability and the attack detection method.

This generalisation holds for Howard's taxonomy. According to [45], an attack taxonomy should be comprehensible, complete, unambiguous and effective. The classification procedure must be clearly defined and mutually exclusive categories must exist. An attack must be mapped to a single category. Terminology must comply with established security terminology. A particular taxonomy may not necessarily meet all of these requirements but all are desirable properties.

2.3.1 Howard's Taxonomy

Figure 2.9 from Howard's Common Language for Computer Security Incidents [43] depicts the five logical steps which an attacker must take during an attack. The attacker uses a *tool* to exploit a *vulnerability* to perform an *action* against a *target* in order to achieve an *unauthorised result*. An *action* directed towards a *target* comprises and *event*.

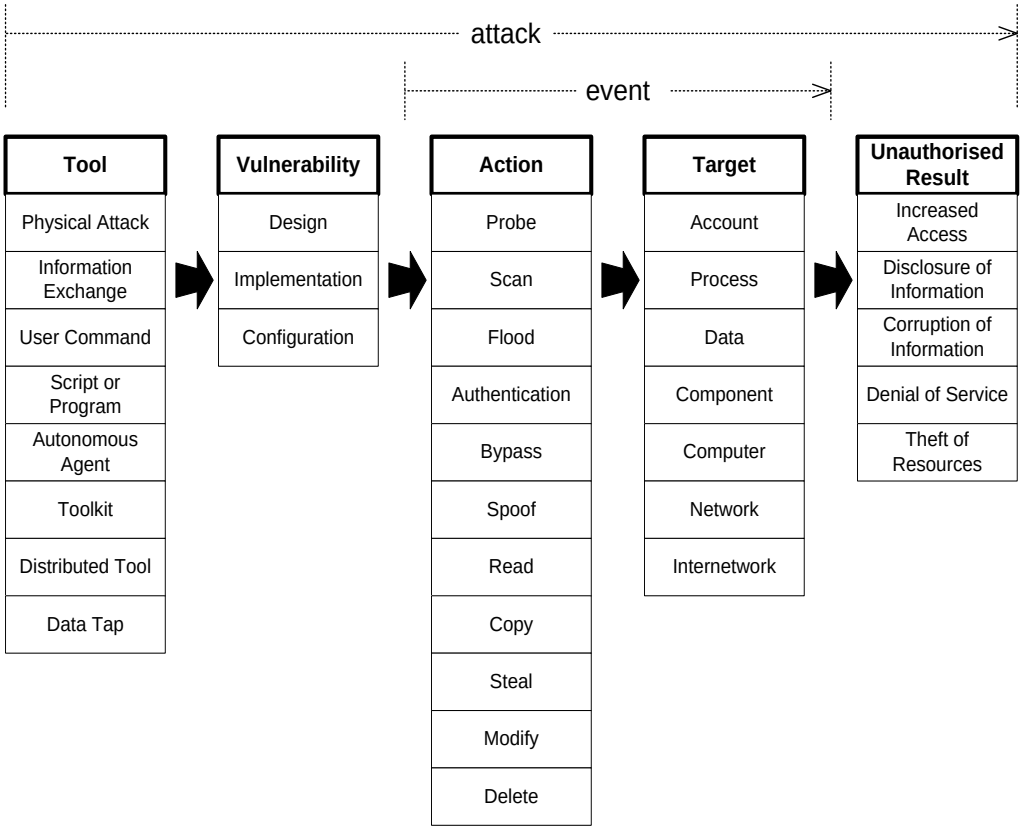

Figure 2.9 (from [43]) - Computer and Network Attacks

An incident may involve multiple attacks which together reach an overall objective (Figure 2.10). For instance an attacker wishing to steal a credit card database from a web server may begin by injecting shellcode into a service which is vulnerable to buffer overflow, providing her with a remote shell. A privilege escalation attack may then be required in order to acquire sufficient privileges to access the database. A data modification attack may be required in order to mask evidence. This group of attacks, which together fulfil the objective of retrieving the database constitute an incident.

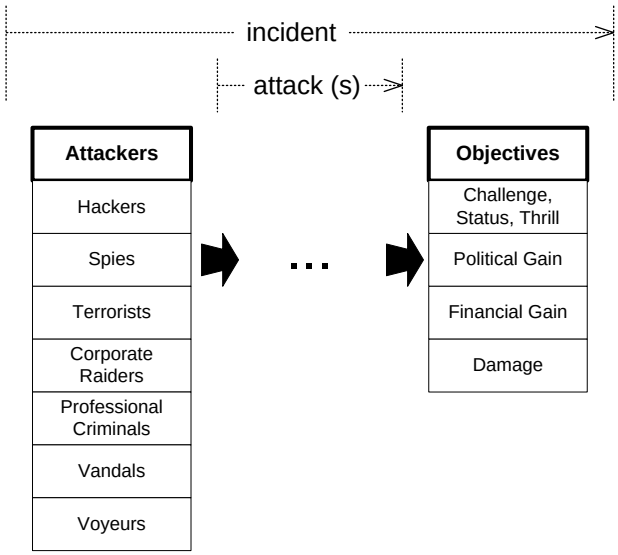

Figure 2.10 (from [43]) – Howard's Computer and Network Incidents

Below we briefly describe each step in an attack in some more detail.

Tools

The first element in Howard's model is the tool used to carry out the attack. More generally it is the method by which the attack is carried out. A physical attack requires the attacker being physically located in proximity of the target where she accesses a company's data storage office and destroys backups for example. Autonomous agents

function independently of user input such as virus or worms while a virus requires user intervention. A toolkit is a collection of functions which centralises the attack process. An example toolkit is the BackTrack [46] live OS which can be run directly from a CD and is specifically designed for penetration testing, forensics and recovery so includes numerous applications of use to attackers.

Vulnerability

A vulnerability is a susceptibility to attack. In computer terms, this translates to a weakness which could allow an attacker to violate confidentiality, availability or integrity in the system. Vulnerabilities may be introduced during the design and implementation stages of a software project. Implementation vulnerabilities may stem from insecure data types, logic flaws or a lack of sanitisation within an application. A third type of vulnerability, the configuration vulnerability, is introduced by the user of an application who selects inappropriate settings, thus allowing an attacker to exploit the system. Insecure settings may sometimes be the default for an application. For example, an email server may by default be configured to allow relaying of emails, thus allowing sending of spam by attackers. The attacker hopes to exploit a vulnerability using one of the aforementioned tools in order to carry out some action.

Action

A vulnerability allows the attacker to carry out some action. This action may include any of the following:

- Probing a system for information related to its setup and/or installed applications and services (possibly with a view to launching other attacks)

- Flooding the target system with data causing it to slow down or cease to function

- Authenticating through deceptive means by masquerading as a valid user (for example, having using a network packet sniffer to catch passwords sent over insecure protocols such as telnet)

- Modifying or deleting information

Target

An action is directed towards a target. The target is a logical or physical entity. A physical entity is a hardware device, for example a computer, network or Internetwork.

Logical entities consisting of informational items (such as the account of a particular user) may also be targets of an attack.

Unauthorised Result

The overall goal of an attack is to achieve some unauthorised result. The goal could be privilege escalation for example. If a low privilege level guest account is available for use with FTP, an attacker may attempt to use this as a starting point from which to launch further attacks to increase privilege level to ultimately control the machine. Disclosure of information involves viewing or stealing of information by persons unauthorised to do so. Denial of service (DoS) involves making a network service or system unavailable or inefficient for users. For critical services which must be online at all times, denial of service attacks can be mitigated through load balancing, and over provisioning but the DoS problem remains a difficult issue to solve.

2.3.2 Case Study: The TJ Maxx Attack

Here we take the example of a security incident which received a considerable amount of media attention, namely the TJ Maxx credit card theft scandal, and apply Howard's model in its analysis.

11 people in the US have been charged in connection with the theft of over 40 million credit and debit card numbers from TJ Maxx stores in what is considered[1] to be the largest ever identity theft case in the US [47]. TJ Maxx is operated by TJX Corporation. Several analysts estimate JTX will incur costs of between 500 and 1 billion usd [48] in handling the incident (investigation, customer notification and lawyers fees).

According to InformationWeek [49], three methods were employed in order to steal the information over a number of years. Firstly, computer kiosks within stores were opened to insert USB drives containing Trojan applications which allowed the attackers to later connect remotely. These kiosks were connected directly to the company network without the protection of a firewall, thereby allowing the attackers full access to this network at a later date. Secondly, many stores used wireless networks protected only by the weak WEP encryption scheme, which, as discussed in section 1.2.3, can easily be fully broken for key retrieval and full network access. Finally, in-store credit and debit card PIN-pad terminals were replaced with altered replicas which would store details of each card processed in a method known as credit card 'skimming'.

[1] As of 5th August 2008.

Howard's Model Applied to TJ Maxx Attack

The overall incident can be classified as follows.

Incident	
Attackers	Professional Criminals - Attacks were carefully planned and conducted by skilled individuals
Objectives	Financial gain - The credit and debit card information was sold on the black market and organisers procured large financial rewards [50]

As mentioned there were three avenues of attack. Accessing the network through kiosks can be broken into two attacks.

Kiosk Trojan Installation	
Tool	Physical Attack - The kiosk was physically accessed
Vulnerability	Design - The acceptance of USB devices should have been disallowed at kiosks by design
Action	Modify - The kiosks had Trojan software installed to allow remote connections
Target	Computer - The kiosks are thereby compromised
Unauthorised Result	Theft of Resources - Kiosk resources can be used surreptitiously

Kiosk Trojan Network Penetration	
Tool	Script or Program - A Trojan
Vulnerability	Design - The internal network should have been firewalled from access by kiosks
Action	Steal - The network is now accessible for retrieval of data
Target	Network - The attacker now has access to the target network
Unauthorised Result	Disclosure of Information - Through accessing the network, credit card data was attainable

WEP Cracking	
Tool	Script or Program - Network sniffer application
Vulnerability	Implementation - Encryption implementation should have been configured to use a more secure protocol such as WPA2
Action	Bypass - Security is totally circumvented by cracking the encryption
Target	Network - Access to the network is achieved
Unauthorised Result	Disclosure of Information - Once on the internal network, sensitive data was attainable

Credit Card Skimming	
Tool	Physical Attack - Hacked PIN pad
Vulnerability	Implementation - PIN-pad terminals were not physically secured and could be exchanged
Action	Modify - Functionality carried out during the PIN authorisation process was altered
Target	Account - Credit and debit card details representing accounts
Unauthorised Result	Disclosure of Information - Credit and debit card details

2.3.3 Discussion

Howard's incident-based taxonomy allows for consideration of the entire attack process. As mentioned by Hansman [45], one of the desirable elements of a taxonomy is mutual exclusion. This property is not held by Howard's taxonomy since classifications can often overlap. For example, within the attacker category for our TJ Maxx implementation above, these 'Professional Criminals' could also be considered to be 'Corporate Raiders'.

2.4 Gauging the Threat with Honeypots

The TJ Maxx incident is one example of a network attack but how common are such attacks? Honeypots and/or honeynets provide some insights into the threat.

A honeypot is a computer which is set up with the intention of luring attackers into launching attacks against it. It may be attached to the Internet, usually carrying default configurations and without the latest security patches (supplementary controls are required however to block successful attackers from reaching other machines on the network). Verbose logging is carried out and all Internet traffic is stored and analysed. Any attempted access is considered suspicious since there is no legitimate reason to connect. A group of honeypots on a network are collectively known as a honeynet.

There are two main reasons for implementing a honeypot or honeynet. Firstly, research honeypots aim only to gather information about the general attack environment and are usually used by universities, governments, the military etc., hoping to learn more about threats [51]. Production honeypots are deployed to mitigate risk through identification of attack patterns within an organisation. The latter are easier to set up and configure than research honeypots, but information gathered is limited to which machine the attacks are coming from and which exploits are being attempted. Research honeypots provide more feedback about the attackers and their tools. Levels of interaction

between attacker and honeypot also vary; ranging from low, to medium, to high. Low interaction systems only simulate services (which thus cannot be exploited to gain full control of the honeypot). Medium interaction systems lack a full OS but services are more complex than in their low interaction counterparts. Finally, high interaction systems are the most complex, involving intricate design and risk exposure of the real operating system and its services. Research honeypots are typically high interaction whereas production honeypots may be low or medium interaction.

An example functioning honeynet in a third level educational institute in Ireland is described in [52]. It allows investigation of recent Internet threats and demonstrates that probes to Internet connected systems are extremely common. The virtual machine based, medium interaction honeypots include Ubuntu 8.04, Windows Server 2003 SP2, Windows 2000 Professional SP4 and Windows XP SP3. Results over a three week testing period [53] uncover some interesting activity. Numbers of unsolicited network packets exceeded 550,000 and included a worm infection. Affected ports and services are listed in Table 2.2 (from [53]).

Service	Port	Protocol	Attacks
Reserved (icmp)	0	ICMP	444
Microsoft-ds	445	TCP	3984
Netbios	135	TCP	349
Netbios	139	TCP	3968
http	80	TCP	722
telnet	23	TCP	16
ssh remote login	22	TCP	52
ms-sql	1435	UDP	118
Unassigned	1026	UDP	3883
Unassigned	1027	UDP	3865
Unassigned	1028	UDP	3714

Table 2-2 (from [53]) - List of Services on Honeypots and Number of Attacks on these Ports

These results make it clear that the Internet is fraught with risk of attack and protection is vital.

2.5 Summary and Conclusion

Having detailed some of the main protocols over which network attacks function, we discussed an example attack taxonomy, through which it was possible to demonstrate what constitutes an attack including the motivations behind it, procedures carried out within and the overall outcome of the attack. Application of the taxonomy in the analysis of a recent, high profile network intrusion incident was presented. Finally it was demonstrated through analysis of honeypot research that network attacks are extremely common, leading to the conclusion that implementation of network defences is vital, given the importance and increasing ubiquity of the Internet in modern society. The following chapter proceeds to discuss defences to these network attacks.

3. The Defences

As has been illustrated in Chapters 1 and 2, there is a clearly serious threat of network attack from which individuals and organisations must strive to protect themselves. This protection takes several forms including firewalls and intrusion detection systems (IDS). In this chapter we look at how both of these defences operate and how they are implemented. We also describe in detail an IDS designed and implemented as part of this research[2]. Results generated from running this IDS on the campus network are presented and analysed.

This chapter is structured as follows. In section 3.1 we describe firewalls along with an overview of some important concepts in the area. Section 3.2 introduces intrusion detection, detailing relevant concepts and system architectures. In section 3.3 we provide a description of our own IDS including test results and future work. A summary and conclusions are presented in section 3.4.

3.1 Firewalls

Firewalls block unwanted network communications and allow only authorised traffic according to a set of rules defined by the user [54]. A firewall can be hardware or software-based and may exist on the host or at the network perimeter where it assesses traffic to the local area network. Network security configuration will vary across organisations but certain considerations always apply in strategically placing firewalls, IDS and other security systems. Following identification of the most vital and valuable systems through risk analysis, the latter may require a higher than average degree of protection. See Figure 3.1 for an example network security configuration for a small to

[2] Development of this system was funded under Enterprise Ireland's "Proof of Concept" scheme.

medium sized company including suitable firewall and IDS locations. Three main generations of firewall exist: first generation - stateless, second generation - application layer and third generation - stateful packet filters.

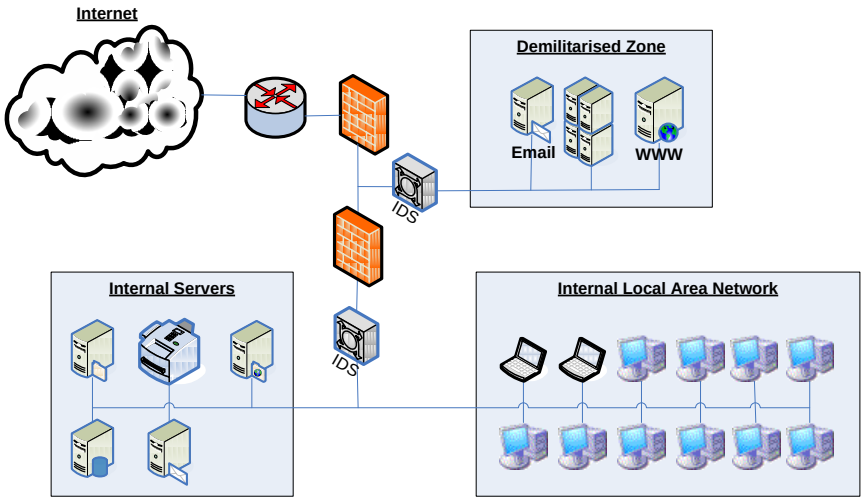

Figure 3.1 - Security Conscious Network Setup Example

3.1.1 First Generation - Stateless Packet Filters

Stateless packet filters are simple in design, inspecting each packet individually. The state of a connection is not taken into consideration in making a decision. An example of a popular stateless packet filter is the Linux-based iptables [55] (stateless by default but can also be configured for stateful analysis). Stateless filtering can provide a combination of low overhead and high throughput. However, decision making is not as fine-grained as with other firewall types.

3.1.2 Second Generation - Application Layer Packet Filters

The application layer packet filter is designed to understand certain application layer networking protocols (see Chapter 2) such as HTTP, FTP etc. This understanding can be applied to block certain websites, monitor for downloading of malicious software,

check whether a disallowed protocol is being used over a non-standard port or whether a protocol is being used in a harmful manner. Examples of application layer packet filters include Zorp [56] and WinRoute [57].

3.1.3 Third Generation - Stateful Packet Filters

Stateful firewalls keep track of sessions and a connection is described by parameters such as source and destination IP address, ports and the current stage of the connection's lifetime i.e. session initiation, in progress, closing or closed. Rules are applied to new connections and once accepted, traffic on this connection is allowed. Two stateful firewall products are Check Point FireWall-1 [58] and Cisco's Adaptive Security Appliance (ASA) [59]. See section 3.3 for further information and the author's implementation of stateful packet inspection in IDS.

3.2 Intrusion Detection

Intrusion detection can be deployed in conjunction with a firewall to provide a complementary layer of network security. Intrusion Detection is the disclosure of attacks made on IT systems with the intention of breaching the confidentiality, integrity or availability of all or part of the system. An intrusion detection system (IDS) will report on attacks whether successful or not in order to provide the network administrator with a view of dangerous traffic traversing their network, adding another important layer to the security of a network [60].

This section is composed as follows. Section 3.2.1 provides an overview of intrusion detection technology. In section 3.2.2 a description of the main architectures and concepts associated with IDS is provided before introducing some example systems in section 3.2.3.

3.2.1 Intrusion Detection Overview

The first intrusion detection model was published in 1986 by Dorothy E. Denning [61] and IDS has become, over recent years [62], a popular addition to the firewall in a network's defensive arsenal. While, as previously mentioned, firewalls are used to block intrusions at the network perimeter based on certain traffic characteristic; they cannot defend against attacks which target legitimate services. For example, organisations often configure their firewall to allow access to port 80 (HTTP) on the Internet web server within their network. If an intruder attacked this web server and gained access to the network, an IDS may detect and flag the resulting internal attack traffic (following an intrusion, attackers or malware may attempt to spread to other hosts within the internal network). IDS take a more detailed view of traffic than firewalls. They apply more sophisticated rulesets and use deep packet inspection which, in addition to examining TCP, UDP and IP header parameters, can inspect the payload of a packet for evidence of attacks and make decisions based on the contents. Security reports are generated on any alarm being raised and delivered to administrators who must respond accordingly in order for the system to be of security benefit.

The main incentives for adoption of IDS are as follows:

- Detection of attacks – The primary function
- Enforcement of security policies – For ensuring that a network is being utilised only in the intended manner
- Audit trail – Administrator has information on how an attack was carried out and the methods used to compromise a system
- Extra security information – Useful for discovering how adequately other security mechanisms are functioning

3.2.2 Intrusion Detection Architectures and Concepts

This section introduces the network, host and hybrid intrusion detection architectures along with strengths and weaknesses of each. Also described are common methods of detection: rule-based and anomaly-based analysis.

Network IDS

ID systems can be categorised according to where the system itself is physically located. Network IDS (NIDS) monitor traffic on the network, looking for evidence of attacks which are then reported to an administrator. The most popular NIDS is Snort [63]. Figure 3.1 in section 3.1 displays possible physical locations for placement of NIDS to complement the firewall and monitor the various areas of a network.

The network IDS has the benefit of monitoring a group of host machines from one physical location on the network, meaning minimal associated maintenance as only one update is required to cover the entire network with a new rule or configuration option. NIDS will usually be invisible to the attacker who either does not realise that this layer of protection exists, or is unaware of the specific rules and therefore whether her presence on the network has been detected. Drawbacks include the associated single point of failure and the possibility of overload with heavy traffic on larger networks. As is the case for any network level security system, NIDS only monitor hosts while they are connected to the associated network, leaving mobile devices open to attack while offsite and operating on less secure networks. Offsite infection with malware presents a real threat to network security as such devices bypass network firewalls on rejoining their home network.

Host IDS

On the host an IDS has access not only to network traffic data, but to all local system calls, OS kernel logs, application logs, network equipment logs etc. Any events occurring on the host can be monitored and therefore a host-based IDS (HIDS) has more information available to detect attacks. This could include local network interface traffic monitoring as in network-based systems (as is the case for our own agent-based network intrusion detection system (see section 3.2.4)).

The main benefit of HIDS is the availability of more information than network traffic alone. Another benefit is continued protection for mobile devices while offsite and on unsafe networks. As more network devices become mobile, it is expected that the number of new attacks threatening mobile devices will also increase. Thus mobile and wireless attack detection techniques will become increasingly important [64]. The distribution of protection across the network is another benefit, resulting in shared processor and memory requirements across hosts and thus allowing for scaling to larger sized networks. Drawbacks include configuration, updates, installation and bug fixes which, when required, will be necessary for possibly hundreds of systems on a network.

Hybrid IDS

A hybrid system merges the previous two architectures, providing the benefits of both types of system for maximum threat coverage. An example hybrid system is Prelude [65] which aggregates results from a number of systems for analysis at one central location, whether host- or network-based.

Signature-Based Detection

Another categorisation method for IDS lies in their mechanism for detecting intrusions. Rule-based detection, also known as signature detection, matches observations against

signatures or rules in a database of known attacks, which in the case of NIDS is in the form of network packet sequences or their contents. Similarly to antivirus signature detection, the disadvantages of the approach are that it can only detect known threats and the signature database must be kept up to date. An advantage is the low number of false positives, since notifications are based on specific rules. False positives may still arise however through inappropriate or poorly constructed rule files. An example signature-based IDS is Snort [21]. An example Snort rule which attempts to detect the SQL slammer worm looks as follows:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"W32.SQLEXP.Worm

propagation"; content:"|68 2E 64 6C 6C 68 65 6C 33 32 68 6B 65 72
6E|"; content:"|04|"; offset:0; depth:1;)
```

This rule specifies that any UDP packet directed towards any internal IP address from any other IP address with the contents specified by the 'content' parameter at the particular 'offset' within the packet results in the associated alert.

Anomaly-Based Detection

Anomaly or behaviour-based detection learns how a system usually behaves and deviations above a certain threshold from the accepted baseline metrics are considered potentially threatening and reported. Thresholds can be set in terms of CPU usage, memory usage, network packet types, user typing rate etc. [66]. An issue with anomaly detection is that a system may be maliciously retrained over time by an attacker to accept anomalous behaviour as normal, thereby removing the possibility of detection. An example tool which can carry out anomaly-based detection is rrdtool [67].

It is possible to combine both signature and anomaly-based approaches into a system that both monitors for matches to signatures and at the same time investigates behaviour patterns. This approach was taken in development of a system by SRI International [68]. Here both benefits and drawbacks of each method are inherited, but the possibility

of monitoring for known signatures while still potentially catching new, unclassified attacks is provided.

False Positives & False Negatives

One problem facing IDS is the volume of erroneous results reported. False positives (false alarms) are alerts generated from benign data by the IDS. False negatives (misses) on the other hand are attacks which are not detected by the IDS. The main requirements of an intrusion detection system are low false positive rate and high true positive rate [64]. Many methods have been proposed in order to reduce false positives but the issue remains a significant problem [69]. Errors will be common when a system is first deployed, but through tuning of rulesets and configuration, can be reduced. As the number of false positives increases, the likelihood of the IDS administrator missing actual attacks simultaneously rises. Decisions on which events warrant a notification are in the hands of the administrator and will vary based on the organisation/network in question [60].

3.2.3 Example Systems

There are many IDS systems, both open source and commercial available today. The two most popular according to Insecure.org [63] are Snort and Open Source Security, Host-Based Intrusion Detection System (OSSEC) [70]. Both of these are free, open source systems, performing signature detection at network and host level respectively.

OSSEC HIDS performs log analysis, integrity checking, root-kit detection, time-based alerting and active response. Architecturally, the system is generally deployed to all monitored clients, forwarding alerts to a management station for analysis and is marketed as an addition to, not a replacement for NIDS.

Snort monitors and performs deep packet inspection of network traffic in real-time in order to match patterns set by an administrator. Rules can be easily updated and can be as broad or as specific as desired although modifications have a direct influence on the number of alerts generated. Snort can be set to perform Intrusion Prevention (IP) if desired, dropping packets which attempt "unacceptable" behaviour such as stealth scanning. Outputs can be long and difficult to decipher and possible add-ons to aid analysis include database visualisation and log file processing tools.

Packet logging tools may be used to capture network packets, leaving the administrator to manually view data for threats. Logging all packets for manual inspection ensures all traffic is caught but manual analysis requires considerable skill. This approach is not practical on a full-time basis due to the volumes of traffic generated by any one device, let alone an entire network, but is useful in monitoring for unknown threats or for forensic evaluation after an attack. Popular packet logging tools include tcpdump [71] and wireshark [72].

Many vendors provide various types of commercial IDS. Cisco ASA offers hardware based IDS/IPS functionality along with many other features.

3.3 Developing a Host-Based, Web-Enabled, Hybrid IDS

Firewall logs routinely show evidence of port scans and attempts to connect to non-running services, especially on untrusted networks such as cybercafé or airport wireless networks. These events, revealing attackers or malware attempting to spread, are often overlooked by many users who are unaware of the dangers they face on the Internet. IDS attempt to remedy this situation by alerting users to attacks, but for the average, untrained user, IDS are usually either overly expensive, overly complicated to install and run or both. Further, with NIDS analysing traffic from one central location, mobile devices may move from monitored to unsafe networks and become infected. These

issues are considered and attempts made to rectify them through development of a host-based, web-enabled hybrid IDS described below.

This section describes in detail this system [73] and is structured as follows. In section 3.3.1 we outline the main requirements of our system at a high level, followed by design and implementation details in section 3.3.2. Results obtained through testing of this system are delivered in section 3.3.3. Future conclusions are presented in section 3.3.4.

3.3.1 Requirements

The approach adopted here is to take administration tasks out of the hands of the host system user, offloading it to a skilled remote authority by transferring pertinent data across the network to an administration server for analysis. The system also takes the approach, unusual in intrusion detection, of providing network traffic monitoring on the host in a distributed environment, thereby sharing the task of packet analysis across the network. The system must be lightweight for transparent use by now popular limited-power mobile devices, while providing an extra level of deep packet inspection by incorporating Snort at the server. Agent processing load is raised only upon instruction from the server following detection of a possible threat.

In the system, users receive feedback on the behaviour of their machine (as seen by others) over the Internet. They are also free to roam with their mobile device, accessing the Internet from potentially threatening locations such as cybercafés, removed from the safety of protected corporate or home networks while still receiving this feedback. User's awareness of Internet threats will be heightened by this process due to the new visibility of the attacks that they are subjected to, thus contributing to their ability to take subsequent steps in protecting themselves and their network (according to OECD [20], raising user awareness is a key factor in defeating malware). The system should be simple to install and maintain in comparison to other IDS. In particular the agent is run by unskilled users and therefore transparency is of utmost importance.

During development of this system, open source tools should be adopted to enhance productivity while avoiding duplication of effort.

3.3.2 Design and Implementation

The system is logically divided into agents and server. This section begins with an overview of the system as a whole, followed by descriptions of the agent and server.

Overview

Installed on each client is a software agent which monitors its own network traffic (see Figure 3.2). Agents submit only minimal information until faced with a perceived threat, in which case more information is submitted. Administrators use a web interface to monitor health of clients and the overall network. All data sent by agents is visible at this web interface where the administrator can also interact with each agent, e.g. to update rulesets or to request more detailed monitoring of a particular host based on events received.

In Figure 3.2 we see a typical deployment with desktops, laptops and PDAs on a network, each running our agent. A PDA sends a data packet (event X) to a laptop. When the laptop receives this data, our analysis engine may deem it suspicious (more on the decision process in section 3.3.2.2) and sends a report to the server where it is stored in a database along with results from Snort, to be viewed later by an administrator. Also in Figure 3.2 one of the group of laptops is receiving an update from the server, perhaps in response to some perceived threat.

Users (owners of client machines running our agent software) can also log in to the server where they are provided with reports on their machine.

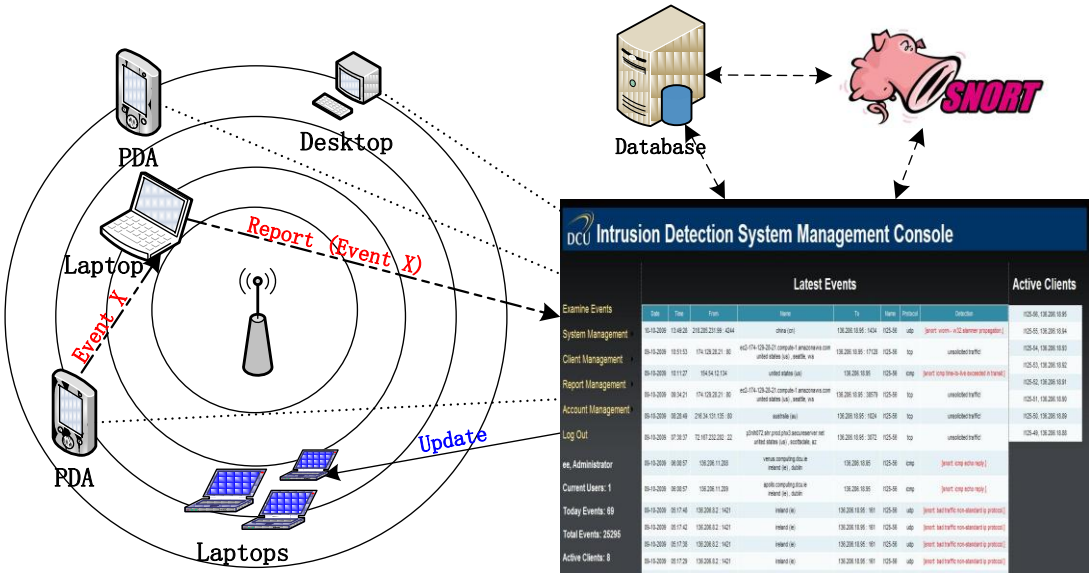

Figure 3.2 - IDS Implementation Displaying a System Under Attack

In this section we describe some of the relevant details of the agent's design and implementation. Agents submit data to the server for analysis and also receive updates from the server.

Design

The primary function of the agent software is to monitor its host's network traffic and report relevant events and packets to the server. Requirements state that agent software must be capable of running on multiple device types. As such, the agent must be simple, flexible and lightweight enough to be ported to PDAs and other resource-limited devices. The agent offers no graphical user interface (GUI), instead users connect to a web application to receive feedback.

Filtering and Reporting

Filtering and reporting are the main functions of the agent application. Figure 3.3 depicts the decision process implemented by the agent for each inbound packet along with the reporting or otherwise which results. These inbound packets could be in the form of TCP, UDP or ICMP traffic and are treated appropriately to the given protocol. Decisions for each inbound packet are handled as follows.

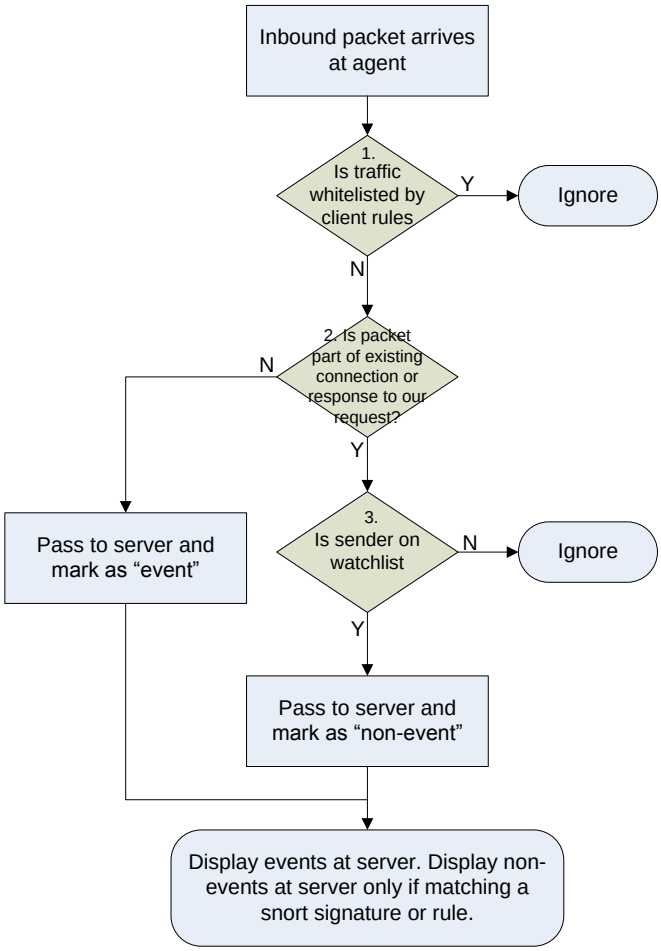

Figure 3.3 - Agent Decision Tree

Decision 1 – Is traffic whitelisted by client rules?

If certain traffic is allowed (whitelisted) on this client, then the ruleset will reflect this and the packet is ignored. For example, if the client is running a web server on port 80, client rules will whitelist traffic to this port and it is therefore ignored and not analysed further by the agent software.

Decision 2 – Is packet part of existing connection or response to our request?

Traffic which is not whitelisted continues to be analysed. As will be discussed further in section 3.3.2.2.2, connection tracking (stateful packet filtering) is implemented by the agent. For TCP traffic, if a connection is initiated by the client, then subsequent traffic on this connection continues for further analysis at decision 3. If it is not part of an existing connection however, the packet is passed to the server and marked as an "event". In the case of ICMP and UDP traffic, no actual connection is established at the transport layer, therefore a response to a request from this host is considered in the same fashion as that of a TCP connection. Incoming traffic not whitelisted by the administrator and not solicited by this host is therefore considered suspicious and is always sent to the server and marked as an "event".

Decision 3 – Is sender on watchlist?

Decision 3 applies to non-whitelisted packets which are part of an existing connection or response to a request made by the client. The watchlist, as discussed further below, is a list of systems held by the agent specifying which clients should be monitored closely. Therefore, at decision 3, if the sender of this packet is on our watchlist, then the packet is passed to the server and marked "non-event".

Arrival at the Server

In any IDS, one of the issues with which the administrator is faced is the vast quantity of data requiring manual analysis. In order to minimise the amount of data at the administration interface, only events are displayed. Upon arrival at the server, both

events and non-events are passed to Snort for further analysis. Following Snort analysis, events are appended with any extra information. Non-events which Snort categorises as security-relevant are promoted to event status and displayed for analysis, otherwise they are ignored.

Client Rules (Whitelisting)

Agents possess a list of rules, specifying which traffic should be whitelisted. Whitelist entries can be based on IP, MAC address, protocol, port or any combination of these. On initial installation of the system, an agent holds a blank whitelist. Remote updating is carried out by the server to populate the whitelist as necessary.

Connection Tracking

TCP, as described in Chapter 2, follows pre-defined mechanisms for connection initiation and termination. By interrogating traffic contents, connections are tracked and treated according to their current status as described above. UDP and ICMP traffic is connectionless at the transport layer and therefore is tracked for a particular client based on whether the communication was initiated by that client i.e. subsequent return communications resulting from outgoing requests are allowed. Timeouts (based on those used by IPTables firewall) along with memory usage thresholds are used to remove dormant connections.

Watchlist

Agents possess a list of IP addresses known as a watchlist. Traffic originating from any of these addresses is monitored closely. The aim is to provide more detailed analysis of systems which are deemed suspicious. As described above, details of all packets originating from a watchlisted client are sent to the server for analysis, whether or not they are part of a locally initiated communication. Again, each agent begins with a blank watchlist which is updated remotely by the server. This watchlist may be configured manually by an administrator or automatically where the server adds to the

watchlist any clients which exceed some suspicion threshold, e.g. a client reported 20 times over a 10 minute period.

Communication

In addition to network traffic, agent and server communicate through heartbeats and updates. Heartbeats are messages sent by agents at configurable time periods to notify the server that they are currently active. The server can thereby list each client as active, providing the ability to interact with it or, conversely, know that a client has gone off-line. Updates are communicated from server to agent in order to modify configuration, whitelist or watchlist.

Implementation

Following are some details on how the described software was implemented.

Communicating with the Server

Each agent which is online and actively communicating with the server sends Extensible Markup Language (XML) based "heartbeat" messages at configurable time intervals to advertise availability to the server. Reported events and non-events are marked up and transferred in XML according to RFC specifications for Intrusion Detection Message Exchange Format (IDMEF) [74] and Intrusion Detection Exchange Protocol (IDXP) [75]. Use of these protocols allows for simplified analysis of data and also for integration with other systems, e.g. the hybrid IDS aggregator, Prelude which allows import and management of IDMEF data (the IDMEF standard was developed with the participation of the Prelude team). Configuration and rule updates from the server are similarly specified using XML.

Stateful Packet Filtering - TCP

For the agent to function correctly, it must distinguish between traffic that belongs to a valid connection and unsolicited traffic. Below we describe how the agent tracks

connections. For packet capture, WinPCap [76] libraries are employed for their proven speed and efficiency. TCP traffic initiated by the local machine is caught in order to detect connection setup and teardown events. Once a connection is established, traffic is allowed since setup has been agreed between both hosts, unless the other party is on a watchlist, in which case intermediate traffic is also monitored.

TCP connections are represented by quadruples of IP address and port numbers of machines involved in the conversation. This combination will always be unique.

As were detailed in section 2.2.2, there are three distinct stages which a TCP connection will go through:

- New

  Connections which are going through the initial TCP three-way handshake are given the 'new' status.

- Active

  An active connection has had a valid three-way handshake carried out and the two machines involved have agreed to communicate. Data passing is allowed between these hosts.

- Closed

  A connection which has gone through the TCP connection teardown sequence is given a status of 'closed'. As we shall see below however, we must track connections for a short time after they are terminated.

Connections within these various stages are tracked by the agent software.

TCP Connection Tracking Issues and Solutions

A number of issues, including the following, arose during implementation.

Solitary SYN Packets

SYN packets represent the initial step in the TCP connection establishment handshake. For this reason a reference to each outbound SYN packet must be stored in order to watch for its companion SYN-ACK and ACK packets. We only track the establishment of outbound connections, since inbound ones must be whitelisted; an allowed service will be reflected in the agent's rules. A solitary SYN packet which does not provoke a responding SYN-ACK could be left in the connection queue with status of 'new', never to achieve 'active' state. Such a situation could arise where traffic is dropped or the target never replies due to firewalling etc. To prevent build-up of 'new' connections, we set a timer and clear out any 'new' connections which are older than the allowed timeframe. To decide on this timeframe we applied the same approach as IPTables [77], two minutes.

Connection Teardown Issues

It was discovered that different OSes can differ in their handling of connection teardown. To cover this, following some experimentation it was decided to consider any FIN packet as ending a connection since all variations of the teardown include at least one FIN. These connections are then moved to a status of 'closed' but allowed to function as active for a period in order to handle subsequent teardown traffic.

RST Issues

RST packets caused similar issues to FIN packets. An RST packet indicates to the receiver that one is not willing to send any more data and that the connection should be terminated. However, often data transfers are still attempted by the RST recipient on a given connection following receipt of an RST packet. Usually this is followed by further RST packet transmissions by the original RST sender. Similarly to our solution for FIN packets, we maintain the connection for a time so that further communications are allowed.

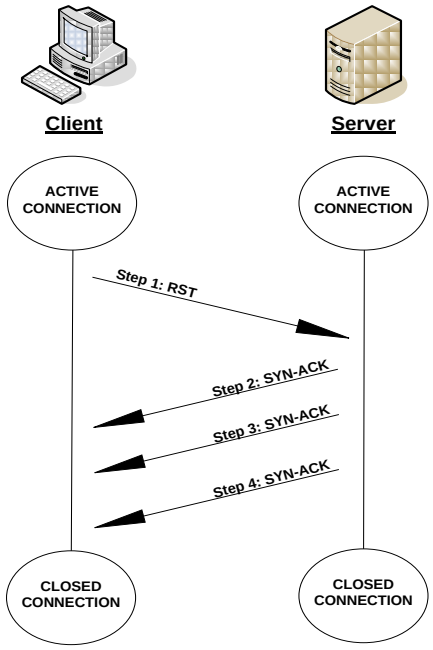

Figure 3.4 - Abnormal RST Sequence

Figure 3.4 demonstrates the problem. An RST packet is sent by the client to the server which responds with three SYN-ACK packets to the same port. Allowing the connection to receive packets for a period following closing solves the problem. Yahoo.com is one example of a site displaying this behaviour.

Stateful Packet Filtering – UDP & ICMP

With UDP and ICMP no connection is established. However, should a client machine send a UDP request, we need to be ready for the reply in order for it not to be reported to the server. Similarly for ICMP.

Connectionless protocols were handled as follows:

- If the initial communication originates from the host machine, then this first outgoing packet causes the "connection" to be flagged as valid and no logging of a reply is made.
- Conversely, when the "connection" originates from a foreign machine, this is considered to be suspicious and sent to the server.

Non-Connection Oriented Tracking Issues

There were some issues with implementing the tracking of UDP and ICMP. It was decided that any outgoing UDP packet is considered as the start of a connection. Also, having no connection teardown sequence means that we cannot easily discard the connection once tracking is no longer required. The resolution was to keep all UDP connections alive until the configurable designated storage capacity is exceeded, at which point, the oldest UDP connection is discarded.

Speed

As with any IDS, speed considerations become important since a high number of the packets passing through a machine's network interface card must be inspected. Hash lists were implemented along with specialised collection types to hold data related to connections to optimise performance. For example sorted classes were employed so that rather than having to search for the oldest connection, this was always the first one. This and other specialised collection classes within the open source C5 generic collection library project [78] were used.

Multithreading was used for further speed enhancements.

Server

Server software provides a remote monitoring and administration system for all agents across a monitored network. Here a brief overview of its design and implementation are provided[3].

Design

A remote administration interface is provided by the server which provides the following functionality to users:

Event Analysis & Response

- Examine Events
- Configure Server Rules
- Configure Client Rules

Client Management

- Update Client Rules
- Update Client Configuration

Watchlist Management

- Configure Threshold Settings

Account Management

- Add Users
- Remove Users

Report Management

- Generate Security Reports

Table 3-1 - Server Functionality

[3] The author was involved in the server's design but not implementation which was carried out by a project colleague.

Implementation

The server provides a web-based management console to the administrator, implemented in ASP.NET through C# and Ajax, integrating a SQL Server database and automatic post-processing of submitted traffic using a local Snort installation. Figure 3.5 displays the main web page of the system showing details of events. Active clients are listed on the right, with options and information to the left. The example in Figure 3.5 contains an instance of the SQL Slammer worm being reported by Snort.

Figure 3.5 - Server Management Console

Reports are received from agents and, if they trigger a server-side rule, the corresponding action is taken. The administrator may decide that an event is harmless and instruct the client agent(s) to ignore such events in future by updating their whitelist.

All events are securely logged for non-repudiation and configurable views allow the administrator to view events by agent and thus rapidly assess the health of a specific client in addition to that of the overall network. Also, owners of client machines can log in to the server and receive feedback on their machine (see Figure 3.6). This feedback details any attacks the client was subjected to and any reports of the client acting suspiciously as observed by other agents in the system.

Figure 3.6 - Client View

Events and non-events are passed automatically to the server side Snort installation for analysis. Results from Snort are added to the server database and included for analysis on the web interface. On applying Snort analysis to non-events generated through watchlisting, these may be given the status of event if security relevant alerts are obtained. Unless they become events, they will not be displayed, an approach that aims to keep the interface uncluttered.

3.3.3 IDS Deployment and Test Results

The agent was installed on two machines, one on the School of Computing wireless network and one on the School of Computing laboratory network. Both agents were left to run over a period of several weeks. Each machine was configured with an empty whitelist in order for maximum vigilance. The host machines do not run any services.

As is the case with a honeypot, all unsolicited traffic to the machines can therefore be regarded as suspicious.

The DCU internal network is protected by a perimeter firewall and monitored by a Snort-based NIDS whose rules are maintained by a third party company. Communications with the Internet from within the University network are through proxy servers.

The system was run for the entirety of the month of October 2009 during which time the following emerged.

No Internal Firewalling

Numerous events were reported to the server from various subnets across the University. Once on any DCU subnet (even open wireless networks) it is possible to communicate with any other subnet in the DCU network.

External probes

Given that our agents are behind a network firewall, external probes should not reach them. However, 216 external IP addresses showed up in the server database, submitted by the agents. This could mean either that the firewall is configured incorrectly, or that internal attackers or malware are spoofing their source address.

Upon investigating whether external IP addresses were being spoofed internally it was found that such spoofing is possible even though an externally source addressed packet arriving on an internal interface should be dropped. Therefore it is possible that some probes reporting external source addresses may have actually originated from internal machines.

On testing from a remote network for receipt of external probes using nmap, it was discovered that, although no responses were obtained, a number of packets did in fact arrive at their destination for certain scan types.

IP	# Probes	Ports	DShield Reports
67.228.177.191	182	80, 1024, 3072	798,467
61.111.114.20	78	61220, 33555	105,948
208.43.231.120	23	18 different ports	63,672
208.43.74.141	15	1024 & 3072	62,812
68.178.232.100	14	1024 & 3072	41,383

Table 3-2 - Top 5 Reported IPs

Table 3.2 summarises the top reported IPs and their targeted ports. DShield [79] is a free, online database which correlates firewall log results submitted by volunteer users. This data can be used for purposes such as analysis of attack trends, and in our case lookup of a particular IP address to discover whether it has been reported as attacking other machines. As we can see, each of these source IP addresses is reported as having attacked a high number of systems across the Internet. All of the top 10 and 54% of the total 216 external IP addresses were reported a number of times to DShield. Three of the top five IP addresses above probe port 1024 which is often utilised by backdoor applications such as NetSpy [80] and MyDoom [81]. Again, firewalls should block these external probes. Probes were seen to a total of 234 ports. Some ports had as few as one but others had up to 900 probes targeted towards them.

Internal Probes

13 internal source IP addresses were reported to the server, varying in numbers of reports from 1 to 5405. These probes may be due to malware, may be the result of active network reconnaissance by internal users or may be due to misconfigured network services.

900 probes target port 161, a port associated with SNMP. Given clients are not running an SNMP server, this activity is suspicious in itself, but moreso given the inherent dangers associated with the SNMP protocol [82]. All of these SNMP probes arrive from a single host which is not attached to the computing domain and are classified by Snort as 'non-standard-protocol' or 'attempted-recon'.

Attempted Malware Infections

Not only is external traffic reaching internal machines, but it is attempting to infect them. One interesting discovery provided by Snort analysis was 63 instances of the Slammer worm [83] attacking our client machines. Figure 3.7 displays an instance of Slammer being detected by Snort. Since infection proceeds over UDP, no connection need be established making this attack particularly dangerous. Here a geo IP database is used to identify the source of the attack as China.

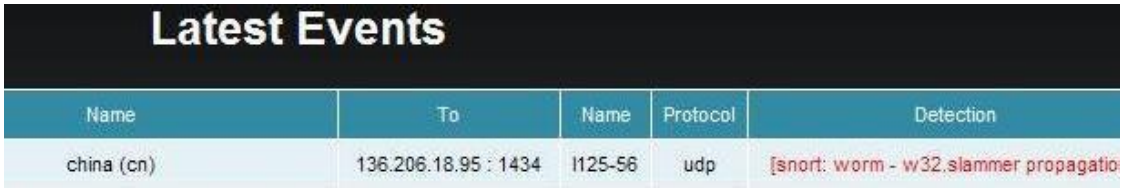

Figure 3.7 - Slammer Detected

Further Results Generated from Snort Feedback

Table 3.3 summarises Snort feedback along with the number of instances of each:

Snort Feedback	Num Results
NULL	6868
BAD TRAFFIC Non-Standard IP protocol	1066
ICMP PING	154
WORM - W32.Slammer Propagation	63
SNMP request udp	30
ICMP Destination Unreachable Port Unreachable	6
ICMP Time-To-Live Exceeded in Transit	3
BAD-TRAFFIC tcp port 0 traffic	1
ICMP PING NMAP	1

Table 3-3 - Snort Feedback

"NULL" represents instances where Snort provided no feedback. Such events are simply marked "Unsolicited Traffic" by the server.

3.3.4 Conclusion

Deployment of the prototype on the DCU campus network has already yielded some interesting results which could prove useful for improving network security. We can see that a serious vulnerability seems to exist which allows probes through the firewall and also internally there is no firewalling across subnets. The presence of external probes on the network has been flagged to network administrators.

We cover possible future extensions in Chapter 6.

3.4 Summary

Having described some of the network security defences currently available, our own implementation of intrusion detection was presented, including design and development details. The system was deployed and tested on the DCU network and some interesting results reported.

Given the vulnerabilities identified through testing our IDS, there is a case for network security assessment of the DCU network. We discuss how the network security process is typically carried out in Chapter 4 and present results of analysing the DCU network in Chapter 5.

4. Network Security Assessment

Having established in the last chapter that some potential insecurities exist in the DCU campus network, a full network security assessment is warranted. In this chapter we describe the means by which network security assessment is carried out. Results of this analysis are presented in the following chapter.

This chapter is structured as follows. In section 4.1 we begin with an introduction to the field of network security assessment. In section 4.2 we detail a selection of the steps involved in a general assessment including gathering publically available data, scanning the network, assessing remote services, assessing web servers and web applications and finally assessing email servers. In section 4.3 we introduce some of the main tools that help in the process before ending with a summary in section 4.4.

4.1 Introduction

Network security assessment, also known as "penetration testing" or "ethical hacking", replicates the activities of attackers in order to discover weaknesses and vulnerabilities within networked computer systems. The aim is to uncover vulnerabilities before an intruder exploits them for denial of service, theft or destruction of data etc. After the assessment process, any findings may then be flagged to the system or network administrator who can then apply corrective measures.

The three main approaches to security assessment are flaw hypothesis [84], attack tree [85] and formal methods [86]. Within this section we describe these approaches along with legal, corporate and practical issues related to the field of network security assessment.

4.1.1 Assessment Approaches

According to McDermott [87], the first published reference to "penetration testing" was by R. R. Linde [84]. His paper detailed the idea of a 'flaw hypothesis' methodology where the software system under investigation is examined in order to identify possible threats. All design, development and user documentation are investigated along with code, in search of potential flaws. The same idea can be translated to network penetration testing where instead of software analysis, all available information relating to network infrastructure is collated and investigated.

Another security assessment approach uses attack trees [88] to graph attack scenarios. The root node represents the overall attack goal with paths from leaf nodes to the root describing particular attacks. Each node may have an associated cost, representing the resources required by the attacker to implement that step in the attack. Software exists for automating creation of attack trees e.g. Amenaza [89].

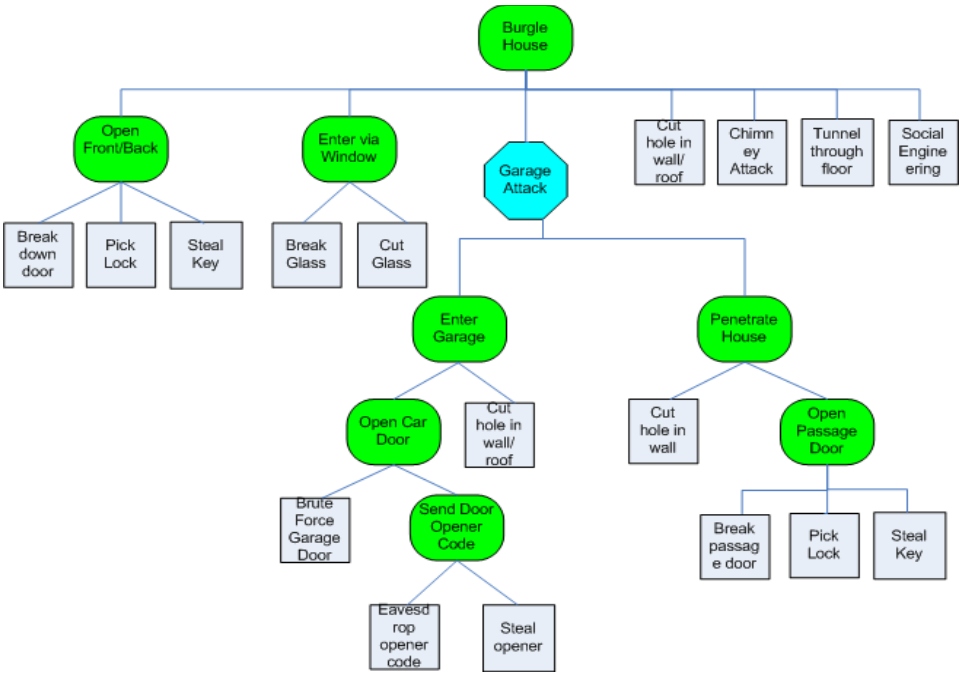

Figure 4.1 (from [90]) – Attack Tree showing ways to Burglarise a House

Figure 4.1 (from [90]) shows an example attack tree where the overall goal is burglarising a house. To implement the attack tree approach in the context of network security assessment we may begin with a top node of 'Data Compromise' or 'Network Intrusion' which is then similarly decomposed. Each of these scenarios is tested to provide a documented, structured approach to the security assessment process.

Formal methods [91, 92] are another method of structuring an assessment. In such approaches, a model based on the organisation's own security policy is constructed, with a formal description language used for policy specification. The model contains information about system entities, their current state and the possible transitions between states. The aim is to prove that the system cannot enter an insecure state.

4.1.2 Issues to Consider in Network Security Assessment

Issues which arise in the area of network security assessment are introduced here. Legal, corporate and practical issues that must be borne in mind before conducting an assessment are reviewed.

Legal Issues

In order to avoid legal complications, all planned assessment activities including IP address ranges, types of attacks to be conducted etc. must be documented in advance and signed by management and the assessment parties. Irish computer crime is covered by the Criminal Damages Act 1991 [93] and the Criminal Justice (Theft and Fraud Offences) Act 2001 [94].

Corporate Issues

The value of a security assessment to a company can be difficult to measure in that there may be no tangible outcome besides feedback on whether the network is considered 'secure' [95]. Whereas being 'secure' means a system is safe from threats, security

compliance means the system conforms to a given set of security requirements [96]. Compliance provides assurance that a certain level of security has been achieved. This compliance can be required by government legislation, industry standards organisations or an organisation's own policies. Major standards with which companies may be required or choose to comply include:

National Institute of Standards and Technology (NIST)

The United States Federal Information Security Management Act of 2002 (FISMA) [97] "provides a framework for ensuring the effectiveness of information security controls over information resources that support Federal operations and assets".

Payment Card Industry Data Security Standard (PCI DSS) [98]

This international standard must be adhered to by any company which handles credit card data. Development of the standard was initially pushed mainly by MasterCard International Inc. and Visa U.S.A. Inc. Compliance requires encrypted data transfers, logical and physical access controls, activity monitoring and logging and relevant to this thesis, regular monitoring and testing of networks.

ISO/IEC 27000 Series [99]

Delivered by the International Organisation for Standardisation (ISO) and the International Electrotechnical Commission (IEC), these standards cover information security management, risks and controls including those of computer networks. The recommendations are broad in scope in order to be applicable to many different types of companies.

Practical Issues

Many of the tests performed within a general security assessment can have dangerous ramifications for the stability of the system being tested. For example, if we test for a vulnerability which could shut down the computer, a successful test could cause denial of service. This test should therefore never be carried out in a functional production environment but would be a very useful undertaking before a system goes live. Even after a network security assessment has been carried out and relevant updates made in response to any results, security is not guaranteed. The most rigorous assessment can only cover known attacks, leaving new, "zero-day" exploits as an unforeseeable threat until they become publically known and can be defended against.

4.2 Assessment Methodology

For this section we follow the flaw hypothesis approach given its prevalence. We also incorporate, to an extent, the technical guidelines for information security testing and assessment provided by NIST [100]. A five step process is followed. In step 1, target information is gathered through online, public domain sources before applying this information to scan the network in step 2, which can glean information related to machines and services accessible on the network. The workings of some of the most popular scan types are explained. Using results from step 2, remote services are assessed in step 3. Web servers and finally email servers are examined in steps 4 and 5 respectively.

4.2.1 Step 1: Gathering Target Information

This first step involves gathering public information related to the target network from the Internet. The aim is to develop a picture of the target network.

Public Websites

Internet search engines [101] are used to locate the organisation's websites, along with those of any subsidiary, parent, auxiliary and associate companies. On company websites, we may find:

- Related companies or entities
- Contact names and email addresses
- Network (wireless and wired) information
- Addresses
- Phone numbers
- Merger or acquisition news

Such information can be used for social engineering attacks. Email addresses may also be harvested for spamming. Downloading the entire corporate website can be useful in order to run fast local searches for useful information. Often a site's HTML and client-side script contains extra information in the form of comments, hidden fields etc.

Details of wireless and/or wired networks available in the company office are sometimes available on the Internet. This information can facilitate gaining access through plugging into a network within the building or attaching to a wireless network from outside, possibly bypassing routers, firewalls, IDS systems etc.

Online newsgroups and forums can be a source of information about users and servers from a particular domain. Threads on network setup issues and questions posed by a company's network administrators can reveal valuable information.

DNS

DNS registration details for a particular company may contain useful information. Contact details for associated administrators along with postal addresses for the organisation are usually provided and can be looked up using tools such as whois.

A company may operate its own DNS servers. Several servers are often used for redundancy – one primary and several backups which are synchronised through zone transfers. Zone transfers should only occur between specified machines i.e. DNS servers for the particular network. Where an attacker can successfully carry out a zone transfer, much sensitive information about network internals is revealed.

Reverse DNS brute force attacks can also reveal much valuable information. Here the attacker knows the IP address range held by the target and can ask its DNS for a hostname for each address in the range. DNS should be configured only to resolve externally accessible hostnames, keeping internal hostnames private but often they are not.

If the target network is large enough, it may have its own AS (autonomous system) identification number for use with BGP routing. BGP separates networks and groups of Internet addresses into AS which communicate across the Internet backbone. AS are usually under the control of a company, university (for example DCU falls under HEAnet AS1213) or service provider. AS numbers and related information can be located and used to discover IP ranges and neighbouring network information [102].

4.2.2 Step 2: Network Scanning

Once a set of IP addresses has been determined using techniques outlined above, the network itself can be explored. The goal of scanning is to discover accessible hosts, services, and firewall configurations.

Scans involve firing packets at networked machines and listening for a response. Many scanning approaches exist and there are tools available to implement each. Often a different scan type returns different information so the choice depends on requirements. Scanning is often carried out using free software such as nmap (see section 4.3.2).

Scan results for a given port may be one of ‘open’, ‘closed’, ‘filtered’ and ‘open|filtered’, depending on the scan type. ‘open’ tells us that the given port is hosting a service with which we can interact from our present location. ‘closed’ means that this port is reachable but offers no service. ‘filtered’ means that access to the port appears blocked by a firewall and we cannot tell whether it is hosting a service or not, but we can tell that the firewall administrator does not want us to have access to this port. ‘open|filtered’ tells us that the port appears either ‘open’ or ‘filtered’ but nmap cannot be certain from the response, however combining this result with that of another scan type may generate a definitive result.

Following is a list of the main scan types along with their specific implementation details. We begin with simple connectionless types before describing TCP connection oriented scans.

Simple Scanning Methods

These scans employ protocols which, while being simpler in nature than their connection-oriented counterparts, have their own idiosyncrasies which we note below.

ICMP Scan

ICMP echo, timestamp, information, address mask and router solicitation requests invoke a response from their target. A response depends on whether the particular device is configured to respond and whether these message types are allowed on the target network.

The ‘ping’ application is available on all major OSes for sending echo request packets.

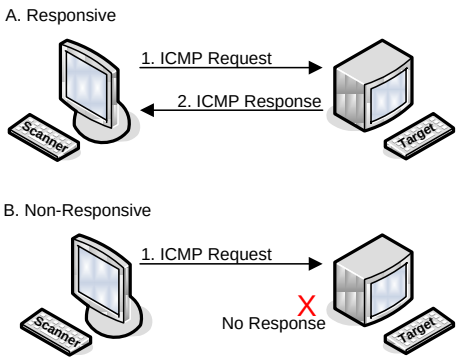

Figure 4.2 - ICMP Scan

Other request packets may be sent using specialised software. ICMP does not have the port abstraction associated with UDP and TCP and therefore this scan, as seen in Figure 4.2, can only discover whether a system is alive. ICMP is often blocked at network border firewalls so when scanning from outside a correctly configured network this scan type is unlikely to be successful.

UDP Scan

UDP is a connectionless protocol which, while not comprising as much of the Internet's overall traffic throughput as TCP, is increasing in volume with the recent popularity of P2P and streaming technologies [37]. There are limitations to this scan and therefore at times it is overlooked by security assessors.

As seen in Figure 4.3 (A), a UDP response to a UDP request indicates an open port. Most UDP implementations respond with an ICMP unreachable error from closed ports as in Figure 4.3 (B). The UDP standard does not require a response from an open port and a filtered port may similarly generate no response. Nmap will thus classify a lack of response as 'open|filtered' as in Figure 4.3 (C), meaning that the port could be open and simply not responding due to inappropriate packet payload or filters are preventing delivery of the UDP packet.

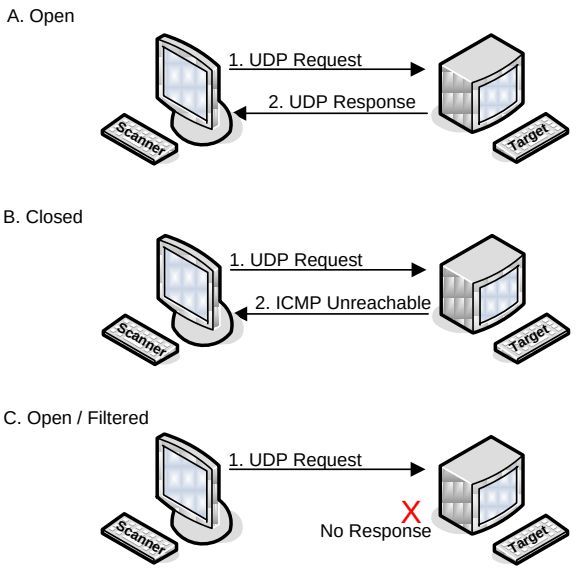

Figure 4.3 - UDP Scan

Speed is an issue for UDP scans. For each unresponsive port, the scanner must wait for a timeout to expire. The problem is that ICMP port unreachable errors are often rate limited. Methods of increasing speed include scanning multiple hosts in parallel or lowering the timeout value, thereby skipping slow hosts.

Furthermore, the simple, connectionless UDP protocol, unlike TCP does not guarantee packet delivery. ICMP port unreachable messages also offer no guarantee of arrival. For these reasons packets may be lost resulting in both false positives and negatives. Retransmission of packets is typically implemented to counteract this issue.

TCP Scanning Methods

Here we look at the more complicated connection-oriented transport protocol of TCP and some scanning methods which it makes available.

Connect Scan

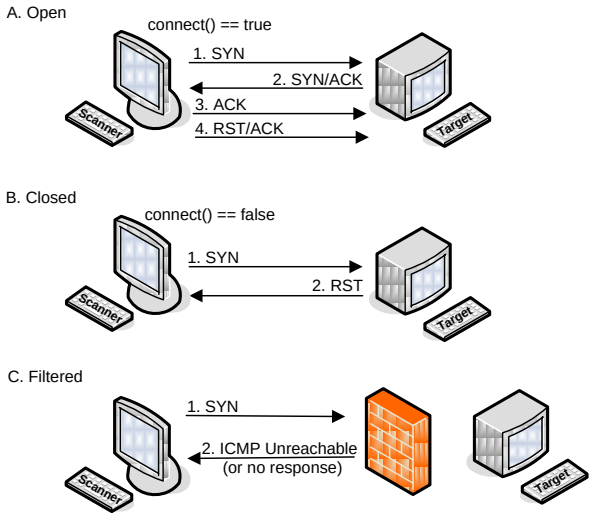

Figure 4.4 - Connect Scan

This scan type does not rely on low level raw packets but instead utilises the OS provided 'connect' system call to attempt connection establishment as seen in Figure 4.4. Responses take the form of the system call's return value rather than network packets. The scanning application does not have the same level of control compared with low level raw packet methods. Also, this method is not stealthy as any full connections created are often logged at the server. More packets and more time are required than for other scan types. The half-open scan below can obtain the same information without most of the drawbacks associated with creating a full connection.

SYN Scan

A TCP packet with the SYN flag set is sent to the target. This mimics the first step in TCP connection setup [39]. The scanning application listens for responses and, as seen in Figure 4.5, understands SYN/ACK to indicate open ports (A), RST to indicate closed ports (B) and a lack of response or an ICMP unreachable packet sent by a firewall to indicate a filtered port (C). SYN scanning is also known as half-open scanning since the scanning machine never responds with the third part of the TCP handshake and the connection is not fully set up. This means that the connection may not be logged by the target system, adding a degree of stealth compared to the connect method. Nmap sends an RST packet which resets the session on receipt of the SYN/ACK.

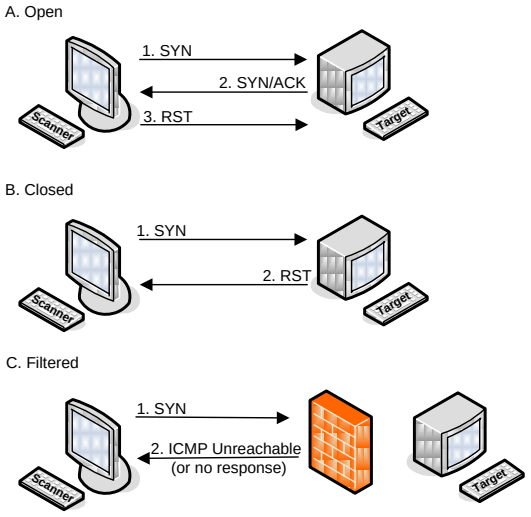

Figure 4.5 - SYN Scan

This scan type is fast with the possibility of probing thousands of ports per second. Simplicity of response (or lack thereof) interpretation is another advantage. SYN scan will be consistently reliable as it relies on TCP protocol standards that cannot be altered by the particular operating system being scanned.

ACK Scan

ACK scanning is used only to discover firewall rulesets and lists ports as filtered or unfiltered. As seen in Figure 4.6, a TCP packet with the ACK flag set is sent to the target host. If the packet arrives (A), in keeping with RFC 793 an RST packet is returned in response. The port is regarded as unfiltered although whether open or closed is not known. Non-responsive ports or those which return ICMP messages are listed as filtered (B).

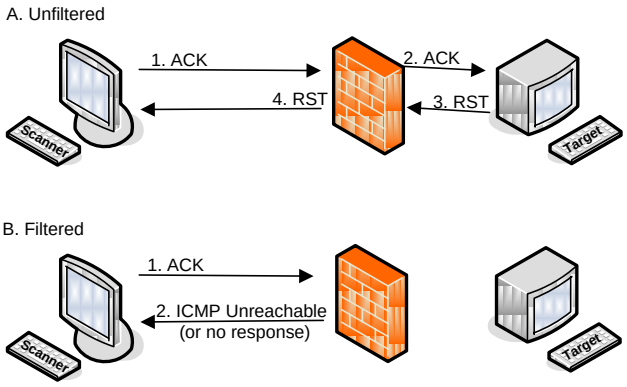

Figure 4.6 - ACK Scan

A variation of the ACK scan is known as Window scan. When a responding RST packet is received, the TCP window field is examined as seen in Figure 4.7. In certain OSes the RST packet's window field will contain a non-zero value if the port is open (A) and zero if closed (B). Unfiltered ports in these cases can be listed as open or closed.

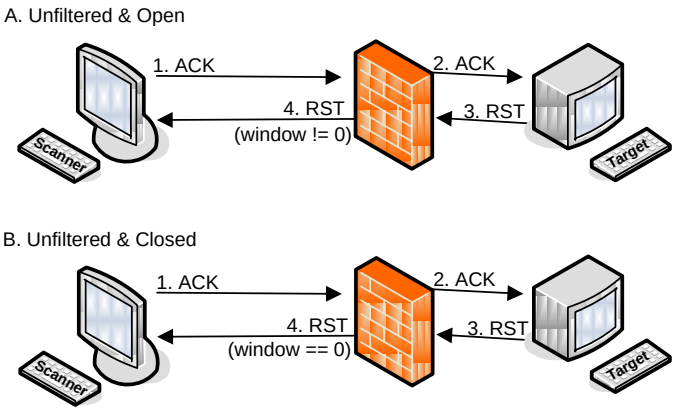

Figure 4.7 - Window Scan

Results, however, are not always reliable. The 'certain operating systems' mentioned above represents a small number of systems and on typical systems this scan will

consider all ports closed. Further investigation is required in order to determine whether results are accurate. For example, a mix of closed and open ports indicates a system is most likely susceptible to the window method. Conversely, a system with all ports showing open or closed is most likely not vulnerable to window scanning.

TCP Scan without SYN, RST or ACK bits set

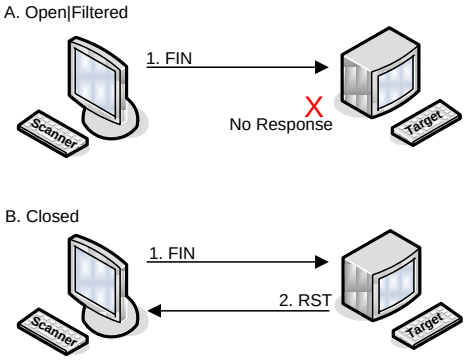

Figure 4.8 - FIN Scan

The TCP standard states that if the destination port state is 'closed', an incoming segment not containing an RST causes an RST to be sent in response. It also states that if the destination port state is 'open' and the packet has none of the SYN, RST or ACK bits set the segment should be dropped. Thus, in RFC compliant systems, as seen in Figure 4.8, a packet with no SYN, RST or ACK bit set will provoke an RST response from closed ports (B) and no response from open or filtered ports (A). Any combination of other flags will produce the same effect. Nmap's NULL (no flags), FIN (FIN flag only, as seen in Figure 4.8) and Xmas (FIN, PSH and URG flags) scans are variations on this approach.

The main benefit to the security assessor is their ability to bypass certain non-stateful firewalls and packet filtering routers (although some IDS systems can be configured to detect these scans).

Indirect Scanning Methods

Three main methods of indirectly scanning hosts are possible [103]. Such approaches offer a high degree of stealth as the scanning machine never communicates directly with the target; instead it arranges to have another machine undertake this function and

gathers any results obtained. The scanned machine therefore sees the intermediate host as the source of any packets it receives.

Zombie Scan

In a Zombie scan (a.k.a. "idle" or "dumb" scan) a remote host (the 'zombie') is probed for the numerical identification field from the Internet Protocol packet header which is used to uniquely identify fragments of an original IP datagram. This number is often predictable in being incremented for each IP packet sent. See step 1 in Figure 4.9. Next the target host is sent a TCP SYN packet, spoofed to the zombie machine's IP address (step 2). If a packet was sent to the zombie (step 3) in response to the spoofed packet, then an RST packet will be sent to the target by the zombie (step 4), incrementing the IP ID. Then the zombie machine is again probed for its IP ID (step 5). Since the IP ID is incremented per packet, an increase of two between probes may reveal the target machine's SYN/ACK in response to the spoofed probe.

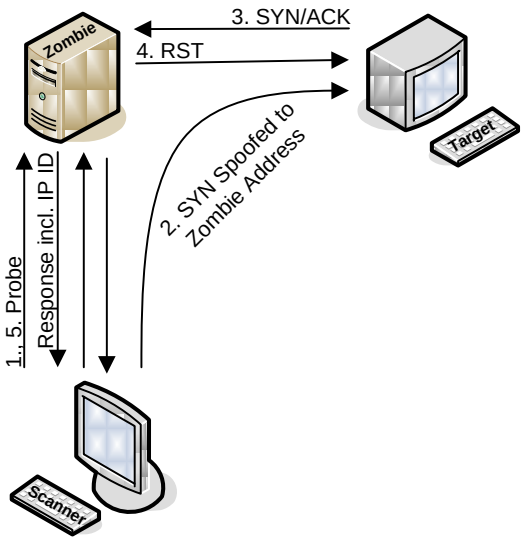

Figure 4.9 - Zombie Scan

The zombie scan has the benefit that a wide range of operating systems produce predictable IP ID sequences and can be exploited in this way, although the latest versions of Linux, Solaris and OpenBSD have corrected the issue [104]. A further requirement for success is that the zombie must also have relatively low traffic throughput at the time of scanning, otherwise its IP ID will be caused to increment by other processes. Even when traffic is low, a number of attempts for each port are necessary to be certain of its state.

FTP Bounce Scan

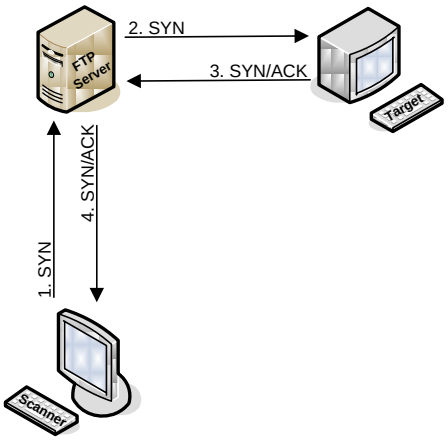

Figure 4.10 - FTP Bounce Scan

The next indirect scanning method is the FTP Bounce Scan. Here FTP systems with vulnerable settings are manipulated to scan a target through usage of the 'PORT' command (CVE-1999-0017). As we will see in section 5.7.1.1, three DCU servers are vulnerable to this misconfiguration weakness. Depicted in Figure 4.10, a vulnerable server acts as an intermediate proxy for any packets sent to it (step 1), passing each to the target (step 2). These messages contain the source IP address of the vulnerable FTP server and the target sees all packets as originating there. The target replies to the vulnerable server (step 3), which forwards the response to the scanner (step 4). Besides masking the packet origin, this method also allows an attacker to access targets blocked by a firewall but accessible to the vulnerable server. Another similar method is the proxy bounce scan which similarly takes advantage of an incorrectly configured proxy server to carry out the scan on behalf of a scanner.

Sniffer Scan

A final indirect method, the Sniffer Scan, captures network traffic destined for an intermediate 'zombie' machine to view responses to spoofed packets. Root access to a machine on the same network segment as the target is required. Consider a simple Ethernet

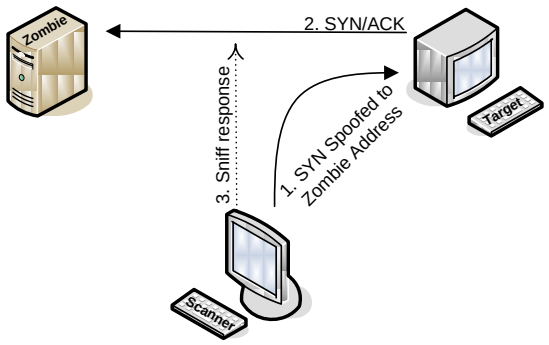

Figure 4.11 - Sniffer Scan

LAN. Each machine can see the reply to the spoofed packet but unless they are the specified recipient, the packet is ignored. However, packet sniffers such as Wireshark, working with a network card in promiscuous mode can capture all packets. The scanner sends a packet to the target with the source address spoofed to that of any other machine on the network segment (a 'zombie') as in Figure 4.11 (step 1). If the target machine transmits a response to the zombie (step 2) the scanner sees it (step 3), and therefore can tell whether ports on the target are open or closed.

On a switched network, an extra step is required. Address resolution protocol (ARP) [33] tables within the switch keep track of which host resides on which port and packets are transmitted only to the port held by the intended destination machine. However, ARP redirection can be used to distort the switch's ARP tables, causing packets which are intended for the zombie to be sent to the scanning machine instead. Another approach is to overload the switch's ARP tables which may result in the switch reverting to hub-like functionality and transmitting messages out all ports. Sniffing may thus be possible in a switched environment.

Indirect scan methods provide the highest possible level of stealth.

4.2.3 Step 3: Assessing Remote Services

Networked operating systems can support a number of remote information services which deliver information over a network connection. Having discovered a number of these services across the target network during step 2, an assessment of their security is now carried out. Some may divulge information of interest to an attacker; others may be insecurely configured, out of date and/or contain vulnerabilities. Services which could be exploited by attackers should be removed or updated as required. Table 4.1 lists commonly available services along with the ports and protocols on which they run.

Acronym	Name	Port
Wins	Windows Internet Name Service	42/tcp
domain	Domain Name System	53/tcp
domain	Domain Name System	53/udp
auth	Authentication Service	113/tcp
ntp	Network Time Protocol	123/udp
snmp	Simple Network Management Protocol	161/udp
ldap	Lightweight Directory Access Protocol	389/tcp
ldaps	Secure (over SSL) LDAP	636/tcp

Table 4-1 - Common Remote Services

Standard tools associated with a particular application protocol can often be used to glean information from the above services. Under some circumstances, protocols may be exploited to allow command execution or exposure of sensitive data. Up-to-date patching of all networked software is vital to avoid exposing vulnerabilities. Simple Network Management Protocol (SNMP) is one example of an application layer technology used to retrieve/configure network device settings. Hardware and software versions and configuration information for the device can be retrieved from its 'management information base' using SNMP. Known for its high number of insecurities [82], SNMP is still widely used and can be a rich source of information to an attacker when accessible.

4.2.4 Step 4: Assessing Web Servers and Web Applications

Given their widespread deployment, both web servers and the web applications they serve deserve specific attention when assessing network security. For this reason we take the web server installations discovered in step 2 for further investigation. Technologies in web site hosting are often complex, with many configuration

parameters available to the administrator. Further complexities arise through the interaction of numerous web programming and markup languages. User input and output validation are essential to secure web applications from SQL injection and cross site scripting attacks. Scope for error is considerable and the wide popularity of the web and web applications means they have become a favourite attack vector for hackers.

OWASP

Open Web Application Security Project (OWASP) [105] is an international security community created with the intention of improving general web application security through the free and open cooperation of security experts around the world. OWASP provides online security training, mailing lists, research papers, security assessment challenges etc. OWASP provide a 'top ten list' which represents a documented consensus on the ten most significant web application security flaws. This list is a good starting point for web application security assessment and can be used to steer the assessment process towards which flaws to look for.

OWASP Top Ten List (2010 release candidate 1 [106]) lists the following:

1. Injection
2. Cross Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Failure to Restrict URL Access
8. Unvalidated Redirects and Forwards
9. Insecure Cryptographic Storage
10. Insufficient Transport Layer Protection

Testing Mechanisms

Web proxy applications, e.g. Paros [107] and Web Scarab [108] are very useful for web application vulnerability assessment. Such applications sit between browser and Internet, and allow the capture and manipulation of all incoming and outgoing web requests and responses. This allows viewing and modification of parameters which may not be available at the browser interface. Web servers themselves may also have configuration errors or vulnerabilities which could be exploited by attackers. Automatic web server scanning tools, e.g. Nikto [109], can aid in their detection.

4.2.5 Step 5: Assessing Email Services

Given the ubiquitous nature of email, email services deserve some specific attention. Email services located within step 2 ought to be investigated further. Many companies rely on this simple and flexible form of communication to carry out routine business and often open this interface on their network to the outside world. Following are the main email protocols:

Acronym	Name	Port
smtp	Simple Mail Transfer Protocol	25/tcp
pop2	Post Office Protocol version 2	109/tcp
pop3	Post Office Protocol version 3	110/tcp
imap2	Internet Message Access Protocol version 2	143/tcp
smtps	Secure Simple Mail Transfer Protocol (over SSL)	465/tcp
submission	Mail Submission Agent	587/tcp
imaps	Secure Internet Message Access Protocol (over SSL)	993/tcp
pop3s	Secure Post Office Protocol version 3 (over SSL)	995/tcp

Table 4-2 - The Main Email Protocols

A number of applications implementing these protocols have had vulnerabilities. The National Vulnerability Database (NVD) Common Vulnerabilities Exposure (CVE) (described in section 4.3.1) can be searched for weaknesses in such applications once the version number is known. Email server applications often run with elevated privileges so access resulting from any vulnerability could allow an attacker privileged control.

Insecure configuration is again an issue here. The first Internet worm, the 'Morris Worm' of 1988 [110], spread by exploiting debug functionality (which should be disabled) in the sendmail server. Common issues with email servers include the following:

- Providing too much information in banners e.g. software versions.
- Insecure configuration can allow servers to function as 'open relays' allowing spammers to spoof messages from any source to any destination.
- The 'EXPN' and 'VRFY' commands may be used to enumerate local users' email addresses.
- Insufficient login security allowing brute-force password attacks on accounts.

4.3 Network Security Assessment Resources

This section describes a selection of resources available to security assessors and attackers alike.

4.3.1 Common Vulnerability Exposure

Launched in 1999, the Mitre Corporation CVE (Common Vulnerability Exposure Database) [111] is a list of information security vulnerabilities. This allows for a common reference method for describing and cataloguing specific vulnerabilities. Currently 300+ products and services and 150+ organisations are using the system. One tool using CVE is Nessus (section 4.3.3). All vulnerabilities within a tested system are listed in Nessus reports, along with a corresponding CVE number, where available. This can be used to index the official CVE web site to find more information, related vulnerabilities, tools and techniques for exploitation etc.

4.3.2 NMap Network Mapper

Originally created by Gordon Lyon (a.k.a. Fyodor) and released in September 1997, nmap network mapper [11] is a highly versatile, free, open source, cross-platform tool comprising among its many functions, host discovery, port scanning, service version detection and operating system detection. While offering many advanced configuration options, a default scan is very simply initiated and often extremely effective. The list of default ports to be scanned is based on research into the most common Internet services carried out by Fyodor. Its ease of use has also added to the popularity of the tool.

4.3.3 Nessus Vulnerability Scanner

The Nessus scanner [112] can be used to test systems for vulnerabilities, misconfigurations, denial of service weaknesses and information leakage. Both safe

non-obtrusive and potentially dangerous tests can be executed, a fact that must be borne in mind during scan configuration, especially in a live, production environment. While charging companies licence fees, Tenable Network Security distribute a free version for personal use. New plugins are regularly added to test for the latest vulnerabilities. As stated above, Nessus uses CVE to identify vulnerabilities reported where available.

4.3.4 Metasploit

"Metasploit provides useful information to people who perform penetration testing, IDS signature development and exploit research" [113]. The freely downloadable Metasploit Framework aids development and execution of exploit code against a target OS or application. The Metasploit Opcode Database provides information on machine language opcodes for use in exploit writing. The Shellcode Database contains a selection of assembly language payloads for use in, for example, buffer overflow exploits.

4.4 Summary

Following an introduction to the field of network security assessment and some associated issues, a five step assessment methodology was outlined. This process begins with gathering of public information, followed by scanning of the network for available computers and network services. Information gathered in these steps is used to initiate assessments of remote services including web servers and web applications and finally email services. A selection of network security assessment resources was then described.

The following chapter uses processes and resources outlined in this chapter to provide a network security assessment of the DCU campus network, then delivers an analysis of results.

5. Implementing Network Security Assessment

In this chapter we present the results of applying the network security assessment procedures described in the previous chapter to the Dublin City University campus network. This investigation follows on from the issues brought to light by our IDS and described in Chapter 3. Our primary aims were firstly to discover what, if any, attack-enabling information is being leaked and secondly to identify any exploitable weaknesses or vulnerabilities on the network. Based on these findings a set of security-enhancing recommendations is made. All assessment was carried out from an external network.

The chapter is structured as follows: In section 5.1 an overview of the campus network is presented. This is followed in section 5.2 by a review of publicly available information on the network. In section 5.3 internal DNS is examined as a potential source of information leakage. Network scanning techniques are then applied to determine what machines and services are externally accessible in section 5.4. Email and web services are assessed in sections 5.5 and 5.6 respectively. Results of an automated vulnerability scan are presented in section 5.7. The chapter concludes with a summary in section 5.8.

5.1 Network overview

DCU operates two centrally administered campus-wide networks: a traditional LAN and an open wireless network. These networks provide students and staff with Internet connectivity as well as access to a number of internally hosted services. According to network administrators[4], network defence is provided by a Cisco firewall and a Snort-

[4] DCU network administrators were interviewed on 18th January 2008.

based intrusion detection system. This IDS is monitored both internally and externally by a third party who also provide configuration and rule updates. There is no monitoring of internal network traffic. External access to internal services is generally restricted and requires the opening of specific IP addresses and port numbers on the DCU firewall. For external connections to the DCU network, Virtual Private Network (VPN) technology is used to encrypt traffic.

In addition to the campus-wide network, a number of Schools and Faculties run their own networks including DNS and email services. One example is the School of Computing which operates both a switched Ethernet LAN and an open wireless network for its staff and students.

5.2 Public Domain Information

The starting point in a network security assessment is to determine to what extent security-relevant information on the target network is available on the Internet.

5.2.1 Network details

Google search revealed that the DCU network is a subset of the HEAnet [114] autonomous system (AS 1213). DCU is given a CIDR (classless inter-domain routing) prefix of 136.206.0.0/16, meaning that 16 bits are available for host addressing, allowing for up to 65,536 IP addresses on the network. HEAnet is a high speed national academic research network, providing Internet and related services to research and educational organisations in Ireland. HEAnet is connected over 2 x 10Gbps links to the Irish Neutral Exchange (INEX) network [115] which connects Irish ISPs. INEX is connected to networks in Europe over a 2.5Gbps link [116].

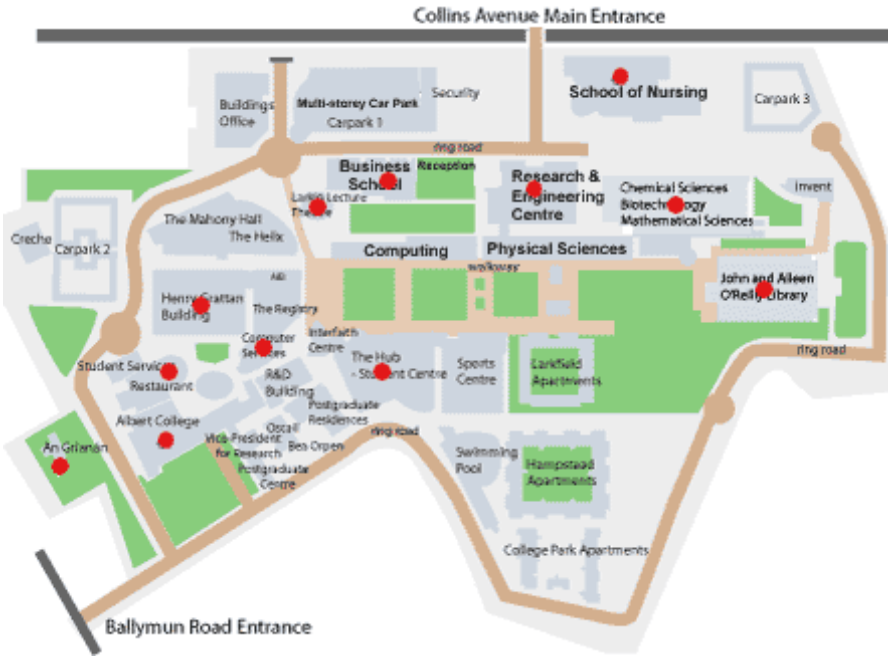

Figure 5.1 (from [117]) - DCU Wireless Network Access Points

Details of the DCU wireless network, as shown in Figure 5.1 (from [117]), are also publicly available e.g. through a Google search for [site:dcu.ie wireless network]. This is an open, unsecured network to which anyone can connect and as noted in Chapter 3, once on a DCU subnet, all DCU subnets are accessible.

5.2.2 Google Groups

A Google Groups search for [dcu.ie network] revealed a thread [118] concerning a user having difficulty connecting a PDA to 'laplan', the campus-wide DCU wireless network. Responses reveal details of the DCU web proxy server and its ports. Another thread [119] gives further proxy server information including the path to the externally available proxy auto-configuration file. This file contains the names of proxy servers, proxy1 and proxy2 along with the port on which they listen. Such information should be on an intranet rather than on the Internet.

Senderbase [120] operates a large database of email and web traffic data ratings which helps ISPs identify spammers. On searching Senderbase for 136.206.0.0/16, 29 host IP addresses are returned (7 of these accompanied an associated DNS hostname), 4 of which are rated as 'good', 25 rated as 'neutral'. A neutral rating indicates that no definitive decision has been made on the host and it may have some negative reports associated with it.

5.2.3 Staff details

A Google search for DCU shows the University's main web site to be http://www.dcu.ie. Available on this web site are all staff details including phone and office numbers, role within DCU and email address. A Google search for [site:dcu.ie inurl:staff administrator] returned details of the NT Administrator, NT Support and Technical Services Managers. Such information may be useful in social engineering attacks and e-mail addresses can be harvested for spamming. While most email addresses take the form *firstname.surname@dcu.ie* others are *firstname.surname@departmentname.dcu.ie* where the latter indicates the presence of an additional, separately administered sub-domain.

5.2.4 Campus Company Details

Searching the online Companies Registration Office database (CRO) [121] along with the main DCU website reveals numerous campus companies and businesses hosted on the DCU network including:

- Campus Residences Ltd.
- Digitary
- The National Centre for Technology in Education (NCTE)
- Investment Intelligence

The subnets associated with such companies should be firewalled from the rest of the DCU network if not done so already.

5.3 DNS Information

A domain's DNS services may be queried for internal IP addresses and machine names. If DNS services are not securely configured an attacker can use leaked information to map a network and its hosts. Below we examine DCU's DNS services for common vulnerabilities.

5.3.1 DNS Interrogation Tools

The ***whois*** tool is used to query DNS databases for information regarding the registrant or assignee of a domain name, IP address or AS number. A whois query can name the network administrator and reveal name servers associated with the domain. For DCU the name servers are (two of which belong to DCU, three are external but cooperate for name resolutions):

- ns1-ext.dcu.ie 136.206.1.1
- ns2-ext.dcu.ie 136.206.1.2
- ns1.tcd.ie 134.226.1.114 (Trinity College Dublin - TCD)
- auth-ns1.ucd.ie 137.43.1.13 (University College Dublin – UCD)
- ns5.univie.ac.at (University of Vienna – Universität Wien)

Additional information returned includes the domain name renewal date of 31 December 2009. If this date were missed domain squatters could potentially register the DCU domain name and charge the organisation to repurchase it.

Using the ***dig*** utility to query DNS also reveals mail exchangers for the DCU zone as:

scan4.dcu.ie (136.206.1.26) scan5.dcu.ie (136.206.1.27)

We assess email services in section 5.5.

5.3.2 Searching for Zone Transfer Weaknesses

According to Kalafut [122] 6.6% of Internet DNS servers are susceptible to external zone transfers. Tools such as 'dig' can be used to check for zone transfer weaknesses, e.g.

```
dig @ns1-ext.dcu.ie dcu.ie axfr
```

A total of 19 machines within the University were found to have port 53 (DNS) open (see section 5.4). Two of these have zone transfer weaknesses and will return all details of their sub-domain, the computing department:

- nuptse.computing.dcu.ie (136.206.11.249)
- nuptse2.computing.dcu.ie (136.206.11.243)

A brief extract from one such zone transfer is presented below:

```
bill@London:~$ dig @nuptse.computing.dcu.ie computing.dcu.ie axfr
computing.dcu.ie.       600     IN      SOA     Mailhost.Computing.DCU.IE.
computing.dcu.ie.       600     IN      MX      10 mailhost.computing.dcu.ie.
computing.dcu.ie.       600     IN      MX      50 scan4.dcu.ie.
computing.dcu.ie.       600     IN      MX      100 bodkin.nuigalway.ie.
computing.dcu.ie.       600     IN      MX      120 mxbackup.esat.net.
computing.dcu.ie.       600     IN      NS      ns2.computing.dcu.ie.
computing.dcu.ie.       600     IN      NS      ns1-ext.dcu.ie.
computing.dcu.ie.       600     IN      NS      mailhost.computing.dcu.ie.
computing.dcu.ie.       600     IN      A       136.206.11.240
alpamayo.computing.dcu.ie. 600  IN      CNAME   mailhost.computing.dcu.ie.
it.computing.dcu.ie.    600     IN      HINFO   "GATEWAY PC" "Solaris 2.6 x86"
ampato.computing.dcu.ie. 600    IN      A       136.206.11.32
aoraki.computing.dcu.ie. 600    IN      A       136.206.11.231
```

The ‘SOA’ (Start of Authority) entry signifies the start of information for a particular zone. Next to be listed are the mail servers ‘MX’ and DNS servers ‘NS’ entries before listing all machines within the zone. ‘A’ entries give the IP address for a particular host name, ‘CNAME’ entries are aliases associated with a host name. ‘HINFO’ entries give hardware details of the machine in question where available. A total of 172 results were returned, as detailed in Table 5.1.

Entry Type	Amount
A	91
CNAME	54
HINFO	18
MX	5
NS	4
Total:	172

Table 5-1 - Zone Transfer Results Distribution

A zone transfer weakness represents a serious flaw as it exposes a complete listing of all hosts and corresponding IP addresses within the domain to an attacker. A denial of service opportunity is also exposed to an attacker who may make multiple zone transfer requests, making the server slow or unresponsive to legitimate requests. Hardware and OS versions leaked may be used to search for vulnerabilities. A number of instances exist of hostnames named after the machine owner which allows for specific targeting of individuals.

5.3.3 Forward DNS Grinding

Forward DNS grinding involves submitting host name guesses as DNS queries and hoping for a hit i.e. the return of a corresponding IP address indicating the host exists within the domain. Identified IP/hostname pairs help in extending the attacker's network map. Forward DNS grinding is similar in approach to a brute-force password cracking uses a dictionary file consisting of common machine names and attempts lookups based on each. Using txdns [123] to forward grind the DCU DNS servers returned 120 machine names for 80 distinct IP addresses (some IP addresses resolve to more than one DNS name). The dictionary files can be continuously augmented with extra information found during the discovery process e.g. figures from Greek mythology are a recurring theme as are mountain names within the computing sub-domain.

5.3.4 Reverse DNS Sweep

Given a domain's IP address range, it is possible to carry out a reverse DNS lookup for each of its IP addresses. ghba [124] is a simple utility which automates this reverse lookup process. For each successful lookup it prints out the corresponding hostname. Using this method 740 successful machine name resolutions were made for the DCU network. The results show a number of naming themes (composers, philosophers etc.) and we see many machines named after their owners. This immediately enables the attacker to target not only specific IP addresses but also specific individuals. Many hostnames were also revealed to be predictable e.g.:

- intel1.physics.dcu.ie , intel2.physics.dcu.ie, intel3.physics.dcu.ie…
- PL86.eeng.dcu.ie, PL87.eeng.dcu.ie, PL88.eeng.dcu.ie…

Such results also allow an attacker to associate departments (physics, electronic engineering etc) with subnets. Knowing which subnets are associated with individual departments again helps the attacker build up a detailed view of overall network

organisation. Further, machines in non-technical departments may be specifically targeted given a possible lack of local security expertise.

5.3.5 DNS Summary

Having obtained a total of 740 distinct hostnames through the ghba reverse DNS sweep, 120 through forward DNS grinding using txdns and 140 through the School of Computing zone transfer, a total of 899 distinct hostnames, associated with 793 distinct IP addresses were gathered.

It is recommended as a security measure that organisations deploy two separate DNS servers: one responding to external requests and another handling internal request which are not to be shared with the public [122]. As the forward grinding and reverse DNS sweeps illustrate, this approach is currently not adopted on the DCU network. Other information provided by DNS tools such as dig and whois is a matter of public record. While DCU's main name servers are not vulnerable to DNS zone transfers, two internal, but externally accessible name servers are vulnerable. This is a serious security weakness and should be fixed.

At this stage we have uncovered much network related information and an overall picture of the network is beginning to emerge.

5.4 Network Scanning

In this section we report our findings on applying a selection of the network scanning techniques detailed in Chapter 4. Scanning was carried out from an external location in order to take the viewpoint of an external attacker.

5.4.1 ICMP Scan

ICMP can be used to discover whether a machine is online or not. Three types of ICMP request are applicable:

- echo
- timestamp
- information

Nmap was used to carry out the ICMP echo scan while the SING (Send ICMP Nasty Garbage) application [125] was used for timestamp and information requests. These scans proved unsuccessful in determining whether machines on the DCU network were online. This is not surprising given many firewalls block ICMP traffic and DCU appears to be a case in point.

5.4.2 TCP Port Scanning

A number of the TCP port scanning techniques described in Chapter 4 were applied against the DCU IP address range to determine which hosts are accessible and which services they are running. Here we report and analyse scan results.

Running a sequential series of scans against DCU's class B network is a time-consuming task. A number of Linux shell scripts were written in order to run scans in parallel, thereby significantly reducing scanning time. Table 5.2 below shows the scans along with the number of hosts which responded to each. This response may be any of open, closed or filtered for at least one port on a particular machine.

A total of 481 distinct IP addresses responded to one or more of the scan types. These results do not reveal which services are running but tell us how many machines are alive, accessible and worth further investigation. Results also reveal which scan types generate the most results. We can see above that ACK scans produce the most responses. Although the numbers of responders to window and maimon scans are identical, different machines responded to each (albeit with considerable overlap).

Scan	Responders	Scan	Responders
ACK	315	Fragmented ACK	8
Window	290	Fragmented SYN	7
Maimon	290	Fragmented Connect	5
Connect	271		
SYN	218	SYN Source 20	64
Null	179	SYN Source 53	92
Xmas	179	SYN Source 80	64
FIN	164	SYN Source 88	115

Table 5-2 - Numbers of Responding Hosts per Scan Type

There were no cases of machines responding to fragmented scans but not to their non-fragmented equivalent. Thus fragmenting packets does not evade firewalls (fragmentation is often used as a firewall evasion technique) on this network.

Source Port Spoofing

As a firewall evasion mechanism, altering the source port on scan can prove a worthwhile undertaking. The chosen ports for our experimentation were 20 (FTP), 53 (DNS), 80 (HTTP) and 88 (Kerberos). Each was chosen because it is often allowed through firewalls given the ubiquitous nature of the underlying service. These scans did provide some supplementary results. Scans with a source port of Kerberos were allowed past the firewall a significant number of times (47) where no other scan types could produce any response (it is unclear why the overall numbers of results obtained

by these scans were significantly lower than for SYN scans with default source ports). From the defensive perspective we should only open the minimum number of ports in the firewall, necessary for normal network functioning. Where possible only traffic associated with internally initiated connections should be allowed.

Scan Results by Host Machine

As mentioned, port scan results were generated for 481 responding machines in total and results for each were aggregated into a spreadsheet for analysis. For each machine a matrix was generated with ports as rows and scan type as columns.

Analysis included calculating the total number of distinct IP/port responses across all machines and scan types. Results for the top responding ports are graphed below. Results for a particular scan on a particular port may be one of 'open', 'closed', 'filtered' and 'open|filtered'. Below we examine the results for both open and filtered ports.

Open Ports

This first set of results covers the reachable and responding ports to which it is possible to connect from outside the DCU network. A total of 141 distinct ports were listed as open on at least one machine across the DCU range. 209 distinct machines reported having at least one port open leaving the remaining 272 machines generating one or more responses other than 'open'. It is unclear why the latter machines should be externally accessible if they offer no accessible services. The top 15 most common of these open ports are listed in Figure 5.2.

A number of these protocols can often be interrogated for further information by attackers so should be carefully configured to be externally inaccessible and/or to give out only a minimum of information. The large number of open ports, totaling 863

across all accessible machines, provides the attacker with an excellent starting point for further investigation and possible break-in. Whether or not each of these ports should be open to external connection should be considered and where left open, they should be sufficiently protected. DCU policy should be to close ports after some period in the absence of a request (by the owner) to have them kept open. Such a policy would ensure services no longer required are closed down rather than remaining indefinitely accessible and running out-of-date software.

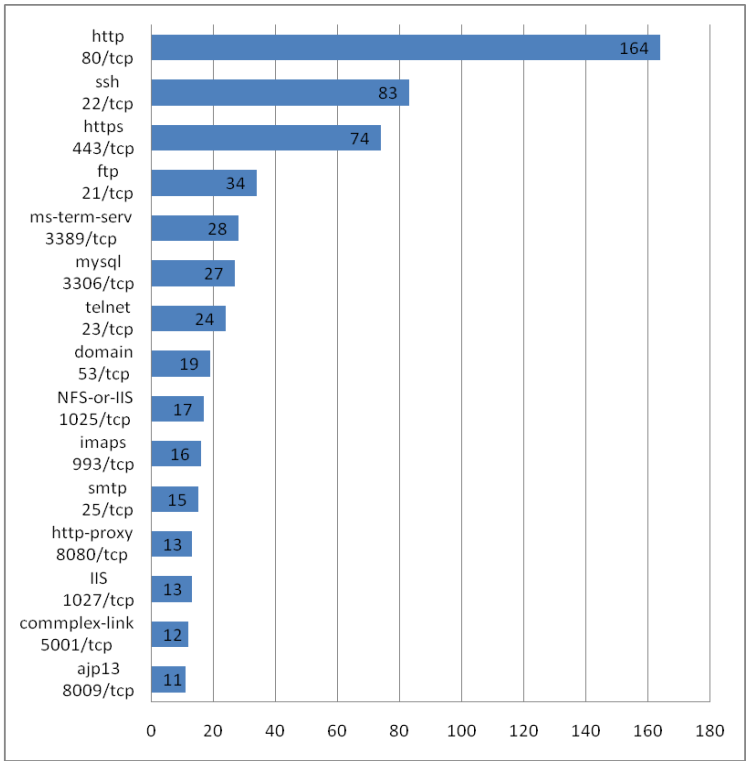

Figure 5.2 - Common ‘open’ Ports

It was discovered that the optimal scan type combination for discovering open ports is the SYN plus connect scans. These two account for 855 of the 863 open ports, leaving only 8 (0.92%) found by other scan types.

This combination of scans, with the addition of SYN source 53 spoof scan will receive an open response from all available systems on the network as follows:

Scan Type	Number of (distinct) Hosts
1. connect	190
2. SYN	17
3. SYN Source 53	2
Total:	209

Table 5-3 - Optimal Scan Combination

As seen in Table 5.3, the connect scan results in an 'open' response from at least one port on 190 different hosts. A subsequent SYN scan provides 17 extra hosts which were not yet discovered by connect. The two remaining hosts were discovered through a SYN scan with source port spoofed to 53, thereby covering all hosts which have at least one 'open' port.

Interesting Ports

An interesting result is the 164 web servers (port 80) accessible to the outside world running on the DCU campus. As mentioned in Chapter 4, web servers and the applications they run are often vulnerable to attack. It is quite possible that a number of these servers are no longer used, maintained or patched for the latest security threats. These are not 'virtual', centrally maintained and hosted servers but individual hardware and software installations. See Section 5.6 for more on web server interrogation.

Protocols known to have security issues associated with them were also revealed. An example is the telnet protocol (TCP port 23) [126], which is running on 24 machines. Telnet transmits unencrypted data including passwords from client to server and should be replaced by Secure Shell (SSH) [127]. Another example is FTP (TCP ports 20 and 21) [26], which is running on 34 machines, and also transmits unencrypted data including passwords and should be replaced with SFTP (FTP over SSH). SSH (and

SFTP) run on port 23, encrypting logins and passwords along with data while otherwise providing the same functionality as their 'insecure' counterparts.

Ports 25 (SMTP email servers) [128] and 53 (DNS domain name servers) are required to be open within most organisations. We looked at DNS vulnerabilities in section 5.3 and in section 5.5 we assess the security of the email servers.

Interesting ports for attackers can also be those which produce a low but positive number of responses. These may be forgotten, unpopular, rarely used or updated technologies which might be vulnerable to attack. One such example is TCP port 6400 (crystalreports) which responded as open on one machine only.

Filtered Ports

Nmap produces different results from different scan types. The FIN, Maimon, Null and Xmas scans on occasion produce the result 'open|filtered' as the scan cannot determine whether the port is actually open or if it is being filtered by a firewall. On the other hand, the ACK, Connect and SYN scans can deliver the more specific 'filtered' result since they can determine that a port is filtered by a firewall and could not possibly be open. When results are compared, the 'filtered' results generally correspond to 'open|filtered' results generated by the other scans and therefore our results for both are amalgamated into one spreadsheet. A total of 137 distinct ports were listed as filtered or open|filtered across all machines on the DCU range. The top 15 most common of these filtered ports are listed in Figure 5.3.

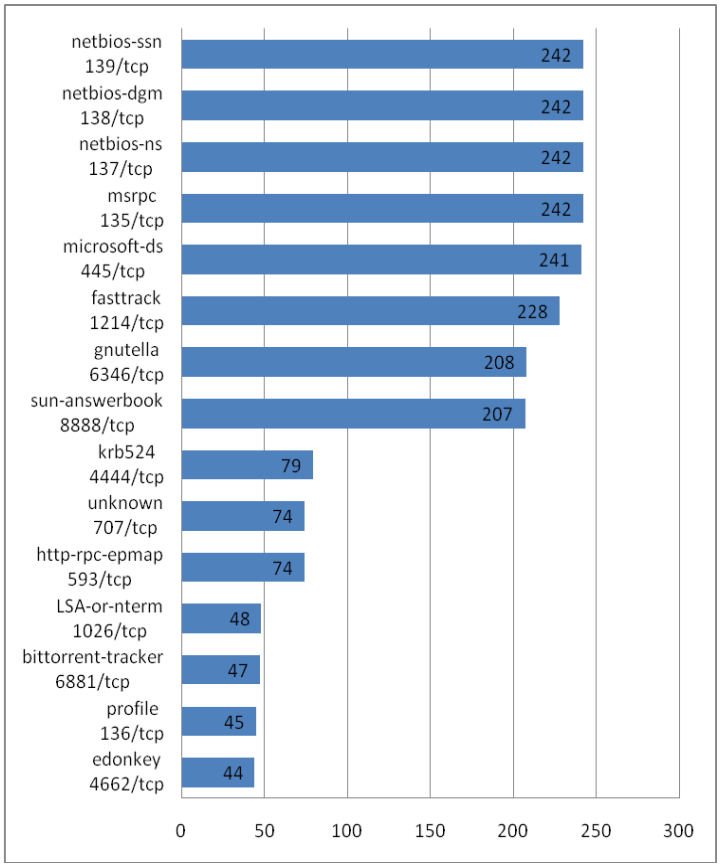

Figure 5.3 - Common 'filtered' Ports

This set of results reveals the protocols which are specifically blocked by firewalls, whether these are on particular machines, or across the entire range. The Microsoft Windows service ports figure strongly in these results, since they could be dangerous if open to the Internet. Ports 4662 (edonkey), 6346 (gnutella) and 6881 (bittorrent-tracker) which represent file sharing protocols known for high network resource consumption, thus reducing network throughput for other users are blocked.

5.4.3 OS Guessing

An operating system guess scan of the responding machines was carried out. Results were obtained only for a number of them. Guesses were provided for 224 machines in

total. Most were not definitive and listed a number of possibilities. OS guesses were as follows:

Guess	# Hosts	Guess	# Hosts
Multiple Guesses	66	OpenBSD	4
Linux	58	FreeBSD	3
Cisco	50	Novell	2
Microsoft	28	Apple	1
Solaris	11	Tandberg Embedded	1
	Total:	224	

Table 5-4 - Operating System Guesses

In Table 5.4, 'Multiple Guesses' indicates nmap was uncertain of the OS in question. From these results the DCU network contains twice as many Linux as Microsoft devices. Cisco operating systems also make up a significant proportion of remotely accessible hosts with Solaris and other Unix based operating systems making up most of the remainder. Cisco OSes run on routers and switches. Some may offer a web interface for configuration. On checking for web interfaces to the Cisco systems, one was found to be open to the public (Cisco Catalyst 2940 switch - 136.206.178.11), allowing access without a password to statistics and configuration updating. This is extremely dangerous as the interface allows an attacker to alter the switch's configuration, revert to factory settings, allow telnet and SNMP access etc.

5.5 Email Server Assessment

DCU's email servers, revealed as open port 25 by nmap, were tested with results described in this section. None were found to act as open relays but other issues did arise, as described below.

5.5.1 Version Fingerprinting

On connecting to port 25, information is often returned in the form of a 'banner'. By default this may contain information regarding the specific email server version. It is possible to telnet to port 25 on ten of the fifteen DCU machines running email servers. Five report to be using Postfix, but do not specify which version, while two do not report the software which they are running at all. Two mention Microsoft ESMPT mail service 6.0.3790.3959 (Exchange 2003). One reports Sendmail 8.13.8/8.12.5.

Scanning applications such as smtpscan [129] can reveal information about SMTP servers when banners are hidden. smtpscan tells us that the above mentioned sendmail installation is actually version 8.11.6. On searching the National Vulnerability Database, at least three vulnerabilities associated with that sendmail version are reported. Firstly the medium severity vulnerability CVE-2009-1490 describes a heap-based overflow allowing attackers to cause denial of service and possibly execute arbitrary code. Secondly the high severity CVE-2002-1337 covers a buffer overflow allowing arbitrary code execution. Finally the high severity CVE-2003-0694 allows both arbitrary code execution and denial of service. Further investigation is necessary to determine whether the actual sendmail version is that reported by the banner or the one reported by smtpscan. None of the SMTP server applications for which results were generated were running the latest version of their particular implementation.

5.5.2 Local User Enumeration

On connecting to an open SMTP server, for example through telnet, it is possible to enter commands. The 'EXPN' and 'VRFY' commands can be used to enumerate local users and should be disabled. Of the ten servers to which we can connect using telnet, eight accepted commands. Six make available the 'VRFY' command while only one, mailhost.computing.dcu.ie, allows both 'EXPN' and 'VRFY'. 'EXPN' was not available on any other SMTP servers. Both 'VRFY' and 'EXPN' can be exploited to

efficiently brute force email addresses. An example of an application which can enumerate mail server users using the 'EXPN' and 'VRFY' commands is smtp-user-enum [130].

5.5.3 Summary

User enumeration on the majority of DCU SMTP servers is achievable, with a number of these allowing the 'VRFY' command, one of which also allows 'EXPN'. These commands allow brute force enumeration of users. Fingerprinting and banner grabbing show that SMTP servers should be upgraded to their latest versions.

5.6 Web Server and Application Assessment

A crawl of the 164 HTTP servers was made, followed by application of the Nikto web server vulnerability assessment tool for further automated inspection.

As we see in Figure 5.4, the majority of available HTTP servers are spread over the 'low' and 'high' range of IP addresses and have no obvious associated department subnet based on hostnames. The remainder comprise of servers within the redbrick (a student run computer user group in the school of computing), computing, electronic engineering (eeng), physics and cse domains. Redbrick's subnet runs the largest obvious grouping of HTTP servers with a total of 22, followed by the School of Computing with 15.

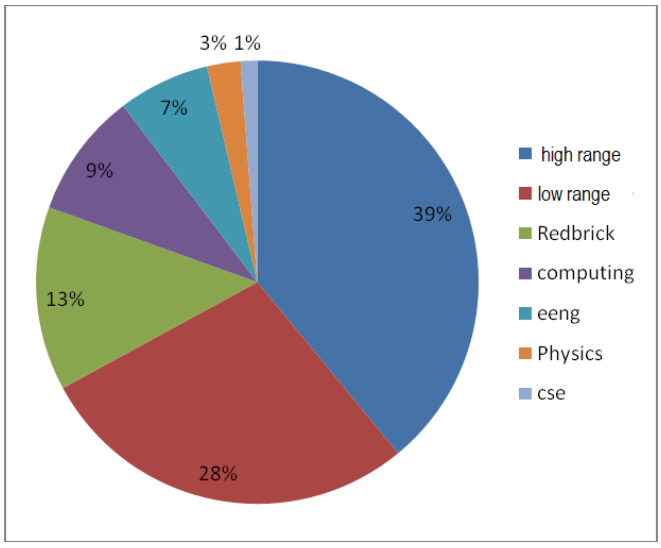

Figure 5.4 - HTTP Server Subnets

5.6.1 Manual Web Viewing

The initial step in any web application assessment is to download and inspect each of the front webpages of the 164 web servers discovered earlier. On carrying out this task we discover one Cisco Catalyst 2940 switch whose administration page is available over the Internet for viewing and configuration without requiring a password, as mentioned in section 5.4.3. Available options include monitoring of statistics, configuration of ports, restarting the device, setting up telnet and SNMP access, and restoring default settings. Two printers are also accessible over the Internet, allowing configuration updates, setting up email alerts and restarting the printer to apply these settings. 11 servers require usernames and passwords but do not use encryption (such as SSL) for data in transit. Therefore usernames and passwords are sent in clear text and can be viewed by anybody sniffing network traffic. Three servers display a directory listing in text based format, allowing the user to see files and folders which exist on the server. Another server provides links to files for download. On following the link to one particular folder, an error message is returned including username 'root' and an associated password. 20 servers, through error messages upon connection attempts or otherwise, provide specific details of web server software in use. Five servers display the default test web page for their particular server application, three of which further provide details of the operating system in use. This may also indicate that a server/site was installed and subsequently abandoned.

5.6.2 Nikto Vulnerability Scanning

The Nikto web server vulnerability scanning tool was run against the 164 HTTP servers. Following are a subset of interesting results obtained.

Vulnerabilities and Outdated Software

23 machines were found to be vulnerable to cross site scripting (XSS) vulnerabilities. 6 servers are listed as running formmail.pl which has numerous remote vulnerabilities

associated with it including file access, information disclosure and email abuse. Outdated versions of Apache, IIS and Oracle Application Server are running on 63, 6 and 1 system(s) respectively. PHP, mod_ssl, OpenSSL, perl and mod_perl versions are outdated on 32, 26, 26, 8 and 10 systems respectively.

Information Leakage

98 servers deliver software version numbers through publicly available banners. 76 systems have a number of accessible files listed as 'interesting' meaning they may contain some information which could be useful to attackers.

Insecure Configurations

54 servers allow directory indexing. 61 machines are reported as supporting the HTTP TRACE command meaning they are vulnerable to Cross Site Trace (XST) attacks. Nine servers allow the 'PUT' and 'DELETE' HTTP commands which permit uploading and deletion of files respectively. Of these nine, six also allow the 'MOVE' command which allows moving of files within the server.

5.7 Automated Network Vulnerability Scanning

The Nessus network vulnerability scanning tool was run against hosts discovered through scanning the DCU network in section 5.4. As mentioned in Chapter 4, unsafe scans were disabled to avoid possible harm to targets. Output provided us with the following information.

5.7.1 Analysis

Machines which contain security holes or give rise to warnings or security notes are reported on a per host basis within Nessus reports. Holes are more serious than warnings which are more serious than notes. Possible fixes are also presented along

with graphs which are provided to aid result interpretation. Table 5.5 provides a summary of the responses generated by Nessus

Hosts which were alive and responding during test	296
Number of security holes found	22
Number of security warnings found	137
Number of security notes found	2542

Table 5-5 - Nessus Results Summary

As seen in Figure 5.5, the most common source of security holes are found in FTP and SSH.

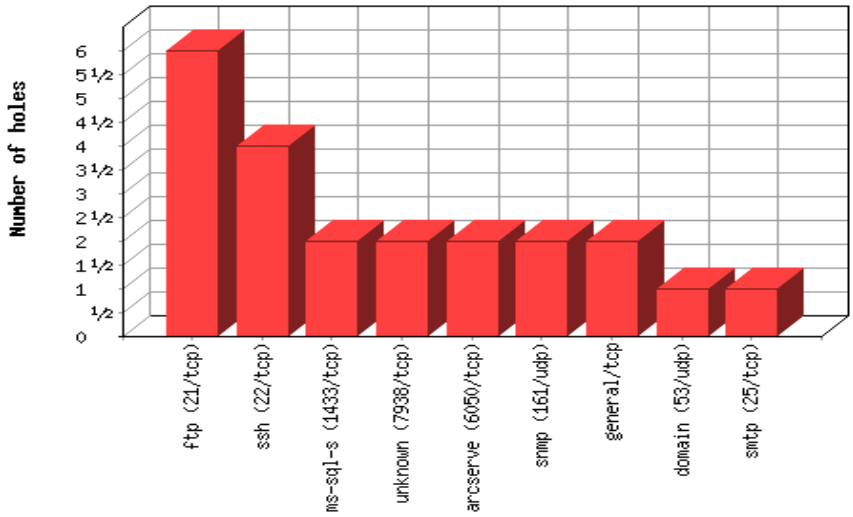

Figure 5.5 - Top 9 Most Vulnerable Services on the Network

Sample security holes, warnings and notes are discussed in the following sections.

Security Holes

Threats with the highest severity, the 'security hole' (also known as 'vulnerability' by Nessus) have a risk factor of critical or high. These could result in compromise of the

machine or service in question. Security holes usually have a CVE number associated with them and possibly published exploits. Table 5.6 lists the systems which have the highest severity level security 'holes'.

136.206.1.27	136.206.161.116	136.206.178.29	eaccess.rince.ie
136.206.19.19	136.206.161.181	136.206.222.20	ebt.cse.dcu.ie
136.206.19.190	136.206.161.234	nuptse.computing.dcu.ie	mail.insero.ie
136.206.160.17	136.206.176.12	pisang.computing.dcu.ie	scan4.dcu.ie
136.206.160.26	136.206.178.4	vinson.computing.dcu.ie	

Table 5-6 - Machines with Security Holes

One example of a security hole which exists on two machines on the network is CVE-2008-0166, which describes an issue with weak SSH host keys. The random number generator associated with these cryptographic keys is predictable and exploits exist (e.g. [131]) which can pre-calculate possible keys to decrease the time necessary to carry out a brute force attack for key retrieval.

Another example security hole exists in three Internet facing FTP servers. CVE-1999-0017 allows an attacker to connect to third party hosts using the PORT command. The attacker can use this vulnerability to bypass a firewall or to carry out an FTP bounce scan, as described in section 4.2.2.3.2, where the server is manipulated to scan other hosts which see the scan as originating from the vulnerable FTP server.

Security Warnings

A 'security warning' can have a high, medium or low associated risk factor. These are not considered as dangerous as the aforementioned 'holes' but some nonetheless could result in exposure of sensitive data, remote access etc.

An example which exists on 14 Internet facing DCU hosts is CVE-2008-1483. This issue could allow hijacking of X11 user sessions due to a server port binding error. Versions of SSH prior to 5.0 are vulnerable and upgrading is advisable.

As previously mentioned in section 4.2.3, the SNMP protocol is known to have a number of associated issues while providing a rich source of information. CVE-1999-0517 describes a predictable community string[5] taking its default value of 'public'. Six hosts may be vulnerable to this issue allowing attackers to view and modify system information.

Security Notes

Finally, 'security notes' provide information which may be indirectly useful to an attacker or results which are inconclusive and require further manual investigation. Also known as informational or low risk threats, as seen in Figure 5.6, these constitute 94% of results obtained. This information may take the form of identified software and hardware versions, hostname resolutions, services accessible etc. Usually these are not serious security issues but nonetheless can denote exposure of excessive system information.

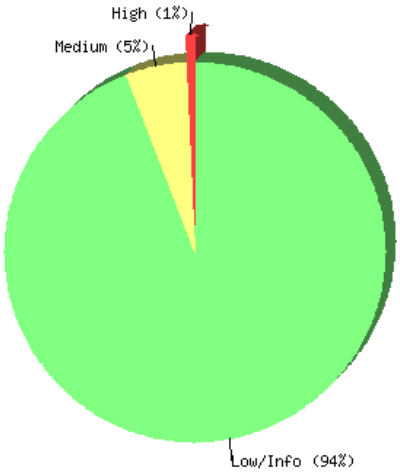

Figure 5.6 - Security Risk Severity Comparison

[5] SNMP passwords are also known as community strings.

5.7.2 Example - 136.206.160.6 (mail.insero.ie)

This host is reported as having one security warning and two security holes. These particular holes could be leveraged to provide the attacker with significant reward. Firstly a high risk rated vulnerability in SQL Server (port 1433) is present allowing arbitrary code execution through a stored procedure which does not check parameters sufficiently (CVE-2008-5416). Arbitrary code execution opens the possibility of spawning a shell on the server and thereby taking full control of this system at the privilege level held by the vulnerable application. Following exploit of this issue, an attacker could take advantage of various memory corruption flaws in this SQL Server version (CVE-2008-0085, CVE-2008-0086, CVE-2008-0106, CVE-2008-0107), which allow privilege escalation on the target host. The attacker now has control with elevated privileges. Neither of these high risk rated issues are associated with any other hosts on the network.

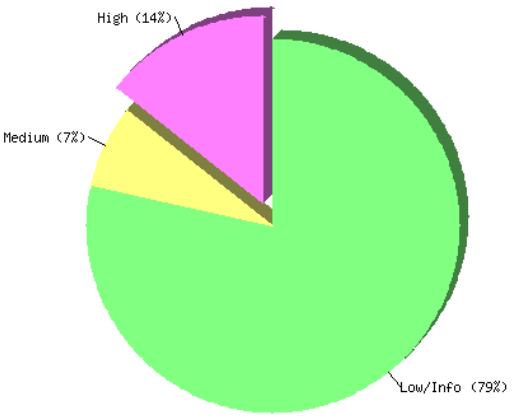

Figure 5.7 - Severity of Security Risks for mail.insero.ie

A medium risk rating warning on this host makes it possible for a remote attacker to gain access to the machine through the functioning version of Remote Desktop Protocol Server (Terminal Services) which is vulnerable to a man in the middle attack (CVE-2005-1794). An attacker could exploit this flaw to decrypt client – server communications and obtain sensitive information including passwords. A total of 30 machines on the network are vulnerable to this particular issue. 11 informational messages are also provided for this host.

5.8 Summary

Having discovered foreign traffic and attempted malware infections on the DCU network through deployment of our IDS, it was decided that a more thorough network security assessment was warranted. Following the five step methodology presented in Chapter 4, we uncovered and reported further security weaknesses within the network. Publically available information related to the network was gathered before DNS was used to source further information related to internal hosts. Next the network was scanned for hosts and services within the associated IP address range. Having located many services across the network, email and web technologies were given specific attention and automated network vulnerability scanning was carried out. A number of security issues, some serious, were uncovered and presented.

6. Conclusions and Further Work

In this final chapter, we sum up this thesis's contribution, suggest future work and make recommendations for improving overall network security.

6.1 IDS Contribution

Below we sum up the work described in Chapter 3 on building a web enabled hybrid IDS and make some suggestions for how it might be improved.

6.1.1 A Web-Enabled Hybrid IDS

The research question we set out to answer in Chapter 1 was whether we could bring much-needed IDS technology to today's "security unconscious" Internet users. Motivations for and the design and implementation of a web-enabled, hybrid IDS aimed at meeting this need were described in Chapter 3. Our IDS has the following features:

- Remotely administered and configured in order to relieve users of the technical complexity involved in typical IDS systems.

- Intrusion analysis is offloaded to a server making it suitable for deployment on mobile devices of limited processing power.

- Feedback is provided to users on the behaviour of their machine as seen by other machines on the network in order to detect malware infections unknown to the user. All agents cooperate in monitoring each other.

- Associated network traffic is increased in volume only in the face of a possible attack through the use of a watchlist.

- Web enabled functioning allows users to freely roam the Internet while receiving constant feedback on the threats they face.

- Exports a web interface so users receive feedback through their web browser rather than having to learn a new application.

- The system is simple to install compared with other IDS, particularly the client agent, which once installed, runs transparently.

- Employs open source Snort for detailed intrusion analysis.

- Demonstrates to users that they are targeted and in so doing, raises their security awareness.

To test our IDS we deployed it on the DCU network where it revealed a number of issues, the most serious of which being attempted Slammer worm infections from foreign IP addresses.

6.1.2 Further Work

A number of possible enhancements not present in the current prototype due to time and/or resource constraints are presented below.

Agent

- **Porting** the agent to a number of other platforms is required since a number of operating systems will coexist on many networks. For the prototype we

concentrated on the most common operating system, Microsoft Windows. Using a language which can be compiled for a large number of OSes, for example, Java, could be an answer to this issue. A Java framework called Java Agent Development Framework (JADE) [132] is intended for development of peer-to-peer agent-based applications and could prove beneficial for future versions.

- Addition of **further processing** at the client could be a possible extension. In design and development the focus was on simplicity, speed and efficiency of the software to make it suitable for devices with limited processing capacity. However on clients with sufficient resources it may be possible to enable extensions such as packet inspection without negatively impacting user experience.

- **Spoofing** is an issue which could potentially hinder the performance of this system. For example, if machine 'Attacker' on the network was to scan another machine 'Scanned', but spoofs the sources of all packets to look as though coming from 'Innocent', then 'Scanned' will report suspicious activity as originating from 'Innocent'. This issue was considered through development of our prototype and one possible solution is to record outgoing traffic at each client for a period of time in a first-in, first-out queue which would be checked when required. Thus, the agent on 'Innocent' would be queried by the server to ensure that the events attributed to it were actually sent from there. This approach would require our agent to be running on all involved clients.

- Addition of **functions to block traffic** would transform this IDS into an IPS. This functionality was considered as an enhancement and modules were developed to allow agents to interact with Windows Firewall on the client machine, blocking certain ports, source addresses etc. The spoofing problem must be solved however before action can be taken against particular hosts.

- **Ruleset profiles** could allow the system to take a stricter approach to events while the user is located within an environment which is considered less secure. For example, Windows folder sharing might be enabled on a home network, but not on a public wireless network. Using profiles, the ruleset could be configured automatically or manually based on current location. Currently this can be implemented manually by having the server update the rules each time the client moves to another network, but this is ultimately impractical.

Server

- Although the server was not the area of focus for the author, its administration interface could benefit from more streamlined visuals, possibly including **Web 2.0 resources**. AJAX is already in use for automatic screen updates, but further enhancements could improve readability. Visualisations including graphs could allow the administrator to quickly interrogate traffic patterns.

- **Aggregating similar data** could reduce the amount of storage space required, but more importantly, aggregating similar data for presentation to the administrator would provide both a more useable interface and simpler data manageability. Currently a scan of 1000 ports on one machine could cause up to 1000 reports to delivered to the administrator and 1000 rows of data displayed on the interface. Server functionality could be enhanced to analyse these events for similarities, displaying the group as one row. The user could click for further details of each individual probe.

Agent-Server Communication

- **Encryption** of data in transit is a vital factor in this system along with **authentication** between client and server. For deployment over the Internet, the current system would not be secure since an attacker could snoop and even

spoof alert or configuration packets to either client or server. Since the XML based format used for configuration updates and alerts is straight forward, an attacker could capture a number of samples before generating valid XML to manipulate the client or server.

- Testing for **scalability** would be useful in discovering how many agents can function together before the server becomes overwhelmed. This will occur as the web server and Snort installation must handle an increasing number of report submissions as the number of agents grows. Database and web servers could be separated across machines to alleviate the problem. Rate-limiting agent submissions could also be implemented.

- Aggregating **similar events** on the client would reduce bandwidth requirements. For example, currently a scan of 1000 ports may result in up to 1000 reports being sent to the server. If the client could buffer data for a period and interrogate it before sending, similarities could be exploited in order to reduce network traffic.

6.2 Network Security Recommendations

In light of the issues uncovered in Chapters 3 and 5, we make the following recommendations for improving network security.

1. Implement internal firewalling to prevent traffic reaching hosts from untrusted wireless networks.

2. Implement internal firewalling to protect campus company hosts and subnets.

3. Keep internal network information private by only making public that which must be shared with the public. An internal intranet should be used for network

setup queries rather than having related newsgroup threads subsequently publically accessible.

4. Remove zone transfer weaknesses from machines identified as being vulnerable.

5. Separate DNS servers should be used for internal and external resolutions. This will keep internal hostnames and network information private.

6. There are 164 externally accessible web servers. This number seems high. DCU policy should be to close externally accessible ports in the absence of a written e.g. annual request to have them kept open. Implementing such a timeout would ensure that services no longer required are not left open indefinitely. Hosting web sites centrally should be investigated as a means to provide those who need them with web servers. Trusting secure administration to individual and unqualified users is not working.

7. Insecure protocols should be disallowed, e.g. telnet and FTP as there is simply no need for them given the existence of secure equivalents.

8. Remove access to the Cisco switch identified in Chapter 5. Password protect it and make it internally accessible only. Also, remove external access to the printers also identified in Chapter 5.

9. Disable ‘EXPN’ and ‘VRFY’ commands on all email servers that currently support them.

10. Disable ‘TRACE’, ‘PUT’ and ‘DELETE’ on all web servers that currently support them.

11. A general recommendation is to upgrade all internet facing services to their latest version. There is a significant issue in the number of outdated software installations running on the Internet.

12. The above recommendations are based solely on the results of external scans. Further assessment should be conducted from inside the DCU network.

13. It should be easier to report security issues than is currently the case. A dedicated email alias should be created to which users can submit queries and faults identified.

References

[1] Miniwatts Marketing Group. Internet World Stats [Online]. http://www.internetworldstats.com/stats.htm [accessed: 2009, 30 Nov]

[2] Pew Research Center. (2009, April 19). Pew internet and american life - spring tracking survey 2009. [accessed: 2009, 11 Nov]. Available: http://pewinternet.org/~/media/Files/Data%20Sets/2009/April_2009_Economy_Crosstab.zip

[3] Commission for Communications Regulation. (2009, 30 Oct). ComReg research report analyses internet connectivity in Ireland. [accessed: 2009, 14 Nov]. Available: http://www.comreg.ie/_fileupload/publications/PR301009.pdf

[4] European Commission. Eurostat [Online]. http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/ [accessed: 2009, 30 Nov]

[5] eMarketer Inc. eMarketer [Online]. http://www.emarketer.com/ [accessed: 2009, 30 Nov]

[6] Commission of the European Communities. (2009, 28 Oct). A public-private partnership on the future internet. [accessed: 2009, 14 Nov]. Available: http://ec.europa.eu/information_society/activities/foi/library/fi-communication_en.pdf

[7] Kaspersky Labs. Malware Evolution 2006: Executive Summary [Online]. http://www.kaspersky.com/malware_evolution_2006_summary [accessed: 2009, 30 Nov]

[8] O. O'Connor. (2008, May). 2nd ISSA/UCD Irish cybercrime survey. [accessed: 2009, 12 Nov]. Available: http://www.issaireland.org/2nd%20ISSA%20UCD%20Irish%20Cybercrime%20Survey%20-%20Results%2017DEC08.pdf

[9] M. Bishop., *Computer Security: Art and Science*, Addison-Wesley Professional, 2002, Dec.

[10] USA CERT. USA Computer Emergency Response Team (CERT) Statistics [Online]. http://www.cert.org/stats/ [accessed: 2009, 12 Nov]

[11] G. Lyon. (1997, Sept). Nmap. [accessed: 2009, 02 Oct]. Available: http://nmap.org/

[12] Metasploit LLC. (2008, 19 Nov). Metasploit framework. [accessed: 2009, 05 Aug]. Available: http://www.metasploit.com/framework/

[13] CERT - Carnegie Mellon University. (2006), Vulnerability discovery: Bridging the gap between analysis and engineering. [accessed: 2009, 15 Nov]. Available: www.cert.org/archive/pdf/CERTCC_Vulnerability_Discovery.pdf

[14] Rootkit.com. Rootkit.com [Online]. http://www.rootkit.com/ [accessed: 2009, 30 Nov]

[15] Caida. The Spread of the Sapphire/Slammer Worm [Online]. http://www.caida.org/publications/papers/2003/sapphire/sapphire.html [accessed: 2009, 29 Nov]

[16] R. Magee. (2007, 27 Aug). 802.ids - wireless intrusion detection. *DCU.*

[17] T. d'Otreppe. Aircrack-ng. [accessed: 2009, 30 Sept]. Available: http://www.aircrack-ng.org/

[18] Facebook. Facebook Press Room: Statistics [Online]. http://www.facebook.com/press/info.php?statistics [accessed: 2009, 12 Nov]

[19] S. Egelman, L. F. Cranor and J. Hong, "You've been warned: An empirical study of the effectiveness of web browser phishing warnings," in *CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1065-1074.

[20] Organisation for Economic Co-Operation and Development (OECD). (2007, 17-18 June). Malicious software (malware): A security threat to the internet economy. [accessed: 2009, 29 Nov]. Available: www.oecd.org/dataoecd/53/34/40724457.pdf

[21] M. Roesch. (1998), Snort. [accessed: 2009, 09 Aug]. Available: http://www.snort.org/

[22] G. F. Coulouris, J. Dollimore and T. Kindberg, *Distributed Systems : Concepts and Design / George Coulouris, Jean Dollimore, Tim Kindberg, 4th Ed.*, Addison-Wesley, Wokingham; Sydney, 2005.

[23] U.S. Department of Defense (DoD). Defense Advanced Research Projects Agency (DARPA) [Online]. http://www.darpa.mil/about.html [accessed: 2009, 05 Oct]

[24] B. M. Leiner, et al, "A brief history of the internet," *SIGCOMM Comput.Commun.Rev.*, vol. 39, pp. 22-31, 2009, Oct.

[25] P. Mockapetris, "RFC 1035 - Domain Names - Implementation and Specification (DNS)," 1987, Nov.

[26] J. Postel and J. Reynolds, "RFC 959 - File Transfer Protocol (FTP)," 1985, Oct.

[27] R. Fielding, et al, "RFC 2616 - Hypertext Transfer Protocol (HTTP)," 1999, June.

[28] R. Thurlow. (2009, May). RFC 5531 - remote procedure call (RPC). [accessed: 2009, 05 Oct]. Available: http://tools.ietf.org/html/rfc5531

[29] J. Case, et al, "RFC 1157 - Simple Network Management Protocol (SNMP)," 1990, May.

[30] University of Southern California. (1981, Sept). RFC 791 - internet protocol (IP). [accessed: 2009, 05 Oct]. Available: http://tools.ietf.org/html/rfc791

[31] Y. Rekhter, T. Li and S. Hares, "RFC 4271 - A Border Gateway Protocol 4 (BGP-4)," 2006, Jan.

[32] IETF Network Working Group, "RFC 792 - Internet Control Message Protocol (ICMP)," 1981, Sept.

[33] D. C. Plummer, "RFC 826 - Address Resolution Protocol (ARP)," 1982, Nov.

[34] ISO/IEC, "ISO 7498 - Open Systems Interconnection Reference Model," 1977.

[35] Wikimedia Foundation Inc. Wikipedia - TCP/IP Model [Online]. http://en.wikipedia.org/wiki/TCP/IP_modelAvailable: http://upload.wikimedia.org/wikipedia/commons/3/3b/UDP_encapsulation.svg [accessed: 2009, 18 Nov]

[36] J. Postel, "RFC 768 - User Datagram Protocol (UDP)," 1980, Aug.

[37] W. John, S. Tafvelin and T. Olovsson, "Trends and differences in connection-behavior within classes of internet backbone traffic," in *9th International Conference, PAM*, 2008.

[38] Internet Corporation for Assigned Names and Numbers (ICANNN). Internet Assigned Numbers Authority (IANA) [Online]. http://www.iana.org/ [accessed: 2009, 17 Aug]

[39] DARPA Information Processing Techniques Office, "RFC 793 - Transmission Control Protocol (TCP)," 1981, Sept.

[40] A. Tanenbaum., *Computer Networks,* Prentice Hall Professional Technical Reference, 2002.

[41] K. S. Killourhy, R. A. Maxion and K. M. C. Tan. A defense-centric taxonomy based on attack manifestations. Presented at DSN '04: Proceedings of the 2004 International Conference on Dependable Systems and Networks.

[42] G. Álvarez and S. Petrovic, "A new taxonomy of Web attacks suitable for efficient encoding," *Comput. Secur.,* vol. 22, pp. 435, 2003.

[43] J. Howard and T. Longstaff, "A Common Language for Computer Security Incidents," 1998.

[44] M. S. Gaderlab, A. A. El Kalam and Y. Deswarte, "Defining categories to select representative attack test-cases," in *QoP '07: Proceedings of the 2007 ACM Workshop on Quality of Protection,* 2007, pp. 40-42.

[45] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Comput. Secur.,* vol. 24, pp. 31, 2005.

[46] M. Moser, et al. (2007, 6 Mar). Backtrack live CD. [accessed: 2009, 19 Oct]. Available: http://www.remote-exploit.org/backtrack.html

[47] BBC News. US cracks 'biggest ID fraud case' [Online]. http://news.bbc.co.uk/2/hi/business/7544083.stm [accessed: 2009, 19 Nov]

[48] Boston Globe. Cost of data breach at TJX soars to $256 [Online]. http://www.boston.com/business/globe/articles/2007/08/15/cost_of_data_breach_at_tjx_soars_to_256m/

[49] InformationWeek. The TJX Effect - Details of the largest breach of customer data are starting to come to light [Online]. http://www.informationweek.com/story/showArticle.jhtml?articleID=201400171 [accessed: 2009, 18 Nov]

[50] Wired.com. TJX Hacker Was Awash in Cash; His Penniless Coder Faces Prison [Online]. http://www.wired.com/threatlevel/2009/06/watt/ [accessed: 2009, 19 Nov]

[51] I. Mokube and M. Adams, "Honeypots: Concepts, approaches, and challenges," in *ACM-SE 45: Proceedings of the 45th Annual Southeast Regional Conference,* 2007, pp. 321-326.

[52] S. Meyer, et al, "Be Aware with a Honeypot," *ITB Research Journal,* pp. 4-16, 2007.

[53] A. Keane and S. Meyer, "A simplified approach for the rapid forensics analysis of a compromised honeypot," in *The IT&T 8th International Conference on Information Technology and Telecommunications*, 2008, pp. 28-35.

[54] W. Formyduval, "Integrating static analysis and testing for firewall policies," in *OOPSLA '09: Proceeding of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, 2009, pp. 749-750.

[55] R. Russell. (1998), Iptables. [accessed: 2009, 04 Nov]. Available: http://www.netfilter.org/

[56] BalaBit IT Security. Zorp. [accessed: 2009, 04 Nov]. Available: http://www.balabit.com/network-security/zorp-gateway/

[57] Kerio. (1998), WinRoute firewall. [accessed: 2009, 04 Nov]. Available: http://www.kerio.com/firewall

[58] Check Point. (1994, April). FireWall-1. [accessed: 2009, 04 Nov]. Available: http://www.checkpoint.com/products/firewall-1/

[59] Cisco Systems Inc. (2005, 03 May). Adaptive security appliance (ASA). [accessed: 2009, 04 Nov]. Available: http://www.cisco.com/en/US/products/ps6120/index.html

[60] K. J. Cox., *Managing Security with Snort and IDS Tools*, Sebastopol CA USA: O'Reilly & Associates Inc, 2004.

[61] D. E. Denning, "An intrusion-detection model," in *IEEE Symposium on Security and Privacy*, 1986, pp. 118-133.

[62] R. U. Rehman and N. Regina, *Intrusion Detection with SNORT (Bruce Perens' Open Source Series): Advanced IDS Techniques using Snort, Apache, MySQL, PHP, and ACID*, Pearson Education, 2003.

[63] Insecure.org. Top 5 Intrusion Detection Systems [Online]. http://sectools.org/ids.html [accessed: 2009, 04 Nov]

[64] Y. Zhang, W. Lee and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wirel.Netw.*, vol. 9, pp. 545-556, 2003.

[65] The Prelude Team. PreludeIDS. [accessed: 2009, 06 Nov]. Available: https://dev.prelude-ids.com/

[66] R. Goss, M. Botha and R. von Solms, "Utilizing fuzzy logic and neural networks for effective, preventative intrusion detection in a wireless environment," in *SAICSIT*

'07: Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, 2007, pp. 29-35.

[67] T. Oetiker. rrdtool [Online]. http://oss.oetiker.ch/rrdtool/ [accessed: 2009, 29 Nov]

[68] SRI International, "A Real-Time Intrusion-Detection Expert System (IDES)," 1992, 28 Feb.

[69] G. P. Spathoulas and S. K. Katsikas, "Reducing false positives in intrusion detection systems," *Comput. Secur.*, 2009.

[70] D. B. Cid. OSSEC v 2.2. [accessed: 2009, 06 Nov]. Available: http://www.ossec.net/

[71] V. Jacobson, C. Leres and S. McCanne. (1987), Tcpdump. [accessed: 2009, 06 Nov]. Available: http://www.tcpdump.org/

[72] G. Combs. (1998), Wireshark. Available: http://www.wireshark.org/

[73] D. Fitzpatrick, H. Y. Luan and D. O'Brien. Agent-based network intrusion detection. Presented at The IT&T 8th International Conference on Information Technology and Telecommunication, 2008.

[74] H. Debar, D. Curry and B. Feinstein, "RFC 4765 - The Intrusion Detection Message Exchange Format (IDMEF)," 2007, Mar.

[75] B. Feinstein and G. Matthews, "RFC 4767 - The Intrusion Detection Exchange Protocol (IDXP)," 2007, Mar.

[76] Network Research Group at Lawrence Berkeley Laboratory. Winpcap. [accessed: 2009, 06 Nov]. Available: http://www.winpcap.org/

[77] O. Andreasson. Iptables Tutorial 1.1.19 - Chapter 4. The state machine [Online]. http://www.faqs.org/docs/iptables/tcpconnections.html [accessed: 2009, 29 Nov]

[78] N. Kokholm and P. Sestoft. (2006), The C5 generic collection library. [accessed: 2009, 07 Nov]. Available: http://www.itu.dk/research/c5/

[79] DShield. DShield port/ip lookup [Online]. http://www.dshield.org/ [accessed: 2009, 09 Nov]

[80] Symantec. Spyware.NetSpy [Online]. http://www.symantec.com/security_response/writeup.jsp?docid=2004-080510-5653-99 [accessed: 2009, 09 Nov]

[81] Symantec. W32.Mydoom.AS@mm [Online]. http://www.symantec.com/security_response/writeup.jsp?docid=2005-021013-2446-99 [accessed: 2009, 09 Nov]

[82] G. Jiang, "Multiple vulnerabilities in SNMP," *Computer*, vol. 35, pp. 2-4, 2002.

[83] McAfee. W32/SQLSlammer.worm [Online]. http://home.mcafee.com/VirusInfo/VirusProfile.aspx?key=99992 [accessed: 2009, 09 Nov]

[84] R. R. Linde, "Operating system penetration," in *AFIPS Joint Computer Conferences*, 1975, pp. 361.

[85] V. Saini, Q. Duan and V. Paruchuri, "Threat modeling using attack trees," *Journal of Computing Sciences in Colleges*, vol. 23, pp. 124, 2008, Apr.

[86] S. Singh, J. Lyons and D. M. Nicol, "Fast model-based penetration testing," *Proceedings of the 36th Conference on Winter Simulation*, pp. 309, 2004.

[87] J. P. McDermott, "Attack Net Penetration Testing," *New Security Paradigms Workshop*, pp. 15, 2001.

[88] Bruce Schneier. Attack Trees - Modeling security threats [Online]. http://www.ddj.com/184411129 [accessed: 2009, 14 July]

[89] Amenaza Technologies Ltd. Amenaza Attack Tree Modelling Software Documentation [Online]. http://www.amenaza.com/documents.php [accessed: 2009, 14 July]

[90] Amenaza Technologies Ltd. Understanding Risk Through Attack Tree Analysis [Online]. http://www.amenaza.com/downloads/docs/Methodology.pdf [accessed: 2009, 14 July]

[91] V. Darmaillacq, et al, "Test Generation for Network Security Rules," *Testing of Communicating Systems*, pp. 341-356, 2006.

[92] W. Mallouli, et al, "A formal approach for testing security rules," pp. 127-132, 2007.

[93] Irish Government, "Criminal Damage Act," vol. 31, 1991.

[94] Irish Government, "Criminal Justice (Theft and Fraud Offences) Act," vol. 50, 2001.

[95] G. Fumey-Nassah, "The management of economic ramification of information and network security on an organization," in *InfoSecCD '07: Proceedings of the 4th Annual Conference on Information Security Curriculum Development*, 2007, pp. 1-4.

[96] K. Julisch, "Security compliance: The next frontier in security research," in *NSPW '08: Proceedings of the 2008 Workshop on New Security Paradigms*, 2008, pp. 71-74.

[97] U.S. Government, "Federal Information Security Management Act of 2002 (FISMA)".

[98] PCI Security Standards Council, "Payment Card Industry Data Security Standard (PCI DSS)".

[99] ISO. International Organisation for Standardisation [Online]. http://www.iso.org/ [accessed: 2009, 02 Oct]

[100] NIST. (2008, Sept). SP 800-115 – technical guide to information security testing and assessment. [accessed: 2009, 29 Sept]. Available: http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

[101] J. Billig, Y. Danilchenko and C. E. Frank, "Evaluation of google hacking," in *InfoSecCD '08: Proceedings of the 5th Annual Conference on Information Security Curriculum Development*, 2008, pp. 27-32.

[102] Fixed Orbit. Fixed Orbit Network Search Tools [Online]. http://www.fixedorbit.com/search.htm [accessed: 2009, 23 Nov]

[103] C. McNab., *Network Security Assessment, 2nd Edition*, O'Reilly, 2007.

[104] G. Lyon. Idle Scanning and related IPID games [Online]. http://www.ouah.org/ipidgames.htm [accessed: 2009, 24 Nov]

[105] The OWASP Foundation. Open Web Application Security Project (OWASP) [Online]. http://www.owasp.org/

[106] OWASP. (2009, 13 Nov). OWASP top 10 list 2010 release candidate 1. [accessed: 2009, 24 Nov]. Available: http://www.owasp.org/images/0/0f/OWASP_T10_-_2010_rc1.pdf

[107] Chinotec Technologies Company. (2004), Paros web application proxy. [accessed: 2009, 04 Oct]. Available: http://www.parosproxy.org/

[108] OWASP. WebScarab [Online]. http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project [accessed: 2009, 27 Nov]

[109] C. Sullo. (2007, 10 June). Nikto. [accessed: 2009, 04 Oct]. Available: http://cirt.net/nikto2

[110] M. W. Eichin and J. A. Rochlis, "With microscope and tweezers: An analysis of the internet virus of november 1988," in *1989 IEEE Symposium on Research in Security and Privacy*, pp. 326-343.

[111] MITRE Corporation. Common Vulnerability Exposure (CVE) [Online]. http://cve.mitre.org [accessed: 2009, 10 Aug]

[112] Tenable Network Security. (2009, May 26). Nessus. [accessed: 2009, 05 Aug]. Available: http://www.nessus.org/nessus/

[113] Metasploit LLC. The Metasploit Project [Online]. http://www.metasploit.com/ [accessed: 2009, 10 Aug]

[114] HEAnet Limited. Higher Education Network - HEAnet [Online]. http://www.heanet.ie.remote.library.dcu.ie/ [accessed: 2009, 10 Oct]

[115] Internt Neutral Exchange Association Limited. INEX [Online]. https://www.inex.ie/ [accessed: 2009, 10 Oct]

[116] T. Bates, et al. CIDR Report [Online]. http://www.cidr-report.org [accessed: 2009, 24 Nov]

[117] Dublin City University. DCU Wireless Network - Laplan3.0 Locations [Online]. http://www.dcu.ie/iss/laplan/index.shtml [accessed: 2009, 24 Nov]

[118] boards.ie. PDA in DCU [Online]. http://www.boards.ie/vbulletin/archive/index.php/t-313862.html [accessed: 2009, 10 Oct]

[119] ubuntuforums.org. Proxy Problems [Online]. http://ubuntuforums.org/showthread.php?t=137668 [accessed: 2009, 10 Oct]

[120] Cisco Systems Inc. Cisco IronPort SenderBase Security Network [Online]. http://www.senderbase.org/ [accessed: 2009, 23 Aug]

[121] Companies Registration Office [Online]. http://www.cro.ie [accessed: 2009, 09 Aug]

[122] A. J. Kalafut, C. A. Shue and M. Gupta, "Understanding implications of DNS zone provisioning," in *IMC '08: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, 2008, pp. 211-216.

[123] A. Silveira. TXDNS [Online]. http://www.txdns.net/ [accessed: 2009, 24 Aug]

[124] l3g3ntz 0f c0de k1dZZzz. ghba.c [Online]. http://examples.oreilly.com/networksa/tools/ghba.c [accessed: 2009, 24 Aug]

[125] A. Omella. SING - Send ICMP Nasty Garbage [Online]. http://sourceforge.net/projects/sing/ [accessed: 2009, 24 Aug]

[126] J. Postel and J. Reynolds, "RFC 854 - Telnet Protocol Specification," May. 1983.

[127] T. Ylonen and C. Lonvick, "RFC 4252 - The Secure Shell Authentication Protocol (SSH)," 2006, Jan.

[128] J. B. Postel, "RFC 821 - Simple Mail Transfer Protocol (SMTP)," 1982, Aug.

[129] Anonymous (2003, 20 Apr). Smtpmap. [accessed: 2009, 29 Oct]. Available: http://www.freshports.org/security/smtpscan/

[130] pentestmonkey. smtp-user-enum [Online]. http://pentestmonkey.net/tools/smtp-user-enum/ [accessed: 2009, 25 Nov]

[131] Milworm. Debian OpenSSL Predictable PRNG Bruteforce SSH Exploit [Online]. http://www.milw0rm.com/exploits/5622 [accessed: 2009, 25 Nov]

[132] Telecom Italia. Java agent development framework (JADE). [accessed: 2009, 16 Nov]. Available: http://jade.tilab.com/