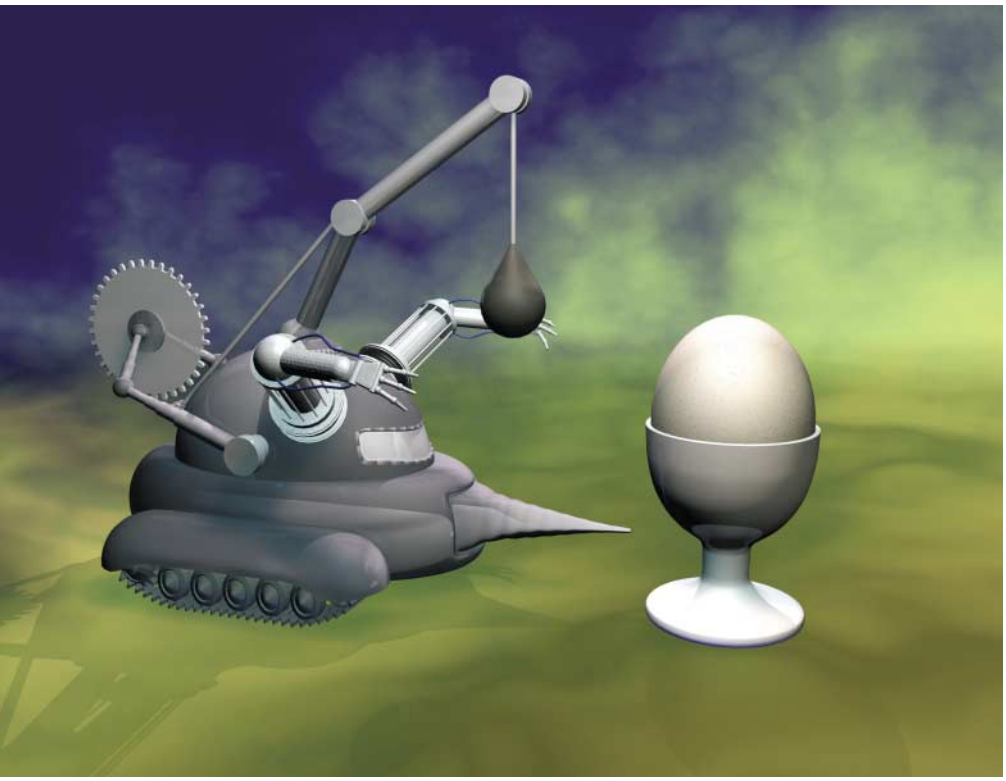


Automatisierte Einbruchtests

Hack-O-Mat

Christoph Puppe

Das Sicherheitswerkzeug Core Impact bildet fast den gesamten Workflow eines Penetrationstests in einer Software ab. Ob mit seiner Hilfe der Administrator den Einbruchstester ersetzen kann, und ob das bekannte Tool auch sonst hält, was es verspricht, zeigt der *iX*-Test.

Dank der automatisierten Durchführung von Einbruchstests – so genannter Penetrationstests – soll das Werkzeug Core Impact der Bostoner Firma Core Security Administratoren in die Lage versetzen, bei diesen Tests ohne die Unterstützung von Spezialisten auszukommen. Überdies soll es flexibel und mächtig genug sein, einen professionellen Penetrationstester zu unterstützen. Derzeit gibt es keine vergleichbare Software, einen ähnlichen Ansatz hat lediglich Canvas von Immunity (www.immunitysec.com) – allerdings in eingeschränktem Umfang mit nur 50 Exploits und weniger Funktionen als Core. Die Open-Source-Alternative Metasploit hingegen (www.metasploit.com) konzentriert sich auf die Exploit-Phase eines Penetrationstests. Im Test befand sich Core Impact Version 3.3. Die nagelneue Version 4.0 hatte der Hersteller nicht freigegeben, lediglich ein Test der neuen Features sowie ein Ausblick durften in den Artikel eingehen.

Vom ersten Scan eines Netzwerkes bis zum Bericht ist alles über eine Oberfläche von einem Rechner aus zu erledigen. Die umfangreicheren unter den Schwachstellenscannern wie Nessus, ISS oder LANGuard sind ähnlich konzipiert, sind in ihren Funktionsumfang aber reduzierter. Auch hat der Tester bei ihnen keine Chance, während des Scanvorgangs einzugreifen, hingegen lässt sich bei Core Impact jeder Teilprozess – manuell oder per Skript – einzeln starten. Das Werkzeug kann ein Netz sondieren, Banner, Ports, Betriebssysteme sowie andere relevante Informationen in Erfahrung bringen und enthält außerdem circa 100 Exploits.

Ein solches Exploit-Modul liefert nicht nur die Information, ob eine Schwachstelle vorhanden ist, sondern führt den Einbruch auch wirklich aus. Der Benutzer erhält auf diese Weise Zugang zum angegriffenen System und kann weitere Sondierungen und Angriffe durchführen. Leider ist die Anzahl der unterstützten Zielsysteme nicht sehr hoch, der Hersteller nennt lediglich Red Hat, OpenBSD, Windows (NT, 2000, XP) sowie Solaris auf Sparc.

Installation und Aufbau

Das Werkzeug ließ sich auf Windows 2000 ohne Schwierigkeiten installieren. Mit im Softwarepaket befindet und installiert sich gleich Version 3.0 der freien Bibliothek *pcap*, die unter Windows die Schnittstelle zum Analysieren und Verarbeiten von Paketen liefert. Eine Update-Funktion ist integriert. Core Impact ist als Framework für Module aufgebaut. Diese sind in Python geschrieben und stehen auf Klick zur Bearbeitung zur Verfügung. Wer eigene Module entwickeln möchte, kann ohne großen Aufwand neue Funktionen und Exploits einbauen und benutzen. Kein anderer Hersteller geht so offen mit dem „Innenleben“ seiner Software um. Der Kunde weiß dadurch immer genau, was diese im System anstellt.

Der Hersteller stellt damit Sicherheitsberatungsunternehmen ein Werkzeug zur Verfügung, das auf allen Arbeitsplätzen eine gleichmäßige Qualität garantiert. Der Aufwand ist zunächst allerdings beträchtlich, da viele spezialisierte Werkzeuge und eine große Zahl von Exploits in Python neu zu schreiben sind. Die Makro-Programmierbarkeit der Module erlaubt es jedoch, dass einmal festgelegte Vorgehensweisen problemlos reproduzierbar sind.

Beim Starten fordert das Werkzeug zum Anlegen oder Öffnen eines „Workspace“ auf, der ein Passwort erfordert. Bei diesem Vorgang erzeugt man außer-

dem mit der Maus eine große Menge Zufallszahlen. Entgegen erster Hoffnungen dienen diese jedoch nur dem Erzeugen eines Schlüssels zur Authentifizierung und nicht der Verschlüsselung der Daten des Workspace. Er präsentiert einen Arbeitsplatz mit fünf Fenstern, der selbsterklärend ist und den der Benutzer auf die eigenen Bedürfnisse anpassen kann.

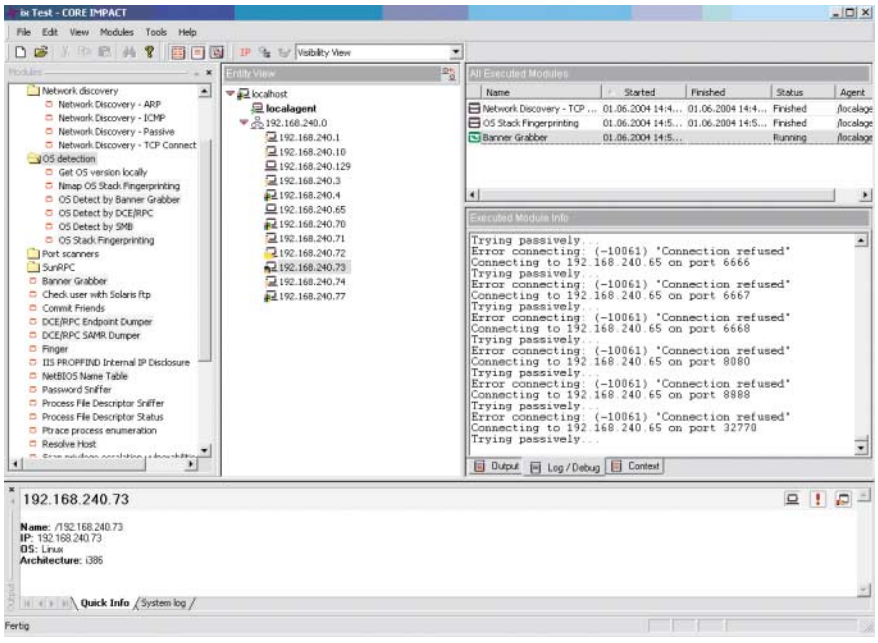
Erste Schritte

Wenn ein Modul ausgeführt werden soll, benötigt es dazu einen Zielrechner. Solange man noch in kein System eingebrochen ist, steht nur der eigene Rechner zur Verfügung. Dieser „localagent“ ist als Source voreingestellt. Das Vorgehen unter Core Impact ist exakt das gleiche wie auf der Kommandozeile. Erst sondiert der Einbruchstester mit den Modulen unter „Network Discovery“ das Netzwerk und sucht die vom eigenen Rechner aus erreichbaren Systeme, die im Entity View dargestellt sind. Nun folgen Portscan, Banner-Grabbing (das Beschaffen von Betriebssystems-, Versions- oder Softwareinformationen anhand der Welcome- und Infobanner von Serverdiensten) und Betriebssystemerkennung, indem er auf der grafischen Oberfläche die jeweiligen Module auf das Netz oder den einzelnen Rechner zieht. Mehrere Module des Tools können mit verschiedenen Methoden auf dem System befindliche Benutzer – teilweise nebst Passwörtern – ausfindig machen und auflisten.

Bis zu diesem Punkt hat Core Impact schon die Arbeit von circa zehn Werkzeugen erledigt und die Ergebnisse in eine Datenbank eingetragen. Die ersten Schwächen treten allerdings auch schon zu Tage. So kann Core Impact zwar das Betriebssystem durch Parsen der gefundenen Banner erkennen, das entsprechende Modul greift aber nicht auf die Datenbank der bereits entdeckten offenen Ports zurück, was zu langen Wartezeiten führt (siehe Abbildung 1, „Error Connecting“). Ein zweites Modul, das ebenfalls das Banner-Grabbing beherrscht, kann das zwar, trägt dafür aber das Betriebssystem nicht in die Datenbank ein.

Das Datenbankfenster ist im Lieferzustand nicht sichtbar, der Benutzer muss es öffnen. Wenn er es für erforderlich hält, kann er sich die Banner anschauen und die darin befindlichen Informationen in der Datenbank manuell ergänzen. Wer Linux als Betriebssystem einträgt, erhält unter „Distribution“ gleich eine Auswahl der fünf bekanntesten Systeme (Suse, Red Hat etc.). Diese manuelle Ergänzungsmöglichkeit kann manchmal sinnvoll sein, etwa wenn die automatische Betriebssystemerkennung versagt und deswegen keine Exploits eingesetzt werden können, da diese sich auf die Datenbank stützen.

Es stehen ausreichend Module zur Verfügung, mit denen der Administrator das Zielnetz erforschen kann. Durch das Ausführen der auf Erkennung ausgerichteten Module – Stichwort „Network Discovery“ – kann man eine Menge herausfinden, auch wenn



Ausführbare Module, Zielsysteme und laufende Tests sind auf der benutzerfreundlichen Arbeitsoberfläche – hier in der Standardeinstellung beim Banner-Grabben – mit einem Blick zu erfassen (Abb. 1).

X-TRACT

- Core Impact automatisiert den Ablauf eines Penetrationstests und erspart dem Benutzer dadurch einige Arbeit und Werkzeuge.
- Das Erforschen des Netzes und Durchführen einfacher Exploits mit dem Einbruchswerkzeug führt zu Erkenntnissen über Schwachstellen im System.
- Überflüssig ist der menschliche Einbruchstester noch lange nicht – Kompetenz lässt sich auch in diesem Fall nicht durch Software ersetzen.

manchmal das Format der Informationen, die in die Datenbank eingetragen werden, unterschiedlich ist und ein Modul die Werte des anderen überschreibt. Bevor das passiert, kann der Benutzer jedoch auswählen, welche Werte er erhalten möchte.

Die Erkennung funktioniert nicht immer ganz zuverlässig, ein ICMP-Scan ergab beispielsweise 130 Systeme, tatsächlich vorhanden waren jedoch lediglich 12. Vorteilhaft ist hingegen, dass der Benutzer sämtliche Eingaben im Fenster Entity View manuell editieren kann. Eine Fehlerkorrektur kann somit in wenigen Sekunden erfolgen.

Das Icon für die Systeme im Entity View passt sich bei erkanntem Betriebssystem sofort an. Router haben leider kein eigenes Symbol. Bis auf FreeBSD 5.1 erkannte das Werkzeug alle getesteten Betriebssysteme, bei Win2k3, Solaris sowie OpenBSD zudem die Major-Versionsnummer und bei Linux die Kernel-Version. Ein anderes Modul, aufbauend auf der *nmap*-Datenbank, zeigt eine vergleichbare Erkennungsleistung, allerdings mit kleineren Fehlleistungen wie das fälschlicherweise als Windows 2000 identifizierte, vom ersten Modul jedoch richtig erkannte Windows 2003.

Zu Bemängeln ist außerdem der fehlende Abgleich zwischen den Systemen und ihren spezifischen Ver-

wundbarkeiten. Das Fingermodul etwa zeigt bei allen Systemen eine Schwachstelle an, die sich jedoch nur im Zusammenhang mit Solaris ausnutzen lässt (Solaris-Bug aus dem Jahr 2000). Bei Windows-Systemen funktioniert die Erkennung, etwa durch Nullsessions, deutlich besser. Die Tatsache, dass das Werkzeug mehr Informationen über Win2k denn über Win2k3 liefert, ist systembedingt.

Die Integration von Ergebnissen anderer Tools wie Nessus oder *nmap* könnte deutlich besser sein. Beim Import von Nessus-NSR-Dateien landet in der Datenbank lediglich die Mitteilung, dass das System existiert, nicht aber die weiteren in der Datei enthaltenen Informationen.

Einbrechen erwünscht

Nach der Erkundung der Systeme geht es ans Eingemachte, an das Ausführen der Exploits. Die Kunst des Einbruchstesters besteht darin, anhand einer Liste von Systemen und ihren Ports einen erfolgreichen Einbruch in die Systeme auszuführen. Dieses Procedere soll Core Impact so weit vereinfachen, dass auch in Sachen Sicherheit wenig erfahrene Administratoren zum Ziel kommen. Je nach gefundenem Betriebssystem sind jetzt auf der Arbeitsoberfläche im linken Fenster bei den Modulen (s. Abbildung 1) die unter dem Stichpunkt „Exploits/Remote“ aufgeführten Exploits farblich markiert; der Administrator kann sie bei selektiertem Zielsystem oder -netz durch Doppelklicken starten oder auf das System oder Netz ziehen. Das erscheint zunächst recht einfach und verständlich, aber der Teufel liegt im Detail.

So erscheinen beispielsweise Exploits, die gar nicht anwendbar sind, weil der Port geschlossen ist. Dagegen kann der Tester bestimmte Exploits gar nicht erst durchführen, wenn Core Impact sie für ein System nicht als gültig anerkennt und anzeigt – selbst wenn sich das Tool „irrt“. Die Anzahl der zur Verfügung stehenden Exploits ist mit circa 100 schon recht hoch, doch noch längst nicht ausreichend. Zudem sind davon nur rund 70 netzwerkbasiert, die anderen dienen dem lokalen Einsatz zum Erweitern der Rechte auf dem System. Hier meldet der Hersteller, dass er gerade vier neue Programmierer eingestellt hat und die Zahl in der nächsten Zeit noch steigen wird. Die Notwendigkeit dieser Maßnahme belegt die Tatsache, dass alleine die Seite www.securiteam.com um die 500 Exploits auflistet. Für einen Penetrationstester ist es zu aufwendig, alle notwendigen Exploits in Python umzusetzen, und ein Administrator dürfte kaum dazu in der Lage sein.

Da die meisten Exploits einen Agenten installieren, der zur Kommunikation einen zusätzlichen Port benötigt, ist dies durch eine gut gepflegte Firewall nur selten erfolgreich. Alle Versuche, mit den angezeigten Exploits einen Server erfolgreich anzugreifen, schlugen fehl. Der Einsatz von Core Impact für externe Penetrationstests erscheint deswegen nicht sehr sinnvoll. Erst als im Test die Firewall komplett deaktiviert wurde, ließen sich die Windows-Systeme mit dem DCOM-Exploit des letzten Jahres erfolgreich angreifen. Nach Installation eines Level-1-Agenten und dessen Festlegung als Source kann man die zweite DMZ scannen und angreifen. Auch hier führten die Angriffe nur zu spärlichen Ergebnissen, obwohl die Server mit den Original-Exploits erfolgreich übernommen werden können. Ursache hierfür ist, dass nicht nur wie erwähnt die Anzahl der Exploits zu niedrig ist, die vorhandenen sind auch zu spezialisiert – was am Beispiel des Solaris-Bug gut zu erkennen war.

Zusammengefasst hinterlässt dieses zentrale Feature von Core Impact keinen guten Eindruck, nicht zuletzt, weil die Exploits zwar auf die Datenbank zurückgreifen, aber es beispielsweise nicht zum Abbruch führt, wenn das Modul gegen einen geschlossenen Port eingesetzt wird. In Abbildung 1 ist zu sehen, dass es das Modul nicht erkennt, wenn der entsprechende Service nicht erreichbar ist.

007 an der Front

Ein erfolgreicher Exploit installiert einen Level-0-Agenten. Dieser ist eine Art Daemon, der im Speicher des Zielsystems läuft und Zugriff auf das System gestattet. Solche Agenten kann man für das Ausführen weiterer Aktionen als Quelle für Module angeben – wobei Level 0 recht eingeschränkt ist, da er nur um die 100 Bytes umfasst. Neu ausgeführte Module laufen dann auf diesem Agenten. Von diesem System erreichbare Rechner, lassen sich nun angreifen, als würde das Notebook des Penetrationstesters im selben Netz

ADRESSEN IM WEB

Produktseite Core Impact	www.coresecurity.com/products/coreimpact/index.php
Produktseite Canvas	www.immunitysec.com/products-canvas.shtml
Metasploit Framework	www.metasploit.com/projects/Framework/
Nessus Vulnerability Scanner	www.nessus.org
ISS Internet Scanner	www.iss.net/products_services/enterprise_protection/vulnerability_assessment/scanner_internet.php
GFI LANGuard	www.gfi.com/lannetscan/
Windows Packet Capture Library	winpcap.polito.it
Python	www.python.org/
Exploit Archiv von SecuriTeam	Securiteam.com/exploits/archive.html

Testumfeld und Konfiguration

Core Impact wurde in VMware auf einem Notebook installiert. Auf einem weiteren Rechner agierten sechs Systeme ebenfalls unter VMware. Diese als DMZ-1 festgelegte Zone war per Router über ACLs (Access Control Lists) mit dem Notebook verbunden und sollte im eigentlichen Aufbau nur über die Ports 21, 22, 23, 25, 80 und 443 vom Notebook aus zu erreichen sein. Verbindungen aus der DMZ zum Notebook waren nicht erlaubt. In dieser DMZ waren installiert: Windows 2003 Server, Windows 2000 Server, OpenBSD 3.1, FreeBSD 5.1, Red Hat 7.3 und Solaris auf Intel.

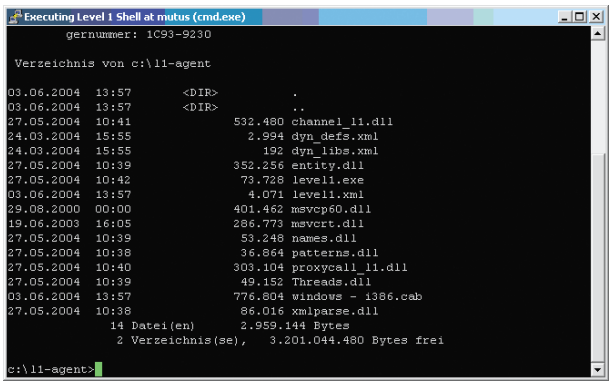
Aus dieser DMZ ging es über eine identische Verbindung in die DMZ-2. Wieder durften nur die Rechner aus der ersten DMZ Verbindungen in die zweite DMZ aufbauen, aber nicht umgekehrt. Damit sollte eine realistische Testumgebung simuliert werden, wie sie häufig in Unternehmen zu finden ist.

Da der Test in dieser Konstellation wenig Erfolg zeigte, wurde erst die Firewall deaktiviert und in einem zweiten Schritt mit einer vereinfachten, einstufigen DMZ gearbeitet. Aus ihr wurden in einem dritten Durchgang fast alle Systeme entfernt, für die der Hersteller keine Exploits liefert.

stehen (siehe Kasten „Testumfeld und Konfiguration“). Auf diese Weise kann man eine komplette DMZ von innen erforschen, nachdem nur ein Server erfolgreich angegriffen wurde.

Wenn der Agent unter einem eingeschränkten Benutzer-Account läuft, weil etwa ein Webserver den Exploit ausgeführt hat, kann man mit ihm auch lokale Exploits ausführen oder Informationen über das System sammeln. Das Installieren eines Level-1-Agenten sorgt dafür, dass man nicht den Einschränkungen von Level 0 unterliegt und so alle Module ausführen kann, die auch vom Arbeitsplatz gestartet werden können. Ein Level-0-Agent ist auf einfache Module beschränkt, die keine *pcap* benötigen. Der Level-1-Agent besteht aus einer Sammlung verschiedener Bibliotheken und einer ausführbaren Datei, die für alle vom Hersteller unterstützten Zielsysteme vorliegt. Er startet als eigener Prozess, öffnet wieder einen Port für Verbindungen und kann auch die *pcap* installieren.

Manchmal kann man eine Verbindung wieder verwenden, die auch schon der Exploit nutzte (nur Level 0). Ebenso lässt sich eine neue Verbindung über einen vom Agenten geöffneten Port auf dem Zielsystem herstellen beziehungsweise kann man den Agenten anweisen, eine Verbindung zum Angreifer aufzubauen. Damit sind die Agenten so flexibel wie ein selbst geschriebener Exploit und in unterschiedlichen Umgebungen einsetzbar. Im Test überzeugte diese Funktion, trotz einiger Schwierigkeiten. So war es nicht möglich, auf dem Win2k-System einen Level-1-Agenten zu installieren, nachdem die Installation am Tag vorher fehlgeschlagen war und der Administrator sie nicht vollständig entfernt hatte. Auch gestaltet sich die Deinstallation des Agenten schwierig, wenn ihm eine Portnummer zugewiesen wird, die nicht vom Angreifer zu erreichen ist. Der Agent läuft dann zwar, aber mangels Kontakt dringt kein Befehl zur Deinstallation zu ihm durch. Zudem kann es passieren, dass sich ein Agent bei Abbruch eines auf ihm laufenden Moduls beendet.



Bei der Durchführung eines Exploit installiert sich auf dem Zielrechner ein Agent, der dem Angreifer neben einem Datei-Browser unter anderem eine Shell zur Verfügung stellt (Abb. 2).

Alle Ereignisse, ausgeführte Tests und die ausgegebenen Resultate der Module für den Zielrechner versieht die Software mit einem Zeitstempel und protokolliert sie. Das hat für so genannte Tiger Teams – von Unternehmen engagierte Einbruchstester – eine nicht zu unterschätzende Bedeutung, da es häufig notwendig ist, dem Kunden genau darzulegen, wer was um welche Uhrzeit von wo aus durchgeführt hat. Auch lässt sich mit diesen Logfiles der Nachweis führen, dass nichts Wichtiges übersehen wurde, wenn etwa ein zweiter Penetrationstester die Arbeit seines Kollegen kontrollieren soll. Die grafische Darstellung dieses Vorgehens ist gut gelungen. Einen darin entdeckten Bug behob der Hersteller innerhalb von zwei Tagen.

In der Protokollfunktion zeigt sich ein großer Unterschied zu den anderen Verwundbarkeitsscannern: Es werden zwar alle Systeme aus dem Entity View aufgelistet, aber als Schwachstelle vermerkt das Tool nur den wirklich ausgeführten Exploit.

Fazit

Die für diesen Test vom Hersteller noch nicht freigegebene Version 4.0 bietet zusätzlich den „Rapid Penetration Test“. Er besteht aus einer Liste mit sieben Schritten für einen Einbruchstest. Auf Klick startet ein komplexes Makro, das nach Eingabe der nötigsten Daten in einen Wizard die Module für den jeweiligen Schritt aktiviert. Tatsächlich sind die Ergebnisse dieser Schritte denen des manuellen Tests der Version 3.3 vergleichbar und aussagekräftig. Die Exploits scheiterten im selben Maße wie bei 3.3. Das Entfernen der Agenten von den gehackten Systemen sowie die Reports funktionierten reibungslos.

Für den engagierten Administrator kann Core Impact interessante Einblicke und auch Erfolgserlebnisse liefern, ebenso wie es das Verständnis für das Vorgehen eines Angreifers fördert. Einen echter Penetrationstest durch einen Spezialisten kann das Werkzeug allerdings nicht ersetzen. Für den Einbruchstester bietet es eine Arbeitsgrundlage, die zahlreiche vorgefertigte automatische Arbeitsabläufe mitbringt, aber auch stark die Flexibilität einschränkt. Als ein Werkzeug unter vielen ist es sicherlich eine sinnvolle Ergänzung und oft nützliche Arbeitserleichterung; besonders bei großen Netzen, da die Module zur Sondierung durchweg überzeugten und die Datenbank gut editierbar ist. Mehr Exploits und ein Import, der auch die ausführlichen Ergebnisse anderer Scanner integriert, würden die Praxistauglichkeit stark erhöhen. (ur)

CHRISTOPH PUPPE

ist Security Consultant bei der Hisolutions AG in Berlin, CISSP und seit mehreren Jahren als Penetrationstester tätig.

DATEN UND PREISE

Core Impact 3.3

Hersteller/Anbieter: Core Security Technologies (www.coresecurity.com)

Plattformen: Windows 2000, XP

Systemvoraussetzungen: 100 MByte freier Festplattenplatz; 256 MByte RAM; Internet Explorer (ab 5.0)

Preise: 2500 US-$ für eine Lizenz für 8 IP-Nummern inkl. Updates für ein Jahr

iX-WERTUNG

- ⊕ flexibles und für den Benutzer transparentes Werkzeug
- ⊕ das modular aufgebaute Framework lässt sich jederzeit ergänzen
- ⊕ automatisierte Abläufe sparen Zeit, Arbeit und Werkzeuge
- ⊖ erlaubt nur die Durchführung angezeigter Exploits
- ⊖ Anzahl der unterstützten Zielsysteme und der Exploits ist zu gering
- ⊖ Stabilität ist verbesserungsfähig