Honeynets und Honeypots

Doris Reim
(doris@net.t-labs.tu-berlin.de)

Seminar „Internet Sicherheit“ ,
Technische Universität Berlin

WS 2006/2007 (Version vom 21. Januar 2007)

Zusammenfassung

Folgender Text faßt verschiedene Arbeiten zum Thema Honeynets und speziell zum Honeynet Project zusammen. Es werden zwei konkrete Projekte an Hochschulen vorgestellt und erklärt, was bei solchen Projekten besonders zu beachten ist. Die vorgstellten Projekte fanden jeweils an der RWTH Aachen und am Georgia Institute of Technology statt und bieten somit auch eine Vergleichsmöglichkeit zwischen USA und Deutschland. Desweiteren wird die rechtliche Situation in Bezug auf solche Experimente beschrieben.

1 Erkenne Deinen Feind - Mit Honig gegen Scriptkiddies

Jeder Rechner, der an das Internet angeschlossen ist, ist ständig der Gefahr von Angriffen ausgesetzt. Ein Angriff kann bereits sein, wenn der Rechner nach offenen Ports, die angreifbar sind, gescannt wird. Findet der Angreifer einen solchen, kann er verschiedene Tools installieren, beispielsweise solche, die benutzt werden, um von dem Rechner aus weitere Angriffe zu starten. Das Scannen der Ports passiert in der Regel mit eigens dafür geschriebener Software, Scripts, die nur gestartet werden müssen und als Ergebnis eine ganze Liste zurückliefern. Das Installieren von Anwendungen auf unsicheren Systemen kann dann zwar von Hand passieren, ist aber ebenso sehr einfach in der Handhabung. Daraus ergibt sich eine große Anzahl von Personen, die dank solcher Skripte sehr einfach fremde Rechner kompromittieren können. Viele Angriffe stammen von relativ unerfahrenen Personen, die sich ausschließlich auf solche automatisierten Werkzeuge verlassen. Diese nennt man auch „Script Kiddies“.

> „Das ’Script Kiddie’ ist jemand, der auf den schnellen, einfachen Erfolg aus ist. Sie sind nicht auf der Suche nach spezieller Information oder zielen auf eine bestimmte Firma. Alles was sie wollen, ist so einfach wie möglich root zu werden.“ [KE 1]

> „Die Methode des ’Script Kiddies’ ist einfach: durchsuche das Internet nach einer bestimmten Schwachstelle und wenn du sie gefunden hast, nutze sie aus. Die meisten Tools, die sie verwenden, sind automatisiert und erfordern wenig Interaktion.“ [KE 1]

Die traditionellen Schutzmechanismen gegen derartige Angriffe sind nach [KEH] allesamt defensive Maßnahmen, wie Firewalls, Intrusion Detection Systeme und Verschlüsselung. Dieses Prinzip birgt einige Probleme. Auf Angriffe kann nur reagiert werden, wenn der Angriff erkannt wird. Insbesondere kann zum Beispiel Anti-Viren Software nur auf bekannte Viren reagieren, neue Methoden bleiben lange unsichtbar. Eine Analyse dessen, was passiert ist, ist nur sehr eingeschränkt möglich und erst dann, wenn der Schaden schon entstanden ist. Solche Angriffe lassen sich in der Regel auch nicht zurückverfolgen. Die Frage ist nun, wie man dieses Verhältnis umdrehen kann, den Jäger also zum Gejagten macht und sich aktiv schützt. Die Idee ist, die Angreifer mit scheinbar unsicheren Systemen zu ködern, um dann ihre Aktivitäten analysieren zu können. Ein Ansatz dafür sind Honeynets.

Folgende Arbeit gibt eine Einführung in das Prinzip der Honeynets, was sie sind und wie sie funktionieren. Außerdem wird kurz auf rechtliche und moralische Fragen in diesem Zusammenhang eingegangen. Die letzten Kapitel geben einen Überblick, wie man ein solches Projekt an einer Hochschule durchführen kann und welche Erfahrungen dabei bereits gemacht wurden.

2 Grundlegendes zu Honeynets

Im folgenden wird die Begriffswelt der Honeynets näher erläutert. Es muß unterschieden werden zwischen Honeypots und Honeynets, woraus sie im einzelnen bestehen und wie sie funktionieren. Dabei wird kurz beschrieben, welche Werkezeuge benutzt werden und vorweg das Honeynet Project vorgestellt, von dem viele der aktuell benutzten Werkzeuge stammen.

2.1 Honeypot und Honeynet - Begriffe

Unter einem Honeypot versteht man zunächst das Prinzip des Köders, nämlich

> „...einen Gegenstand, von dem eine gewisse Attraktivität ausgeht, die bestimmte, nicht nur tierische, Interessenten anzulocken vermag. {...} Wissenschaftler haben neuerdings begonnen, das Prinzip der Köder auch im Bereich der IT-Sicherheit anzuwenden. Hier werden elektronische Köder ausgelegt, um das Verhalten von Angreifern leichter zu studieren. Elektronische Köder sind Netzwerkressourcen (Computer, Router, Switches), deren Wert darin besteht, angegriffen und kompromittiert zu werden. Die Köder haben keine spezielle Aufgabe im Netzwerk, sind aber ansonsten nicht von regulären Komponenten zu unterscheiden. Sie sind mit spezieller Software ausgestattet, die die anschließende Forensik eines Angriffs deutlich erleichtert. Im Gegensatz zu einer herkömmlichen forensischen Untersuchung erlauben beispielsweise gezielte Veränderungen im Betriebssystem das direkte Mitschneiden aller Aktivitäten eines Angreifers.“ [Dorn 04]

.

Das heißt, es wird ganz real ein Rechner mit bestimmten Diensten im Netz angeboten, der dabei allerdings überwacht wird. Zusätzlich benötigt man eine spezielle Software. Diese kann unterschiedliche Aufgaben erfüllen. Die wichtigsten sind, sich vor dem Angreifer zu verstecken und Daten zu sammeln. Auf diese Weise kann man die Aktionen der Angreifer weiter verfolgen und später analysieren. Näheres zu den Tools wird in Kapitel 2.3 beschrieben.

Außer die als Köder dienenden Dienste anzubieten und Daten zu sammeln, erfüllt ein Honeypot keine weiteren Aufgaben. Ein Honeypot darf nicht auf einem Produktivsystem eingerichtet werden, da er dazu da ist, kompromittiert zu werden und dadurch

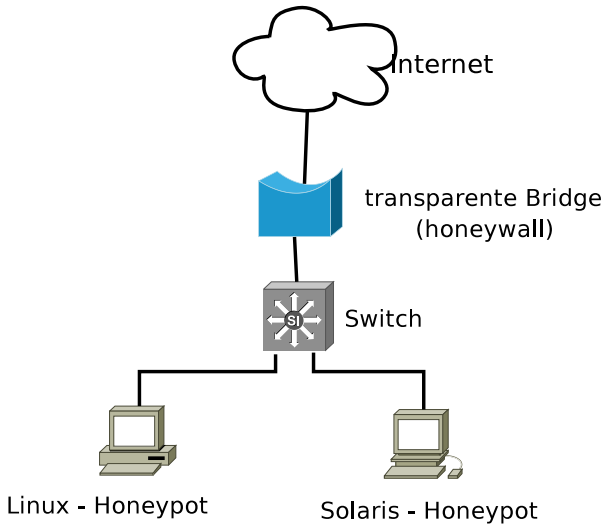

Abbildung 1: Beispielaufbau eines Honeynets [Dorn 04]

das Produktivsystem gefährden würde. Ein Honeypot ist also eine einzelne Maschine, die mit besonderer Software ausgestattet ist. Zusätzlich benötigt man Geräte, die der Sicherheit dienen, wie etwa eine transparente Bridge. Die transparente Bridge, in diesem Zusammenhang als Honeywall bezeichnet, wird als eine Art umgedrehter und unsichtbarer Firewall eingesetzt. Umgedreht, weil es nicht ihre Aufgabe ist, Angreifer auszusperren, sondern sie im Gegenteil durch zu lassen, danach aber nicht mehr aus dem Netz herauszulassen. Die Honeywall soll den Rest des Netzes vor dem dann kompromittierten System schützen, dabei aber unsichtbar bleiben.

Im Gegensatz dazu bezeichnet der Begriff Honeynet eine Topologie, wie sie zum Beispiel in Abbildung 1 zu sehen ist. Mehrere Honeypot-Rechner werden hinter einer Honeywall zu einem Netz zusammengeschlossen. Das hat den Vorteil, daß verschiedene Betriebssysteme verwendet werden können oder auch auf unterschiedlichen Systemen einzelne, unterschiedliche Dienste angeboten werden können.

Die gezeigte Art des Aufbaus ist nicht die einzige Möglichkeit. Jede Topologie, die andere Teile des Internets nicht gefährdet, ist möglich. Eine weitere Variante bilden virtuelle Honeynets. Dabei werden mehrere virtuelle Maschinen auf einem Rechner installiert.

2.2 Das Honeynet Projekt

Bevor die für Honeynets verwendeten Werkzeuge näher beschrieben werden, soll kurz die Quelle vieler dieser Programme vorgestellt werden.
Unter `www.honeynetproject.org` stößt man auf eine ganze Sammlung an Literatur und Software zum Thema Honeynets. Unter [Pro1] wird das Projekt beschrieben als ein gemeinnütziger, internationaler Zusammenschluß verschiedener (Forschungs-) Gruppen und Einzelpersonen.

Die Ziele des Projekts sind klar umrissen. Einerseits gilt es, ein Bewußtsein für die Problematik zu schaffen, also sowohl dem einzelnen Nutzer wie ganzen Organisationen zu vermitteln, daß sie Ziel von Attacken sind, warum, wie die Attacken funktionieren und wie man sich dagegen wehren kann.

Das zweite Ziel gilt der Information. Hinter diesem sehr allgemeinen Begriff steckt die Idee, jenen, die sich bereits der Gefahren bewußt sind, bessere Schutzmaßnahmen an

die Hand zu geben und zum anderen weiter zu forschen an den Motiven der auch Black-Hats genannten Angreifer, an deren Kommunikationswegen und Vorgehensweisen.

Das dritte Ziel ist, jedem, ob er sich nur schützen oder gar selbst an dem Thema forschen möchte, alle nötigen Werkzeuge zur Verfügung zu stellen. Zu allen diesen selbst gesteckten Zielen existiert eine ganze Reihe von Whitepapers und Software, auf die später noch weiter eingegangen wird.

2.3 Die Werkzeuge

Wie in Kapitel 2.1 bereits erwähnt, braucht man für die Realisierung eines Honeypots zusätzliche Software. Die Funktionsweise solcher Tools wird hier kurz anhand von Sebek[1] [KES] erläutert, welches auch im Projekt an der RWTH Aachen im Einsatz ist.

Die Aufgabe von Sebek ist es, Daten über den Angreifer zu sammeln und dabei unerkannt zu bleiben. Das heißt, das System muß sich aus Sicht des Angreifers unauffällig verhalten.

Die erste Idee, die Aktionen der Angreifer zu verfolgen, war, die dazugehörigen Netzwerkdaten zu sammeln. Diese Methode funktionierte nur solange, bis die Black-Hats anfingen, ihre eigenen Kommunikationskanäle zu verschlüsseln, wobei sie auf kompromittierten Hosts SSH-Clients und ähnliches installierten, falls diese nicht vorhanden waren. Um also weiter Daten sammeln zu können, mußte die Verschlüsselung umgangen werden. Aus dieser Situation heraus begann das Honeynet Project mit Kernelbasierten Rootkits zu experimentieren, was zur Entwicklung von Sebek führte.

Sebek wird vollständig im Kernel ausgeführt und ist in der Lage alle oder nur einen Teil der Daten aller Benutzer auf dem System zu sammeln. Bei verschlüsselten Daten werden unter anderem durch Mitschneiden der Shelleingaben Schlüssel und Passwörter aufgezeichnet. Dadurch können die Daten später wieder entschlüsselt werden.

Sebek selbst besteht aus zwei Komponenten, einem Client und einem Server. Der Client wird auf dem Honeypot installiert, der Server oft auf der zum Honeynet gehörenden Firewall.

Der Client sammelt die Daten mittels read() Systemaufruf auf einem Honeypot und schickt sie über das Netz an den Server. Da Sebek ein Kernelmodul ist, hat es Zugriff auf den Kernel-Space des jeweiligen Honeypots. Dadurch können alle Aktivitäten und Daten des read() Aufrufs abgefangen werde. Dies wird realisiert, indem die read() Funktion in der Systemaufrufstabelle durch eine neue ersetzt wird. Die neue Funktion ruft die Originale auf, kopiert deren Inhalt in einen Paketpuffer, hängt einen Header an und verschickt das Paket an den Server.

Die read() Funktion ist eine der Schlüsselfunktionen von Sebek und dient hier als Beispiel, welche Tricks angewendet werden, um das Ziel, alles mitverfolgen und doch unerkannt zu bleiben, zu erreichen.

Die Kommunikation wurde, laut der Autoren, auf eine Weise realisiert, die für einen Angreifer schwer zu entdecken ist. Dabei kann der Server Pakete direkt vom Netz sammeln oder er bekommt ein Archiv mit Paketdaten. Die Daten erreichen den Server in einem plattformunabhängigen Format, so daß das Betriebssystem der einzelnen Honeypots keine Rolle spielt. Danach können die Daten direkt entpackt und analysiert oder an eine Datenbank weitergeleitet werden.

Neben diesem und ähnlichen Tools zur Datensammlung benötigt man noch geeignete Software, um die Daten später zu analysieren. Neben Programmen wie tcpdump, gibt es zum Beispiel Honeysnap [KEA], ein Kommandozeilenprogramm, das einzelne oder mehrere pcap-Dateien parsen und zusätzlich eine erste Analysezusammenfassung erzeugen kann.

[1] http://www.honeynet.org/tools/sebek/

3 Darf man das?

Wer ein Honeynet installiert, bietet bewußt einen Angriffspunkt für Black-Hats und sammelt eine große Menge teils kritischer Daten. Dabei treten rechtliche und moralische Fragen auf, die in diesem Kapitel betrachtet werden sollen.

> „Die Verantwortlichkeit des honeynet-Betreibers ist in zwei Bereichen besonders diskussionswürdig:
>
> - Wie ist das honeynet in Bezug auf das gesamte Internet und wie sind insbesondere Angriffe von honeypots aus auf andere Systeme zu beurteilen? Hierbei sind ethische, straf- und zivilrechtliche Aspekte von Interesse.
> - Wie ist es zu bewerten, daß die Black-hats ohne ihr Wissen zum Teil eines Experimentes gemacht werden? Hierbei kommen insbesondere datenschutzrechtliche Aspekte zum Tragen. “ [Dorn 04]

3.1 Rechtliches

In diesem Abschnitt wird auf gesetzliche Fragen eingegangen. Dabei werden zum Vergleich die entsprechenden Gesetze in Deutschland und USA zusammengefaßt.

3.1.1 Deutsches Recht

Aus Sicht von [Dorn 04] müssen drei Gesetze dabei betrachtet werden. Das eine ist das Gesetz über Beihilfe zu einer Straftat in dem Fall, daß für einen Angriff auf andere Systeme das Honeynet benutzt wird. Dieses Gesetz kommt dann nicht zum tragen, wenn man das Honeynet durch geeignete Schutzmaßnahmen absichert. Ein Honeynet ist in der Regel sogar wesentlich besser abgesichert als viele andere mit dem Internet verbundene Systeme, wodurch dieser Paragraph entfällt.

Das zweite betroffene Gesetz bezieht sich auf die Frage nach einer eventuellen Haftung, wenn anderen vom Honeynet aus Schaden entsteht. Diese Frage ist nicht fundamental geklärt, der Trend gehe jedoch nach [Dorn 04] dahin, daß deutsche Gerichte davon absehen, Schäden, die durch ein nicht abgesichertes System verursacht werden, zivilrechtlich zu verfolgen.

Mit dem kürzlich gefallenen Urteil des Landgerichts Hamburg[2] bzgl. unverschlüsselter Wireless Netzwerke scheint es jedoch, daß diese beiden Fragen nicht mehr so eindeutig verneint werden können. Letzten Endes wird hier nur bleiben, sich möglichst weit abzusichern und diese Dinge weiter zu verfolgen.

Beim dritten Gesetz handelt es sich um die Frage des Datenschutzes, wenn die Aktivitäten eines Black-Hats ohne sein Wissen aufgezeichnet werden.

[Dorn 04] kommt schließlich zu dem Ergebnis, daß das Datenschutzrecht nicht verletzt wird. Aus meiner Sicht stellt sich die Frage aber eigentlich überhaupt nicht. Es wäre dieselbe Frage, ob ein Einbrecher den Besitzer eines Hauses verklagen kann, weil dessen Überwachungskamera ihn beim Einbrechen in das Haus gefilmt hat.

Ein anderer Punkt, auf den [Dorn 04] leider nicht eingeht, wäre an dieser Stelle die Frage, ob ein Honeynet mit dem Provozieren einer Straftat zu vergleichen ist. Ein Beispiel dafür wäre ein Polizist, der an der Straße Dorgen anbietet. Aus meiner Sicht ist das jedoch nicht unbedingt vergleichbar, da jemand, der nicht sowieso nach verletzbaren Systemen sucht, auch keine findet. Er kann nicht zufällig darauf stoßen und dadurch zum Einbruch in ein System animiert werden.

[2] Aktenzeichen 308 O 407 / 06

3.1.2 Amerikanisches Recht

In dem Whitepaper [HU] des Honeynet Projects über Honeynets an Universitäten taucht wie bei dem deutschen Projekt die Frage nach rechtlichen Problemen auf.

In USA existiert ein allgemein gefaßtes Gesetz zu Daten, die über Daten - und Telefonleitungen gehen, der sogenannte Wiretap Act [US]. Dieses Gesetz bietet einen gewissen Rahmen, in den das Honeynet gefaßt werden kann, ist aber - ähnlich der deutschen Rechtslage - bei weitem nicht präzise genug, beim Aufbau eines solchen Projekts alle Aspekte eindeutig handhaben zu können.

Demnach dürfen Daten, die über das Netz gehen, egal ob ein Telefonat oder Daten, die über das Internet gehen, also alle Daten, die nicht gespeichert werden, niemals ohne Wissen des Users abgehört werden. Allerdings gibt es verschiedene Ausnahmen und Unklarheiten, so herrscht etwa Uneinigkeit über Emails, da diese eben gespeichert werden. Zudem gibt es eine eigene Klausel für Provider, die Daten „for mechanical or service quality“ überwachen und nutzen dürfen.

Das Georgia Institute of Technology (Georgia Tech) hat nach [HU] im Fall des Honeynets ihre Rechtsabteilung befragt und einen Policy Katalog erstellen lassen, um sich rechtlich abzusichern. So wird beispielsweise darauf geachtet, daß sich alle Teile des Honeynets im Adressraum der Universität und damit im Einflußbereich eines Providers befinden, damit die Klausel für Provider Geltung finden kann. Außerdem werden aus datenschutzrechtlichen Gründen keine IRC-Server im Universitätsnetz mehr zugelassen, da diese wie ein Telefonat nicht heimlich abgehört werden dürfen. Auch wenn kein präzises Gesetz besteht und der Wiretap Act in mancher Hinsicht sehr kompliziert wirkt, scheint die rechtliche Lage im Bezug auf ein Honeynet - zumindest im Hochschulbereich - eindeutiger zu sein, als das in Deutschland der Fall ist.

3.2 Ethisches

Die Idee des Köders, wie sie beispielsweise auch von der Polizei eingesetzt wird, wirft seit jeher moralische Fragen auf. Wie weit darf man die Gelegenheit zu einer Straftat geben, durch die Taten provoziert werden, die ohne diese Gelegenheit nie stattgefunden hätten. Diese ganz spezielle Fragestellung kann im engen Rahmen dieser Arbeit mit Sicherheit nicht im nötigen Umfang diskutiert werden.

Eine andere Frage, die [Dorn 04] stellt, ist die der Verantwortlichkeit gegenüber anderen, die vom eigenen Honeynet aus angegriffen wurden. Entgegen dessen Antwort, daß ein Honeynet durch das Überwachen allen Verkehrs, der darüber geht sogar mehr Sicherheit für andere biete, sollte vielleicht eher diese Verantwortung ernst genommen werden. Produktivsysteme, auf denen womöglich sensible Daten liegen, dürfen auf keinen Fall durch noch so positiv motivierte Forschungsprojekte der Gefahr einer Kompromittierung ausgesetzt werden. Insofern sollte es sich der Betreiber eines solchen Projekts zur Pflicht machen, sein Honeynet immer weiter zu entwickeln und zu jeder Zeit die Kontrolle darüber zu behalten.

Die Frage nach den moralischen bzw. ethischen Aspekten beim Betrieb von Honeynets erörtert [Dorn 04] teilweise unter Zuhilfenahme zweifelhafter statistischer Überlegungen, daß ein zusätzliches unsicheres System, den Anteil der unsicheren Systeme insgesamt erhöht und somit die Chance für den Einzelnen, Opfer eines Angriffs zu werden, vermindert.

4 Honeynets an Hochschulen

Hochschulen sind aufgrund der Gestalt und Größe ihrer Netze ein gut geeigneter Ort für den Aufbau eines Honeynets. [HU] beschreibt neben der Möglichkeit zur Forschung, die sich unter anderem aus den Daten, die durch ein solches System gesammelt werden können ergibt, zwei der wichtigsten Gründe, die für diese Ortswahl sprechen. Zum einen bietet ein Honeynet zahlreiche Möglichkeiten zur Lehre, wie etwa Internet-Sicherheits-Veranstaltungen, in denen anhand der aus den Honeynets gewonnenen echten Daten das Vorgehen von Angreifern vorgeführt werden kann. Zum anderen kann das Netz als zusätzliches Sicherheitswerkzeug dienen. Im Folgenden wird kurz umrissen, was nach [HU] die wichtigsten Punkte beim Aufbau eines Honeynets an einer Hochschule sind.

4.1 Vorarbeiten

Bevor man mit der Installation beginnt, muß das Projekt vorbereitet werden. Der erste Schritt ist, sich eine Genehmigung von Seiten der Hochschule zu holen. Wie in Abschnitt 3 dargelegt, gibt es einige rechtliche und moralische Bedenken, die im jeweiligen Umfeld geregelt werden müssen, zumal das Netz der Hochschule gehört.

Das Honeynet darf keine zusätzliche Gefahr darstellen, von dem aus Attacken auf die Universität gestartet werden können. Insofern müssen verschiedene Stellen befragt und vor allem überzeugt werden. Den Administratoren und der Hochschulleitung muß der Nutzen klar gemacht werden, wie eben zusätzliche Sicherheit, leichteres Entdecken von Angriffen, Entdecken von Sicherheitslücken im System und nicht zuletzt auch die immensen Möglichkeiten in Forschung und Lehre. Zudem sollten gegebenenfalls die Administratoren des Universitätsnetzes in den Aufbau mit einbezogen werden. Man braucht unter anderem eigene IP Adressen aus dem Hochschulnetz für das Projekt.

4.2 Benötigte Hardware

Je nachdem, mit welchen Betriebssystemen man arbeiten möchte, wie groß das Netz sein soll und welche Topologien man testen will, gibt es sehr unterschiedliche Anforderungen an die Hardware. Insgesamt ergibt sich der Eindruck, daß für ein initiales Projekt kaum spezielle Hardware erforderlich ist.

Der Aufbau an der RWTH Aachen umfaßte um 2004 laut [Dorn 04] lediglich zwei Workstations, einen Switch und eine transparente Bridge. Für das Projekt am Georgia Institute of Technology (GIT) gibt [HU] eine detaillierte Auflistung der Geräte, aus der ersichtlich ist, daß zum Beispiel die eingesetzten Rechner Maschinen sind, wie sie an einem Lehrstuhl als Workstations eingesetzt werden. Besonders ist dabei nur, daß für jeden Rechner je zwei zusätzliche Festplatten zum Austausch vorhanden sind. Diese geben die Möglichkeit, den Zustand des jeweiligen PCs im Fall eines Angriffs in diesem Augenblick „einzufrieren“ und die Daten offline zu analysieren.

4.3 Erster Aufbau eines Honeynets

[HU] rät dazu, bei einem Erstversuch mit Honeynets, klein anzufangen. In [HU] wird dieses Prinzip GEN 1 genannt, die erweiterte Variante dann GEN II, wobei leider nicht explizit definiert wird, welchen Umfang ein Honeynet hat, das zum Typ GEN I gehört.

Der Idee nach ist es wohl sinnvoll, für den Anfang kleine Topologien mit vielleicht nur zwei oder drei Rechnern, wie es das Projekt an der RWTH Aachen getan hat, zu verwenden und die Software auf vertraute Dinge, wie dem von den Projektmitgliedern täglich benutzten Betriebssystem und einer ersten Honeynet Software einzuschränken. Erst wenn genug Erfahrung da ist, können je nach Ziel, das verfolgt wird, Software und

Hardware erweitert werden. Immer unter der Prämisse, daß andere Systeme geschützt werden müssen. Dabei sollte auch bedacht werden, daß auf eine Stunde aufgezeichneter Daten im Schnitt 40 Stunden Analyse kommen, die viel persönliche Interaktion benötigt. Daher ist ein Honeynet ein sowohl zeit- als auch personalintensives Projekt.

5 Die Realität - Zwei Projekte an Universitäten

Als konkretes Beispiele werden im Folgenden zwei Universitätsprojekte, der RWTH Aachen und des Georgia Institute of Technology (GIT) vorgestellt. Zu beiden Projekten wird außerdem eine kurze Zusammenfassung des Aufbaus und ihrer bisherigen Ergebnisse gegeben.

5.1 RWTH Aachen

In [Dorn 04] wird das Projekt an der RWTH Aachen beschrieben. Der Aufbau wurde in Abbildung 1 gezeigt. Einen Eindruck der konkreten Umsetzung gibt folgender Auschnitt aus dieser Arbeit. Einer der Honeypots ist ein Rechner mit Linux.

> „Auf dem Linux-*honeypot* läuft eine SUSE Linux 8.0 Professional Installation, die die Dienste HTTP (Apache 1.3.23 inklusive PHP 4.1.0), FTP (vs. FTPd 1.0.1) und SSH (open-SSH 3.0.2p1) anbietet. Außerdem wurde PHP-Nuke in Version 5.0 sowie MySQL 3.23.53 installiert, um beobachten zu können, ob Angreifer auch spezielle Webapplikationen angreifen. Gegenüber einer Standardinstallation wurden nur wenige Änderungen durchgeführt: Die wichtigste Änderung war die Installation des Sebek-Client. Zudem wurden einige sogenannte *honeytokens* auf dem System hinterlegt. Dies sind verschiedene Arten von Daten (beispielsweise Mails, Tabellenkalkulations- oder verschlüsselte Daten), die das Interesse des Angreifers wecken und die den Ablauf der Informationsgewinnung nach einem erfolgreichen Angriff nachvollziehbar machen soll. Desweiteren wurde das System so eingerichtet, daß es einem 'normalen' System mit drei Benutzern gleicht.“ [Dorn 04]

Die hier erwähnten speziellen Tools, wie zum Beispiel Sebek wurden bereits in Kapitel 2.3 näher erklärt. Neben einigen Anlaufproblemen beim Aufsetzen des Systems und der Konfiguration aller notwendigen Werkzeuge, traten laut dem Autor vor allem Probleme bei der Durchführung des Projekts seitens des Rechenzentrums der RWTH Aachen auf. Dort gab es tatsächlich die in Kapitel 4 beschriebenen Vorbehalte, was die Gefährdung des übrigen Netzwerks der Universität betrifft. Im Fall der RWTH einigte man sich auf ein eigenes /26-Subnetz, das vom restlichen Netz der RWTH physisch getrennt ist. Soweit das aus der Arbeit hervorgeht, besteht über das Rechenzentrum lediglich die Anbindung ans Internet.

Das Honeynet in Aachen wurde 2003 aufgesetzt, [Dorn 04] stammt von 2004. Im Zeitraum dazwischen gab es zwar keine Kompromittierung des Systems, wohl aber massenhaft Port-Scans, die ersten bereits nach 10 Minuten. Weiterhin konnte beobachtet werden, daß Dienste wie SSH oder PHP kaum beachtet wurden und daß lediglich nach FTP-Servern, die oft für illegales File Sharing benutzt werden, gesucht wurde.

Unter dem Codenamen NoSEBreaK [Nose 04] gelang es der Projektgruppe jedoch, das eigene System anzugreifen. Man fand ein Verfahren, Sebek zu entdecken und zu umgehen, wobei die Logs, die der Analyse von Angriffen dienen sollten unbrauchbar gemacht werden konnten. Da Sebek ein Kernelmodul ist, hinterläßt es Spuren seiner Existenz im Speicher, die zur Entdeckung führen können. Eine der Methoden, Sebek außer Kraft zu setzen ist ein schlichter Reboot, da Sebek keine Option bietet, automatisch beim booten gestartet zu werden. Eine Integration der Installation in den Bootprozess

würde wiederum zu viele Spuren hinterlassen. Dieses Beispiel zeigt, welche Probleme die Implementierung eines Tools wie Sebek aufwirft. Auf der anderen Seite ist ein Honeynet kein kurzfristiges Projekt. Selbst nach einer Laufzeit von über zwei Jahren gab es außer Portscans und Scans nach einfachen Schwachstellen keinen ernsthaften Angriff von außen. Aber auch aus diesen Mengen an Scans konnten Ergebnisse gezogen werden.

Diese Art Scans ist genau die eingangs erwähnte automatisierte Suche nach verbreiteten Schwachstellen. Die Analyse dieser Daten kann immer wieder neue Erkenntnisse über die alltäglichen Sicherheitslücken bringen. Zu Sebek sei noch gesagt, die betreffende Arbeit stammt von 2004. Inzwischen ist Sebek in der dritten Generation und viele der Probleme konnten behoben werden.

5.2 USA

Ein Projekt, das nach eigener Aussage schon mehr Ergebnisse geliefert hat, findet am Georgia Institute of Technology(GIT) [HU] statt. Der Aufbau dieses Projekts ist dem an der RWTH Aachen ähnlich, hat aber im ganzen acht Honeypots. Ein Ergebnis des Projekts ergab sich in einer Dissertation über neue Methoden, Rootkits zu entdecken. Desweiteren konnten neue Arten von Rootkits entdeckt und analysiert werden.

In der Lehre wurde das Honeynet eingesetzt, um Praktika für Studenten anzubieten. Dabei konnten sie anhand der Beobachtung und Analyse der Angriffe selbst und der entdeckten Rootkits Dokumentationen und Analysen schreiben, wie ein Angriff abläuft und wann welche Tools eingesetzt werden.

Für die Autoren selbst eine Überraschung, erwies sich das Experiment als reales Sicherheitswerkzeug. Seit dem Beginn wurden im internen Netz der GIT zahlreiche kompromittierte Systeme entdeckt. Eine dieser Entdeckungen machte schließlich auch das zuvor skeptische Rechenzentrum zu Befürwortern des Projekts.

Im Honeynet wurde ein kompromittiertes System entdeckt. Um genauere Daten zu bekommen, ließ man das System online und konnte den Angriff auf einen weiteren Rechner im internen Netz der Universität zurückverfolgen. Zu diesem Zeitpunkt wurde das zuständige Büro informiert. Dort versuchte man zunächst mit den vorhandenen Sicherheitsprogrammen, den Fehler zu finden, was nicht gelang. Die Maschine zeigte keinerlei Hinweise auf einen Angriff, trotzdem wurden von ihr aus andere Systeme im Netz der GIT angegriffen. Schließlich konnte der Fehler entdeckt werden. Der Angreifer hatte sich irgendwie, vermutlich durch einen Brute Force Angriff oder über eine Dummy Webseite, das Passwort beschafft. Der User, dessen Passwort gestohlen worden war, wurde sofort informiert und veranlaßt, ein neues, diesmal sicheres Passwort zu wählen und dieses nicht für Accounts auf Webseiten zu benutzen. Da dieser Angriff ohne das Honeynet wahrscheinlich nicht entdeckt worden wäre, bekam das Projekt ab diesem Zeitpunkt von Seiten der GIT bessere Unterstützung und wurde weiterentwickelt.

6 Zusammenfassung

Gesamt betrachtet sind Honeynets ein mächtiges Werkzeug in der Erforschung von Angriffen auf Rechner im Internet und bieten neue Ansätze, wie man Systeme schützen kann.

Die technische Umsetzung ist vergleichsweise einfach, da eine Menge an Werkzeugen bereits vorhanden ist und der Hardware Aufbau aus gängigen Komponenten zusammengesetzt werden kann. Aus rechtlicher Sicht gibt es wenig Einschränkungen. Die Verpflichtung, das Netz vor dem Honeynet zu schützen ist eine eher moralische. Kommt es jedoch zu Angriffen, kann sehr wohl der Betreiber des Honeynets verantwortlich gemacht werden. In der Praxis ist es daher sinnvoll, Schutzmechanismen zu verwenden, gerade was die Umsetzung an Hochschulen betrifft.

Auch wenn ein Honeynet nicht direkt spektakuläre neue Angriffe aufzeichnet, liefert es doch genug Material, um beispielweise die Entwicklung von Schutzsoftware voranzubringen.

Literatur

[Dorn 04] Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz: *Ermittlung von Verwundbarkeiten mit elektronischen Ködern. In: Proceedings of the Detection of Intrusions and Malware and Vulnerability Assessment. S. 129-141, (Lecture Notes in Informatics)*; IMVA 2004, Tagungsort: Dortmund, 07.07.2004.

[Pro1] `http://www.honeynet.org/misc/project.html`.

[US] `http://www.law.cornell.edu/uscode/18/usc_sec_18_00002511----000-.html`.

[HU] Honeynet Project: *Know Your Enemy: Honeynets in Universities*; 2004.

[KE 1] Jochen Berner: *Den Feind erkennen 1*.

[KEH] Honeynet Project: *Know Your Enemy: Honeynets*; 2006.

[KES] Honeynet Project: *Know Your Enemy: Sebek*; 2003

[Nose 04] Maximillian Dornseif, Thorsten Holz, Christian Klein: *NoSEBrEaK - Attacking Honeynets. In: the proccedings of the 5th Annual IEEE Information Assurance Workshop*; Tagungsort: West Point, 7. - 9. Juni 2004

[KEA] `http://www.honeynet.org/tools/honeysnap/index.html`