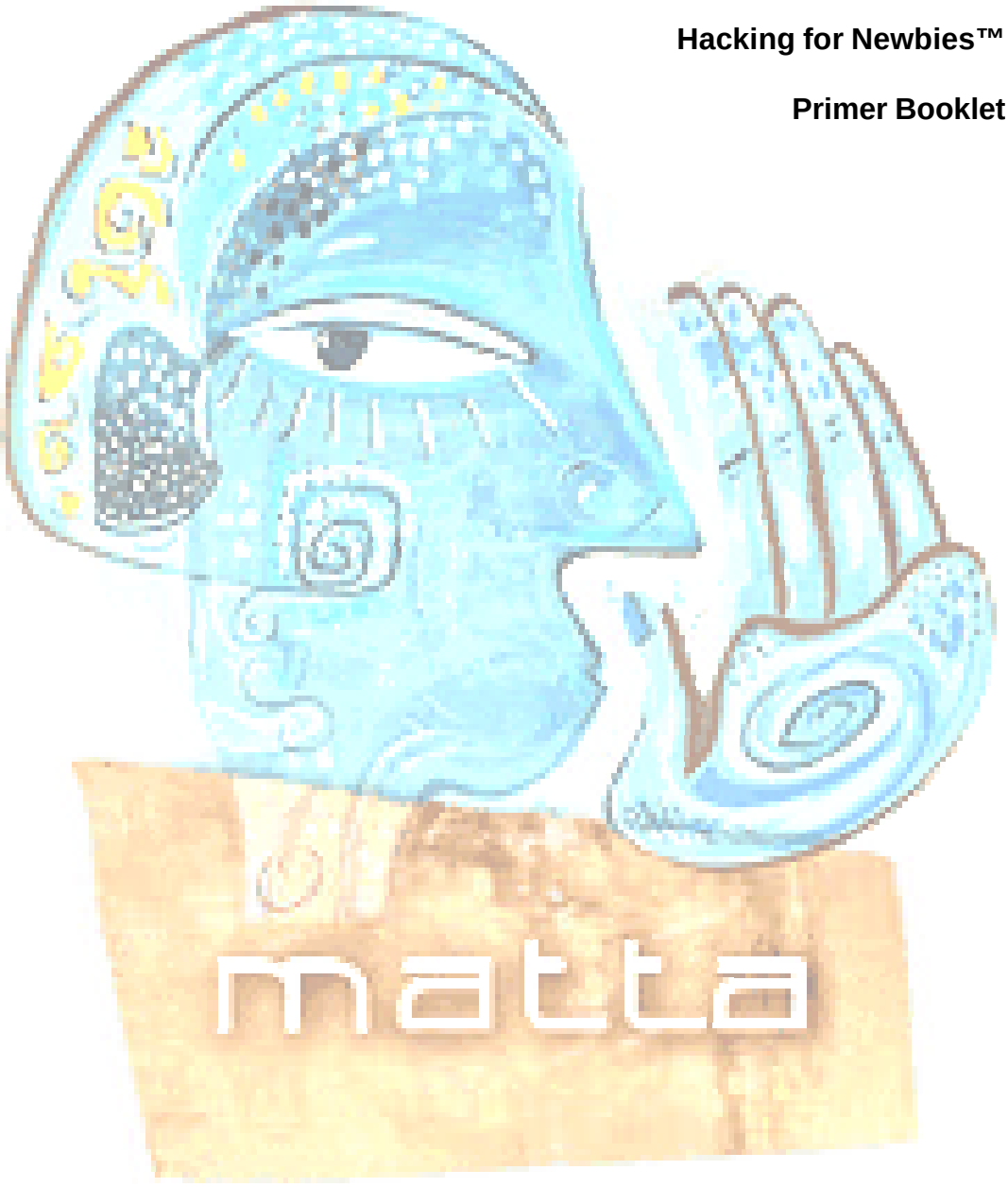
An Introduction to Internet
Attack & Penetration

Hacking for Newbies™

Primer Booklet



Matta Security Limited

16 – 19 Southampton Place
London WC1A 2AX

+44 (0) 8700 77 11 00

courses@trustmatta.com

http://www.trustmatta.com

IMPORTANT - COPYRIGHT AND TRADEMARK NOTICE

Attack & Penetration Group
Matta Security Limited
16 – 19 Southampton Place
London WC1A 2AX

courses@trustmatta.com

+44 (0) 8700 77 11 00

Matta Security Limited is registered in England, company number 4235114.

An Introduction to Internet Attack & Penetration

Table of Contents

Introduction 4

- Course welcome and introduction 4
- An overview of the risks and threats 5
- The anatomy of a hack 6

Hacking Concepts and types of Internet-based Attacks 7

- The fundamental hacking concept 7
- The reasons why software is vulnerable 8
- Network service vulnerabilities and attacks 9
- Password guessing and brute force attacks 12
- Viruses, worms and Trojan horse attacks 13

Closing Comments 14

Introduction

Matta is a fiercely independent information risk management company. Since early 2001, the firm has been operating primarily within the UK and Europe – offering bespoke security services, ranging from security assessment, to deployment and integration of authentication, VPN and firewall products.

The latest Matta offering to the already fertile marketplace is an independent, hands-on applied hacking course (with maximum class sizes of 6), broken down into 3 levels –

Level 1 – Hacking for Newbies™ (1 day course)
Level 2 – Hacking for Techies™ (2 day course)
Level 3 – Hacking for Spooks™ (3 day course)

This document is a structured and to-the-point introduction to Internet-based attack & penetration. Extracts are taken from our Hacking for Newbies™ course material, of which more information is available from http://www.trustmatta.com/services/courses.htm. In a bid to increase awareness of information security issues and show just how easy it is to break into computer systems and networks, Matta has decided to openly publish this information and further technical papers into the future. This document is intended as a primer, allowing a structured insight into attack & penetration techniques to be realised.

Matta actively create and present bespoke training programmes to clients with high requirements for information security expertise in-house, allowing them to assess internal network space and other elements themselves. Matta clients that can be mentioned include high street banks, stockbrokers and other financial companies with global footprints. Through it's strong and trusted background, Matta can deliver peace of mind.

For the information of those reading this document and looking to approve it for posting to public forums such as BugTraq, we do not plug our applied hacking courses (or any other commercial offering) at any further stages in this document.

An Overview of the Risks and Threats

The risks and threats to organisations with networked computer infrastructures are endless. Companies are being forced to embrace the Internet and the electronic channels that are built between un-trusted Internet-based hosts, and publicly accessible corporate servers. The Internet makes the World a much smaller place, allowing businesses to realise –

- An effective information presentation medium (corporate websites, et al)
- Online revenue streams (e-commerce and online ordering of products)
- Inexpensive global communications (e-mail, video conferencing, even VoIP)

By the same token, organisations are opening their networked environments up to the likes of Internet-based attackers (hackers, script kiddies & hacktivists) through deploying Internet-based points of presence for traffic to flow both to, and from the Internet into corporate network space. The very makeup and protocols on which the Internet has been built have inherent security weaknesses, which are exploited by hackers in many different situations.

Corporate network environments are often complex, and security weaknesses will almost certainly exist at one level or another. The problem companies and networked organisations have, is that security poses a great cost. Banks spend millions annually ensuring that their operating environments are secure, but they have a lot to lose. It is difficult for companies to see the return on investment (ROI) regarding security, as it often does not contribute directly to the bottom line.

In terms of large organisations that have fallen foul to Internet-based attack, the following are worth mentioning –

www.yahoo.com	www.infowar.com/hacker/hack_121397a.html-ssi
www.nasdaq.com	www.zdnet.com/zdnn/stories/news/0,4586,2334751,00.html
www.playboy.com	cnn.com/2001/TECH/internet/11/20/playboy.hacked/index.html
www.cduniverse.com	www.internetnews.com/ec-news/article/0,,4_289221,00.html

It is extremely difficult for companies especially to keep intruders out of their Internet-based networks, due to the nature of the networks deployed and their openness to allow Internet traffic onto them. Throughout this course, we will highlight the threats to such networks, and give you a structured insight into hacking techniques.

The Anatomy of a Hack

A typical Internet-based attack against an organisation involves the following being undertaken –

1. Identification of Internet-based points of presence
2. Network scanning & reconnaissance
3. Accessing the target system or network
4. Carrying out objectives
5. Manipulation of logs and system files

A successful attack can take anything from seconds, to months, even years to complete. In less direct cases, involving compromising trusted hosts and networks. In complex cases, the attacker will use tools such as 'trojans' and 'sniffers' on the server, then wait for a user to access a trusted system or network, and piggy-back on his connection or use his credentials to later access the target system.

Such an indirect network-based attack against an organisation involves the following being undertaken –

1. Identification of the target host and network space
2. Network scanning & reconnaissance of the target host and network space

3. Identification of trusted hosts and networks and servers
4. Network scanning & reconnaissance of trusted networks and servers
5. Accessing the trusted networks and servers
6. Deploying network manipulation or sniffing technologies
7. Logging user activity by the deployed hacker technologies

8. Accessing the target system or network using valid user credentials
9. Escalation of privileges to ensure a decent level of access
10. Carrying out objectives
11. Manipulation of logs and system files

Many 'secure' networks nowadays are compromised using indirect network-based attacks. Examples include the banner server at SecurityFocus.com, where 'Fluffy Bunni' changed all of the banners displayed to his own slogan. Attackers with knowledge of unpublicised vulnerabilities such as the SSH vulnerabilities that have only been realised over the last year, but known in some circles since the first releases of SSH 1, pose great danger to networks that are considered safe from attack, as many IT managers and systems staff simply trust their network services.

Hacking Concepts and types of Internet-based Attacks

The Fundamental Hacking Concept

> "Hacking is the process of influencing a computer system in such a way that it performs an action that is useful to you."

A simple example of this is to think of a search engine, which is programmed to accept a query, cross-reference it with a database, and provide a list of relevant sites. The search engine processes the query locally on the server to generate a result. Through understanding the potential security vulnerabilities in search engines and the way that they are developed, a hacker could manipulate the search engine to look for the root entry of the /etc/passwd file if the search engine did not perform sanity checking of the queries and arguments passed to it.

Not long ago, The main US Pentagon, Air Force and Navy web servers (www.defenselink.mil, www.af.mil and www.navy.mil) were all vulnerable to a very similar attack, as they used a search engine called multigate, where a string such as the following could be passed to the engine resulting in the server password file being presented –

http://www.defenselink.mil/cgi-bin/multigate/search?SurfQueryString=root&f=/etc/passwd

Since then the multigate system has been superseded, although information is still available about multigate and its uses, from Google –

http://www.google.com/search?q=cgi-bin%2Fmultigate&hl=en

In this environment, these high-profile military websites were properly protected at network level by firewalls and other security appliances. However, by the very nature of the massive amount of information presented by these sites, a search engine was deployed (presenting vulnerabilities at system and application level). A key point to remember regarding attack & penetration is this –

> "It is not impossible to compromise a system, only improbable"

The Reasons why Software is Vulnerable

In a nutshell, software is vulnerable because there are costs associated with ensuring that software is secure to an acceptable level. Corporations such as Microsoft have historically marketed insecure operating platforms (the Windows series, Outlook, Internet Explorer), which are in turn deployed by corporations in business-critical environments.

Many software programmers are pushed to write code that works well, not code that works and is robust and secure when attacked. Programmers are not aware of techniques that can easily be used at the development stage to ensure that arguments passed to routines are sanity checked and controlled. In order for a secure program to be developed, the interaction of that program with the environment in which it is run should be controlled at all levels, no data passed to the program should be trusted or assumed to be correct. 'Bounds checking' is a simple term that would go a long way in the development community to creating more secure software.

So a developer writes a network service daemon (such as telnetd or ftpd), but does not include routines for bounds checking of the length of commands and arguments that are sent to that service by the end user. Our hacker undertakes an attack against the code, and realises that by issuing a certain command with an extremely long argument (over 8000 characters), the service crashes. Upon testing the vulnerability a little further, he finds that his long argument is being written onto the executable stack, which is causing the service to crash. Eventually an 'exploit' is written by the hacker for this vulnerability, allowing him to run arbitrary code on any host running the vulnerable service.

If money was invested by developers to ensure that their source code is pro-actively audited and assessed, operating platforms which are deployed in mission-critical environments of thousands of businesses, would be at far less risk from attack.

Network Service Vulnerabilities and Attacks

This paper concentrates on Internet-based network service vulnerabilities, in the way that software running at kernel and system daemon level processes data received. Network service vulnerabilities can be categorised into the following groups –

- Process manipulation, where crafted data fed to network service software affects the logical program flow, resulting in the system being compromised or affected by an attacker launching one of the following types of attack –
    - Overflow-based attacks, where incorrect bounds checking of data processed allows attackers to manipulate data in memory, resulting in either arbitrary code being run directly, or system processes being indirectly compromised.
    - Non overflow-based attacks, where the system is compromised in a less complex manner by abusing insecure network service ‘features’, allowing attackers to usually run system commands and access data without the need to manipulate memory.
    - Denial of Service (DoS) attacks, where a network service is not necessarily vulnerable to a degree where a system compromise occurs, however crafted data being sent to the service can cause system resource starvation or an outright system crash.
- Information leaks, where network services are not vulnerable to the extent of a direct system compromise occurring, but present attackers with information that is useful. It is not fair to group information leak vulnerabilities under the process manipulation group, as in many cases network services will leak information as a ‘feature’.

Process Manipulation – Overflow-based Attacks

An example of a simple remote buffer overflow is the Microsoft Internet Information Services (IIS) 4.0 and 5.0 vulnerability, regarding the .IDA ISAPI filter. Insufficient bounds checking is performed by the ISAPI filter for arguments passed to .IDA files, resulting in a buffer being overflowed and arbitrary code potentially being run on the system. The Code Red worm primarily abused this vulnerability to spread, issuing the following web request to vulnerable servers the world over –

```
http://www.example.com/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u68
58%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
```

In the above example, the first 224 N’s of the argument (in purple) passed to default.ida file (in blue) are used to fill up the buffer. The raw machine code afterwards (in red) is injected into the executable stack area of memory, resulting in the code being run on the server itself.

Process Manipulation – Non Overflow-based Attacks

BSD 4.4 Routed Trace File Exploit

A good example of a non-overflow based attack is that of the BSD 4.4 routed trace file exploit, where a debugging option can be specified within an RIP (routing information protocol) packet being sent to a vulnerable routed daemon (listening on UDP port 520), resulting in any file on the file system being appended to or created, containing debug and tracing information.

This option was obviously intended for network debugging purposes within the routed system, but can be abused because sanity checking of the filename being specified to write the debugging information to is not performed.

More information about the BSD 4.4 routed trace file vulnerability is available from –

http://www.insecure.org/sploits/routed.tracefile.html
http://ciac.llnl.gov/ciac/bulletins/j-012.shtml

Web-based CGI Exploits

A second example of a non-overflow based attack would be of a vulnerable CGI script running on a web server, where a command such as 'cat /etc/passwd' can be run on the server itself. A classic example of this is the PHF attack which was used in 1996 to compromise the CIA web server by issuing the following request from a standard web browser –

http://www.cia.gov/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd

Here is a breakdown of this timeless attack –

http://www.cia.gov	the transfer protocol and target server address
/cgi-bin/phf	the path to the vulnerable script, phf in this instance
?Qalias=x	the initial query sent to the phf script
%0a	hex for a carriage return, used as a 'shell escape'
/bin/cat	the command we want to run on the server, /bin/cat
%20	hex for a space, so that an argument can be passed to cat
/etc/passwd	the /etc/passwd file is specified

Through using a simple shell escape (the %0a carriage return), we are able to run any command locally on the server as the 'nobody' user. In this instance we have decided to view the /etc/passwd file, containing user information (and encrypted passwords in insecure cases).

There are now over 250 similar CGI script attacks, of which more information can be found on security web sites such as www.SecurityFocus.com and www.PacketstormSecurity.org. A Unix-based application level vulnerability scanner exists called 'CGIchk', of which details can be found in the tools listing at the end of this paper. CGIchk tests for the presence of over 250 vulnerable CGI scripts on a given server.

Process Manipulation – Denial of Service (DoS) Attacks

An example of a DoS vulnerability is in the Linux kernel (2.0.34 and before), where an 'off-by-one IP header bug' exists in the IP fragmentation code. An attacker sending a series of specially crafted IP packets containing malformed IP header information can crash an entire Linux server at kernel level by exploiting this vulnerability, resulting in Denial of Service (DoS).

Rhino9 released an advisory regarding this problem some time ago, available from –

http://www.technotronic.com/rhino9/advisories/06.htm

There are many vulnerabilities like this present in many different platforms and pieces of software running at either kernel or service daemon level. Software developers have a complex job ensuring that all data passed to a network service is sanity checked and at the same time, ensuring that networking code is fast and efficient.

Information Leaks - Solaris Fingerd

A simple example of an information leak is a vulnerability in the Solaris finger daemon, where a request of "1 2 3 4 5 6 7 8 9" results in user details being revealed. Even the latest release of the Solaris operating platform (version 8 at the time of writing) is vulnerable to this attack, and below is a working example of this attack launched from a Unix command prompt –

```
$ finger "1 2 3 4 5 6 7 8 9 0"@mail.example.com
[mail.example.com]
Login       Name               TTY         Idle    When    Where
root     Super-User            console      <Jun  3 17:22> :0
admin    Super-User            console      <Jun  3 17:22> :0
daemon          ???                         < .   .   .   . >
bin             ???                         < .   .   .   . >
sys             ???                         < .   .   .   . >
adm      Admin                              < .   .   .   . >
lp       Line Printer Admin                 < .   .   .   . >
uucp     uucp Admin                         < .   .   .   . >
nuucp    uucp Admin                         < .   .   .   . >
listen   Network Admin                      < .   .   .   . >
nobody   Nobody                             < .   .   .   . >
noaccess No Access User                     < .   .   .   . >
nobody4  SunOS 4.x Nobody                   < .   .   .   . >
informix Informix User                      < .   .   .   . >
crm      Chris McNab           pts/0          1 Tue 09:08  onyx
axd      Andrew Done           pts/4         3d Thu 11:57  194.6.18.2
```

From gleaning user details such as this, the attacker can either launch a brute force password guessing attack directly against the server, or attempt to compromise trusted hosts where users are known to log in from, such as 194.6.18.2 in the above example.

Information Leaks - Sendmail

A second example of information being leaked by a network service is Sendmail, an SMTP e-mail relay system used by hundreds of thousands of networks globally to transfer email. Even if 'switched on' systems administrators disable the EXPN and VRFY functionality that is traditionally used by attackers to glean username and GECOS information, valid system logon details can still be gleaned in the following fashion –

```
$ telnet mail.example.com 25
Trying 194.6.18.10...
Connected to mail.example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.9.3/8.9.1; Mon, 23 Nov 2001 14:21:17
+0200 (CEST)
HELO example.com
250 example.com Hello onyx.example.com [194.6.18.3] (may be forged),
pleased to meet you
EXPN root
502 Sorry, we do not allow this operation
MAIL FROM: test@test.org
250 test@test.org... Sender ok
RCPT TO: test
550 test... User unknown
RCPT TO: sybase
550 sybase... User unknown
RCPT TO: informix
250 informix... Recipient ok
```

Through abusing the Sendmail service at mail.example.com, we have found that 'informix' is a valid system account. Leaking valid account information can result in a system compromise occurring, through brute force password guessing being undertaken by determined attackers.

Password Guessing and Brute Force Attacks

Coupled with information leak vulnerabilities described earlier in this paper, password guessing and brute force attacks are becoming a more favourable way for attackers to compromise servers that are not directly vulnerable to remote process manipulation attacks.

When undertaking a brute force password guessing attack against a server, you are assessing the security policy in place on that host, and the standard of passwords used within the target organisation. The Matta Attack & Penetration Group has undertaken security assessment work in the past where username / password combinations such as test / test are found in use on mission-critical servers, where a weak security policy is in place, and access controls mechanisms are almost useless.

The important factor when launching a brute force attack against a host is the efficiency with which you can launch the attack, which is usually determined by how many failed login attempts you can accumulate before you are disconnected and have to reconnect. Many POP3 services act as an effective point for brute force attacks to be conducted against, as they often allow users to enter the wrong password many times, and do not log failed login attempts. SSH service daemons will log all failed logon attempts, as will some telnet service daemons when attacked in this manner.

Viruses, Worms and Trojan Horse Attacks

Viruses, worms and Trojan horse programs are traditionally thought of as an indirect Internet-based attack, but can be used to devastating effect directly by highly determined attackers aware of the inner workings of their target network, including areas of trust and weakness.

When used directly by determined attackers against a specific organisation, a virus or Trojan horse can be planted onto a target network by adopting techniques to circumvent anti-virus systems in place. The Matta Attack & Penetration Group are aware of vulnerabilities in SMTP virus scanning mail gateway software such as MAILsweeper, which can result in viruses and malicious code being passed straight through the gateway without being checked. Extreme business downtime and costs can be incurred by victims of determined attacks such as this, as the attacker will combine anti-virus circumvention with mail spoofing, and other techniques.

Below is an example of an effective virus attack as conceptually outlined above, including MAILsweeper vulnerabilities being exploited, and mail spoofing techniques being adopted –

- The attacker undertakes reconnaissance to identify
    - The victim and his e-mail address
      (e.g. "John Smith" <john.smith@example.com>)
    - The Internet e-mail gateway for that domain and associated security
      (e.g. mail.example.com running MAILsweeper content checking)
    - A trusted party to spoof the e-mail from
      (e.g. "Tom Jones" <tom.jones@example.com>)

- The attacker creates a highly potent new virus strain to specifically 'detonate' and damage the target internal network space when the victim opens the e-mail. An example would be to take the 'Bad Trans' virus or one of its variants, modify it heavily, and also attach 4MB worth of A's to the end of the virus as a payload. Upon the virus being opened and spreading, thousands of 4MB e-mail messages will start to be sent around the target network, soon crashing Microsoft Exchange mail servers and Outlook clients company-wide.

- The attacker builds the email message to be sent to the victim, including spoofed sender information and the virus attachment itself, concealed by building a non-standard email message which will not be opened and scanned by MAILsweeper, but will be opened on the victim's workstation using Microsoft Outlook.

  Information about exploiting MAILsweeper and building malformed e-mail messages such as this can be found on the SecurityFocus Vuln-Dev mailing list, at http://lists.insecure.org/vuln-dev/2001/Jul/0093.html. The Matta Attack & Penetration Group is currently testing SMTP content checking products for circumvention vulnerabilities such as this, and will be posting a full analysis to the BugTraq security mailing list in 2002.

- The attacker connects to the Internet email gateway and plays a script containing specific SMTP spoofing commands, and the bespoke e-mail message that has been built. With any luck, the message should be sent through the MAILsweeper system and to the victim, it's then just a case of him opening it.

In this instance, four specific vulnerabilities are being exploited sequentially by the attacker to achieve his goal –

- The MAILsweeper e-mail gateway does not check malformed e-mail messages.
- The SMTP protocol (which most Internet-based e-mail is sent and received by) allows Internet-based attackers to send spoofed e-mail from the Internet into internal corporate addressing space, appearing to have originated internally.
- The victim's email client (Microsoft Outlook) happily opens malformed and spoofed e-mail messages.
- The victim trusts e-mail content and attachments from his friends and anti-virus systems that are deployed to prevent e-mail viruses being received.

Seemingly small vulnerabilities such as these can be abused in many cases to present attackers with an opportunity to cause damage. Security systems such as MAILsweeper should not be ultimately trusted, and second or even third-line defences should be considered in-line with corporate security policy, to prevent determined attackers from being successful.

Closing Comments

At the time of publishing this primer booklet presenting a structured introduction into Internet Attack & Penetration, the following Matta white papers are available publicly, giving clear technical insight into the issues at hand –

- IP Network Scanning & Reconnaissance Technical Primer
- Denial of Service Technical Primer
- Using DNS to Effectively Map Networks

Available from the Matta website at http://www.trustmatta.com, along with other information and security white papers which may be of interest. Into the future, technical information regarding specific attack types and hacker strategies will become available, so keep posted.