網路入侵偵測系統之研究

Study on Network Intrusion Detection System

期末報告

國立交通大學資訊工程學系

中山科學研究院資訊通訊研究所指管通情組

執 行 時 間　：90 年 01 月 01 日 至 90 年 12 月 31 日

計畫執行單位　：　國立交通大學　資訊工程學系

計畫主持人　　：　蔡文能　副教授

日　　期　：中華民國九十年十二月三十一日

NSC 90-2623-7-009-013

目錄

計畫摘要

網際網路(Internet)的蓬勃發展給企業帶來許多商機，給個人帶來了許多便利。虛擬私有網路（Virtual Private Network，簡稱 VPN）可以利用 Internet 結合多個分區網路，讓企業得以享受資源共用的好處，省去租用專線的昂貴費用，可以提升網路資源的利用率，提高企業內的生產力。然而，既然 Internet 是開放的，它也引來許多新問題、新風險，例如非法入侵(Intrusion)、網路攻擊(Attack)、電腦病毒(Computer virus)，甚致電子郵件遭受網路廣告疲勞轟炸等等。政府機構之網路若遭到攻擊或入侵，輕則人民的權益受影響，重則影響國家安全。

在 RFC1636 文件中正式將防火牆列入資訊安全機制；本計畫就是要研究類似防火牆的網路攻防技術。使得用此技術所架設的網路在受到網路攻擊或入侵時更有機會正常運作。在此我們把意圖癱瘓系統或網路的行為歸類為攻擊，把意圖進入系統竊取資料或進入系統做一些事的歸類為入侵。

常見的網路攻擊是 Denial of Service (DoS)。就是利用系統或 TCP/IP 的漏洞，讓網路塞滿了垃圾封包，而導致系統或網路無法正常服務。例如"Ping of Death" 和 "Teardrop" 攻擊，"SYN Flood" 和 "LAND" 攻擊，都可能造成被攻擊系統當機。又如"smurf"攻擊，利用不斷的對路由器發出 ICMP 要求封包，可能使網路充滿 ICMP 要求與回應封包而讓網路交通中斷。

網路的入侵者通常利用系統的漏洞或是後門來入侵系統。後門也可能是我們網路內部人員自己抓回的特洛依(Trojan)病毒程式而不自知。特洛依(Trojan)病毒程式一般又稱後門程式。

本計畫報告結構如下：1、簡述國內外有關本計畫之研究情形與相關背景知識；2、弱點資料庫收集；3、各類攻擊技巧；4、入侵偵防系統雛形；5、結論。

1. 國內外有關本計畫之研究情形與相關背景知識

自從1990年WWW出現後網際網路(Internet)的蓬勃發展給企業帶來許多商機，給個人帶來了許多便利，因此網路已經成為日常生活不可獲缺的一項工具。但它也引來許多新問題、新風險，如非法入侵(Intrusion)、網路攻擊(Attack)、電腦病毒(Computer virus)，甚致電子郵件遭受網路廣告疲勞轟炸等等。

因此，除了良好的防火牆之外，還要需要網路攻擊的偵測與阻絕的輔助，以確保網路的安全。在此，我們把意圖癱瘓系統或網路的行為歸類為攻擊，把意圖進入系統竊取資料或進入系統做一些事的歸類為入侵。

常見的網路攻擊是 Denial of Service (DoS)。許多攻擊必須先以 DoS 阻絕正常主機或網路服務再發動其他弱點探測，如 TCP 序號攻擊（Sequence Number Attack）。

1.1 常見的 DoS 攻擊

目前常見的有三種型態的 DoS 攻擊，他們皆是利用 TCP/IP 的漏洞或是網路程式的漏洞，讓系統當機或是讓網路塞滿了垃圾封包，而導致網路停擺。

1.1.1 利用主機系統的 TCP/IC 漏洞

第一種是利用主機系統的 TCP/IC 漏洞，例如"Ping of Death" 和 "Teardrop" 攻擊。

"Ping of Death"利用"ping"這支工具程式來產生超過 IP 協定所能夠允許的最大封包。當這個封包送到沒有檢查功能的系統，則可能會造成系統當機。

Teardrop 攻擊則是利用 IP 封包重組的漏洞。當資料經由網路傳送，IP 封包經常會被切割成許多小片段。每個小片段和原來封包的結構大致都相同，除了一些記載位移的資訊。而 Teardrop 則創造出一些 IP 片段，這些片段包含重疊的位移值。當這些片段到達目的地而被重組時，可能就會造成一些系統當機。

1.1.2 利用 TCP/IP 規格本身的漏洞

第二種是利用 TCP/IP 規格本身的漏洞，例如"SYN Flood" 和 "LAND" 攻擊。一般來說，甲端想和乙端的應用程式溝通，甲端會先送出 SYN 封包給乙端。當乙端收到之後，會回應一個 SYN-ACK 封包給甲端，最後甲端會送出一個 ACK 封包給乙端當作確認，這稱為 three way handshake。在完成這些程式之後，甲端和乙端才可以開始收發資料。

"SYN Flood"攻擊會針對欲攻擊的系統發送一連串的 SYN 封包，每個封包會讓系統回應一個 SYN-ACK 封包，然後系統會等待對方送出 ACK 封包。系統貯列裡的 SYN-ACK 封包必須等到接收到對方的 ACK 封包或是超過逾時時間之後才會移除。最後系統貯列會因為充滿了 SYN-ACK 封包而造成無法再處理其他使用者的要求。

"LAND"攻擊因出現在一支叫 LAND.C 的程式而得名，它會送出一連串的 SYN 封包給網路上的系統，並且利用"IP Spoofing"的技術讓系統以為這些封包都是他自己發送的。(就是偽造一個 src.addr/dst.addr 都是被攻擊者的 IP，src.port 與 dst.port 均為 xxx 的封包且 SYN flag on，如此簡單到不能再簡單的封包就能癱瘓網路!)當系統在處理這些封包時，由於它自己並不能回應給自己，而造成系統當機。

1.1.3 廣播大量封包

嚴重的 "smurf"攻擊。它會對網路直接廣播，造成網路很快的充滿垃圾封包而中斷。"smurf"會不斷的對路由器發出 ICMP 要求封包(即 ping)，而且會將封包目的位元址改為廣播位址，讓路由器對網路上每一台機器發出 ICMP 要求封包，造成網路充滿 ICMP 要求和回應封包。如果"smurf"將 ICMP 要求封包的來源位址改為另一個網路的主機，則結果兩個網路都會充滿 ICMP 封包而讓網路交通中斷。

1.2 分散式 DoS 攻擊

更進一步網路 DoS 攻擊為分散式阻斷服務攻擊 Distributed Denial of Service (DDoS)，攻擊者在已經遭受入侵的 Internet 主機上安裝攻擊程式，同時發動一波又一波的攻擊，造成目標主機所提供之服務癱瘓甚至中斷，如圖 1 所示。常見的攻擊方式有 Trinoo（或是 Trin00），另一種稱為 Tribe Flood Network（簡稱 TFN）和 TFN2K。

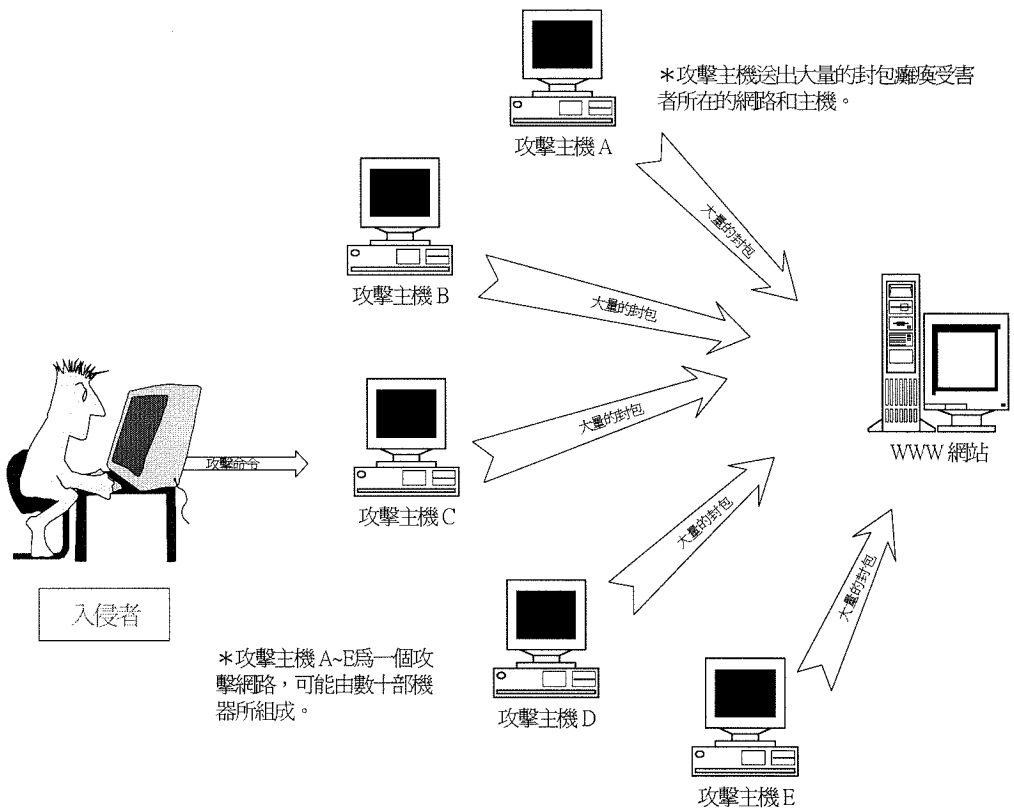


圖 1. 分散式攻擊示意圖

1.2.1 Trin00

一個 Trinoo 網路包含數個主控端（masters，或稱之為 servers）和大量的常駐程式（daemons，或稱之為 clients）。入侵者先連接上主控端並啟動程式，然後下達攻擊指令，告訴它攻擊哪些目標的 IP 位址後，主控端就會跟所有的常駐程式溝通，由常駐程式發動攻擊。

1. 入侵者➔主控端：目的連接阜 27665/TCP

2. 主控端➔常駐程式：目的連接阜 27444/UDP
3. 常駐程式➔針對攻擊目標的 IP 位址，不同的連接阜，發送大量的 UDP 封包，造成對方的主機系統負荷過重，而無法提供正常的服務。

1.2.2 TFN(Tribe Flood Network)與 TFN2K

TFN 和 Trinoo 是相當類似的攻擊技巧，只不過它除了 UDP flood 的攻擊外，它還可以進行 TCP Syn flood，ICMP echo request flood，和 ICMP directed broadcast（例如 Smurf）的阻斷攻擊。TFN 網路的架構和 Trinoo 一樣，不過 TFN 的主控端是透過 ICMP echo reply 的封包（內含 16-bit 的二進位數字放在 ID 的欄位裡面，而資料欄位可能會擺放任何參數）來和 TFN 常駐程式溝通。二進位的數字代表不同的控制指令，並且在編譯的時候已經決定了。

在啟動 TFN 主控端時，需要入侵者提供所有可利用的 TFN 常駐程式的列表。某些報告指出最近版本的 TFN 主控端可能使用 blowfish 的加密方法來封裝列表。而且 TFN 可能有遠端檔案拷貝的能力（remote file copy，例如 rcp），藉此可以自動的散布新的 TFN 常駐程式，或是有新的軟體版本時可以自動更新。

1.3 路由器攻擊

尚有一些攻擊方式是針對 Router 來進行破壞，而一旦 Router 被癱瘓之後，所造成的將是整個 Subnet 無法正常工作，目前有幾種攻擊方式 如：在內部閘道協定中使用的 OSPF protocol 目前就存在幾種攻擊模式。

1.3.1 Max Seq. Attack：

因為，在 Link State Advertisement(LSA)中，有一個欄位是用來區別哪一個 LSA 為較新的，而若有一個惡意的 Router 將這個欄位設為最大值（表最新的），並更改 LSA 的資料，則在這個 Subnet 其他的 Router 就會根據這個 LSA 來更新

他的 Link-state 的資訊，且不管其他的 LSA 的資訊。需等原來發出該 LSA 的 Router 發現該 LSA 資訊錯誤後，再發出一個更新的 LSA 資訊，而若此時該惡意的 Router 依然繼續使用此方法來攻擊該網路，則整個網路架構會很不穩定，因為，網路的 Link-state 會一直變，而造成網路變慢。

1.3.2 Max Age. Attack：

在 Link State Advertisement(LSA)中，有一個欄位是用來決定 LSA 的 Age，當此欄位設為最大時，表示要清除這個 LSA，因此，若有一個惡意的 Router 一直將 LSA 的這個欄位設為最大值(表要清除)，則在這個 Subnet 其他的 Router 就會從資料庫中刪除這個 LSA 的資訊，而原來發出該 LSA 的 Router 發現該 LSA 錯誤後，再發出一個新的 LSA，讓其他 Router 在加入該 LSA 的資訊。而若此時該惡意的 Router 依然繼續使用此方法來攻擊該網路，則整個網路架構也會很不穩定，因為，其他 Router 一直刪除 LSA 而不知道 LSA 真正的資訊。

2. 弱點資料庫收集

許多網路攻擊方式都是針對通訊協定本身實作上的缺陷，尋找可能的弱點。因此，建立系統弱點資料庫, 可協助系統管理者找尋系統安全相關資訊；藉由補強程式的收集,可方便系統管理者更新系統。但是換另一個角度來看，因為這些弱點資料庫以公開的方式發佈在各種媒介上，因此攻擊者也可以利用這些資料來尋找受害主機，減少花在猜測與探測弱點的時間。所以，身為一個安全管理人員，不能不時時刻刻注意網路上公布的弱點資料，隨時更新以防系統暴露在毫無防禦的公開網路環境下。

這類的資料庫大致上可分為：作業系統弱點資料庫、通訊協定弱點資料庫、病毒資料庫與其他類型的弱點資料庫。

2.1 作業系統弱點

先來看一則新聞：

> [Yahoo 新聞電子報 2001/12/22 作業系統有蟲　不稀奇]
>
> 微軟又出紕漏了嗎？其實，這次 Windows XP 裡的安全漏洞，已經不是微軟產品第一次被人抓到瑕疵，幾乎微軟的每一代作業系統，都曾發生好多次被有心人士找到漏洞加以利用的事件。也因此不斷推出產品更新程式，幾乎已經成為微軟例行性工作；定時關心有無需要下載更新程式，也已經成為使用者不得不關心的重要事情。
>
> 號稱有史以來測試最嚴格的微軟作業系統 Windows XP，還是難逃被人抓到漏洞的命運，而使用者似乎也早已經習慣了面對微軟產品有「蟲(Bug)」的新聞。之前嚴重肆虐全球的 Nimda、情書等病毒，也都是利用微軟 IE 或 IIS 的程式設計漏洞，所設計的新一代病毒。甚至連屬於.NET 世代的微軟 Passport 單一帳號，也在日前被發現有讓使用者機密財物資料曝光的漏洞。
>
> [省略...]

其實這一類的報導早已司空見慣，也說明了軟體開發者並無法顧慮到所有可能發生的錯誤情形。尤其使用者對於軟體品質上的要求，使得現今軟體研發的複雜度逐漸提高，程式碼動輒數千數萬行以上，作業系統更為其甚者。在本

計畫中我們分別針對不同的作業系統，收集並整理其弱點資料庫。

2.1.1 Windows 2000

2.1.1.1 標題 : Window 2000 漏洞

大部分的 Window 2000 Server 服務就算是面對簡單的攻擊，也可能受創。這些簡單的攻擊幾乎會讓 Window 2000 Server 在很短的時間內提高 CPU 使用至百分之百。然而對攻擊者來講，他並不需要很大的成本。

易受攻擊的 Port 如：

- TCP port 7、9、21、23、7778。
- UDP port 53、67、68、135、137、500、1812、1813、2535、3456。

這些攻擊沒有立刻鎖住機器，他只過度利用 CPU 的資源。

攻擊方法是利用 Linux 的 netcat 指令僱用/dev/zero 當 input，

例如： nc target.host 7 < /dev/zero (攻擊 TCP port 7) 或者

nc -u target.host 53 < /dev/zero (攻擊 UDP port 53)。

由於很多服務會受到影響, 這會允許很快及簡單的遍撒攻擊。

參考：http://www.microsoft.com/technet/security/bulletin/MS00-049.asp

2.1.1.2 標題：針對微軟 Exchange server 所做的阻絕攻擊

攻擊者可以使用 Exchange 的 RPC（Remote Procedure call）服務來對此伺服器發動阻絕攻擊。而藉由一些常用的程式（如舊版的 Outlook 或 Windows95）便能輕易地將這項弱點散播出去。

參考：http://www.securiteam.com/windowsntfocus/
Denial_of_Service_attack_against_MS_Exchange_servers__DCOM__ncacn_http_.html

2.1.2 Windows 9x

2.1.2.1 標題：路徑名稱中包含 DOS 裝置名稱

在微軟 Windows95 以及 98 中，在解析檔案路徑名稱時可能發生錯誤。對於 DOS 來說，裝置名稱屬於保留字，所以不能用來當作檔案或者是目錄名稱。假如使用者試圖利用某個具有 DOS 裝置名稱的檔案路徑來存取檔案或是目錄，則此舉動會被視為不合法的行為，系統回傳錯誤訊息。然而，假如路徑名稱包含多個 DOS 裝置名稱，則系統可能會當掉。

參考：http://icat.nist.gov/icat.cfm?cvename=CVE-2000-0168

2.1.2.2 標題：微軟 Windows98/2000 Folder.htt 弱點

微軟Windows系統可以透過 web-based 檔案夾顯示的能力有個致命的弱點，就是透過偷改Folder.htt 當案(用ActiveX控制項ShellDefView的函數或方法InvokeVerb)可以塞入特洛伊木馬程式。

參考：http://www.securityfocus.com/bid/1571

http://icat.nist.gov/icat.cfm?cvename=CAN-2000-0790

2.1.3 Windows NT

2.1.3.1 標題：WinNT spooler 服務漏洞

在Windows NT 中的spooler服務讓本地端的使用者能夠新增自己的 .dll 檔案，並且讓spooler以系統權限執行這些 .dll檔。其中最主要的問題發生在AddPrintProvider()這個函數，這可能使得權限躍升到管理者的權限。

參考：http://icat.nist.gov/icat.cfm?cvename=CVE-1999-0899

2.1.4 Red Hat Linux

2.1.4.1 標題：Linux UMB scheme 有漏洞

根據 Linux Advisory , UMB scheme 含有一些檔案提供 world-writable 權限. 甚至會被利用拿到 root 的權限.

參考：http://www.securityfocus.com/bid/1551

2.1.5 Sun Solaris

2.1.5.1 標題：認出 Solaris 的 OS type

使用 sing 送正常 Address Mask Request 給 SUN Solaris 2.7 機器 , 結果如下：

```
# ./sing -mask IP_Address
SINGing to IP_Address (IP_Address): 12 data bytes
12 bytes from IP_Address: icmp_seq=0 ttl=236 mask=255.255.255.0
12 bytes from IP_Address: icmp_seq=1 ttl=236 mask=255.255.255.0
12 bytes from IP_Address: icmp_seq=2 ttl=236 mask=255.255.255.0
12 bytes from IP_Address: icmp_seq=3 ttl=236 mask=255.255.255.0
12 bytes from IP_Address: icmp_seq=4 ttl=236 mask=255.255.255.0

--- IP_Address sing statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

所有系統會使用 ICMP Address Mask Reply 送 Address Mask 的資訊

現在我們用 sing 送 fragmented ICMP Address Mask Request 給 SUN Solaris 2.7 機器, 結果有很大差異：

./sing -mask -c 2 -F 8 IP_Address

SINGing to IP_Address (IP_Address): 12 data bytes

12 bytes from IP_Address: icmp_seq=0 ttl=241 mask=0.0.0.0

12 bytes from IP_Address: icmp_seq=1 ttl=241 mask=0.0.0.0

--- IP_Address sing statistics ---

2 packets transmitted, 2 packets received, 0% packet loss

若他是 SUN Solaris 機器, 他就會 reply 一個 0.0.0.0 的 Mask Address

2000-06-15 08:08

參考：http://sourceforge.net/projects/sing

2.1.5.2 標題：lpset -r Buffer Overflow 弱點

參考：http://sunsolve.sun.com/securitypatch

2.1.6 SuSE Linux

標題：**wuftpd 2.6.0-121 之前版本的安全漏洞**

在執行 SITE EXEC 指令時，wu-ftpd 程式未做好正確的 bounds checking。導致入侵者可能可以以 root 權限在系統上執行任意程式碼。

參考：http://www.suse.de/de/support/security/suse_security_announce_53.txt

2.1.7 Slackware Linux

標題：Fdmount 緩衝區溢位

Slackware Linux 版本 4.0 和 7.0 中的 Fdmount 程式含有一個緩衝區溢位的問題。Fdmount 程式被預設安裝為 suid-root，並且這個程式通常只能被"floopy"群

組中的使用者使用。Floppy 群組中的使用者可以利用緩衝區溢位的方法得到系統管理者的權限。

參考：http://csrc.nist.gov/icat/vulnerabilities/CVE-2000-0438.html

2.1.8 Linux Mandrake

標題：Xlockmore 4.16 的緩衝區溢位問題

在 Xlockmore 版本 4.16 中的 xlock 程式有一個緩衝區溢位的漏洞，這個漏洞使得本地端的使用者利用一個足夠長度的-mode 參數讀取機密的記憶體資料。

參考：http://csrc.nist.gov/icat/vulnerabilities/CVE-2000-0455.html

2.2 通訊協定弱點

2.2.1 telnet

2.2.1.1 弱點一：密碼竊取

密碼未編碼，可以網路監聽軟體，監聽某一個 IP 位置，擷取其 telnet 的封包並且分析之，即可取得使用該台主機的使用者在其使用 telnet 連線登入的主機上的使用者 ID 與密碼，進而偷取資料或是做為其他 hack 用途的跳板。

2.2.1.2 弱點二：緩衝區溢位(buffer overflow)

在一些比較舊的平台上，會有緩衝區溢位的問題，由於 telnet 連線的時候會需要傳送使用者名稱(ID)、終端機型態(例如 VT 100)、使用者密碼，通常伺服器端會有一塊 buffer 負責放置這些收入的字元，但是有些系統的設計上，這些緩衝區的大小是固定的，而且也沒有作例外的處理，所以在那三個情況下，可能可以輸入大量的字元，讓緩衝區產生溢位(overflow)藉以 crack 目標系統，或是進入系統竊取資料，在許多平台上都有此問題，例如 red hat 6.0。

2.2.1.3 弱點三：telnet Daemon 環境弱點

一些支援標題為"Telnet Environment option"的 RFC 1408 或是 RFC 1572 規格的 telnet daemon 程式，被發現有一些弱點可供侵入，支援這項延伸功能的系統提供了一樣功能，可以將系統的環境變數傳遞到另一個系統上去。假如遠端或目標系統皆支援 RFC 1408 或是 RFC 1572 的規格並且分享同樣的物件資料庫，在這種情況下對於 telnet daemon 呼叫的登入程式有影響的環境變數，就有可能藉由連線被傳遞出去，在這種情況下一個普通 user 可以藉著一般權限的登入以及一些步驟變成該系統的 root(系統管理者權限)，進行他想要作的任何事情。在該系統上有帳號的使用者可以輕易的突破這項弱點，而在該系統沒有帳號的使用者也可以利用這項弱點，只要他們先放入一個變更過的分享物件函示庫到目標系統上即可。因此無論此用者在該目標系統上是否有 local 的帳號，他都可以利用此弱點進行入侵。

2.2.1.4 弱點四：telnet 加密弱點

由於 telnet 並未對送出的封包加以編碼 mask，所以使得使用者密碼可以在網路上輕易的被擷取(弱點一)，因此就有所謂的 encrypted telnet 出現，這是利用一些數學方法，對 telnet 送出的封包加以編碼加密，其中一套就是所謂的 Kerberos V4，這項編碼法被使用於 Berkely telnetclient 上，我們所知所有支援 Kerberos V4 的 BSD telnet 都會受到影響，使得編碼加密的效果減少，讓使用該加密方法的資料的安全性降低。這使得任何能夠存取使用這種加密法傳送的封包的使用者可以輕易的解密，得到其中的資訊。

2.2.1.5 弱點五：telnet 破解

有許多的 UNIX 電腦，用他們自己版本的 telnet 不正當的更換了原本的 telnet 版本，這會使得一些登錄著使用者資料的 log 檔(其中包括遠端使用者的使用者

名稱與密碼)會被傳播出去，這使得有心者可以利用此弱點，取得許多在該電腦上登入的遠端使用者的資料，並進行資料偷竊之類的行為。

2.2.2 IGMP

針對媒體伺服器所引發 denial of service，可以影響系統的 performance 或直接來 crash 系統。在 Microsoft Windows 裡，TCP/IP Stack 並沒有把 fragmented IGMP packets 處理好。IGMP 在 TCP/IP 裡算是一個新的 Protocol，它可以讓機器加到 multicast 的 network 裡。一種稱作"fawx"的 exploit script 可以送一些壞的 packet 到 windows 的機器中讓系統當掉。

2.2.3 HTTP

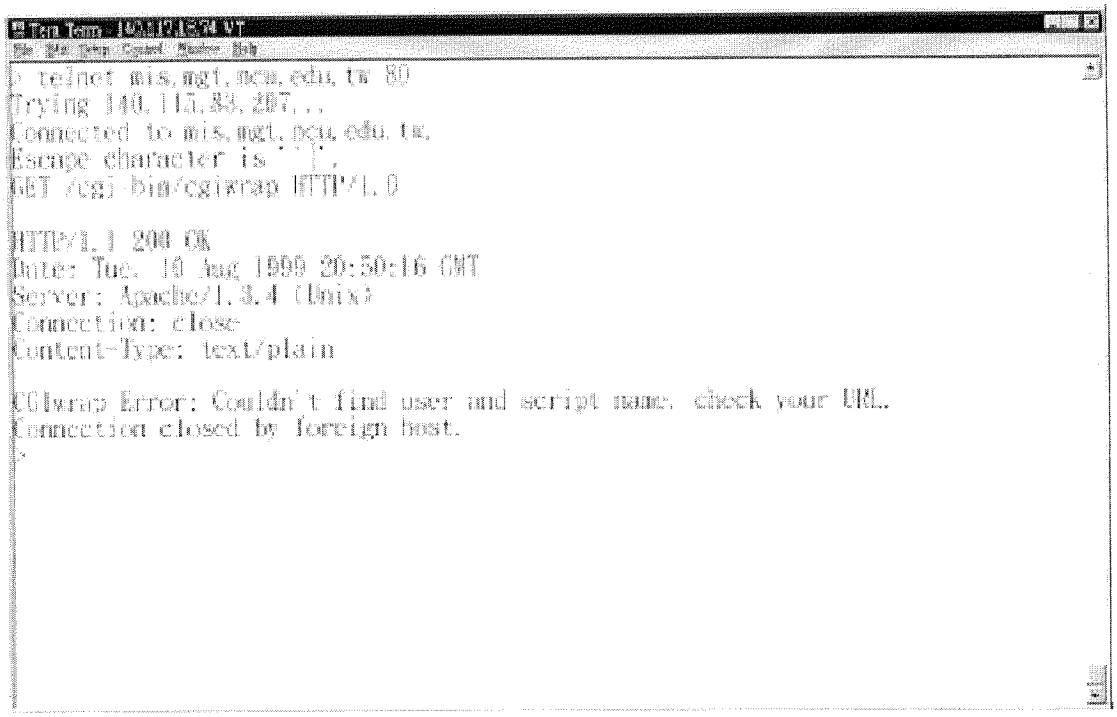
2.2.3.1 DoS/DDos attack

各類的阻絕攻擊或是分散式阻絕攻擊都能達到癱瘓一個網站的目的。

2.2.3.2 CGI attack

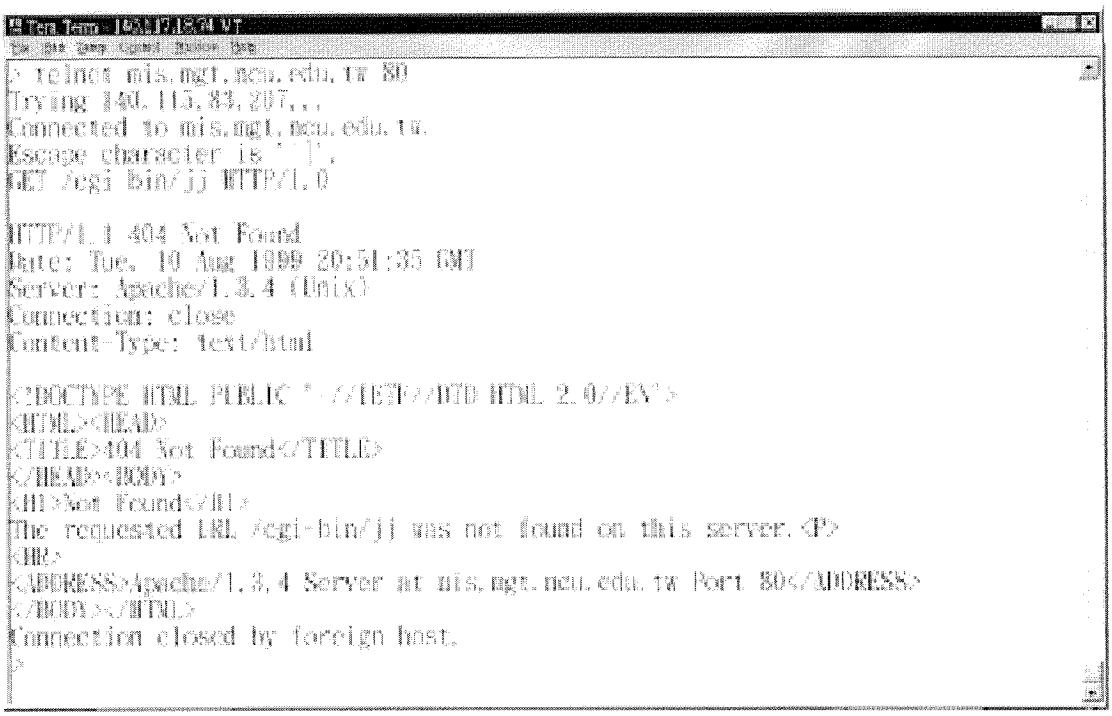
CGI 安全調查

我們可以針對 Web 伺服器送出 CGI 的服務要求，如果伺服器裝有該 CGI 程式，便會回傳額外資訊，如下圖所示。



CGI 探測回傳結果

如果該主機沒有安裝此一 CGI 程式，則會回傳錯誤訊息，如下所示。



CGI 探測回傳錯誤訊息

2.2.3.3 ASP attack

ASP 漏洞調查

在沒有打 Services Pack6 補丁的 NT server 上，至少有 6 種方法可以看到 ASP 程序的源代碼,它們是:

1. http://www.someserver.com/msadc/Samples/SELECTOR/showcode.asp?source=
   /msadc/Samples/SELECTOR/showcode.asp
2. http://somewhere/something.asp::$DATA
3. http://somewhere/something.asp%2e
4. http://somewhere/something.asp.(加一個點)
5. http://somewhere/something%2e%41sp
6. http://somewhere/something.asp%81

2.2.4 SMTP

2.2.4.1 弱點一：pipe

傳送 e-mail 過去後，嘗試在 server 上跑一些程式，它有可能是在 server 端執行一個 script 檔，來跑一些 code，除此之外，pipe symbol 可以附加在 MAIL TO: command 這種形式。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default1.htm

2.2.4.2 弱點二：DEBUG

舊的管理漏洞，存在於舊的 mail server，因為它支援此命令，這個命令允許一些不速之客來 access serve。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default2.htm

2.2.4.3 弱點三：HELO very long

一個 buffer-overflow 的漏洞，通嘗發生於 user name 或 passwd 太長以致於 server 上的 buffer overflow。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default4.htm

2.2.4.4 弱點四：EXPN

入侵者藉由 EXPN 這個指令掃描 mail server 並嘗試找出使用者的帳號。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default6.htm

2.2.4.5 弱點五：too many recipients

spammer 濫用你的 e-mail server 來發廣告信。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default8.htm

2.2.4.6 弱點六：corrupted MAIL command

入侵者將發送一些不正確的 fomatted command 來危駭你的系統。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default9.htm

2.2.4.7 弱點七：email name very long

入侵者嘗試用 buffer-overflow 來攻擊。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default10.htm

2.2.4.8 弱點八：corrupted RCPT command

入侵者將藉由 RCPT 這個指令所產生出的 buffer-overflow 來入侵你的系統，並取得系統的控制權。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default11.htm

2.2.4.9 弱點九：command very long

入侵者利用 buffer-overflow 來攻擊你的系統漏洞。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default20.htm

2.2.4.10 弱點十：mail to decode alias

藉由舊的 email alias 在你的 server 上執行某些 code。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default12.htm

2.2.4.11 弱點十一：mail to undecode alias

藉由舊的 email alias 在你的 server 上執行某些 code。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default13.htm

2.2.4.12 弱點十二：MIME file name very long

入侵者藉由 server 上的 buffer-overflow 來 crash 或 break-in 你的系統，使用者名稱，密碼，及檔名太長都可能產生 buffer-over flow。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default14.htm

2.2.4.13 弱點十三：uucp-style recipient

舊的 uucp-style address 過去被當成可接收 mail 的 address，有些 anti-spam system 無法去 check address 的型態，以致於入侵者入侵你的機器並藉由你的機器送發廣告信。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default16.htm

2.2.4.14 弱點十四：login-failed

藉由 login 的漏洞來 try 出密碼。

參考：http://www.gsn-cert.nat.gov.tw/doc/doc/protocol/SMTP/Attack/default17.htm

2.2.5 PPTP

微軟的 Windows NT 安全問題再度引起爭議，這次問題是出在 Windows NT 所提供的虛擬私有網路(Virtual Private Network 簡稱 VPN)，VPN 是一種利用網際網路來連接公司各據點，以達到互相連通的技術，如此一來公司就好比有自己內部的網路，費用要比各據點接專線相連的成本低了許多，但是安全性相對的也就低了很多。微軟的 Windows NT 針對 VPN 的需求，提供了 PPTP (Point-to-Point Tunnelling Protocol)的服務，這項服務當然也有注意到安全的問題，所以在這些 PPTP 的連線中，微軟都有加密保護，可是最近一位安全顧問 Schneier 指出，這個加密的保護簡直是太薄弱而不堪使用，並指出微軟 PPTP 中的五項弱點，可是微軟馬上反擊，認為這五項所謂的弱點中，有些只是理論，再不然就是早已經推出修補程式，同時也有其他的安全專家認同微軟所言，認為 Schneier 提出的問題其實早就已經在論壇上討論過，並有些也已經解決，不過大家都一致認同這樣的爭端可以給微軟壓力，讓微軟多重視安全問題。

3. 各類網路攻擊技巧

3.1 Sniffer

電腦網路是共享通訊通道的。「共享」的意思就是電腦能夠接收到送給其他電腦的資訊。捕捉在網路中傳輸的資訊就稱為 sniffing（窺看）。在區域網路中的每一台電腦所送出的 frame 都會有標頭（header），標頭裡面會包含所欲傳送的目的端位址（address）。一般情形下，每一台電腦只會接收兩種 frame：

1. 目的端位址和本身硬體位址（MAC address）相同的 frame。
2. 目的端位址為廣播位址（broadcast address）的 frame。

在接收到上述兩種 frame 時，網路卡會透過 CPU 產生一個硬體中斷來通知作業系統，使系統將 frame 之資料做進一步的處理。

但網路卡也可以轉換成雜亂模式（promiscuous mode）。在此狀態下的網路卡具備「廣播位址」，因此會接收流經此網路卡的所有封包。Sniffer 即是將主機的網路卡狀態進入雜亂模式的軟體，其工作在網路環境的底層，它會攔截所有正在網路上傳送的資料，並且透過對應的軟體處理，及時分析這些資料的內容，來取得有用的資訊。

通常 sniffer 所要收集的資訊可以分成下列幾種：

1. 帳號與密碼
2. 金融帳號
3. 使用者之間交換的通訊內容
4. 底層之通訊協定

假如使用者又沒有採取適當的加密程序來保護傳輸的資料，則監聽者很容易地就能夠分析擷取到的資料，進而得到有用的資訊。

參考：http://www.china-pub.com/computers/emook/0364/info.htm

3.2 Ping of Death

此攻擊屬於阻絕攻擊（DoS）的一種，主要是利用"ping"這支工具程式來產生超過 IP 協定所能夠允許的最大封包（即超過 65535 位元組的長度）。當這個封包發送到沒有檢查功能的系統時，可能會造成系統當機。

"ping"原本是藉由發送 ICMP Echo request 封包，來探測目的地主機是否存在。所以 ping 能夠傳送的最大資料量為 65535-20-8=65507 位元組（其中 20 位元組為 IP 表頭，8 位元組為 ICMP 表頭）。因此若要利用 ping 來做攻擊，只要想辦法讓封包夾帶的 Data 長度超過 65507 位元組就可以達到目的。然而在 Ethernet 下的 MTU（Maximum Transmission Unit）預設值為 1500 位元組，超過 MTU 的封包都會經過切割（fragmentation）再傳送出去給目的端，到達目的端後再重組（reassemble）成為原來的封包。所以 Ping of Death 的攻擊方式就是產生超量的 ICMP 封包，使其最後一個切割片段的 offset+size>65507，造成不正常封包長度而達到攻擊效果。

參考：http://www.insecure.org/sploits/ping-o-death.html

3.3 Teardrop

此攻擊方式也是屬於阻絕攻擊的一種，它的目的並不是要偷取資料，而是要利用網路主機系統 IP 封包重組的漏洞，讓網路主機癱瘓而無法正常運作，導致使用者無法存取網路資源。

Teardrop 會假造出一些 IP 片段，這些片段包含錯誤的位移植（offset）。當這些片段到達目的地而被系統主機根據錯誤的位移植重組時，因誤判封包的大小，而造成運算錯誤而使得整個程式當掉或重新開機。

參考：http://cyber.cs.ntou.edu.tw/~station/hcb/attack_form.htm

3.4 Smurf

Smurf也是阻絕攻擊的一種。要完成Smurf攻擊得要有兩個重要的條件：第一、必須要假造ICMP echo request封包；第二、要將ICMP echo request封包送到廣播位址（broadcast address）。

在Smurf的攻擊過程中，可以分成三大角色：

- 攻擊者（attacker）
- 攻擊媒介者（intermediary，用來造成擴大效應的網路）
- 受害者（victim）

Smurf的攻擊方式是，攻擊者從遠端利用ICMP echo request 封包送給IP 廣播位址，並且將這個ICMP封包的來源位址填上欲攻擊對象的IP位址。當媒介者收到這個廣播的ICMP echo request封包時，假如媒介者沒有過濾這些廣播的ICMP封包，則他就會處理這個ICMP echo request，並且回應ICMP echo reply給受害主機。當這個用來當作攻擊媒介的網路都回應這個ICMP echo request時，會造成受害主機嚴重的網路擁塞問題。假設媒介網路有一百台電腦，則攻擊者只要送出一個封包，就可產生一百個回應，擴大效應增加一百倍，對攻擊者來說，是相當方便有效的阻絕攻擊。

攻擊者甚至可以自己研發自動攻擊工具，讓工具自動的將這些廣播ICMP echo request封包送給多個媒介網路，造成這些媒介網路回應封包給受害者。或者是尋找會回應廣播封包的網路（也就是那些網路的路由器不會過濾broadcast封包），成為下次攻擊行動的攻擊媒介者。

參考：http://www.cert.org/advisories/CA-1998-01.html

3.5 IP Spoofing

IP Spoofing 就是在產生 TCP/IP 封包時，使用他人的 IP 位址當作封包的來源。利用 IP Spoofing 攻擊的方式大致上可以分為二類：

1. 侵入網路主機：

   藉由不需使用者名稱和密碼的認證漏洞，侵入網路主機獲得主機上的存取權限。在一個 spoofing 攻擊中，入侵者發送一個訊息給目的主機，並聲稱來自於可信任的主機，為了要讓這個攻擊成功，入侵者必須先找出可被信任主機的 IP 位址，然後偽造封包的 IP 表頭讓封包看起來是從可信任主機發送出來的。藉由欺騙遠端主機，讓遠端主機以為攻擊者為合法的可信任主機而建立連線，攻擊者就可以藉由這個連線獲得遠端主機的管理權限，之後就可以利用此權限值入後門程式。

   此類行的攻擊有：

   - Man-in-the-middle：攻擊者入侵至用戶端與伺服端之間傳輸的通路中，轉接用戶端與伺服端交換的資訊而不被發覺。對用戶端而言，攻擊者偽裝成伺服端；而對伺服端而言，攻擊者偽裝成用戶端。
   - Routing re-direct：改變封包的路由訊息，從原來的主機變成到攻擊者的電腦。
   - Source routing：重新導向經由攻擊者主機的個別封包。
   - Blinding spoofing：猜測從主機的回應訊息，允許命令被傳送，但無法立即得到回應。

2. 癱瘓網路主機：

   這種攻擊方式屬於阻絕攻擊的一種，利用假造的來源位址讓別人無法反查攻擊者的來源，或者是利用被偽裝的 IP 位址來發送大量的訊息，使得受害主機忙於回應，進而造成網路擁塞，使得整個網路上的主機所提供的服務無法正常運作。

   此類的攻擊方式有：

- SYN flooding：使用隨機來源位址的連線要求來塞滿受害主機的 receive queue，讓服務主機資源不足而無法再和正常的要求建立連線。
- Ping flooding：使用隨機來源位址發送 Ping request 封包，讓受害主機忙於回應 ICMP echo reply 封包給假造的 IP 位址主機。
- UDP flooding：利用一些會回應 UDP 封包的 well-known port，藉由偽裝來源使得兩台不知情的主機陷入互相回應的迴圈中。
- Smurf：偽裝成受害主機的 IP 位址，並發送 ICMP echo request 封包到廣播位址，使得受害者收到大量的 ICMP echo reply 封包。
- Fraggle：如同 Smurf，但改使用 UDP 封包。

參考：

http://www.networkice.com/Advice/Underground/Hacking/Methods/Technical/Spoofing/default.htm

3.6 Port Scan

Port scan 是試著連接目標系統的 TCP 與 UDP port，以瞭解其*所提供的服務種類*或者是其所聆聽（listen）的狀態。藉由得知對方所聆聽的 port，可以猜測該系統所用的*作業系統種類*與所執行的*應用程式*，如果對方的設定有瑕疵或者正在執行的軟體有安全上的漏洞，那麼未經授權的使用者就可以連接其聆聽的 port 去存取該系統。

目前用來進行 port scan 的技術有以下幾種：

- Vanilla TCP connect() scanning：利用 connect()系統呼叫，針對目標主機的 service port 進行連線。如果目標主機的某個 service port 有在聆聽，則連線就會成功，反之失敗。
- TCP SYN scanning：這個技術又稱"half-open"掃瞄，因為沒有完成 TCP

的 three-way handshake 連線動作。此方法就是對目標主機送出 SYN 封包，假如收到 SYN/ACK 的回應封包，則表示這個 port 有開啟。但假如收到 RST/ACK 的回應封包，則表示此 port 沒有開啟。

- TCP FIN scanning：根據 RFC 793 的規範，當關閉的 port 收到 FIN 封包時，必須回應適當的 RST 封包。而開啟的 port 收到 FIN 封包時，理所當然會忽略此封包。雖然這是 TCP 的規範，但是微軟的 Windows 作業系統並沒有遵照這個規範，不管 port 是否開啟，當收到 FIN 封包時一律都會回應 RST 封包，所以這種技術並不適用在 Windows 上。但是這個技術卻可以用來分辨目的端主機的作業系統是屬於 Windows 還是 Unix like 系統。
- Fragmentation scanning：此方法將原本要送出的封包（FIN 或 SYN）分段，如此一來，原本的 TCP 表頭就被分段到一個個的 IP 封包中，這些 IP 封包在流經防火牆或是封包過濾器時就比較難以被察覺。
- UDP ICMP port unreachable scanning：此方法是用來掃瞄開啟的 UDP port，因為 UDP 協定是非連線的（connectionless），開啟的 UDP port 不必對攻擊者偵測的封包回應，關閉的 UDP port 也不需要回應 RST 封包。但是大部分的主機若是其關閉的 UDP port 收到封包，會回應 ICMP port unreachable 封包，所以攻擊者可以利用這項特徵，來偵測某一個 UDP port 是開啟還是關閉的狀態。

參考：

http://www.networkice.com/advice/Underground/Hacking/Methods/Technical/Port_Scan/default.htm

3.7 SYN Flooding

當一個 TCP 連線要建立時，必須要完成 three-way handshaking 的協議動作。若 A 要與 B 做連線時，在正常的情況下，A 會從一個特定的 port 發

出 SYN 封包，送給 B 正在聆聽的特定 port 上。當 B 收到這個 SYN 封包後，系統 B 會進入 SYN_RECV 狀態中，並試圖回送一個 SYN/ACK 封包給 A。如果 A 順利收到 B 所回應的封包，會再回應一個 ACK 封包，然後整個連線的狀態就進入 ESTABLISHED。然而最大的問題在於，一個系統會分配一定量的資源，來建立這個潛在的連線，但是有時候連線的建立動作無法順利完成。雖然一個系統通常可以支撐數百個連線同時連接到同一個 port，但是或許只能同時支援一、二十個連線要求，再多就會把系統資源都耗在建立連線的需求上。這正是 SYN flooding 用來摧毀系統的主要機制。

在進行 SYN flooding 時，攻擊會從系統 A 送出一個 SYN 封包到系統 B，但是攻擊者會把來源 IP 位址偽裝成一個不存在的 IP 位址。而 B 收到這個 SYN 封包後，會嘗試送回 SYN/ACK 封包到這個假造的 IP 位址。如果假冒的 IP 位址是一個存在的系統，那麼此系統就會回應一個 RST 封包。因此在進行 SYN flooding 時，偽裝的來源 IP 位址必須是一個永遠無法到達的位址，使得 B 送出的 SYN/ACK 封包後，永遠等不到 ACK 封包或是 RST 封包，那麼這個連線狀態就會一直保持在 SYN_RECV，並且被放進連線佇列（connection queue）中等候，一直要等到 timeout 以後才會將這個連線從佇列中刪除。一旦這類的假造 SYN 封包數量一多，那麼連線佇列很快就會用光，造成受害系統無法接受正常的連線要求。

參考：http://www.cert.org/advisories/CA-1996-21.html

3.8 Land Attack

此攻擊為阻絕攻擊的一種，藉由發送偽造的 TCP 連線封包，使得有缺陷的網路服務主機當機或降低效率。

Land attack 將 TCP SYN 連線封包的 source address 與 source port 偽裝成

目的端主機的 address 與 port，並發送給欲攻擊的網路主機，只要網路主機本身有提供 TCP 服務（如： telnet、FTP、Http），那麼當封包送達時，便欺騙網路主機讓它以為是自己要和自己建立連線。此時，若此主機的作業系統無法處理這個錯誤，則可能會發生當機。

參考：http://www.insecure.org/sploits/land.ip.DOS.html

3.9 DDoS

此類攻擊主要是指植入後門程式後，從遠端遙控發動攻擊，攻擊者可從多個已入侵的跳板主機控制數個代理攻擊主機，所以攻擊者可同時對控制下的代理攻擊主機啟動攻擊命令，以對受害主機進行大量攻擊。

目前，攻擊者最常使用的分散式攻擊軟體有四種：

- Trinoo
- TFN
- TFN2K
- Stacheldraht

這四種攻擊都是利用相同的原理，並且前三項已經在第一部份有做說明，因此這裡就不在累贅。

參考：http://magazine.nsfocus.com/detail.asp?id=273

Stacheldraht: http://magazine.nsfocus.com/detail.asp?id=53

Trinoo：http://magazine.nsfocus.com/detail.asp?id=54

TFN2K：http://magazine.nsfocus.com/detail.asp?id=57

3.10 UDP Bomb

一般來說，關於 UDP bomb 有下列三種說法：

1. UDP bomb 是 UDP flood：攻擊者利用大量偽造來源的 UDP 封包送往受

害者的 random port。當受害主機收到 UDP 封包後，會取得目的 port 的資料，決定是否有應用程式正在那個 port 等待，當系統知道沒有應用程式在等待時，會回應一個 ICMP destination unreachable 給偽造的來源。如果有足夠的封包攻擊受害者系統，系統有可能發生當機。

2. 利用會產生回應的 UDP service 使得兩台主機進入迴圈的狀態：假如有兩個 UDP service，而這兩個 service 都是會產生輸出的 service，將這兩個 service 建立起連線將會產生大量的封包，形成阻絕攻擊。
3. 將 UDP 表頭中的 length 欄位填入不合法的值：UDP 表頭中有 length 欄位，其目的是要填入 UDP 封包的長度。其合法的範圍為：

$$\text{length} = \text{total length} - \text{IP header length}$$

如果 length < total length – IP header length，則有可能使得 SunOS version 4.11 ~ 4.13 無法處理這個封包，並且造成主機重新啟動。這個阻絕攻擊算是利用系統的漏洞所造成的攻擊。

參考：http://www.cert.org/advisories/CA-1996-01.html

3.11 Fraggle

此攻擊方式屬於阻絕攻擊的一種，基本攻擊概念和 Smurf 相近，同樣是傳送封包到廣播位址，其相異點是：Smurf 傳送的是 ICMP 封包，而 Fraggle 傳送的則是 UDP 封包。

在做攻擊之前，攻擊者必須收集一些可用來當作媒介攻擊的廣播位址清單。有了這些媒介位址以後，攻擊者送出封包到這些廣播位址，並將封包來源 IP 偽造成受害者的 IP，而封包目的 port 設為 7（echo port，收到封包的主機會回應給來源）或 19（chargen port，收到封包的主機會產生字串回應給來源）。當媒介者收到這些廣播封包後，假如媒介者有開啟 echo

service 或者是 chargen service，那麼便會回應封包給受害者，一旦媒介者的數量很大的時候，便會造成受害者的網路擁塞，使其無法提供服務。

攻擊者甚至可以將攻擊來源 port 設為 7 或 19，那麼媒介收到封包後，會送往受害主機的 port 7 或 19。假如受害者的 port7 或 19 有開啟，則會回應給媒介者，反覆來回這樣的封包造成雙方主機進入迴圈的狀態。

參考：http://www-arc.com/sara/cve/Possible_DoS_problem.html

3.12 ICMP Unreachable

因為 ICMP unreachable 封包的格式很簡單，其前 8 Bytes 為 ICMP 表頭，其中 Type 欄位為 3，表示屬於「封包無法到達」之類的錯誤訊息；而 Code 欄位用來描述這個功能程度上或細節上的不同。除此之外，ICMP unreachable 封包所包含的資料也很少，所以攻擊者利用這點發送假的 ICMP unreachable 封包，來欺騙受害主機，使得主機認為封包無法到達。

攻擊的時候必須先竊聽封包，利用得來的封包發送 ICMP unreachable。只要攻擊者聽取並發送的速度比原本封包目的主機的回應還快，先一步到達來源主機，那麼來源主機就會認為目的主機無法到達。這樣可能的結果是造成原本已有的連線斷線，或是欺騙伺服器或是用戶端認為本身的網路不通，而使得所有封包都無法送達。

參考：http://www.securiteam.com/unixfocus/5SP062K40G.html

4. 入侵偵防系統雛形

在所有的網路攻擊入侵事件中，最常見也最令人頭痛的是 SYN Flooding 攻擊，因此本計畫首先針對此種攻擊做深入研究以建立一有效的入侵偵防系統雛型。在 SYN Flooding 網路攻擊中，攻擊者把大量的 SYN 封包傳送到欲攻擊的伺服器，並且在這些 SYN 封包中的來源位址皆填入假造的 IP 位址。因此，被攻擊的伺服器其連線佇列中充滿了 SYN+ACK 封包，原因是無法收到相對應的 ACK 封包。此時受到攻擊的伺服器就不能繼續提供服務，因為它的連線佇列已滿並且不能接受合法的 SYN 連線要求。

本計畫研究了現有的一些防禦方法並且比較其優缺點。我們透過修改 SYN Cache 的機制改進 SYN Cookie 的方法。配合我們的構想提出一個新的解決方案，並在 FreeBSD 平台上實作一個雛型系統，然後實測其效能表現。實驗的結果說明了我們所提出的方法能夠有效地防禦 SYN Flooding 攻擊。

4.1 SYN Flooding 防禦原理

一種最簡單的存活方法就是Random Drop，其運作原理是當伺服器的backlog串列已滿，但又收到 SYN 封包時，伺服器會隨機選擇一個 backlog 串列的登錄點來替換。伺服器會送給被替換掉的用戶端一個 RST 訊息。

假如被替換掉的登錄點是由惡意的 SYN 封包所產生的，則此 RST 封包當然不會到達這個假的目的端。

但假若被替換掉的登錄點是由合法的用戶端所產生，那麼這個 RST 封包就會造成用戶端重新建立連線(如圖 4.1 所示)。

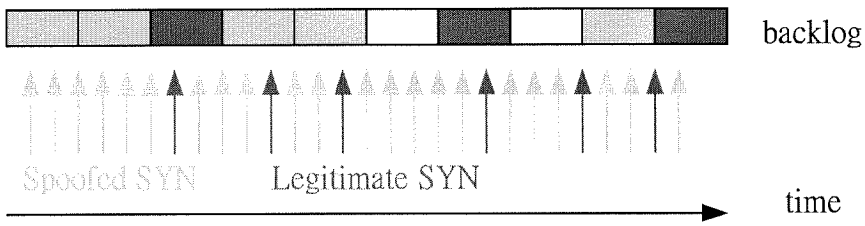


圖 4.1. Random Drop 抵抗網路攻擊

許多連接到 Internet 的主機使用防火牆來防禦外來攻擊。其原理是將所有欲送往防火牆內部主機的封包，先經過防火牆過濾來阻擋有攻擊意圖的封包。然而這個方式會造成封包經過防火牆時，需要一些額外的處理而延遲傳送。主要的方式有：

- Firewall as a Relay
  - 優點 ： 防火牆內部的伺服器能夠免於假造的 SYN 封包攻擊。
  - 缺點：額外的處理會造成封包延遲傳送。

- Firewall as a Semi-transparent Gateway
  - 優點：正常連線一旦建立後，並不會有額外的處理造成封包延遲傳送。
  - 缺點：在受到攻擊時，伺服器端會有大量的非法連線。

- SYN Cookie

  就是本系統所參考的主要方法，把對方的 SYN 序號以及 IP 配合一個秘密數值設計一個 hash 函數，用來作為新的 SYN 序號以驗證封包。

4.2 網路攻擊偵防系統雛形設計與實作

圖 4.2 顯示出整個系統架構。過濾器(Filter)部份就是我們的偵防系統，我們把過濾器與伺服器設定為相同的 IP 位置，並且關閉伺服器的 ARP Reply。這樣一來，就能隱藏伺服器而保護伺服器以避免直接受到 SYN 封包的攻擊。

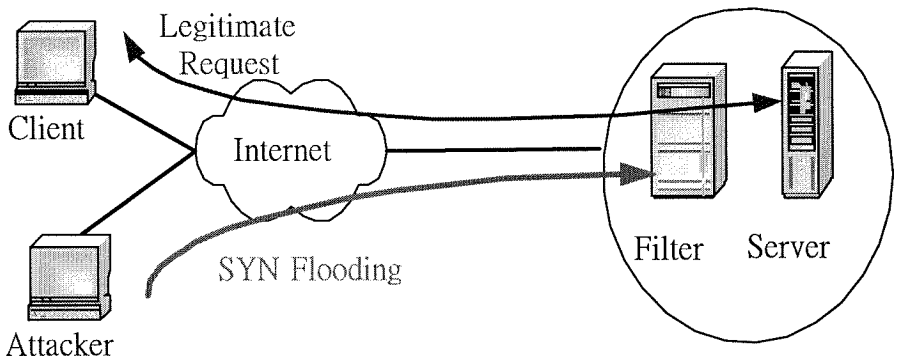


圖 4.2. System Architecture

首先，利用 SYN-Cookie 的方式來對欲存取伺服器的用戶端做認證。第二，使用 SYN-cache 的方式來解決 SYN-Cookie 所造成的封包流失的問題。第三，藉由防火牆的架構來保護伺服器不受直接攻擊。最後，讓過濾器與伺服器擁有共同的 IP 位置來隱藏伺服器。

藉由修改 FreeBSD 核心碼來發展這個系統的雛形。主要修改的部分分為過濾器以及伺服器兩大部分。詳細的細節可以參考研究生黃韜維(指導教授蔡文能)的碩士論文。(如附錄)

4.3 系統效能測試

本報告將會比較對照組系統，以及我們系統的效能評比，對照組系統為 FreeBSD kernel build-in firewall 與 ipfw (IP Firewall) 之組合。

接著會使用一些開放式軟體套件，包括 apache benchmark 及 http performance 來，測量此系統的 HTTP service 效能。

測試環境

scenario 1: FreeBSD kernel build-in firewall, ipfw

scenario 2: FreeBSD kernel build-in firewall ,our prototype system

測試步驟

在不同的攻擊速率下 (1000 ~ 10000 packets per second)，開啟不同大小的網頁(1 ~ 200 Kbyte)。

效能測試結果

產能

在未受到攻擊時，我們的系統會有 25%的額外負擔。但是在受到 SYN Flooding 攻擊時，我們系統的 throughput 大約會是現有防火牆系統的 4 倍左右。並且在不同 SYN-Flooding 的攻擊速率下，我們的系統顯得比較穩定。

執行時間

在受到 SYN-Flooding 攻擊時，我們系統與現有防火牆系統比較之下，會有較短的執行時間，並且不管攻擊速率如何，執行時間維持穩定。

每秒處理的請求個數

雖然在未受到攻擊時，我們系統所處理的連線請求個數較少，但是在受到攻擊時，在不同攻擊速率甚至是最大的 10000pps 的攻擊速率下，所能處理的請求個數維持與未受攻擊前所能處理的請求個數相差不遠。但是在現有的防火牆系統

下，一旦受到 SYN-Flooding 攻擊，效能會急速下滑。

此系統之特點

在我們系統中，過濾器只負責過濾封包，驗證用戶端身份，以及根據 Layer2 來轉送用戶端與伺服器之間的封包。這個過濾器可以說是專門設計來防禦 SYN Flooding 攻擊，而不提供另外的服務，因此可以有較強的抵擋能力。

與其他系統的比較

	Our Approach	SYN Cookie	RESET Cookie	Random Drop
Guarantee Service	YES	YES	YES	NO
Memory Immunity	YES	YES	YES	YES
Computing Immunity	NO	NO	NO	YES
Packet Retransmission	YES	NO	NO	YES
Good Performance	YES	YES	NO	YES

表格 4.1. SYN Flooding Defense Mechanism Comparison (1)

	Our Approach	Firewall	Cisco TCP Intercept
Connection Establishment	NO	YES	YES
Sequence Number Conversion	NO	YES	YES

表格 4.2. SYN Flooding Defense Mechanism Comparison (2)

5. 結論

俗語說: 知己知彼，百戰百勝。如前所述，網路攻擊大抵皆是利用系統漏洞或是網路通訊協定的漏洞，我們網路入侵偵測系統的建置平臺可以選定有公開原始碼的 FreeBSD 或 Linux 系統比較能夠修改系統內部。不過要知道網路上則可能連接有各種系統; 所以各種系統的漏洞都可能被用來試圖攻擊我們的網路入侵偵測系統。因此我們首先建立了弱點資料庫，然後分析弱點資料庫並研究各種攻擊模式。一般有兩種攻擊的歸類方法。第一種是正面表列:正面表列規範正常行為，凡違反此規範之行為都視為侵犯。負面表列訂定不正當行為、已知之攻擊行為特徵，凡在此列表之行為皆視為侵犯，這是「攻擊特徵」的偵測方法，傳統電腦病毒偵測就屬於此類方法。第一種方法容易錯誤拒絕正常網路連線，第二種方法容易錯誤接受攻擊的行為。如何界定「異常」、與「不當」的清楚規範，以找出攻擊行為是入侵偵防最重要的一環。

在本計畫中，我們除了蒐集並研讀系統弱點以及攻擊入侵防禦等技巧，並且在每個月的開會討論中與中科院參與同仁共享心得，此外，我們也實作了一個網路攻擊偵防系統雛形如報告中第四章所述，該系統目前能夠有效防禦網路 SYN Flooding 的攻擊，在往後研究中，可以擴充以解決其他的網路入侵問題。

參考資料(References)

[1] 黃韜維, 蔡文能, "A A Study on SYN Flooding," 國立交通大學資訊工程研究所，碩士論文, June, 2001.

[2] 張舜理, 謝續平, "A Security Testing System for Vulnerability Detection," 國立交通大學資訊工程研究所，碩士論文, June, 1999.

[3] Jou, Y.F.; Gong, F.; Sargor, C.; Wu, X.; Wu, S.F.; Chang, H.C.; Wang, F. Design and implementation of a scalable intrusion detection system for the protection of network infrastructure DARPA Information Survivability Conf erence and Exposition, 2000. DISCEX '00. Proceedings , Volume: 2 , 1999

[4] Feiyi Wang; Gong, F.; Wu, F.S.; Narayan, R. Intrusion detection for link st ate routing protocol through integrated network management Computer Com munications and Networks, 1999. Proceedings. Eight International Conferenc e on , 1999

[5] B.R.Smith ,S. Murthy,and J.J. Garica-Luna-Aceves.Securing. Securing Distan ce-vactor Routing Protocols. In IEEE/ISOC Symposium on NetWork and Di stributed System Security,San Diego,CA,February 1997.

[6] B. Vetter,F.Wang,and S.F. Wu. An Experimental Study of Insider Attacks fo r the OSPF Routing Protocol. In IEEE International Conference on Network Protocols,October 1997

[7] B.R.Smith and J.J. Garica-Luna-Aceves.Securing the Border Gateway Routin g Protocol. In Global Internet, November 1996

[8] Landwher, C. E, Bull, A. R., and McDermott J. P.,"A Taxonomy of Compu ter Security Flaws.", ***ACM Computing Surveys***, Vol 26, No.3, September 1 995, pp211-254

[9] Bishop,M. "A Taxonomy of Unix System and Network Vulnerabilities", ***Tec hnical Report CSE-95-10***, Department of Computer Sciences, University of California at Davis, 1995

[10] Nicholad J. Puketza, Kui Zhang, Mandy Chung, "A Methodology for Testi ng Intrusion Detection Systems," ***IEEE Trans. on Software Engineering***, V ol. 22, No.10.October, 1996.

[11] David Detlefs, K. Rustan M. Leino, Greg Nelson, and James B. Saxe. "E xtended Static Checking," ***SRC research report #159***, December 1998

[12] Farmer D., Spafford E. H., "The COPS Security Checker System" , ***Purd ue University Report***, 1991

[13] Dorothy E. Denning, "An Intrusion-Detection Model," ***IEEE Trans. On So ftware Engineering***, Vol. SE-13, No. 2, February, 1987.

[14] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. Levitt, "Network Intrusion Detection, " ***IEEE Network***, May/June 1994.

[15] Koral Ilgun, etc., "State Transition Analysis: A Rule-Based Intrusion Detection Approach," ***IEEE Trans. on Software Engineering***, Vol. 21, No. 3, March 1995.

[16] Erland Jonsson, and Tomas Olovsson, "A Quantitative Model of the Security Intrusion Process Based on Attacker Behavior," ***IEEE Trans. on Software Engineering***, Vol. 23, No. 4, April, 1997..

[17] Vern Paxson, "Bro:A System for Detecting Network Intruders in Real-Time,"***Proceedings of the 7th USENIX Security Symposium***, San Antonio, TX, January 1998.

[18] Charlie Kaufman, etc., "Network Security: Private Communication in a PUBLIC World," **PTR Prentice Hall**, New Jersey, 1995.

[19] Shih-Kun Huang and Shiao-Rong Tyan, "Intrusion Detection and Vulnerability Analysis for GCA Service, " 1999 Project for Institute of Telecommunication.

[20] Stuart Staniford-Chen and Brian Tung "Common Intrusion Detection Framework,"
http://olympus.cs.ucdavis.edu/cidf/, http://gost.isi.edu/projects/crisis/cidf.html

[21] Krsul, I., Spafford, E. and Tripunitara, M. "Computer Vulnerability Analysis,"
ftp://coast.cs.purdue.edu/pub/COAST/papers/ivan-krsul/krsul9807.pdf , May. 1998

[22] Michael Sobirey, "Currently 83 Intrusion detection Systems,"
http://www-mks.informatik.ut-cottbus.de/~sobirey/ids.html, 1999

[23] Elliott, J. IT Professional, "Distributed denial of service attacks and the zombie ant effect" Volume: 2 Issue: 2 , March-April 2000

[24] Chen, Y.W. "Study on the prevention of SYN flooding by using traffic policing" Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP , 2000

[25] T. F. Lunt, et al., " IDES: A Real-Time Intrusion Detection Expert System (IDES)," Interim Progress Report, Project 6784, SRI International, May. 1990

[26] D. E. Denning, "An Intrusion Detection Model," IEEE Trans. On Software Eng. 13-2, pp. 222-232 Feb, 1987.

[27] T. Escamilla, Intrusion Detection, Network Security Beyond the Firewall, John Wiley & Sons, Inc. 1998.