7 Introduction

On-line lecture: http://buchananweb.co.uk/adv/unit01.html

7.1 Objectives

The key objectives of this unit are to:

- Provide an outline of risk, and the terminology used.
- Provide an outline to a range of threats.
- Understand the usage of client/server connections.
- Outline the usage of services on Windows and Linux, and provide an introduction to service-oriented infrastructures.
- Provide a practical background in Windows and Linux for services, logging and auditing.

7.2 Introduction

ISO 27001/2 is a key standard and defines best practice related to information security. An important element of this standard is the assessment of risks, and their associated threats. This chapter outlines some of the key classifications of threats, and Unit 2 discusses how vulnerabilities can be assessed. It also aims to define some of the terminology that can be used to define a security incident, and outlines a formal taxonomy which defines common definitions for key terms. The ISO 27001/2 standard started life as "Information Security Code of Practice" from the UK (DTI), and was published in the 1990. It recently changed from ISO/IEC 17799 to ISO/IEC 27001/2 and provides a benchmark for most areas of security. Overall it defines 12 main areas, which are defined in Appendix A.

The key focus of this chapter is to provide a background to key areas, including the integration of services within computing environments, as this will provide a core to the module. It thus covers the key principles of client-server communications, where a service is advertised at a known Internet address, on a defined TCP port. A client can thus easily find the service, if it knows both these parameters. This type of communication is the standard one used on the Internet, and is also the basis of service-oriented architecture, where services are distributed around a network. In this way it becomes easier to provide robustness and maintenance, while allowing devices of limited capacity access to services which they might not be able to normally support on a local basis. Much of modern security and network forensics are also build around the usage of services, thus they will provide a key focus of the rest of the module.

7.3 Security Taxonomy

Along with defining an ontology to allow a wide range of professionals to use a standardized method of defining risk, there needs to be clear classifications for

threats and their mitigation procedures. Abbas (2006), for example, defines a taxonomy for Internet security with a number of attack strategies:

- **Manual Penetrating the System and/or Individual Privacy**. This includes methods and techniques that define the manual system penetration and includes password cracking, social engineering, and masquerading.
- **Data Interception, Interruption, and Replaying**. This includes methods of intercepting information or communication processes, and includes the tampering of transmitted data.
- **Biometrics and Physical Token**. This includes attacks related to physical and biometrical methods, such as for copying a biometric scan, and their associated processes.
- **Defeating Mechanisms and Policy**. This includes comprising the authentication, authorization, and access control infrastructure and their associated policies.
- **Malicious Code**. This includes malicious software, viruses, malfeasant code, bugs, coding problems, and so on.
- **Distributed Communication Systems**. This includes methods to attack that uses network communication protocols, such as for Distributed Denial of Service (DDoS).

A fairly abstract representation is provided by Hollingworth (2009) which classifies a security incident in terms of threats, attack tools, vulnerabilities, access methods, results and objectives. For example (Figure 7.1):

A **Threat:**
- Hacker.
- Spies
- Terrorists.
- Corporate Raiders.
- Professional Criminals.
- Vandals.
- Military Forces.

is achieved with **Attack Tools**:
- User command.
- Script or program.
- Autonomous Agent.
- Toolkit
- Distributed Tool.
- Data Tap.

for **Vulnerabilities**:
- Implementation vulnerability.
- Design vulnerability.
- Configuration vulnerability.

with **Access**:
- Unauthorized Access of Processes for:
- Unauthorized Use of Processes for:
  - Files.

- Data in transit.
- Objects in Transit.
- Invocations in Transit.

which **Results** in:

- Corruption of Information.
- Disclosure of Information.
- Theft of Service.
- Denial-of-Service.

for **Objectives**:

- Challenge/Status.
- Political Gain.
- Financial Gain.
- Damage.
- Destruction of an Enemy.

Thus, with the Russian attack on Estonia a denial-of-service attack might be defined with:

[Government Agents] used [Autonomous Agents] to exploit a [Design Vulnerability] for [Unauthorized Use] of [Processes] for [Invocations in Transit] which resulted in a [Denial-of-Service] for [Political Gain]

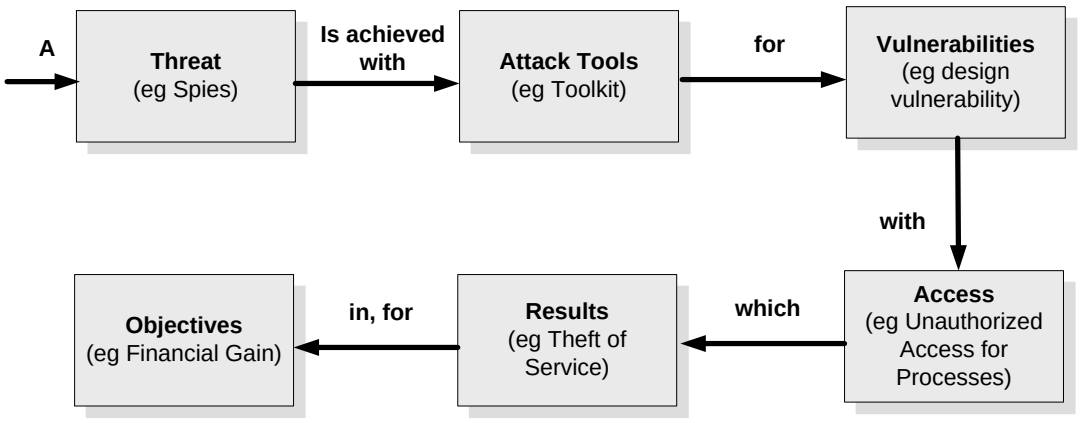


Figure 7.1 Security incident taxonomy

7.4 Threats

Kent (2003) defines that a threat relates to an adversary motivated and capable of exploiting a vulnerability. From now on, an adversary will be known as an intruder. There are various properties which can relate to this threat, including:

- **Accessibility/Opportunity**.
- **Asset Financial Value**.
- **Asset Security Value**.
- **Attacker Sophistication/knowledge**.

- **Coordination/synchronisation**. This might involve the amount of co-ordination or synchronisation required to exploit the vulnerability, such as related to the co-ordination of bots in a Distributed Denial-of-Service attack.
- **Knowledge**. This might relate to the amount of knowledge that an intruder requires for exploit the vulnerability.
- **Progression**. This might relate to the setup of backdoors and Trojans.
- **Threat characteristics**.

The following defines some relevant threats.

Hardware misuse

There are many forms of hardware misuse including:

- **Logical scavenging**. This involves scavenging through discarded media (Figure 7.2).
- **Eavesdropping**. This involves intercepting communications (Figure 7.2).
- **Interference**. This involves the actual interference of communications, such as in jamming communications, or modifying it in some way (Figure 7.3).
- **Physical attacks**. This involves an actual physical attack on the hardware (Figure 7.3).
- **Physical removal**. This involves the actual physical removal of hardware.

Andy Jones and Prof Andrew Blyth from the University in Glamorgan, in 2008/2009, found that an analysis of 300 discarded disk showed that 34% of the disks still had personal information, which included health care and banking information (including a 50 billion Euro currency exchange). One of the disks even contained details of the test launch procedures for the THAAD (Terminal High Altitude Area Defence) ground-to-air missile defence system, which is a system designed to destroy long-range intercontinental missiles. As much as possible organisations should have a predefined plan on how they discard of their equipment, especially for electronic media (normally USB sticks) and for hard disks. The term used in the US for logical scavenging is dumpster diving.

Physical attacks and removal can cause major problems with security, and thus organisations require a strong policy on protecting hardware, including padlocking equipment within locked cabinets, within restricted zones.

Environmental risks

Computer equipment is fairly sensitive to changes in environmental conditions, especially related to temperature and humidity changes. This could involve risks related to ventilation, humidity control, and system cooling. Computer equipment can also be affected by power supply changes, including power spikes and surges, power blackouts, and reduced power levels (sags and brownouts). Often these environmental risks are caused by operating conditions (especially the weather and local supplies) or human/process failure, but may be part of a computer environment attack from an intruder.

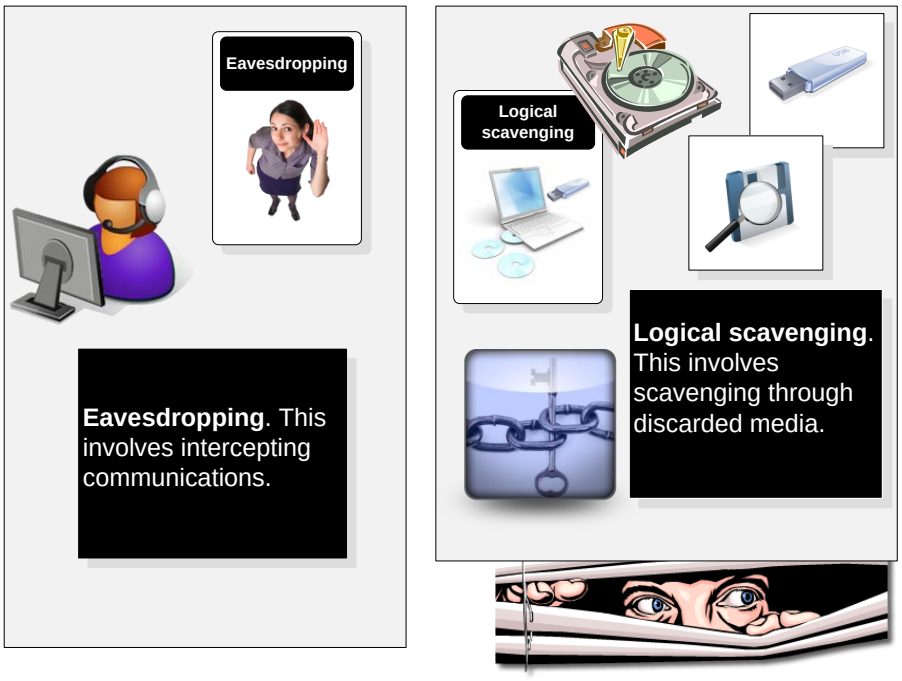


Figure 7.2 Eavesdropping and logical scavenging

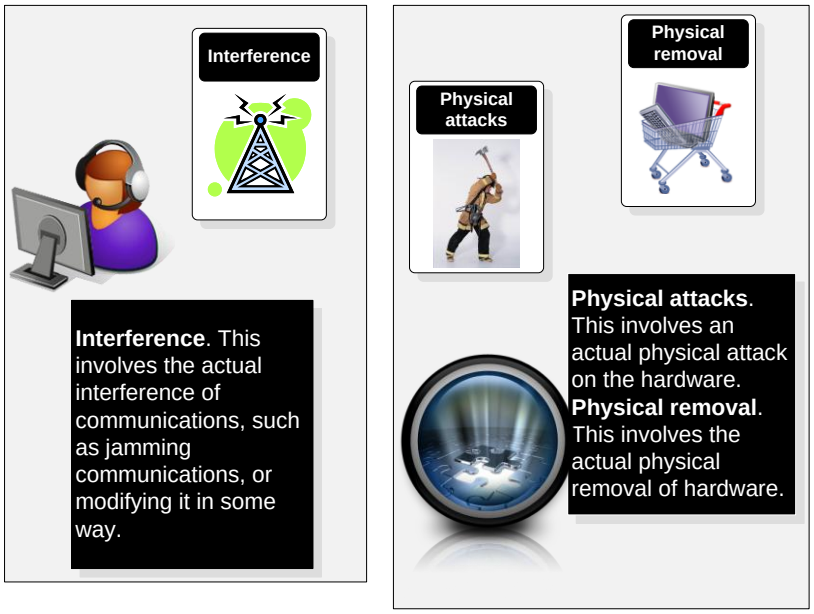


Figure 7.3 Interference and physical attacks

External misuse

External misuse involves some form external interference with internal users within an organization, normally involving some form of social engineering attack. This might include:

- **Visual spying**. This is the actual physical viewing a user's activities, such as their keystrokes or mouse clicks (Figure 7.4).
- **Misrepresentation**. This involves the actual deception of users and system operators, and is a form of social engineering (Figure 7.4).

Often misrepresentation occurs through a telephone call, a meeting in-person, an observation of the person, or through email. Typical methods include gaining a background knowledge of an individual, such as through dumpster diving or on-line searching (such as *Facebook*), and then focusing on them to mine them for information. For example if an intruder targets a certain user, a search on the Internet might give them their mother's maiden name. They could then phone the *target*, saying they were their mobile phone company, and saying there was a problem with their direct debit, and thus gain the bank account of the user. Armed with this, and a knowledge of the target, could allow them to gain access to privileged information.

Typical method used to mitigate against social engineering attacks are to:

- Improve training for users.
- Define strict policies about the kind of information that is allowed to be given over communications channels. For example a bank will never ask for a full password, and will typically only ask for two or three characters from a pass-phase.
- Enforce a system of badges and ID cards.

In many cases, it is people who are the weakest link in a security infrastructure, and not the actual technological design/implementation.

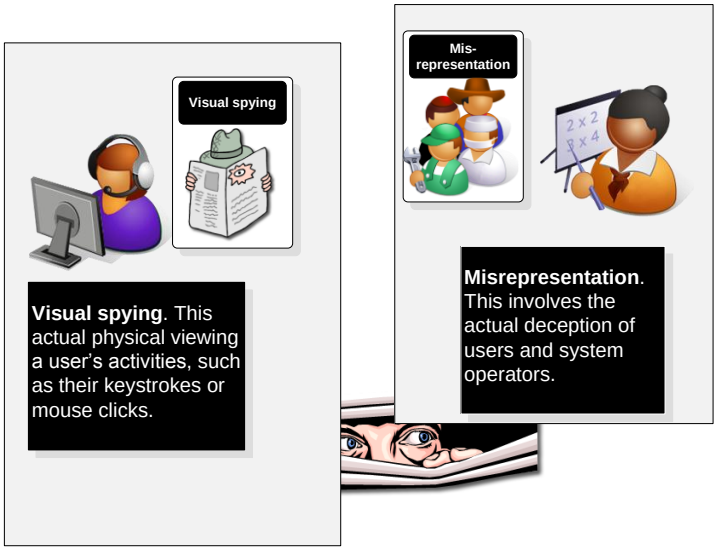


Figure 7.4 Visual spying and misrepresentation

Pests

Pest are malicious pieces of software which have been created for the purpose of exploiting a host machine. These types of threat are often detected using a signature scanning program, such as for a virus scanner or an intrusion detection system. Examples include:

- **Trojan horses**. This involves users running programs which look valid, but install an illicit program which will typically do damage to the host (Figure 7.5).

- **Logic bombs**. This involves the installation of a program which will trigger at some time in the future based at a given or pre-defined event, such as for a certain user logging-in, or a message from a remote system (Figure 7.5).
- **Malevolent worms**. This involves a worm program which mutates in a given way which will eventually reduce the quality-of-service of the network system, such as using up CPU resources, or taking up network band-width (Figure 7.6).
- **Viruses**. This involves malicious programs attaching themselves and which self replicate themselves (Figure 7.6).

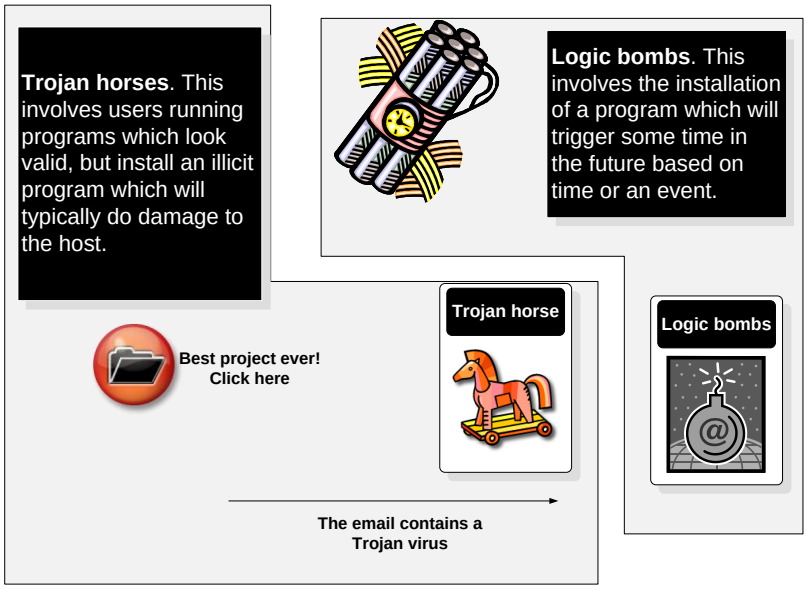


Figure 7.5 Trojan horses and logic bombs

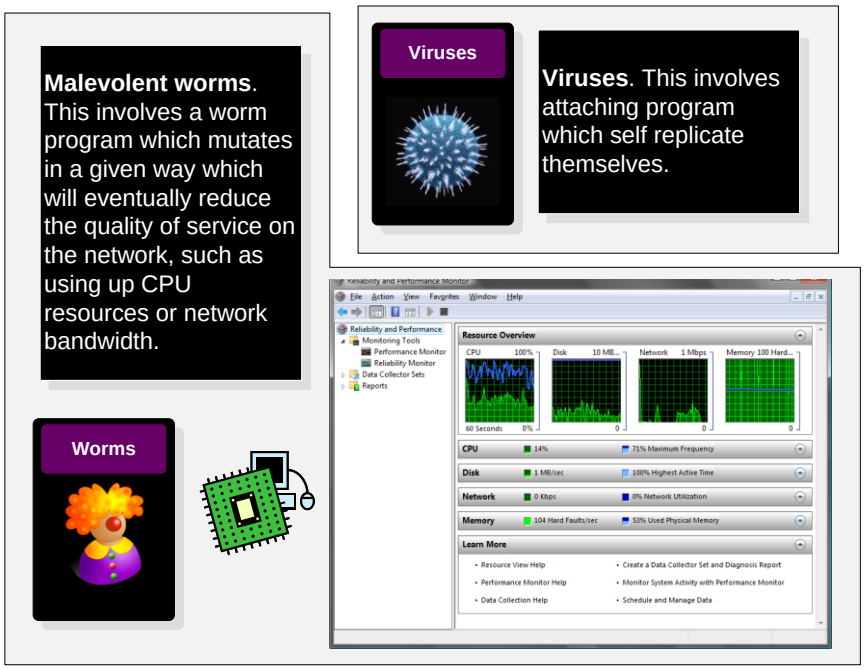


Figure 7.6 Malevolent worms and viruses

Active misuse

An active misuse threat normally involves some form of proactive method, such as for:

- **Incremental attack**. This involves damaging a system using an incremental approach.
- **Denial-of-service (DoS)**. This involves attacking a host with continual requests for services, which eventually reduces its performance (Figure 7.7). With a distributed DoS (DDoS), remote hosts make continual accesses to hosts, such as continually requesting a connection (SYN flood).
- **Active attack**. This typically involves entering incorrect data with the intention to do damage to the system (Figure 7.7). An example threat with this involves buffer overflows, where the intruder intentionally enters data into a program, which overfills the memory area reserved for input variables. This overflowing can affect other data, or can even affect the actual program code. The major problem involves older programming languages such as C and C++, which allocates defined areas for their variables, and does not typically check for the size of the data input. Modern languages such as Java and C# use dynamic variable sizes, where more checking is done on the input data, in which one variable cannot affect another one.

Passive misuse

Passive misuse is a threat where an intruder attacks a system using methods that look fairly passive in their scope, but where they try to exploit a weakness in protocols/methods. This includes:

- **Browsing**. This issues random and/or selective searches for information.
- **Interference/aggression**. This involves exploiting database weaknesses using inferences (Figure 7.8). In the example in Figure 7.8, the user is not allowed to see the individual marks of students, but is allowed to see the average of a number of students. It can be seen that for three students, and three queries for an average mark of each of each group of two students, results in the inference of their individual mark. Inference is difficult to defend against, as there are an almost infinite number of ways that someone may view data, and the only way to overcome it is to make sure that the queries allowed on a system is limited to valid ones.
- **Covert channels**. This involves hiding data in valid communications, such as with network traffic or by passing information through the creation of timed events (Figure 7.9). Many of the modern Internet protocols were created at a time where security was not a major factor. They are thus often flawed, and there are many fields where data can be hidden. For example, the IP header contains a TTL field, which is ignored by more hosts, and can be used to hide data.

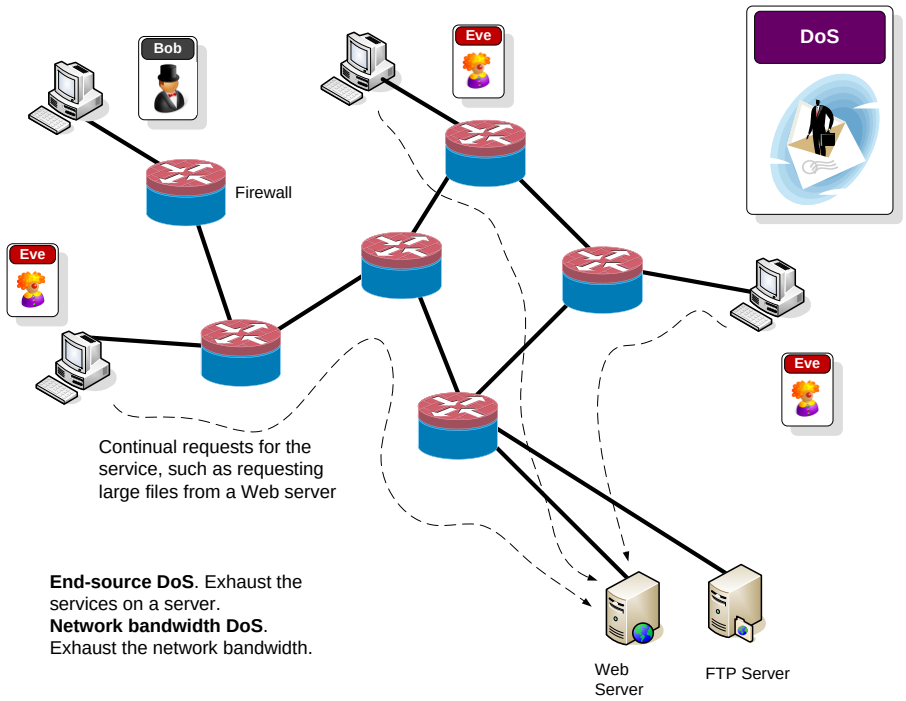


Figure 7.7 DDoS

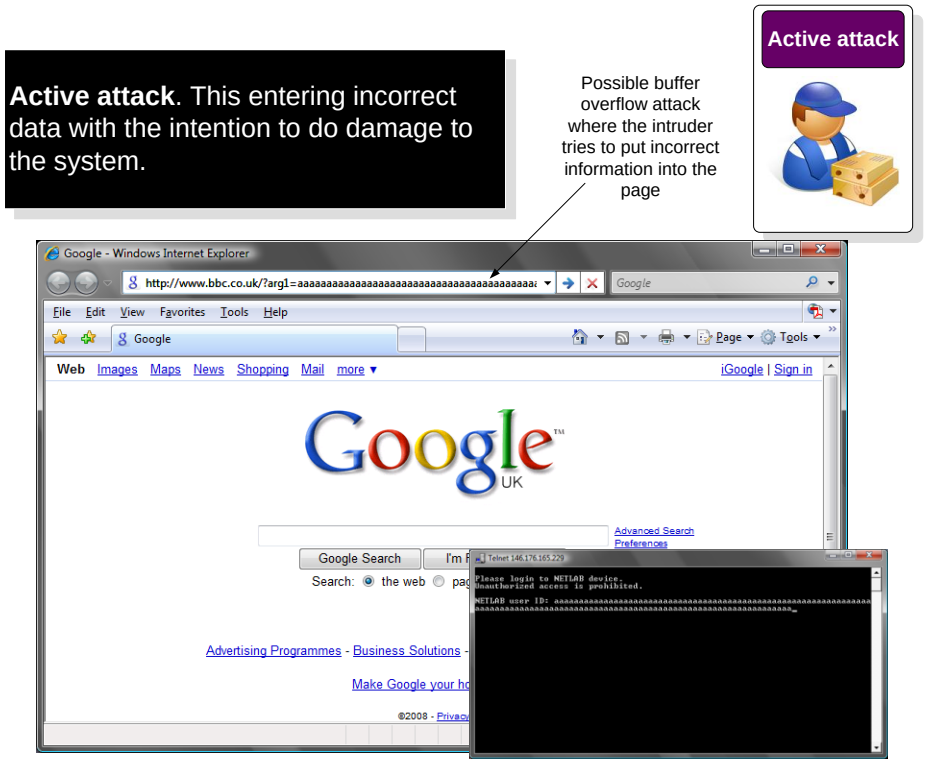


Figure 7.8 Active attack

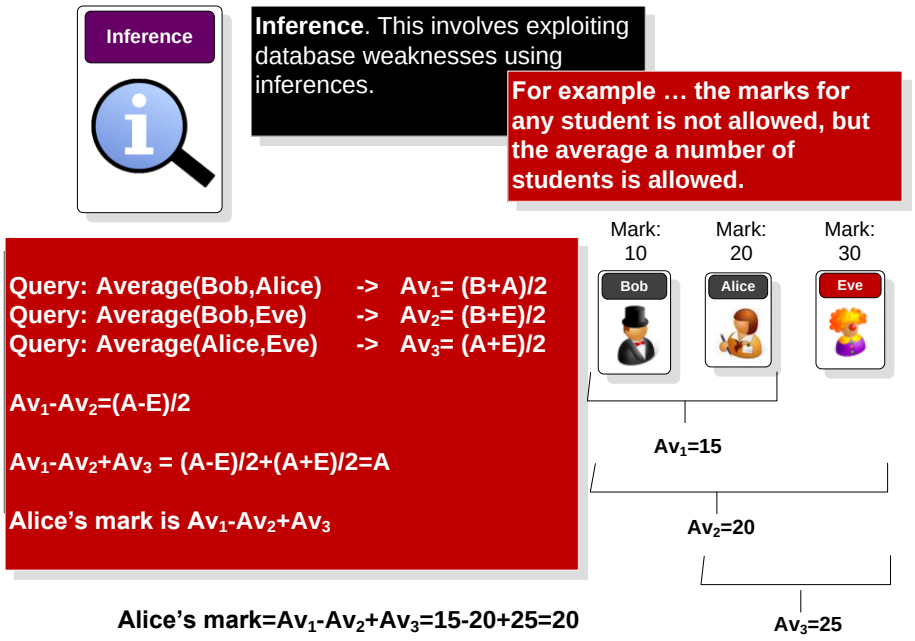


Figure 7.9 Inference

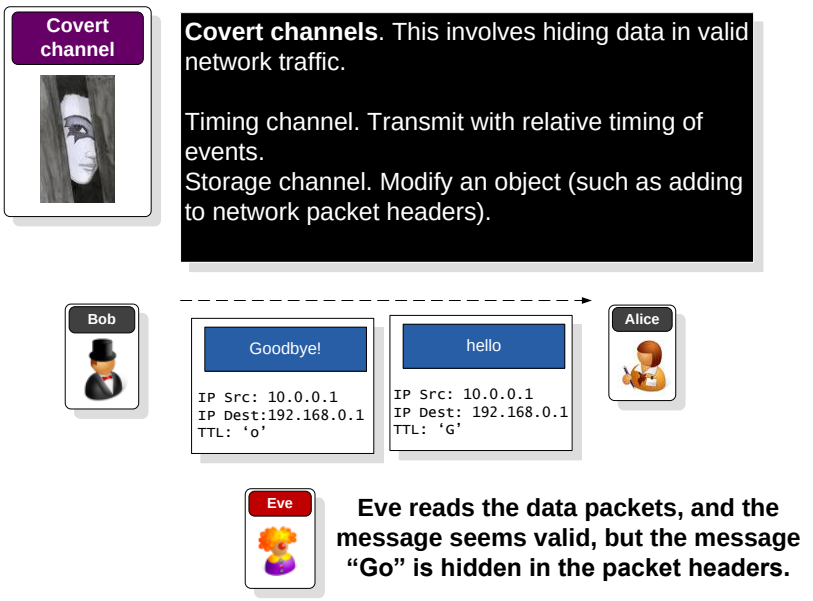


Figure 7.10 Covert channels

Masquerading/spoofing

These types of threats normally involve some form of impersonation of a device or a user, or involve obfuscation of the source of the threat. Examples of this include:

- **Impersonation**. This involves the impersonation of a user/device (Figure 7.11).
- **Spoofing**. This involves the spoofing of devices (Figure 7.11).
- **Piggy back attacks**. This involves adding data onto valid data packets (Figure 7.12).
- **Network weaving**. This involves confusing the system as to the whereabouts of a device, or confusing the routing of data. A typical technique used for this is to

hide operations behind NAT (Network Address Translation), or to use a proxy agent to perform an action (Figure 7.12).

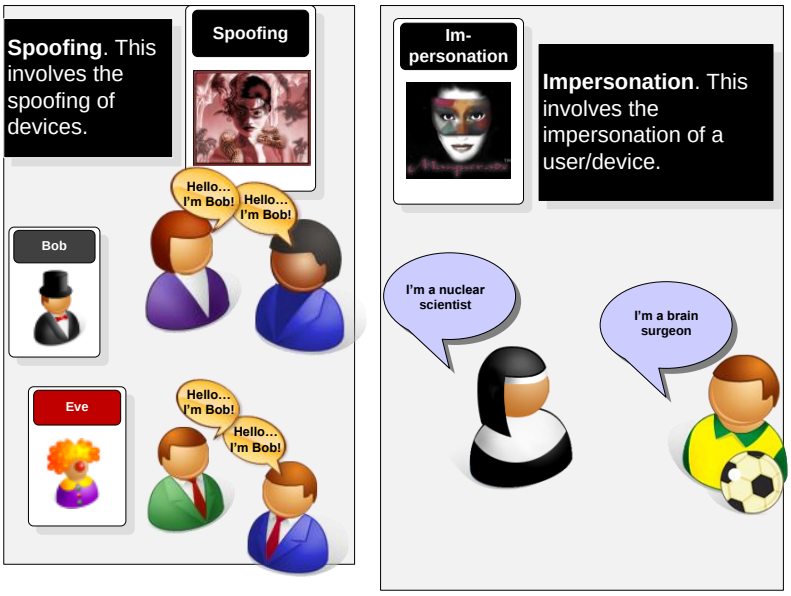


Figure 7.11 Spoofing and impersonation

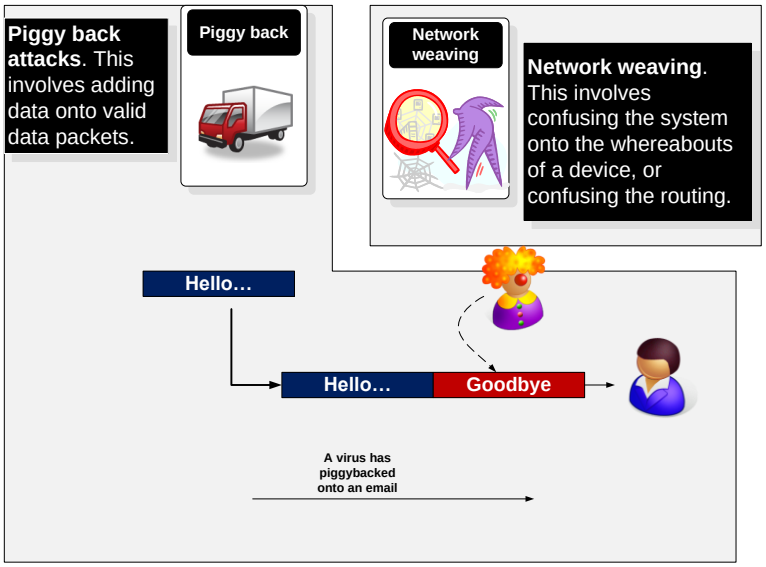


Figure 7.12 Piggy back and network weaving

Bypasses

These type of bypasses include:

- **Trap door impersonation**. This involves the creation of pages or login screens which look valid, but are used to gain information from a user, such as for their bank details, or login password (Figure 7.13).
- **Authorization attacks**. This involves trying to gain access to a higher level of authorization than is valid for the user, such as with password attacks.

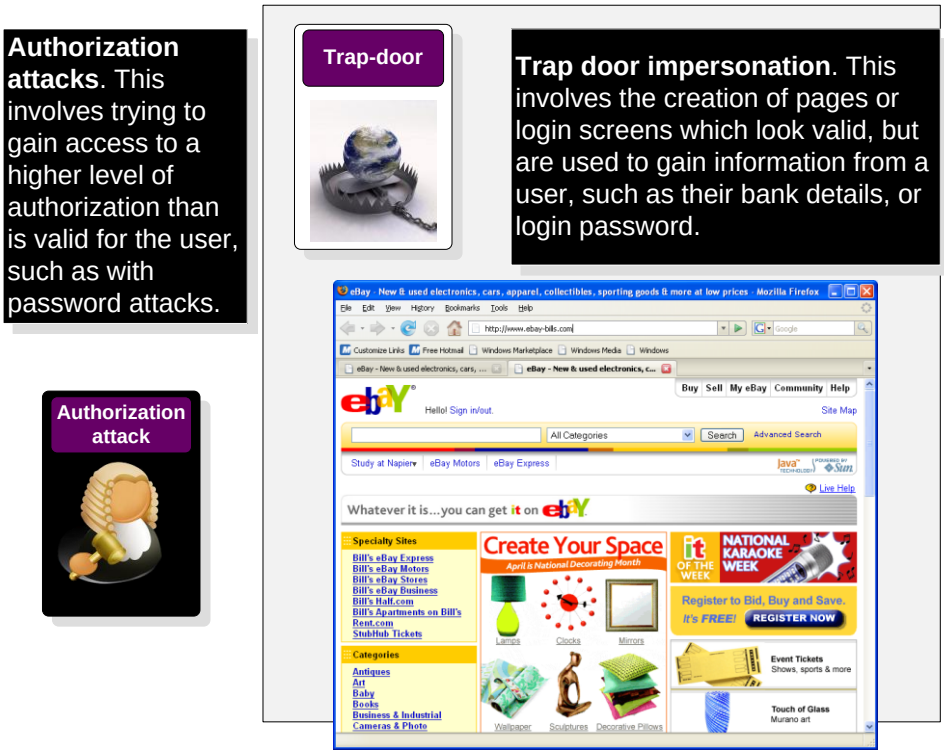


Figure 7.13 Authorization attacks and trap door impersonation

7.5 Service-oriented infrastructures

The first few generations of computers were based around centralized systems, with terminals which accessed central mainframe, which ran processes for them. With the introduction of the IBM PC, more software was installed locally on machines, and centralized services were used for key applications such as for network storage and email. As networks and, often, servers, were initially unreliable, most PCs were setup to run most of their applications locally. As the reliability and spread of networks has increased, there has been a tendency to move applications and services away from the local machine to distributing them around a network. This makes it easy to update software without requiring the local installation of updates. The main benefit, though, is that distributed services provide a fairly robust infrastructure, and it allows for the processing to move from the localized machine to networked services, which increases the scope of the devices that can access applications. For example, a mobile phone typically has limited processing and storage facilities, but with a service-oriented architecture, it is possible for them to access large scale processing facilities, along with access to mass storage.

Figure 7.14 shows an example of creating a service-oriented architecture, where services can either be local, domain-, remote- or cloud-based. Firewalls can block the access to services, including host-based, local and domain firewalls, each of which can block services at their different levels. Typical services include proxy services, directory services, Web services, file transfer services, email services, and so on. Generally, to aid robustness, there is a move towards having multiple service points, so that the infrastructure can cope with network outages.

Servers are normally used to provide networked services, as these are typically built in order to be robust for their power supplies, their storage, their network connection, and so on. In the past, servers have often run a range of services, such as email and file services, but generally most organizational servers are now setup to focus on providing one type of service, thus distributing services across a range of servers.

A service connection has a client connecting to a service. The definition of the service is normally found using an IP address to find the server, and a TCP/UDP port to define the connection point on the server in which to connect to. The connection thus becomes:

IP(host)Port(host) -> IP(service)Port(service)

Normally the IP(service) and Port(service) are well know so that hosts can access them. On the Internet, each of these mapping must be unique, thus Port(host) requires to change in order to make the mapping unique (Figure 7.15). Typical service ports are 80 (HTTP), 21 (FTP), 23 (Telnet), 25 (SMTP), 110 (POP3), but increasing the port used is 80, and the services are tunnelled through this port as firewalls tend not be block HTTP access. The services can also run locally on a host, and can access their own services in the same way as they are done remotely.

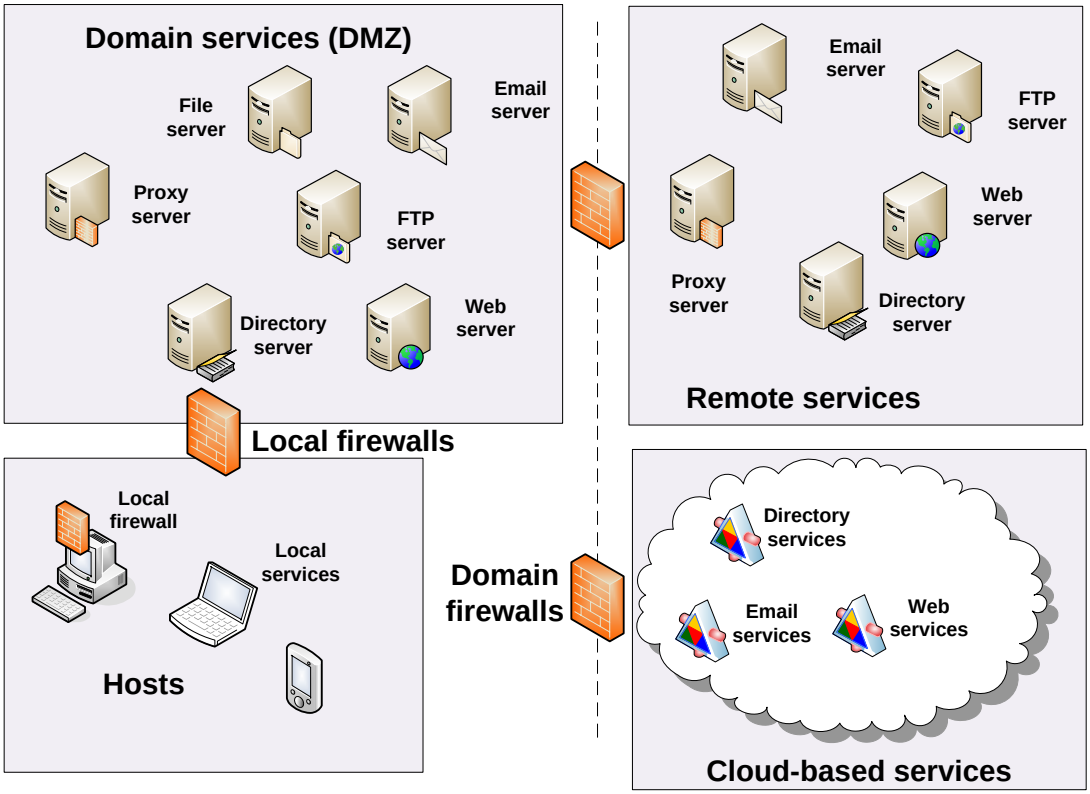


Figure 7.14 Security policy definition, implementation and verification

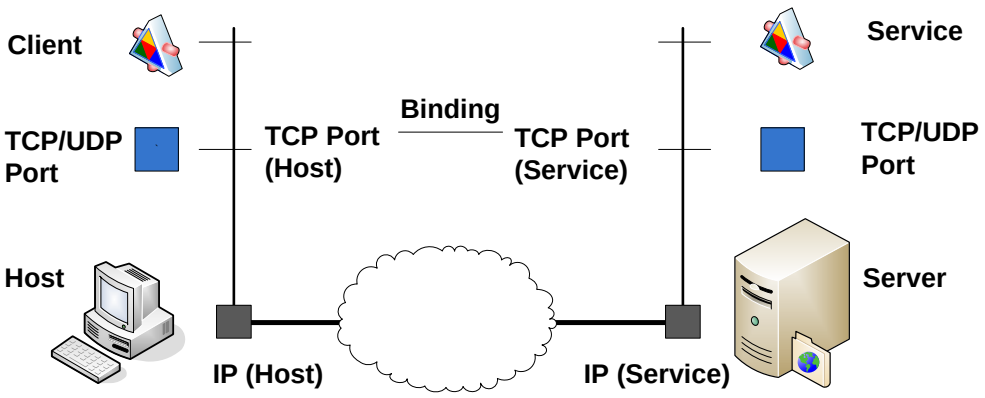


Figure 7.15 Port binding for services

http://buchananweb.co.uk/adv_security_and_network_forensics/dotnetclient/dotnetclient.htm

7.6 Security policies

Often military analogies are used in security, and the equivalent of having internal and external attacks is equivalent to fighting an external army at a defensive line, while also being attacked from behind a defensive line. In networked systems, with security, there is no attack, as the goal is purely to defend. Overall the key factor in any security system is that the aims and objectives of the organization must map directly onto the implementation of the security policy.

A networked system is a complex entity, composed of many elements, such as hardware devices, operating systems, application programs, file systems, and users. In a highly secure system the overall system should be also be broken down into entities, each of which have security policies for individual users, and also for groups of users. For example, a networked printer should have a policy which restricts access to individual users, and also groups of users. Often in a hierarchal network the entities should inherit security policies from the hierarchy above them. For example with file directories, the subdirectories will often inherit their security policies from the level above, unless otherwise stated. This type of approach typically simplifies the security policy for the overall system.

Often the key elements of any security policy are to:

- **Deter**. This is where the system is designed and implemented in order to initially deter intruders from attacking the system in the first place.
- **Log**. This is a key element in modern systems which require some form of logging system. It is important that the data that is logged does not breach any civil liberties, and is in a form which can be used to enhance the future security of the system.
- **Detect**. This is where detection agents are placed within the network to detect intrusions, and has some method of tracing the events that occurred in an intrusion, so that it can be used either in a forensic computing investigation, and/or to

overcome a future intrusion. Organisations often have many reasons for detecting network traffic.

- **Protect**. This is where policies are created which protect systems, users and data against attack, and this potential damage. A key element of this is to protect them against accidental damage, as accidental damage is often more prevalent than non-accidental damage.
- **React**. This is where a policy is defined which reacts to intrusions, and defines ways to overcome them in the future. Often organisations do not have formal policies for this type of activity, and often rely on *ad-hoc* arrangement, where the method of reacting to a security breach is created after the event.
- **Recover**. This is where policies are defined to overcome any system damage, whether it is actual physical damage, the abuse of users; or the damage to data.
- **Audit/verify**. It is important that the security policy allows for auditing and for the verification that it achieves its requirements.

Security, typically, focuses on the detection, protection and recovery from an attack, whereas forensic computing focuses on not just the malicious activity, but also in capturing the after-affects of an attack, as well as for non- malicious behaviour. A key component is that security tends to focus on the assumption of guilt within attacks, whereas forensic computing must focus on both malicious and non-malicious data so that a fair case can be presented for an investigation. This a forensics policy will typically focus on the detection of events, and the associated procedures. The key focus for the forensic computing parts of this module will be on:

- **Log**. This will define the data that is recorded, and, possibly, the rights of the data to be viewed by certain individuals within an organisation.
- **Detect**. This would be the activities which were to be detected for forensic investigations.
- **React**. This is where a policy is defined which reacts to malicious activities, and, especially in a forensic computing investigation, the procedures involved.
- **Audit/verify**. It is important that the forensics policy allows for auditing and verification that it achieves its requirements.

7.7 Defining the policy

A key element of security is to have a policy which is defined at the highest-level in the organisation, and which is well-known to the employees in the organisation. Also, if there is public access to the network, they should also be informed as to the security restrictions placed on the network. Figure 7.16 shows a transparent and auditable system where the policy is defined at the highest level, and includes: the aims and objectives of the organisation; the legal moral and social responsibilities of the organisation; and the technical feasibility of the policy. These are then decided upon, and a policy is implemented by technical staff. A key feature is that this policy should be audited in some way, and also verified that it achieves the policy requirements.

There are possibly many different types of network/user activity that should be detected and which could breach the aims and objectives of the organisation, or

which breach the social, moral and legal responsibilities of the organisation. Examples of classifications for attacks might be:

- Attempted administrator privilege gain.
- Attempted user privilege gain.
- Denial-of-service.
- ICMP event.
- Information leak.
- Network scan.
- Non-standard protocol.
- Policy violation.
- Suspicious string detection.
- Suspicious login.
- Trojan activity.
- Unusual client-port connection.
- Web application attack.

There are many examples of network traffic/user activity that might be monitored with an intrusion detection system (IDS). It can be seen that it is not just treats to the network, but also activities that might be wasteful in resources, or which breach social and moral rules. It can often be just as embarrassing for a user in an organisation to be involved in an immoral activity, than it is to have a network intrusion. Thus applications such as peer-to-peer file sharing, such as Kazaa, should be avoided in organisations, as they have many copyright issues. Along with this audio and video streaming, such as from news sites, may be wasteful on bandwidth, and, if this type of traffic was great enough, it might swamp traffic which is important for the organisation.

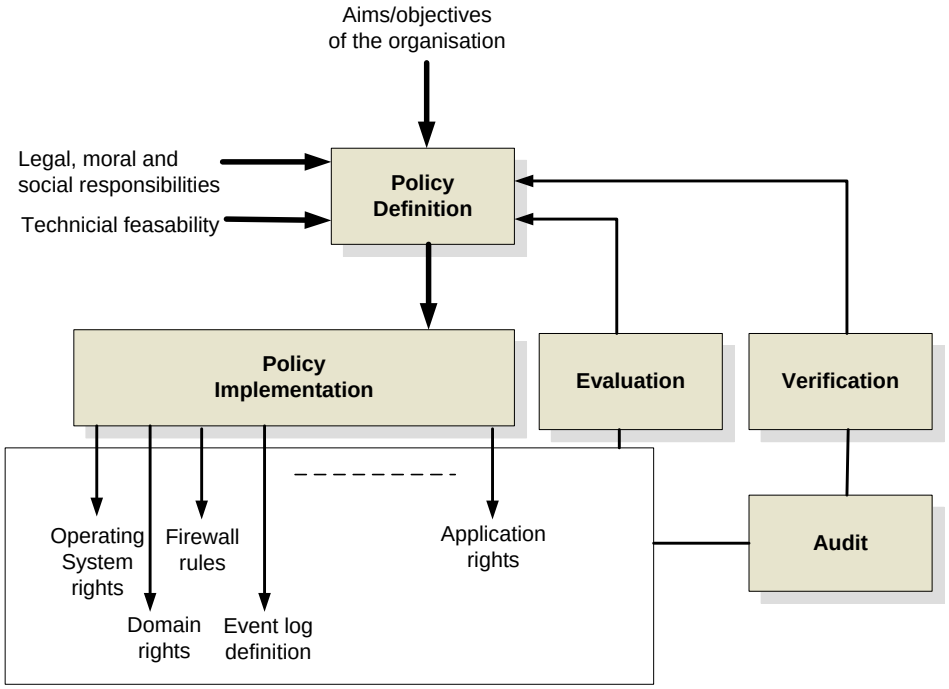


Figure 7.16 Security policy definition, implementation and verification

7.8 Tutorial

The main tutorial is at:

- On-line tutorial: http://buchananweb.co.uk/adv01.html

7.9 Windows Service tutorial

- On-line demo: http://buchananweb.co.uk/adv_security_and_network_forensics/threat01/threat01.htm

Note: The labs in this section require a virtual image defined in Appendix A.

Part A. This part of the lab has two elements: the host machine (DESKTOP) and the Windows virtual image (WINDOWS2003).

7.1 Run the Windows Server 2003 virtual image (User name: Administrator, Password: napier). Within the virtual image, run the command prompt and determine its IP address using IPCONFIG.

☞ What are the IP addresses of the server and the network address which will be used to connect to the virtual image:

7.2 From DESKTOP, ping WINDOWS2003, and vice-versa.

SERVICE: Web

7.3 In WINDOWS2003, go to the folder c:\inetpub\wwwroot.

☞ What are the names of the files in this folder:

7.4 From WINDOW2003, run netstat –a, and determine the services that are running.

☞ List some of the services:

7.5 From your host, connect to the Web Server from DESKTOP using http://w.x.y.z, where w.x.y.z is the IP address of WINDOWS2003 (Figure 7.17). Repeat this using:

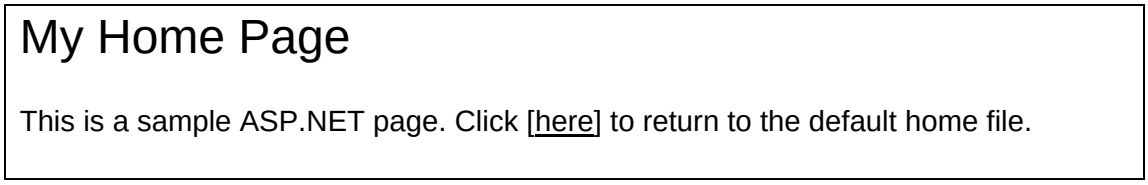
telnet w.x.y.z 80

and then enter:

GET /iisstart.htm

☞ What is the result, and how does it relate to accessing the home page of the Web server?

7.6 On WINDOWS2003, using Microsoft Web Developer Express, open up the c:\inetpub\wwwroot Web folder, and Add a New Item to create your own home page. Next modify iisstart.htm so that it has a link to your home page. The home page should be:



☞ Can you access this page from the host (DESKTOP)?

☞ On WINDOWS2003, go to C:\WINDOWS\system32\LogFiles\W3SVC1. What is the contents of the folder, and what do the files contain?

☞ How might these log files be used to trace malicious activity?

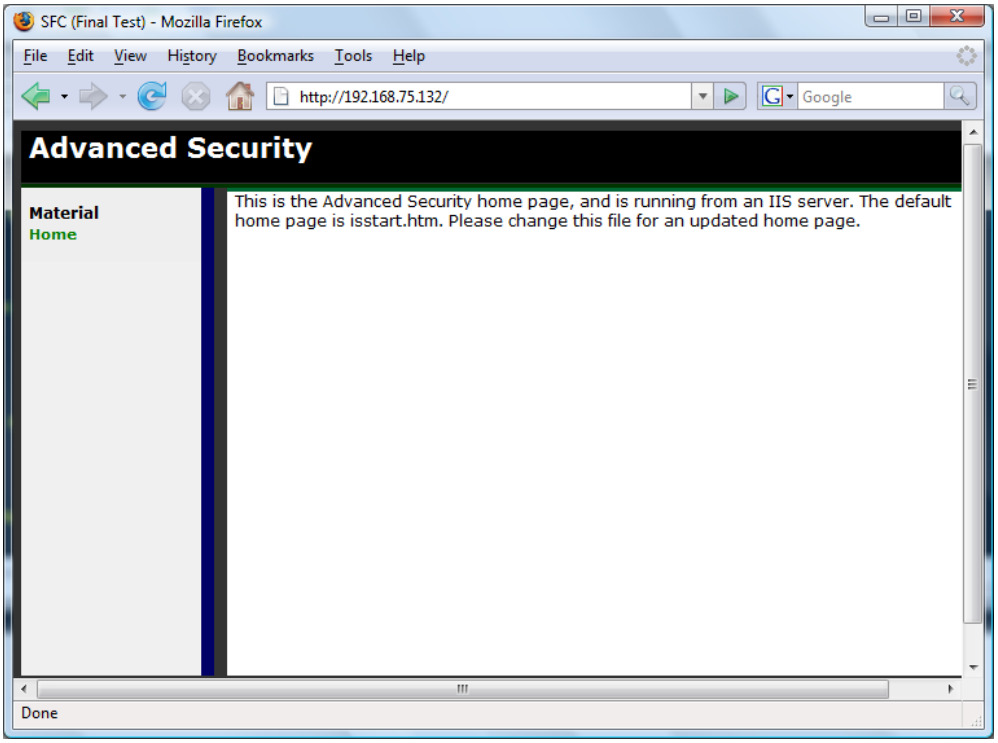


Figure 7.17 HTTP connection

SERVICE: Telnet

7.7 From your host, connect to the Telnet Server from DESKTOP using telnet x.y.y.z, where w.x.y.z is the IP address of WINDOWS2003. Login in with Administrator (password: Napier).

☞	What is the default home folder for Telnet on WINDOWS2003:

Quit from Telnet, using the "exit" command.

SERVICE: FTP

7.8 From your host, connect to the FTP Server from DESKTOP using ftp://w.x.y.z where w.x.y.z is the IP address of WINDOWS2003 (Figure 7.18). Repeat this using:

telnet w.x.y.z 21

and then enter the commands in bold (and note the commands that you get beside the sample return ones):

220 Microsoft FTP Service **USER Administrator** 331 Password required for Administrator.

PASS napier
230 User Administrator logged in.
SYST
215 Windows_NT
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (192,168,75,132,4,65).
LIST

Now FTP opens up a port for the data transfer. This is calculated from the last two digits of the Passive Mode response (227 response). It is calculated as four times the second last digital plus the last digital. So, in this case, it is:

Port = 4*256+65 = 1089

Next open up the data transfer by creating a new Telnet connection, such as:

telnet w.x.y.z 1089

- ☞ Can you access this page from the host?
- ☞ On WINDOWS2003, go to C:\WINDOWS\system32\LogFiles\MSFTPSVC1. What is the contents of the folder, and what do the files contain?
- ☞ How might these log files be used to trace malicious activity?

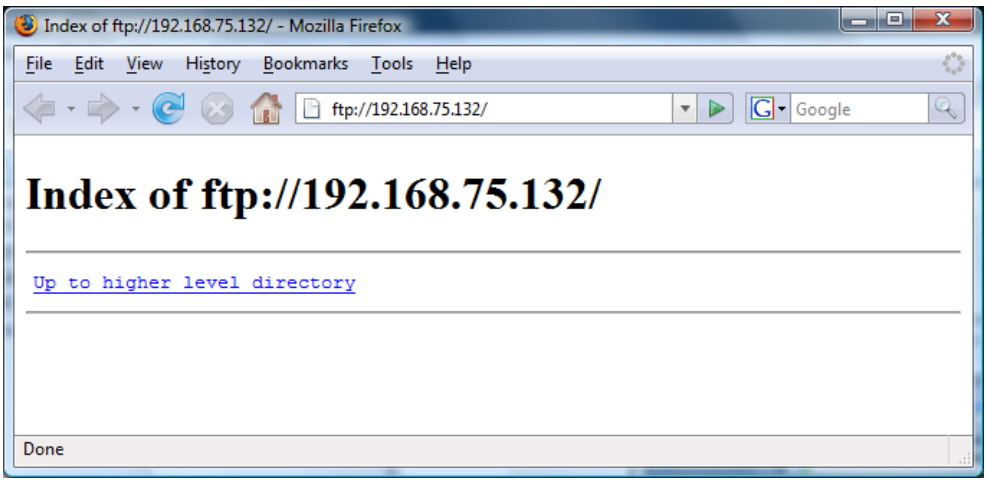


Figure 7.18 FTP connection

SERVICE: SMTP

7.9 From your host, use the following command:

telnet w.x.y.z 25

and connect to the SMTP server. Next enter the commands in bold:

```
220 napier Microsoft ESMTP MAIL Service, Version: 6.0.3790.3959 ready at  Sun,
0 Dec 2009 21:56:01 +0000
help
214-This server supports the following commands:
214 HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HELP AUTH TURN
      ETRN BDAT VRFY
helo me
250 napier Hello [192.168.75.1]
mail from: email@domain.com
250 2.1.0 email@domain.com....Sender OK
rcpt to: fred@mydomain.com
250 2.1.5 fred@mydomain.com
Data
354 Start mail input; end with <CRLF>.<CRLF>
From: Bob <bob@test.org>
To: Alice <alice@ test.org >
Date: Sun, 20 Dec 2009
Subject: Test message

Hello Alice.
This is an email to say hello
.
250 2.6.0 <NAPIERMp7lzvxrMVHFb00000001@napier> Queued mail for delivery
```

7.10 Go to WINDOWS2003, and go into the C:\Inetpub\mailroot\Queue folder, and view the queued email message.

☞ Outline the format of the EML file:

SERVICE: Remote Desktop

7.11 From the host, connect to WINDOWS2003 using the Remote Desktop (Figure 7.19).

☞ Which is the service which is running on WINDOWS2003, that allows the remote connection to happen?

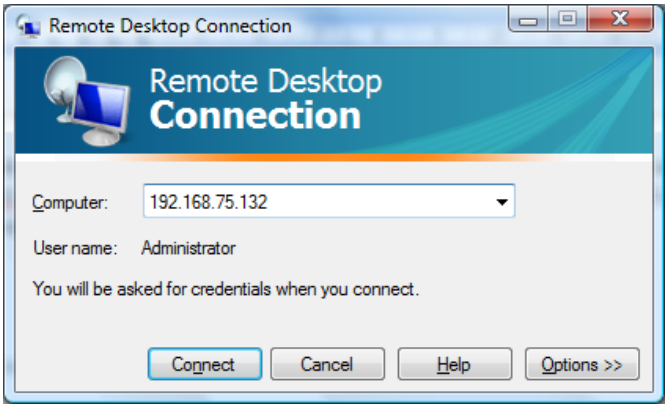


Figure 7.19 Remote desktop connection

SERVICE: Find the service?

7.12 From your host connect to Port 7 using the client.exe program from (Figure 7.20):

http://www.dcs.napier.ac.uk/~bill/dotNetClientServer.zip

☞ What is the service?

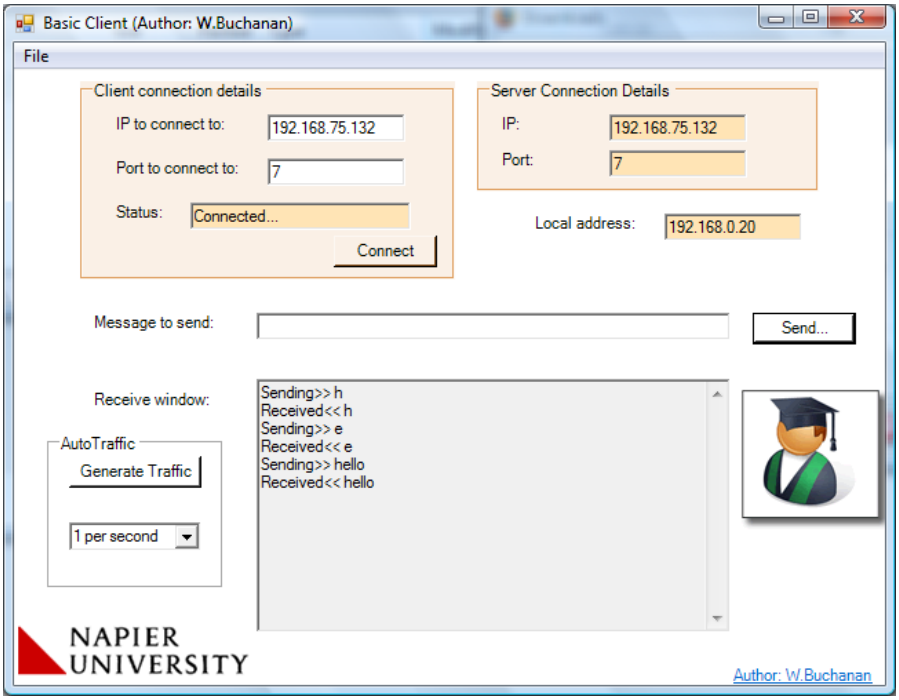


Figure 7.20 Port 7 access

7.13 Run the server (on port 1024) on WINDOWS2003, and run the client from DESKTOP, and make a connection between them.

AUDIT/LOGGING

7.14 Auditing and logging are important in terms of tracing activities.

- ☞ Check in the Event Viewer in WINDOW2003 (Figure 7.21), that the logon event has been added. How might this be used to trace activity?

- ☞ From Local Security Policy, find the option to change option so that Privileged Access is audited. What is the option:

- ☞ From Local Security Policy, find the option to change option so that the Guest Account cannot login. What is the option:

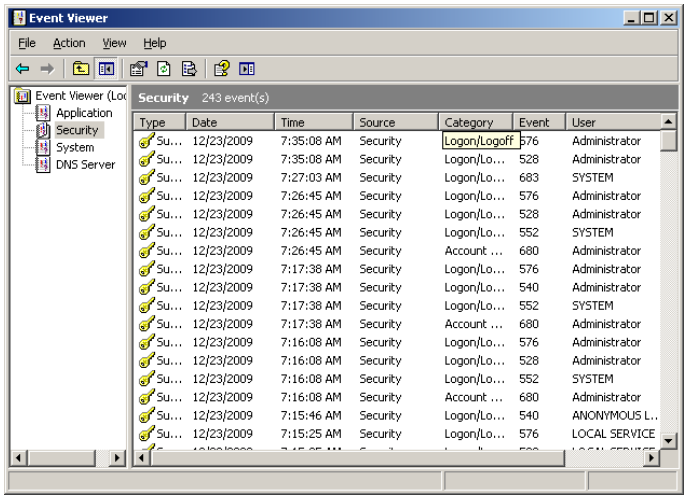


Figure 7.21 Event viewer

7.10 Linux Service tutorial

On-line demo:
http://buchananweb.co.uk/adv_security_and_network_forensics/unix/unix.htm

Note: The labs in this section require a virtual image defined in Appendix A.

PART B. This part of the lab has two elements: the host machine (DESKTOP) and the Linux virtual image (UBUNTU).

7.15 Run the Linux virtual image (User name: Administrator, Password: napier123). Within the virtual image, run a Terminal and determine its IP address using **ifconfig**.

What are the IP addresses of the server and the network address which will be used to connect to the virtual image:

7.16 From DESKTOP, ping UBUNTU, and vice-versa.

SERVICE: Web

7.17 In UBUNTU, go to the folder /var/www.

What are the names of the files in this folder:

7.18 From UBUNTU, run netstat –l, and determine the services that are running.

List some of the services:

7.19 From your host, connect to the Web Server from DESKTOP using http://w.x.y.z, where w.x.y.z is the IP address of UBUNTU (Figure 7.22). Repeat this using:

telnet w.x.y.z 80

and then enter:

GET /index.html

☞ What is the result, and how does it relate to accessing the home page of the Web server?

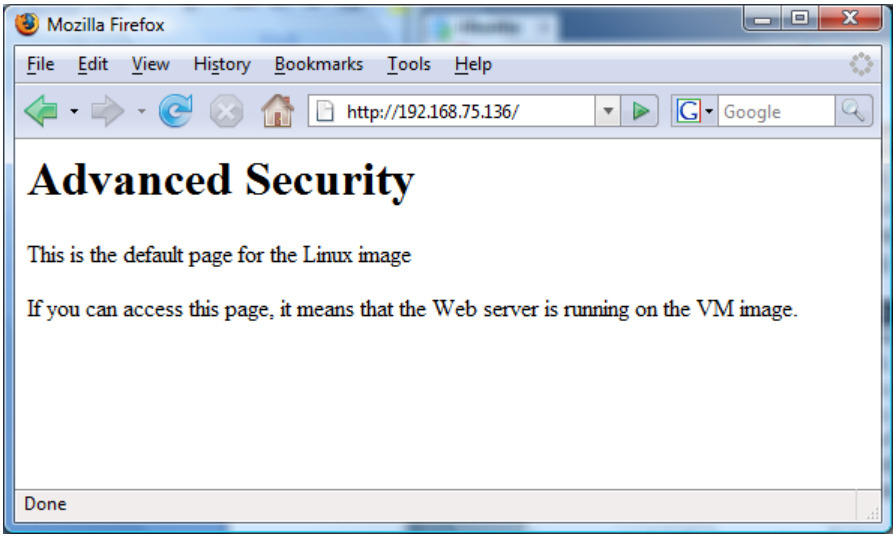


Figure 7.22 HTTP connection

7.20 On UBUNTU, using Screem HTML/XML Editor, open up the /var/www and create a new page, such as:

My Home Page

This is a sample ASP.NET page. Click [here] to return to the default home file.

☞ Can you access this page from the host (DESKTOP)?

☞ On UBUNTU, go to /var/log/apache2. What is the contents of the folder, and what do the files contain?

☞ How might these log files be used to trace malicious activity?

SERVICE: Telnet

7.21 From your host, connect to the Telnet Server from DESKTOP using telnet x.y.y.z, where w.x.y.z is the IP address of UBUNTU. Login in with napier (password: napier123).

☞ What is the default home folder for Telnet on UBUNTU (use pwd to determine the current directory):

Quit from Telnet, using the "exit" command.

SERVICE: FTP

7.22 From your host, connect to the FTP Server from DESKTOP using ftp://w.x.y.z where w.x.y.z is the IP address of UBUNTU (Figure 7.23). Repeat this using:

telnet w.x.y.z 21

and then enter the commands in bold (and note the commands that you get beside the sample return ones):

USER napier
331 Password required for napier.
PASS napier123
230- Linux ubuntu 2.6.31-14-generic #48-Ubuntu SMP Fri Oct 16 14:04:26 UTC 2009 i686
230-
230- To access official Ubuntu documentation, please visit:
230- http://help.ubuntu.com/
230-
230 User napier logged in.
PWD
257 "/home/napier" is current directory.
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (192,168,75,136,146,31)

LIST

Now FTP opens up a port for the data transfer. This is calculated from the last two digits of the Passive Mode response (227 response). It is calculated as 146 times the second last digital plus the last digital (31). So, in this case, it is:

Port = 146*256+31 = 37397

Next open up the data transfer by creating a new Telnet connection, such as:

telnet w.x.y.z 37397

- ☞ Can you access this page from the host?
- ☞ On UBUNTU, go to /var/log. View the syslog file (such as with **cat syslog**). What is its contents?
- ☞ How might these log files be used to trace malicious activity?
- ☞ View the contents of /etc/inetd.conf file. How is this used to enable services?

SERVICE: Remote Desktop

7.23 Download the VNC Client from:
http://www.realvnc.com/cgi-bin/download.cgi

then from the host, connect to UBUNTU using the Remote Desktop (Figure 7.24).

- ☞ Which is the service which is running on UBUNTU that allows the remote connection to happen?

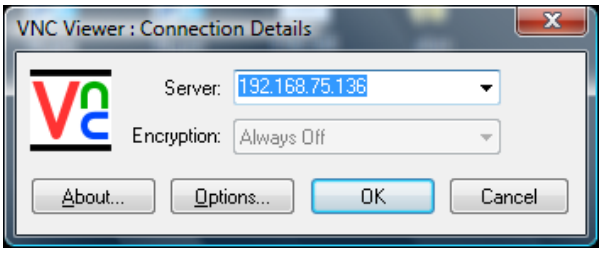


Figure 7.23 VNC Viewer