Virtuelle Leimruten

Honeypots zur Analyse von Angriffen auf Clients und Server

Honeypots sind bewusst unsicher konfigurierte IT-Systeme, die traditionell vor allem Angreifer von Server-Systemen in die Falle locken sollen. Seit kurzem widmen sich jedoch mehrere Projekte auch der Realisierung und Analyse von clientseitigen Honeypots.

Von Sebastian Wolfgarten, Dublin

Im Logfile-Rauschen des Alltagsbetriebs gehen Warnzeichen für Angriffe schnell einmal unter – ein Ansatz, hier Abhilfe zu schaffen, sind Intrusion-Detection-Systeme, ein anderer, mit weiter reichenden Zielen die so genannten Honeypots: speziell eingerichtete, bewusst angreifbare Systeme, auf denen es eben keinen Alltagsbetrieb gibt und die als leichter zu überwachende Falle für Angreifer dienen.

Der genaue Ursprung von Honeypots lässt sich heute nur noch schwer ermitteln. Grundsätzliche Überlegungen zu diesem Thema gab es bereits Mitte der 1980er-Jahre, eine erste öffentliche Erwähnung erfolgte durch die Buchautoren Clifford Stoll und Bill Cheswick in „The Cuckoo's Egg" und „An Evening with Berferd". Es dauerte jedoch bis 1999, dass der amerikanische IT-Sicherheitsexperte Lance Spitzner die Idee erneut aufgriff und einer breiteren Öffentlichkeit bekannt machte: Spitzner gründete damals das Honeynet-Projekt [1], das als die weltweit größte nichtkommerzielle Vereinigung zur Erforschung von Honeypots gilt. Einen Honeypot definiert Spitzner allgemein als „a resource whose value is being attacked or compromised" [2].

In der Praxis handelt es sich bei einem Honeypot in der Regel um ein speziell vorbereitetes und absichtlich unsicher konfiguriertes Computersystem, das mit einer oder mehreren Sicherheitslücken (z. B. veraltete Softwareversion) ausgestattet ist und gezielt in einem Computernetzwerk platziert wird, um potenzielle Angreifer in eine digitale Falle zu locken. Ein Honeypot steht damit eigentlich im krassen Gegensatz zu den klassischen Bestrebungen, die Sicherheit eigener IT-Systeme zu maximieren. Trotzdem gibt es zahlreiche Gründe, Honeypots nicht nur zu Forschungszwecken einzusetzen (vgl. [3]):

_____ *Ablenkung:* Ein vermeintlich leichtes Ziel kann einen (Gelegenheits-)Angreifer von wirklich wichtigen Systemen ablenken.

_____ *Lerneffekt:* Honeypots können helfen, neue Angriffsmuster und -strategien zu identifizieren und zu erforschen; selbst die Entdeckung nicht öffentlich bekannter Angriffswerkzeuge (z. B. Exploits) ist möglich.

_____ *Prävention:* Mithilfe eines Honeypots können Administratoren zusätzliches Wissen über den Ablauf eines Angriffs erlangen und damit die eigene Reaktionsfähigkeit sowie Sicherheitsvorkehrungen verbessern.

_____ *Beweiskraft und Identifikation:* Unter Umständen können die im Honeypot gewonnenen Daten als Beweismittel in einem Prozess dienen und die strafrechtliche Verfolgung eines Angreifers ermöglichen.

_____ *Typisierung:* Eine bessere Einschätzung von Angriffen ermöglicht die Unterscheidung zwischen gefährlichen und weniger gefährlichen Attacken.

_____ *Profiling:* Die Erstellung eines Täterprofils kann Hinweise über die Motivation eines Angreifers liefern.

Neben den genannten Gründen lassen sich die die speziellen Eigenschaften eines Honeypots auch nutzen, um eigene IT-Sicherheitsbestrebungen zu unterstützen: Beispielsweise sollte sämtlicher Netzwerkverkehr, der von einem Honeypot ausgeht oder an diesen gerichtet ist, als ungewöhnlich eingestuft und infolgedessen kritisch begutachtet werden.

Auf der anderen Seite sind Honeypots auch mit einer Reihe nicht zu unterschätzender Nachteile behaftet, die einen Einsatz in vielen Bereichen limitieren oder gänzlich verhindern:

_____ *Beschränkte Sicht:* Honeypots können nur solche Angriffe protokollieren, die gegen sie selbst gerichtet sind. Angriffe auf andere Systeme werden weder verhindert noch bemerkt.

_____ *Risiko:* Jeder Einsatz eines IT-Systems oder einer neuen Technik bedeutet immer auch ein zusätzliches Sicherheitsrisiko. Ein Honeypot ist sogar besonders risikoreich, da das System ja primär dazu gedacht ist, kompromittiert zu werden.

_____ *Rechtliche Konsequenzen:* Nach einem Einbruch in einen Honeypot wird wahrscheinlich versucht, diesen zum Angriff auf weitere Systeme zu verwenden – der Angriff auf Dritte kann hier schwerwiegende rechtliche Konsequenzen haben.

_____ *Aufwand:* Der Betrieb eines Honeypots bedeutet einen enormen Arbeitsaufwand, da das System bei der Installation und Konfiguration sorgfältig vorbereitet werden muss, um einen maximalen Nutzen erzielen zu können. Außerdem muss man es während des Betriebs ständig überwachen, um möglichen Schaden von anderen Systemen abzuwenden. Schließlich ist die Analyse nach der erfolgreichen Kompromittierung sehr zeitaufwändig und sollte auf keinen Fall unterschätzt werden.

Honeypot-Typen

Die Typisierung eines Honeypots erfolgt in der Regel gemäß dem Umfang der Interaktionsmöglichkeiten, die das System einem Angreifer zur Verfügung stellt. Die einfachste Form wird als Low-Interaction-Honeypot bezeichnet: Das System beschränkt sich dabei zumeist auf die softwarebasierte Emulation eines Betriebssystems oder Netzwerkdienstes und umfasst in der Regel nur einen Bruchteil der Funktionen, die durch ein reales Produkt zur Verfügung gestellt würden. Für einen Angreifer besteht dabei im Normalfall keine Möglichkeit, auf das dem Honeypot zugrunde liegende Betriebssystem zuzugreifen.

Durch eine besonders einfache Konfiguration und Inbetriebnahme ermöglichen Low-Interaction-Honeypots wie honeyd [4] oder Back Officer Friendly [5] es auch „Einsteigern", eigene Honeypots zu betreiben und erste Erfahrungen zu sammeln. Ein weiteres Einsatzgebiet dieser Kategorie ist das (automatisierte) Sammeln von Würmern, die in einem Netzwerk ihr Unwesen treiben. Nachteile von Low-Interaction-Honeypots sind unter anderem die per Definition geringen Interaktionsmöglichkeiten für einen Angreifer sowie die rudimentären Protokollierungsfunktionen, die sich in der Regel auf die Speicherung allgemeiner Verbindungsdaten beschränken (z. B. Quell- und Ziel-IP, Datum, Uhrzeit, etc.).

Die zweite Kategorie von Honeypots stellt eine Art Zwischenlösung dar: Realisierbar ist ein solcher Medium-Interaction-Honeypot beispielsweise mithilfe einer chroot-, Jail- (BSD-Unixe) oder User-Mode-Linux-Umgebung, in der einem Angreifer in einem abgeschotteten Umfeld eine partiell oder vollständig isolierte Kopie von Teilen des Betriebssystems zur Verfügung steht. Innerhalb dieser isolierten Umgebung kann sich ein Angreifer wie in einem echten System „frei bewegen".

Problematisch sind bei der Implementierung eines Medium-Interaction-Honeypots die Überwachung und Analyse sowie der allgemeine Aufwand für die Inbetriebnahme des Systems (z. B. Erstellung der chroot-Umgebung etc.). Zusätzlich müssen Sicherheitsvorkehrungen getroffen werden, damit ein Angreifer nicht aus der Sicherheits-Umgebung ausbrechen und auf das darunter liegende Betriebssystem zugreifen kann. Die Bedeutung von Medium-Interaction-Honeypots ist in der jüngsten Vergangenheit in Literatur und Praxis deutlich gesunken.

Bei einem High-Interaction-Honeypot handelt es sich um ein voll funktionsfähiges und in der Regel eigenständiges System, das einem Angreifer ohne Einschränkungen zur Verfügung steht. Dadurch erhält dieser Zugriff auf ein reales System, mit dem er nach Belieben interagieren kann. Der Betreiber eines solchen Honeypots kann dadurch eine Fülle an Informationen über Angreifer, Motivation, Verhalten, Wissensstand, Identität sowie geographische Position erhalten. Die so gewonnenen Daten können dazu dienen, ein besseres Verständnis über reale Angriffsmethoden zu entwickeln und eigene Systeme effektiver zu schützen.

Nachteile von High-Interaction-Honeypots sind allerdings die ressourcenintensive Installation und Überwachung eines solchen Systems sowie die zeitaufwändige Analyse eines kompromittierten Honeypots: Lance Spitzner vertritt beispielsweise die These, dass für jeweils 30 Minuten, die ein Angreifer auf einem kompromittierten System zugebracht hat, eine Analysezeit von etwa 40 Stunden notwendig ist [6]. Zudem gehen von High-Interaction-Honeypots große Sicherheitsrisiken für das umgebende Netz aus und es drohen rechtliche Konsequenzen, sofern ein solches System zum Angriff auf weitere Computer verwendet wird.

Clients im Visier

In den letzten Jahren zeigte sich neben den klassischen Angriffen auf Serversysteme ein deutlicher Trend hin zu professionellen, zielgerichteten Attacken auf Endbenutzer-PCs. Symantecs Internet Security Threat Report [7] berichtete beispielsweise für den Zeitraum von Januar bis Juni 2005, dass sich immer mehr Angreifer auf fokussierte Atta-

cken gegen Client-Systeme konzentrieren und großflächige Angriffe auf (Unternehmens-)Netzwerke zunehmend unterlassen. Diesen Trend bestätigt auch ein aktueller Bericht der University of Washington [8].

Die klassischen Angriffsmotive (Neugier, Langeweile, Aufzeigen technischer Virtuosität etc.) scheinen verstärkt in den Hintergrund zu treten und durch kriminelle Energie sowie die Aussicht auf schnelle finanzielle Bereicherung ersetzt zu werden. Daher sind beliebte Betätigungsfelder vor allen Dingen Betrug, Erpressung und der Missbrauch fremder Zugangskennungen (Identity Theft). Diese Entwicklung wird zusätzlich durch eine Vielzahl von Schwachstellen in populären Betriebssystemen und diversen Client-Programmen begünstigt (Web-Browser, E-Mail-Clients etc.).

Daher ist es auch nicht verwunderlich, dass laut Symantec eine große Menge an Spyware sowie acht der zehn bekanntesten Adware-Programme via Web-Browser installiert werden. Zusätzlich sorgen frei verfügbare Viren-, Exploit- und Phishing-Kits dafür, dass auch unerfahrene Angreifer Zugang zu recht professionellen Werkzeugen haben und somit sehr leicht eine Vielzahl möglicher Opfer ins Visier nehmen können.

Klassische Honeypots sind für die Analyse derartiger Angriffe eher ungeeignet, da sie sich primär auf Serversysteme konzentrieren. Um mit dem Honeypot-Ansatz die Untersuchung von Angriffen auf Clients zu ermöglichen, war daher die Schaffung einer neuen Art von Honeypots notwendig. Die Idee stammt wieder einmal von Lance Spitzner, der im Juni 2004 in der Honeypots-Mailingliste den Inhalt einer Securityfocus-Meldung kommentierte: "What would be interesting is using a 'client' honeypot. Take a clean install of a Win32 system, then have IE on it connect to hundreds of random websites. See if any of the websites makes 'unauthorized' modifications to your 'client' honeypot :)"

Während es sich bei einem klassischen Honeypot um ein relativ passives Gebilde handelt, das nur darauf wartet angegriffen zu werden, durchsucht ein „Client-Honeypot" das Internet aktiv nach bösartigen Webseiten und davon ausgehenden Angriffen. Das System überwacht und analysiert hierzu die Auswirkungen der besuchten Seiten auf ein Client-System. Neben zufällig ausgewählten Adressen eignen sich vor allem eher zwielichtige Angebote, die man mithilfe geeigneter Anfragen (z. B. crackz, serialz, warez, pr0n etc.) in diversen Suchmaschinen (z. B. Google, Astalavista) findet.

Führt der Besuch einer solchen Seite auf dem Client beispielsweise zur Erzeugung oder Ausführung einer (möglicherweise neuen) .exe-Datei oder zu einem neuen Eintrag im Autostart-Ordner, so ist dies ein starker Hinweis auf bösartige Absichten. Des Weiteren kann man durch die Benutzung verschiedener Betriebssystem- und Browser-Versionen mit jeweils unterschiedlichen Patch-Leveln feststellen, auf welche Systemkonfigurationen und Applikationen eine Internetseite abzielt.

Honeymonkeys

Auch der Softwareriese Microsoft hat das Potenzial clientseitiger Honeypots erkannt und Anfang 2005 das so genannte Honeymonkey-Projekt ins Leben gerufen [9]. Ein Honeymonkey ist laut Microsoft ein clientseitiger Honeypot, der in einer Virtualisierungsumgebung (Microsoft Virtual PC) die Tätigkeiten eines Benutzers beim Surfen im Internet mithilfe automatisierter Programme simuliert. Dabei besuchen zahlreiche Windows-XP-Clients, die mit verschiedenen Patch-Ständen ausgestattet sind, eine Liste von mehr als 5000 vorgegebenen Internetadressen und warten darauf, angegriffen zu werden.

Um eine höhere Effektivität zu erreichen, werden dabei vor allem solche Webseiten besucht, die in der Vergangenheit bereits negativ aufgefallen sind. Laut einem Microsoft-Bericht [10] wurden so innerhalb der ersten Monate nach Inbetriebnahme des Systems 752 Internetadressen entdeckt, die sich auf insgesamt 287 verschiedene Server zurückverfolgen ließen und die allesamt eine nicht aktualisierte Standardinstallation von Windows XP erfolgreich angreifen konnten. Durch die Verwendung unterschiedlicher Versionen und Patch-Level konnte Microsoft die betroffenen Systeme exakt identifizieren und feststellen, dass keiner der Angriffe im Untersuchungszeitraum Mai/Juni 2005 imstande war, ein zu dieser Zeit vollständig gepatchtes Windows-XP-SP2-System erfolgreich anzugreifen.

Im Juni/Juli 2005 hat das System jedoch erstmals einen Angriff gegen eine Schwachstelle entdeckt, für die es zu diesem Zeitpunkt noch keinen Patch und keinen öffentlich verfügbaren Exploit gab. Es ist sicherlich diskutabel, ob man dies – wie im Microsoft-Bericht – wirklich als Zero-Day-Exploit bezeichnen möchte, da der Sicherheitsdienstleister SEC-Consult bereits zuvor ein entsprechendes Advisory veröffentlicht hatte [11]. Auf jeden Fall belegt diese Analyse aber die Wirksamkeit und das Potenzial clientseitiger Honeypots. Als kleiner Wermutstropfen sei angemerkt, dass Microsoft die Honeymonkey-Erkenntnisse bisher nur sehr eingeschränkt veröffentlicht hat. Umso erfreulicher ist, dass es mittlerweile auch ein Open-Source-Projekt in dieser Richtung gibt.

Honeyclient

Beim Honeyclient-Projekt [12] handelt es sich um die erste Open-Source-Implementierung eines clientseitigen Honeypots. Die Software, die etwa zu derselben Zeit entstanden ist, wie das Honeymonkey-Projekt, wurde von Kathy Wang entwi-

ckelt und unter der BSD-Lizenz veröffentlicht. Anders als der Microsoft-Ansatz ist sie jedoch nicht auf mehrere Systeme verteilt, sondern besteht aus zwei Perlskripten, die unter Microsoft Windows 2000 oder XP automatisiert Webseiten ansteuern und die Folgen analysieren. Nach dem Besuch jeder einzelnen Seite werden das lokale Dateisystem sowie die Registry nach Änderungen durchsucht und ein Vorher-Nachher-Vergleich angestellt. Gibt es Unterschiede, so wird eine Warnmeldung erzeugt, der man dann manuell nachspüren muss.

Dabei zeigte sich erneut ein allgemeines Problem beim Betrieb von clientseitigen Honeypots: das Finden bösartiger Internetseiten. Momentan fehlt eine zentrale Anlaufstelle, wo solche Webadressen gemeldet und koordiniert analysiert werden können. Die Schnelllebigkeit derartiger Seiten erschwert eine erfolgreiche Suche zusätzlich. Einen sehr interessanten Lösungsansatz für dieses Problem fanden indes Aidan Lynch und Daragh Murray von der Dublin City University: Die beiden Studenten haben im Dezember 2005 eine Erweiterung für Honeyclient veröffentlicht, die Webadressen aus (Spam-)E-Mails extrahieren und mithilfe der Honeyclient-Software analysieren kann (Download über [12]). Obwohl noch keine konkreten Ergebnisse vorliegen, sind die beiden Studenten zuversichtlich, dass sich so eine klare Beziehung zwischen Spam-Mails und Angriffen auf Clients aufzeigen lässt.

Fazit

Honeypots sind und bleiben interessante Untersuchungsobjekte, die besonders im wissenschaftlichen Umfeld enormen Nutzen entfalten. Die durch die Veränderung der Bedrohungsszenarien im Internet hervorgerufene Entwicklung von Client-Honeypots steckt momentan noch in den Kinderschuhen und wird bisher nur durch Microsoft intensiver genutzt. Das von Kathy Wang ins Leben gerufene Projekt Honeyclient ist zwar ein guter Anfang, verfügt jedoch bislang nur über eingeschränkte Funktionen und Analysemöglichkeiten und benötigt noch wesentlich tiefgreifendere Werkzeuge, um die Auswirkungen von Internetseiten auf einen Client festzustellen. Solche Werkzeuge müssten beispielsweise auch den Inhalt des Arbeitsspeichers überwachen und den Zugriff auf zentrale Systemfunktionen kontrollieren. Eine vielversprechende Lösung für dieses Problem könnte im Rahmen einer Diplomarbeit an der Universität Mannheim entstehen [13].

So oder so darf man gespannt sein, wie sich Client-Honeypots in Zukunft entwickeln und ob beziehungsweise wann sie womöglich auch Einzug in das Risikomanagement von Unternehmen halten und dort die Gefährdungssituation von Mitarbeiter-PCs ergründen helfen. ■

Sebastian Wolfgarten (sebastian@wolfgarten.com) studiert „Security and Forensic Computing" an der Dublin City University und arbeitet bei Ernst & Young Irland im Bereich Technology & Security Risk Services.

Literatur

[1] The Honeynet Project, http://project.honeynet.org

[2] Lance Spitzner, The Value of Honeypots, Part One: Definitions and Values of Honeypots, www.securityfocus.com/infocus/1492

[3] Lukas Grunwald, Jochen Schlichting, Süße Falle, Honey-Techniken zur Einbruchsvorsorge, iX 6/2003, S. 102

[4] Niels Provos, Developments of the Honeyd Virtual Honeypot, www.honeyd.org

[5] (nfr)(security), Back Officer Friendly, www.nfr.com/resource/backOfficer.php

[6] Lance Spitzner, Honeypots, Tracking hackers, Addison-Wesley, ISBN 0-3211-0895-7

[7] Symantec Internet Security Threat Report, September 2005, http://ses.symantec.com/content.cfm?articleid=1539 (Registr. erf.)

[8] University of Washington, Alexander Moshchuk, Tanya Bragin, Steven D. Gribble, and Henry M. Levy, A Crawler-based Study of Spyware on the Web, www.cs.washington.edu/~gribble/papers/spycrawler.pdf

[9] Microsoft Research, Strider HoneyMonkey Exploit Detection, http://research.microsoft.com/HoneyMonkey/

[10] Yi-Min Wang, Doug Beck, Xuxian Jiang, Roussi Roussev, Chad Verbowski, Shuo Chen, and Sam King, Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities, in: Proc. Network and Distributed System Security (NDSS) Symposium, February 2006, http://research.microsoft.com/honeymonkey/NDSS_2006_HoneyMonkey_Wang_Y_camera-ready.pdf

[11] SEC-Consult Security Advisory <20050629-0>, IE6 javaprxy.dll COM instantiation heap corruption, www.sec-consult.com/184.html

[12] Kathy Wang, Honeyclient Development Project, www.honeyclient.org

[13] Bing Yuan, Thorsten Holz, Diploma Project: Client-Side Honeypots, Universität Mannheim, http://pi1.informatik.uni-mannheim.de/diplomas/show/27