

Beispielsweise ist der SMB-Exploit MS08-067 für Windows-Systeme in folgendem Verzeichnis aufzufinden:

```
/MSF-Path/msf3/modules/exploits/windows/smb
```

Metasploit lädt zusätzlich Module, die im Userverzeichnis `~/.msf4/modules` abgelegt sind. Weitere Pfade für Module lassen sich in der Metasploit-Konsole mit dem Parameter `-m` angeben.

■ `./plugins`

Im `plugins`-Verzeichnis befinden sich unterschiedliche, nachladbare Erweiterungen des Frameworks. Diese lassen sich mit einem `load <PLUGIN>` zur Laufzeit in das Framework einbinden. Zu diesen Erweiterungen zählen neben der Nessus-Bridge auch die NeXpose- und Wmap-Erweiterungsmodule. Mit dem Kommando `show plugins` lassen sich jederzeit die aktuell geladenen Plugins anzeigen.

■ `./scripts`

Das Verzeichnis `scripts` umfasst die Post-Exploitation-Skripte von Meterpreter. Diese Skripte lassen sich in einer aktiven Meterpreter-Session mit dem Kommando `run` ausführen.

■ `./tools`

Dieses Verzeichnis beinhaltet weitere Tools und Skripte, die bei unterschiedlichsten Aufgaben im Pentesting- und Research-Bereich unterstützen können. In diesem Verzeichnis findet man somit unterschiedlichste hilfreiche Skripte, die die Exploit-Entwicklung vereinfachen können. Einige dieser kleinen Helfer werden im Rahmen der Exploit-Entwicklung in Kapitel 10 genutzt.

2.5 Benutzeroberflächen

Das Metasploit-Framework bringt mehrere BedienungsOberflächen mit. Neben der häufig eingesetzten Metasploit-Konsole (`msfconsole`) gibt es mittlerweile zwei unterschiedliche grafische Oberflächen. Lange Zeit gab es zusätzlich ein sehr einfaches Webinterface, und es gibt zudem noch ein Kommandozeileninterface, das sich direkt über die Linuxkonsole steuern lässt. Jede dieser BedienungsOberflächen hat gewisse Stärken, aber auch Schwächen, die eine Bedienung von Metasploit in Teilbereichen vereinfachen oder teilweise auch erschweren oder gar unmöglich machen können.

Um die unterschiedlichen Oberflächen und deren Stärken wie auch deren Schwächen kennenzulernen, wird jedem Leser empfohlen, zumindest den einen oder anderen Exploiting-Vorgang mit den unterschiedlichen Benutzerschnittstellen durchzuführen.

2.5.1 Metasploit-GUI(s)

Metasploit umfasst derzeit zwei aktiv entwickelte grafische Oberflächen, die im Rahmen von Exploiting-Vorgängen zur Verfügung stehen. Bei beiden Oberflächen handelt es sich um Java-Anwendungen, die eine funktionale Java-Umgebung voraussetzen. Armitage ist die jüngere GUI und wird im folgenden Abschnitt 2.5.2 und zudem in Abschnitt 6.4 im Bereich der Automatisierungsmöglichkeiten detailliert betrachtet.

Die Kommunikation zwischen der GUI und Metasploit erfolgt über eine definierte Metasploit-API, die über die XML-RPC-Schnittstelle angesprochen wird. Das Framework umfasst einen Daemon, der nach erfolgreichem Start von der GUI über das Netzwerk oder lokal angesprochen wird.

```
root@bt:~# msfrpcd -h
```

```
Usage: msfrpcd <options>
```

```
OPTIONS:
```

```
-P <opt> Specify the password to access msfrpcd
-S       Disable SSL on the XMLRPC socket
-U <opt> Specify the username to access msfrpcd
-a <opt> Bind to this IP address
-f       Run the daemon in the foreground
-h       Help banner
-n       Disable database
-p <opt> Bind to this port instead of 55553
-t <opt> Server type, [Basic|Web]
-u <opt> URI for Web server
```

Listing 2-9 Startvorgang des msfrpcd

Der Startvorgang des XML-RPC-Daemons lässt sich folgendermaßen initiieren.

```
root@bt:~# msfrpcd -f -U msf -P test -t Basic
[*] XMLRPC starting on 0.0.0.0:55553 (SSL):Basic...
[*] XMLRPC ready at 2011-06-24 13:26:40 +0200.
```

Der in Abbildung 2-15 dargestellte Konfigurations-Wizard benötigt entweder die Verbindungsdaten des Daemons oder er startet einen eigenen lokalen Daemon und verbindet sich automatisch zu ihm.

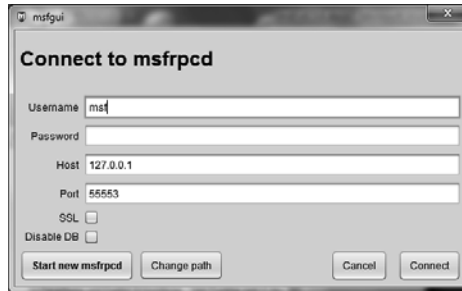


Abb. 2-15 Connection Settings

Nach einem erfolgreichen Startvorgang der Metasploit-GUI stellt sie sich wie in Abbildung 2-16 dar:

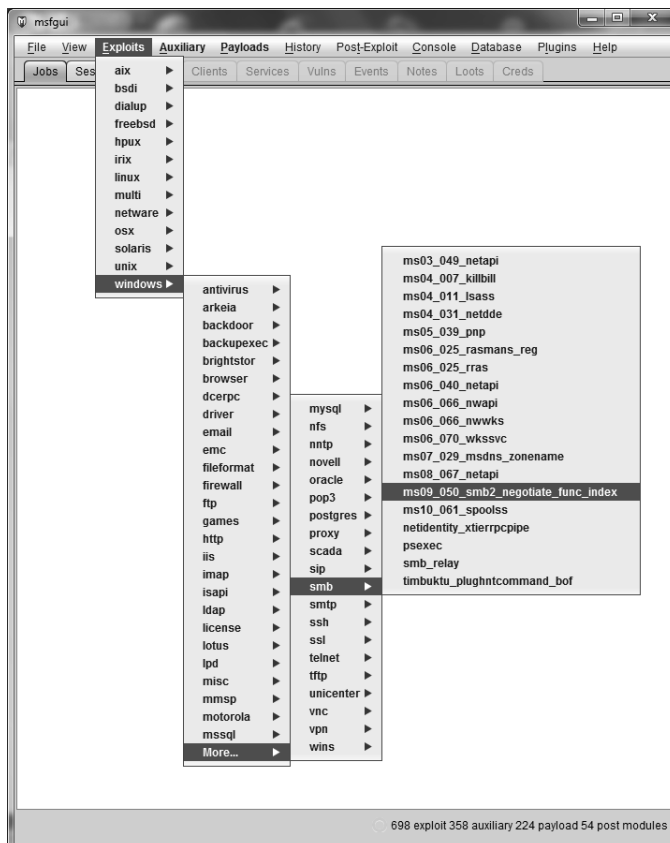


Abb. 2-16 msfgui-Hauptfenster

Das Hauptfenster weist unterschiedliche Tabs auf, die je nach Umfang der vorliegenden Informationen aktiviert werden. Exploits und Auxiliary-Module lassen

sich über das Menü auswählen und anwenden. Eine grafische Abfrage der benötigten Optionen vereinfacht die Anwendung der einzelnen Module.

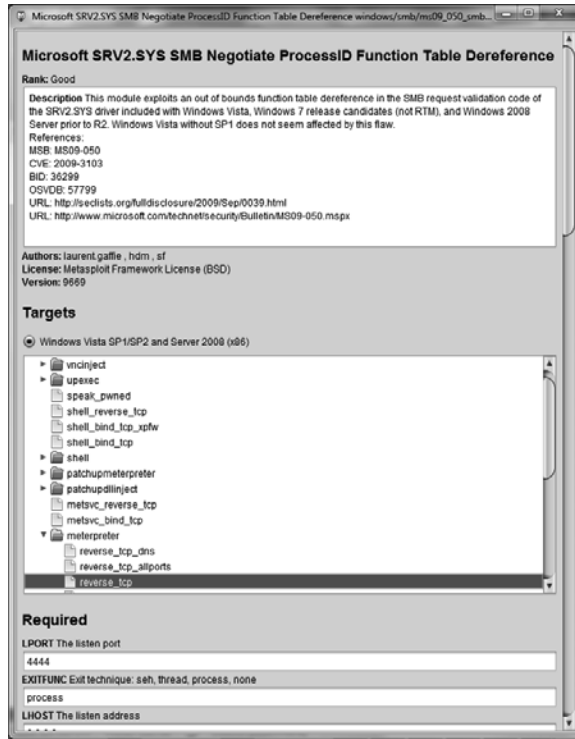


Abb. 2-17 Abfrage der benötigten Optionen

Neben einer allgemeinen Beschreibung des Moduls, wie sie auf der Konsole über den Befehl `info` dargestellt wird, muss der Payload ausgewählt werden. Abschließend müssen zumindest die Grundkonfigurationsoptionen gesetzt werden. Die erweiterten *Advanced Options* sind unter *Optional* konfigurierbar.

2.5.2 Armitage – seit Version 3.6.0

Bei Armitage handelt es sich um die zweite Java-basierte grafische Oberfläche, die eine einfache und optimierte Anwendung des Frameworks ermöglichen soll. Armitage wurde mit Version 3.6.0 in das Metasploit-Framework integriert und bedarf seitdem keiner weiteren manuellen Installation.

Armitage liegt im Metasploit-Verzeichnis `/MSF-Path/msf3/data/armitage` und lässt sich mit dem Ruby-Skript `armitage` im Metasploit-App-Verzeichnis aufrufen. Das Armitage-Skript startet den bei der Metasploit-GUI bereits dargestellten Connection-Wizard, der für den Verbindungsaufbau zum Metasploit-Server zuständig ist. Wurde der `msfrpcd` zu diesem Zeitpunkt noch nicht auf der Konsole gestartet,

lässt er sich an dieser Stelle über den Punkt Start MSF starten, wodurch nach erfolgreichem Startvorgang der Verbindungsaufbau über die XML-RPC-Schnittstelle automatisch erfolgt.

Nach dem Start von Armitage wird der Anwender von einer dreigeteilten Oberfläche empfangen. In der linken Spalte werden die vorhandenen Metasploit-Module mit einer funktionalen Möglichkeit der Suche in einer ausklappbaren Baumstruktur dargestellt. Neben dieser Moduldarstellung gibt es einen großen schwarzen Bereich, der im Laufe der Sicherheitsanalyse die erkannte Systemstruktur aufbaut bzw. darstellt. Direkt unter diesen beiden Fensterbereichen findet sich ein weiterer Bereich, der sich über Registerkarten bzw. Tabs erweitern lässt. Dieser umfasst nach dem Startvorgang die bekannte Metasploit-Konsole, die den durchgeführten Pentest jederzeit durch manuelle Optimierungen ergänzen kann.

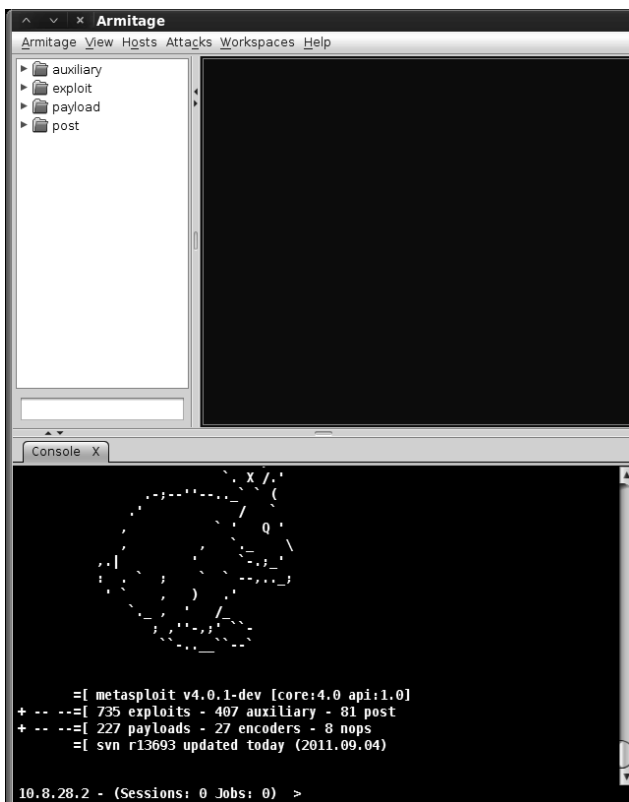


Abb. 2-18 Startbildschirm von Armitage

Das Menü des Armitage-Fensters beheimatet weitreichende Optionen und Konfigurationsmöglichkeiten. Neben der Möglichkeit, detaillierte Einstellungen vorzunehmen (Armitage → Preferences), lassen sich unter dem Menüpunkt *View* ermittelte Credentials oder die aktuell laufenden Jobs darstellen. Im Menüpunkt *Hosts*

finden sich zudem unterschiedliche Optionen, um neue Systeme hinzuzufügen. Neben dem Import verschiedener Quelldateien, beispielsweise von Nmap, Nessus, NeXpose oder MSF-Express, lassen sich Nmap-Portscans durchführen. Zudem lassen sich über den Menüpunkt *MSF Scans* unterschiedlichste Auxiliary-Module nahezu automatisiert zur Anwendung bringen.

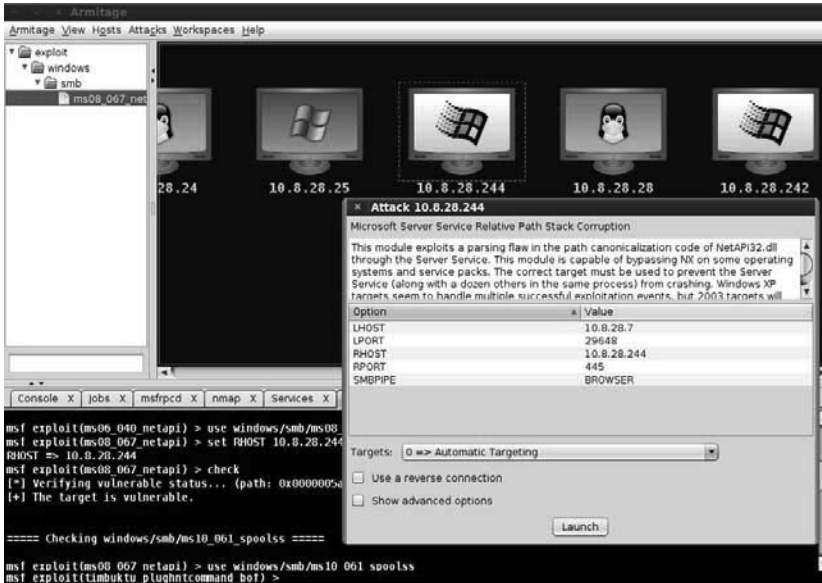


Abb. 2-19 Armitage im Einsatz

Vorhandene Module lassen sich mit einem Doppelklick auswählen und unter Zuhilfenahme des einfachen Wizard auf die ausgewählten Systeme anwenden. Der Wizard umfasst die typischen Optionen, die in der Metasploit-Konsole mit `show options` und `show advanced` abrufbar sind (siehe hierfür auch Abschnitt 2.5.4). Wurde vorab ein System in der grafischen Ansicht ausgewählt, übernimmt dieser Wizard automatisch die Option `RHOST` bzw. bei Auxiliary-Modulen die Option `RHOSTS`. Mit einem Rechtsklick in der grafischen Darstellung ist es möglich, ein weiteres Kontextmenü aufzurufen. Dieses Menü ermöglicht, je nach Status der Analyse, weitere Aktionen, angefangen von Exploiting-Vorgängen bis hin zu umfassenden Tätigkeiten der Post-Exploitation-Phase.

Kommt es im Rahmen der Penetration zu erfolgreichen Exploiting-Vorgängen, werden die betroffenen Systeme rot markiert, sind dadurch sofort erkennbar und lassen sich in der grafischen Oberfläche weiter analysieren.



Abb. 2-20 Erfolgreiche Übernahme eines Systems

Die dargestellte Oberfläche macht einen Pentest mit Metasploit in vielen Fällen transparenter und anschaulicher. Einige Features wie die durchzuführenden Port- und Service-Scans lassen sich nahezu automatisch und oftmals erheblich einfacher als manuell auf der Kommandozeile durchführen. Trotz der scheinbar einfachen Bedienung und dem intuitiven Handling dieser grafischen Oberfläche sollte jeder Anwender umfangreiches Metasploit-Know-how mitbringen.

Wurde für einen Pentest in erster Linie die Metasploit-Konsole mit Datenbankbindung eingesetzt, ist es möglich, die ermittelten Informationen in Armitage zu laden und weiterzuverwenden. Beispielsweise lässt sich die grafische Aufbereitung der Scanergebnisse für die abschließende Reporterstellung nutzen.

Hinweis: Da bei der Anwendung dieser Oberfläche sehr häufig Performanceprobleme auftreten, wird sie derzeit bei typischen Pentests kaum eingesetzt.

2.5.3 Metasploit-CLI – Command Line Interface

Neben den bislang behandelten, vorwiegend grafischen Benutzeroberflächen spielt Metasploit seine volle Stärke erst auf der Kommandozeile aus. Das mitgelieferte Kommandozeileninterface (Command Line Interface, CLI) lässt sich sehr gut direkt von der Linux-Konsole aus einsetzen und dementsprechend auch überaus einfach in eigene Skripte einbinden. Prinzipiell handelt es sich bei der CLI um einen Wrapper für die eigentliche Metasploit-Konsole (msfconsole), der eine et-

was abgewandelte Befehlsyntax benötigt, die msfconsole aufruft und abschließend wieder auf die Linux-Konsole zurückkehrt. Wird die Metasploit-CLI ohne Parameter aufgerufen, kommt es zur Anzeige der Hilfsfunktion mit einer Auflistung aller vorhandenen Module.

```
root@bt:~# msfcli -h
Usage: /MSF-Path/msf3/msfcli <exploit_name> <option=value> [mode]

=====
Mode           Description
----           -
(H)elp         You're looking at it baby!
(S)ummary      Show information about this module
(O)ptions      Show available options for this module
(A)dvanced     Show available advanced options for this module
(I)DS Evasion  Show available ids evasion options for this module
(P)ayloads     Show available payloads for this module
(T)argets      Show available targets for this exploit module
(AC)tions     Show available actions for this auxiliary module
(C)heck        Run the check routine of the selected module
(E)xecute     Execute the selected module
```

Listing 2-10 Hilfsfunktion der CLI

Eine integrierte Suchfunktion ist nicht direkt vorhanden. Diese lässt sich folgendermaßen mit einem einfachen grep auf der Linux-Kommandozeile realisieren:

```
root@bt:~# msfcli | grep ms08_067
[*] Please wait while we load the module tree...
    exploit/windows/smb/ms08_067_netapi  Microsoft Server Service Relative
                                         Path Stack Corruption
```

Listing 2-11 Suchmöglichkeit mit der CLI

Wurde dabei ein passendes Modul gefunden, lassen sich dessen Optionen mit einem abschließenden 0, welches auf der Metasploit-Konsole einem show options entspricht, ausgeben. Die erweiterten Optionen lassen sich mit einem A für show advanced oder mit einem S, welches die Ausgabe des info-Befehls darstellt, anzeigen.

```
root@bt:~# msfcli exploit/windows/smb/ms08_067_netapi 0
[*] Please wait while we load the module tree...

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

Listing 2-12 msfcli-Moduloptionen

Im nächsten Schritt müssen die benötigten Optionen gesetzt werden. Dies erfolgt auf der Kommandozeile in der Form `OPTION=Value`:

```
root@bt:~# msfcli exploit/windows/smb/ms08_067_netapi RHOST=10.8.28.1 0
```

[*] Please wait while we load the module tree...

Name	Current Setting	Required	Description
RHOST	10.8.28.1	yes	The target address
RPORT	445	yes	Set the SMB service port
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

Listing 2-13 Setzen einer Option über die Kommandozeile

Wurden alle Optionen korrekt gesetzt und ein passender Payload definiert, ist es möglich, das Modul mit einem abschließenden `E` für Execute zur Anwendung zu bringen.

Bereits bei ersten Tests dieser Eingabeoberfläche wird der größte Nachteil dieser Benutzerschnittstelle deutlich spürbar. Es ist die Ladezeit, die bei jedem Absetzen eines Befehls anfällt und eine möglichst schnelle und zeitlich optimierte Arbeitsweise enorm erschwert.

2.5.4 Einführung in die Metasploit-Konsole (msfconsole)

Bei der Metasploit-Konsole handelt es sich um die am häufigsten eingesetzte Oberfläche des Frameworks. Im Rahmen dieses Buches werden zwar von Zeit zu Zeit auch die anderen Oberflächen verwendet, aber den Hauptbestandteil wird die im folgenden Abschnitt dargestellte Metasploit-Konsole ausmachen. Dies liegt zum einen an der sehr einfachen und schnellen Anwendung, zum anderen lassen sich mit einem Grundverständnis dieser Benutzeroberfläche auch die kommerziellen Produkte von Rapid7 wesentlich besser anwenden und evtl. auftretende Fehler einfacher interpretieren.

2.5.4.1 Startvorgang und Hilfsfunktion der Konsole

Die Konsole wird auf der Linux-Kommandozeile mit dem Befehl `msfconsole` aufgerufen. Nach einem kurzen Ladevorgang wird der Anwender von einem ASCII-Splash-Screen mit weiteren Details begrüßt (Abb. 2-21).

Diese Details umfassen den Zeitpunkt des letzten Updates (30.08.2011), die gestartete Version (4.0.1-dev), das SVN-Release (r13664) sowie die Anzahl der Exploits, Auxiliary-Module, Payloads, Encoders und Nops. Sollen im Laufe der Arbeit mit der Metasploit-Konsole die beim Startvorgang dargestellten Informationen erneut abgefragt werden, lässt sich der Befehl `banner` nutzen.



Abb. 2-21 Msfconsole

Nach dem erfolgten Start der Konsole lässt sich mit dem help-Befehl bereits ein erster Überblick der vorhandenen Umgebung bzw. des Befehlssatzes einholen.

2.5.4.2 Tab-Completion und Show-Kommando

Das erste überaus hilfreiche Feature der Metasploit-Konsole ist die TAB-Completion. Die meisten Leser werden diese Autovervollständigung bereits von der Linux-Konsole kennen. Bereits nach der Eingabe des ersten Buchstabens und dem einfachen Drücken der TAB-Taste zeigt Metasploit alle möglichen Befehle an, die mit dem eingegebenen Buchstaben starten. Geben wir beispielsweise show TAB+TAB (zweimal die TAB-Taste drücken) ein, erhalten wir folgende Ausgaben:

```

msf > show + <TAB> + <TAB>
show all          show exploits  show payloads
show auxiliary   show nops      show plugins
show encoders    show options   show post

```

Listing 2-14 TAB-Completion

Geben wir anschließend an den show-Befehl ein Leerzeichen und ein ex, gefolgt von TAB TAB ein, wird unsere Eingabe zu show exploits vervollständigt. Diese Auto-Vervollständigung funktioniert nicht nur bei Befehlen, sondern auch bei Modulen, Payloads und den zu setzenden Optionen.

Mit show exploits kommt es zur Auflistung aller verfügbaren Exploit-Module.

```
msf > show -h
```

```
[*] Valid parameters for the "show" command are:
      all, encoders, nops, exploits, payloads,
      auxiliary, plugins, options
```

```
[*] Additional module-specific parameters are:
      advanced, evasion, targets, actions
```

```
msf > show exploits
```

```
Exploits
```

```
=====
```

Name	Disclosure Date	Rank	Description
----	-----	----	-----
windows/sip/aim_triton_cseq	2006-07-10	great	AIM Triton
windows/sip/sipxezphone_cseq	2006-07-10	1.0.4 CSeq Buffer Overflow great	SIPfoundry
windows/sip/sipxphone_cseq	2006-07-10	great	SIPfoundry
windows/smb/ms03_049_netapi	2003-11-11	good	Microsoft
windows/smb/ms08_067_netapi	2008-10-28	great	Microsoft
windows/smb/ms10_061_spoolss	2010-09-14	excellent	Microsoft
windows/smb/psexec	1999-01-01	manual	Microsoft
windows/smb/smb_relay	2001-03-31	excellent	Microsoft

```
<snip>
```

Listing 2-15 Show-Kommando

Bei den Exploits werden als weitere Information zusätzlich das *Datum der Veröffentlichung* und ein sogenanntes *Ranking* angeführt. Im Ranking spiegeln sich Faktoren wider, wie beispielsweise die Wahrscheinlichkeit, ob der Service oder gar das ganze Betriebssystem bei der Anwendung des Exploits abstürzt, wie zuverlässig der Exploit ist und ob er eine Codeausführung ermöglicht.

2.5.4.3 Search-Kommando

Die mit dem show-Kommando ausgegebene Liste kann sehr umfangreich sein (derzeit über 700 Exploits). Soll diese Informationsflut etwas eingedämmt werden, eignet sich dafür das search-Kommando. Dieser Suchbefehl unterstützt verschiedene Filtermethoden, wie beispielsweise die unterstützte Plattform oder die Art des Moduls.

- **name:** Innerhalb des Modulnamen suchen name:Microsoft
- **path:** Innerhalb des Modulpfades suchen path:windows/smb
- **platform:** Nach einer speziellen Plattform suchen platform:linux
- **type:** Nach einem Modultyp suchen (exploit, auxiliary, oder post) type:exploit
- **app:** Client- oder Server-Module durchsuchen app:client
- **author:** Nach einem speziellen Autor suchen author:hdm
- **cve:** Nach einer CVE-ID suchen cve:2011
- **bid:** Nach einer Bugtraq-ID suchen bid:31874
- **osvdb:** Nach einer OSVDB-ID suchen osvdb:49243

Wird beispielsweise ein Exploiting-Modul für den Windows-SMB-Service gesucht, lässt sich folgender Aufruf einsetzen:

```
msf > search type:exploit name:Microsoft path:smb

Matching Modules
=====

   Name                               Disclosure Date Rank
Description
----                               -
-----
  exploit/windows/smb/ms03_049_netapi  2003-11-11      good
Microsoft Workstation Service NetAddAlternateComputerName Overflow
  exploit/windows/smb/ms04_007_killbill 2004-02-10      low
Microsoft ASN.1 Library Bitstring Heap Overflow
  exploit/windows/smb/ms04_011_lsass    2004-04-13      good
Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflow
  exploit/windows/smb/ms04_031_netdde   2004-10-12      good
Microsoft NetDDE Service Overflow
  exploit/windows/smb/ms05_039_pnp      2005-08-09      good
Microsoft Plug and Play Service Overflow
  exploit/windows/smb/ms06_025_rasmans_reg 2006-06-13      good
Microsoft RRAS Service RASMAN Registry Overflow
  exploit/windows/smb/ms06_070_wkssvc    2006-11-14      manual
Microsoft Workstation Service NetpManageIPCCconnect Overflow
  exploit/windows/smb/ms07_029_msdns_zonename 2007-04-12      manual
Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
<snip>
```

Listing 2-16 Search-Kommando in der Anwendung

Eine Suche nach einer speziellen CVE-Nummer ist folgendermaßen durchführbar:

```
msf > search cve:2008-4250
```

Name	Disclosure Date	Rank	Description
----	-----	----	-----
exploit/windows/smb/ms08_067_netapi	2008-10-28	great	Microsoft Server Service Relative Path Stack Corruption

Listing 2-17 Suche nach einer CVE-Nummer

Um alle Windows-Reverse-Meterpreter-Payloads darzustellen, lässt sich folgende Suchanfrage starten:

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

Name	Rank	Description
----	----	-----
payload/windows/meterpreter/reverse_tcp	normal	Windows Meterpreter (Reflective Injection), Reverse TCP Stager
payload/windows/meterpreter/reverse_tcp_allports	normal	Windows Meterpreter (Reflective Injection), Reverse All-Port TCP Stager
payload/windows/meterpreter/reverse_tcp_dns	normal	Windows Meterpreter (Reflective Injection), Reverse TCP Stager (DNS)
payload/windows/patchupmeterpreter/reverse_tcp	normal	Windows Meterpreter (skape/jt injection), Reverse TCP Stager
payload/windows/patchupmeterpreter/reverse_tcp_allports	normal	Windows Meterpreter (skape/jt injection), Reverse All-Port TCP Stager
payload/windows/patchupmeterpreter/reverse_tcp_dns	normal	Windows Meterpreter (skape/jt injection), Reverse TCP Stager (DNS)
payload/windows/x64/meterpreter/reverse_tcp	normal	Windows x64 Meterpreter, Windows x64 Reverse TCP Stager

Listing 2-18 Search-Kommando in der Anwendung – Reverse-TCP-Meterpreter-Payloads für Windows-Systeme

Soll ein Suchvorgang über alle Module, ohne Einschränkung auf eine spezielle Modulgruppe, durchgeführt werden, ist es möglich, mit search Suchbegriff eine Suchanfrage über den vollständigen Modulbereich durchzuführen.

Hinweis: In früheren Metasploit-Versionen war anstelle der dargestellten Möglichkeit eine Einschränkung auf einzelne Modulkategorien mit dem Parameter -t möglich. Seit Version 3.7.1 kommt die hier dargestellte Methode zum Einsatz.

2.5.4.4 use- und back-Kommando

War es über die show- und search-Methode möglich, ein passendes Modul zu finden, muss dieses im nächsten Schritt ausgewählt und konfiguriert werden. Ein benötigtes Modul wird mit dem Befehl use <Modulpfad/Modulname> ausgewählt. Mit diesem Befehl wird in die Modulebene gewechselt; dort kommt es zur Konfiguration und Anwendung des ausgewählten Moduls. Diese Ebene ist an der neuen Eingabeaufforderung erkennbar:

```
msf exploit(<MODULNAME>) >
```

Hier getroffene Einstellungen gelten im Normalfall ausschließlich für dieses Modul. Mit dem Befehl back ist es möglich, die Modulebene zu verlassen und zurück in die Hauptebene zu wechseln.

```
msf > use windows/smb/ms09_050_smb2_negotiate_func_index
msf exploit(ms09_050_smb2_negotiate_func_index) > back
msf >
```

Listing 2-19 use- und back-Kommando in der Anwendung

Kommt es im Anschluss erneut zu einem Aufruf des bereits konfigurierten Moduls, sind die davor getroffenen Einstellungen erhalten geblieben. Diese bleiben bis zum Beenden der aktuellen Metasploit-Sitzung oder zum Neusetzen bzw. zum Löschen (siehe Abschnitt 2.5.4.6) der Optionen. Beim Setzen von Optionen ist zu beachten, dass die mit dem set-Befehl gesetzten Optionen eines Moduls ausschließlich für das jeweilige Modul Gültigkeit haben.

2.5.4.5 Anzeige von Moduloptionen und weiteren Informationen

Jedes Metasploit-Modul bringt unterschiedlichste Konfigurationsmöglichkeiten in Form von Optionen mit. Bei Auxiliary-Modulen muss typischerweise die Zieladresse bzw. der Adressbereich des Zielsystems angegeben werden. Bei Exploits müssen zudem noch ein Payload und unterschiedliche weiterführende Payload-Optionen gesetzt werden. Die möglichen Optionen lassen sich nach der Auswahl des Moduls mit dem in Listing 2-20 dargestellten show options anzeigen.

```
msf exploit(ms09_050_smb2_negotiate_func_index) > show options
```

Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):

Name	Current Setting	Required	Description
RHOST		yes	The target address
RPORT	445	yes	The target port
WAIT	180	yes	The number of seconds to wait for the attack to complete.

Exploit target:

```

  Id  Name
  --  ---
  0   Windows Vista SP1/SP2 and Server 2008 (x86)

```

Listing 2–20 *show options*

Das dargestellte Kommando stellt die für eine Anwendung des Moduls essenziellen Optionen dar. Häufig ermöglicht ein Modul weitere, sogenannte »advanced options«. Diese Optionen umfassen unterschiedlichste Möglichkeiten, den Exploit und den Payload für spezielle Umgebungen oder Anwendungsfälle zu optimieren. Häufig kann man dabei den Payload-Handler deaktivieren, Timeouts definieren oder SSL und ähnliche Optionen anpassen. Diese erweiterten Optionen lassen sich mit dem Befehl `show advanced` aufrufen.

Weitere Modulinformationen lassen sich zudem mit dem Befehl `info` abrufen. Die mit diesem Befehl dargestellten Details umfassen neben den bereits bekannten Grundoptionen eine Beschreibung, weiterführende Informationen wie Internetressourcen und Details zur Version, zur Plattform und zur Verlässlichkeit des Moduls.

2.5.4.6 Setzen (set) und löschen (unset) von Optionen

Im Anschluss an die Auswahl des benötigten Moduls und die Ermittlung der benötigten Optionen müssen diese Parameter konfiguriert werden. Metasploit bringt für die Konfiguration solcher Parameter den `set`-Befehl mit.

set PARAMETER option

Sobald der `set`-Befehl abgesetzt wurde, stellt Metasploit den neu konfigurierten Befehl dar und kehrt zum Eingabeprompt zurück.

```

msf exploit(ms09_050_smb2_negotiate_func_index) > set RHOST 10.8.28.1
RHOST => 10.8.28.1
msf exploit(ms09_050_smb2_negotiate_func_index) > show options

Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     10.8.28.1       yes       The target address
  RPORT     445              yes       The target port
  WAIT     180              yes       The number of seconds to wait for
                                         the attack to complete.

```

Exploit target:

```

  Id  Name
  --  ---
  0   Windows Vista SP1/SP2 and Server 2008 (x86)

```

Listing 2–21 *set-Kommando*

Wird ein Parameter falsch gesetzt oder wird er nicht weiter benötigt, lässt er sich mit dem Befehl `unset` löschen bzw. in seinen Originalzustand zurücksetzen. Mit `show options` und `show advanced` ist es jederzeit möglich, den aktuellen Status aller Optionen abzurufen.

Hinweis: Mit dem Befehl `set` ohne weitere Parameter ist es möglich, den Status aller Optionen abzufragen (siehe Abschnitt 2.6).

```
msf exploit(ms09_050_smb2_negotiate_func_index) > unset RHOST
Unsetting RHOST...
msf exploit(ms09_050_smb2_negotiate_func_index) > show options
Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):
  Name      Current Setting  Required  Description
  ----      -
  RHOST     445              yes       The target address
  RPORT     180              yes       The target port
  WAIT      180              yes       The number of seconds to wait for
                                     the attack to complete.
```

<snip>

Listing 2-22 *unset-Kommando*

Die bereits dargestellten `show`- und `search`-Funktionen lassen sich in der Modulebene ebenso anwenden wie in der Hauptebene. Es ist somit kein Zwischenschritt in die Hauptebene nötig.

Gibt man in der Exploit-Ebene den `help`-Befehl ein, so ist die Hilfe um eine weitere Sparte, die exploitspezifischen Kommandos, erweitert.

```
<snip>
Exploit Commands
=====
  Command      Description
  -
  check        Check to see if a target is vulnerable
  exploit       Launch an exploit attempt
  rcheck       Reloads the module and checks if the target is vulnerable
  reload       Just reloads the module
  rexploit     Reloads the module and launches an exploit attempt
```

Listing 2-23 *Hilfe für Exploit-Module*

2.5.4.7 Externe Kommandos

Im Rahmen von Sicherheitsanalysen mit Metasploit kann es zu der Situation kommen, dass möglichst schnell bestimmte Daten oder Informationen vom loka-

len Betriebssystem benötigt werden. Prinzipiell stellt diese Situation kein Problem dar, da sich eine weitere Shell öffnen lässt und sich dort der Befehl absetzen lässt. Metasploit bietet hierfür die wesentlich schnellere Möglichkeit, direkt aus der Metasploit-Konsole Systembefehle abzusetzen.

Das beste Beispiel zur Anwendung von externen Kommandos ist die Konfiguration eines Reverse-Shell-Payloads, für den wir die lokale IP-Adresse benötigen. Um dabei möglichst rasch und ohne Zeitverzögerung weiterarbeiten zu können, lässt sich der Linux-Befehl `ifconfig eth0` direkt in der Metasploit-Konsole absetzen:

```
msf exploit(ms09_050_smb2_negotiate_func_index) > ifconfig eth0
[*] exec: ifconfig eth0

eth0      Link encap:Ethernet  HWaddr 00:0c:29:cf:6a:ba
          inet addr:10.8.28.9  Bcast:10.8.28.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6a:64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1482745 errors:0 dropped:8019 overruns:0 frame:0
          TX packets:409353 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:621640900 (621.6 MB)  TX bytes:57410895 (57.4 MB)
          Interrupt:18 Base address:0x2000

msf exploit(ms09_050_smb2_negotiate_func_index) > set LHOST 192.168.1.102
<snip>
msf exploit(ms09_050_smb2_negotiate_func_index) > ls /MSF-
Path/msf3/data/wordlists
[*] exec: ls /MSF-Path/msf3/data/wordlists

db2_default_pass.txt
db2_default_userpass.txt
db2_default_user.txt
hci_oracle_passwords.csv
```

Listing 2–24 Externe Unix-Kommandos in Metasploit ausgeführt

In Listing 2–24 zeigt die Zeile mit dem `exec:-`Befehl, dass es sich um einen Systembefehl handelt, der von der Metasploit-Konsole an das Betriebssystem übergeben wird. Diese Vorgehensweise hilft ungemein, wenn es um eine rasche und effektive Arbeitsweise geht.

2.5.4.8 Kommandozeilen-Prompt

Metasploit bietet seit der Version 4 die Möglichkeit, den typischen Prompt der Metasploit-Konsole an die eigenen Bedürfnisse anzupassen. Dies umfasst neben der Anzeige der aktuellen Zeit und des Datums auch farbliche Anpassungen und beispielsweise die Anzeige aktueller Statusinformationen wie die aktiven Sessions oder die gerade laufenden Jobs. Folgendes Listing formatiert im ersten Schritt die Datums- und Uhrzeitanzeige. Im zweiten Schritt wird der Kommandozeilen-

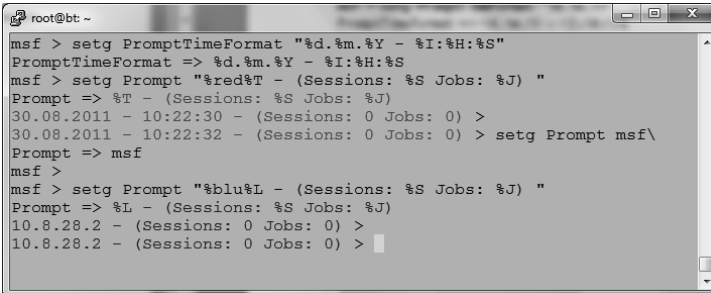
Prompt mit weiteren relevanten Informationen wie %S für die aktuellen Sessions und %J für die aktiven Jobs ausgestattet.

```
msf > setg PromptTimeFormat "%d.%m.%Y - %I:%H:%S"
PromptTimeFormat => %d.%m.%Y - %I:%H:%S

msf > setg Prompt "%red%T - (Sessions: %S Jobs: %J) "
Prompt => %T - (Sessions: %S Jobs: %J)
22.07.2011 - 11:23:25 - (Sessions: 0 Jobs: 0) >
```

Listing 2-25 Prompt anpassen

Mit einem zusätzlichen %L würde der Prompt in Zukunft auch automatisch die lokale IP-Adresse anzeigen. Dieser Prompt lässt sich auch dementsprechend farblich anpassen. Im dargestellten Beispiel wurde Rot für den Prompt gewählt.



```
root@bt: ~
msf > setg PromptTimeFormat "%d.%m.%Y - %I:%H:%S"
PromptTimeFormat => %d.%m.%Y - %I:%H:%S
msf > setg Prompt "%red%T - (Sessions: %S Jobs: %J) "
Prompt => %T - (Sessions: %S Jobs: %J)
30.08.2011 - 10:22:30 - (Sessions: 0 Jobs: 0) >
30.08.2011 - 10:22:32 - (Sessions: 0 Jobs: 0) > setg Prompt msf\
Prompt => msf
msf >
msf > setg Prompt "%blu%L - (Sessions: %S Jobs: %J) "
Prompt => %L - (Sessions: %S Jobs: %J)
10.8.28.2 - (Sessions: 0 Jobs: 0) >
10.8.28.2 - (Sessions: 0 Jobs: 0) >
```

Abb. 2-22 Anpassen des Kommandozeilen-Prompts

Hinweis: Mit dem save-Kommando lassen sich die getroffenen Einstellungen in der Grundkonfiguration ablegen und dementsprechend bei jedem Startvorgang automatisch laden.

2.6 Globaler und modularer Datastore

Jedes Modul benötigt diverse Optionen, die vor einer erfolgreichen Anwendung gesetzt werden müssen. Beispielsweise müssen typischerweise bei Scanning-Modulen die Optionen RHOSTS und THREADS angepasst werden.

Werden diese Optionen nun bei Modul A mit set XYZ angepasst und wird anschließend zu Modul B gewechselt, dann müssen alle Anpassungen erneut durchgeführt werden. Die Optionen wurden durch die Verwendung von set ausschließlich im lokalen Datastore (im Modul-Datastore) von Modul A abgelegt.

Um solche Optionen nicht in jedem Modul erneut setzen zu müssen, gibt es den globalen Datastore. Optionen, die in diesem Datastore gesetzt sind, gelten für alle Module und müssen nicht jedes Mal neu gesetzt werden.