DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Analyse und Implementierung eines Honeypot-Systems

Studienarbeit

für die Prüfung zum

Bachelor of Engineering

des Studienganges Informationstechnik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Steffen Kurstak und Julian Kühn

Abgabedatum 1. April 2013

Bearbeitungszeitraum	12 Wochen
Matrikelnummer	
Kurs	TINF11B3
Gutachter der Studienakademie	Dr. Ralf Brune

Erklärung

Gemäß §16 (3) der „Studien- und Prüfungsordnung für den Studienbereich Technik“ vom 1.11.2007.

Ich habe die vorliegende Studienarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ort Datum Unterschrift

Zusammenfassung

Hier steht eine Kurzzusammenfassung.

Sperrvermerk

Die vorliegende Arbeit beinhaltet interne vertrauliche Informationen der z.B. Firma EDEKA Handelsgesellschaft Südwest mbH.

Die Weitergabe des Inhaltes der Arbeit und eventuell beiliegender Zeichnungen und Daten im Gesamten oder in Teilen ist grundsätzlich untersagt.

Es dürfen keinerlei Kopien oder Abschriften – auch in digitaler Form - gefertigt werden. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma EDEKA Handelsgesellschaft Südwest mbH in Abstimmung mit dem/der Verfasser/in.

Die vorliegende Arbeit ist nur den Korrektoren sowie ggf. den Mitgliedern des Prüfungsausschusses zugänglich zu machen.

(Stempel)

(Ort, Datum)

(Unterschrift des Betreuers bzw. Ausbildungsleiters)

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abk.	Vollständige Abkuerzung

Kapitel 1

Einleitung

1.1 Einführung

Kommt noch

Um die technischen Gegebenheiten zu verstehen, werden in dieser Arbeit auch die verschiedenen Anwendungs-Möglichkeiten und Implementierungsarten von Honeypots und Honeynets erläutert. Das Thema Hacking wird ebenfalls beschrieben und erklärt.

1.2 Ziele der Arbeit

Im Rahmen dieser Arbeit soll ein Honeypot implementiert werden und über das Internet als potentielles Angriffsziel für Hacker dienen. Anschließend sollen die Angriffe ausgewertet und diverse Statistiken erstellt werden. Zudem sollen ein oder zwei Angriffe, mit Hilfe von Tools, detailliert aufgearbeitet werden. Für die Implementierung des Honeypots ist auch ein Softwarevergleich nötig, um eine passende Software zu finden. Möglich ist auch der Einsatz eines kommerziellen Honeypots. Für die Analyse der Angriffe ist ebenfalls der Einsatz mehrere Tools möglich und erwünscht.

1.3 Stand der Technik

Zwischen 2000 und 2006 war der Honeypot am populärsten. Neben Firmen haben auch Privatnutzer mit Hilfe der Honeypots angefangen Hacker zu studieren. Da ein Honeypot allerdings gerade in der Industrie nur Geld kostet und oftmals keinen messbaren Mehrwert bringt, ist der Einsatz von Honeypots zurückgegangen. Zudem sind auch die Communitys zurückgegangen und der damit einhergehend Support von OpenSource Projekten. Die Honeyport-Software Honeyd ist hierfür ein passendes Beispiel, da das letzte Release der Software bereits aus dem Jahre 2007 ist. Somit ist bei dem Einsatz von OpenSource Produkten in dieser Studienarbeit, mit Problemen bezüglich Dokumentation und Support zu rechnen. Hilfreich wäre hier eine kommerzielle Lösung, um diesen Problemen aus dem Weg zu gehen.

Kapitel 2

Honeypots

2.1 Definition

Einleitend und noch oberflächlich erklärt ist die Aufgabe eines Honeypots, einen Angreifer von einem bedeutsamen Ziel abzulenken. In der Informationstechnik erfüllen Honeypots, je nach Einsatzgebiet und Ansicht des Entwicklers, verschiedene Aufgaben.

Einige Unternehmen sehen den Einsatz eines Honeypots als ein Intrusion Detection System. Für andere ist es nur eine Täuschung für Hacker, welche dadurch von den produktiven Systemen abgelenkt werden sollen. Ein weiterer Nutzen eines Honeypots ist es, die angreifenden Hacker analysieren zu können und somit neue Vorgehensweisen und Trends der Hacker zu erkennen. Diese Informationen sind vor allem für Sicherheitsfirmen relevant, da sie dabei neue Viren, Trojaner oder weitere Schadsoftware erkennen können.

Trotz dieser verschiedenen Einsatzmöglichkeiten hat Lance Spitzner eine passende Definition gefunden: „A honeypot is security resource whose value lies in being probed, attacked, or compromised.“ [**?**]

Der Honeypot ist nach Spitzner eine Sicherheits-Ressource deren Wert darin liegt, erforscht, attackiert und kompromittiert zu werden. Durch diese Definition ist offen gelassen, welchen Nutzen der Anwender daraus zieht. Mit dieser allgemein gehaltenen Definitionen können nun Ziele eines Honeypots im nächsten Kapitel beschrieben werden.

2.2 Ziele

Die Ziele eines Honeypots lassen sich in zwei Kategorien einteilen. Das sind zum einen die „Research Honeypots“und zum anderen die „Production Honeypots“.

Die unterschiedlichen Ziele dieser Honeypots werden in den folgenden Kapiteln beschrieben.

2.2.1 Research

Ein Research Honeypot wird für die Untersuchung und Forschung der Angriffe eingesetzt. Der Honeypot wird meist getrennt von allen Produktivsystemen aufgebaut. In Abb. 2.1 wird der Honeypot vor die Firewall des Produktivnetzes installiert. Dadurch wird versucht mögliche

Angreifer weit weg von dem Produktiv-Systemen zu halten.

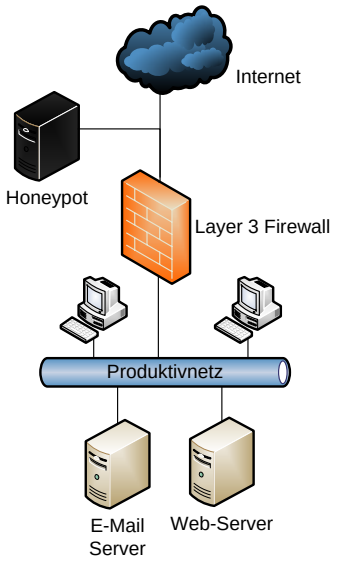

Abbildung 2.1: VResearch Honeypot - Variante I

Eine weitere Möglichkeit eines solchen Aufbaus ist in Abb. 2.2 dargestellt. Hier wird der Honeypot durch die Layer 3 Firewall von dem Produktivnetz getrennt. Bei diesem Aufbau muss genau auf die Firewall-Einstellungen geachtet werden, um den Angreifern nicht ungewollt einen Sprungserver zu den produktiven Systemen zur Verfügung zu stellen.

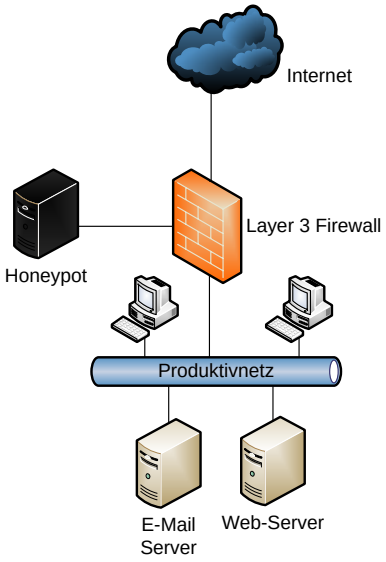

Abbildung 2.2: Research Honeypot - Variante II

Es geht hier primär um die Erkennung von neuen Würmern, Trojanern und weiterer Schadsoftware. Neben der Schadsoftware werden auch Angriffe von Hackern sowie der nicht so qualifizierten „Skript-Kiddies“untersucht.

Bei jedem Angriff werden Daten und Informationen über den Angreifer gesammelt. Dabei

lässt sich häufig ermitteln wer, womit und evtl. auch warum er angegriffen hat. Setzt man mehrere Honeypots an verschiedenen Standorten ein, lassen sich mit den gesammelten Daten neue Trends der Angreifer erkennen. Durch diese Informationen versuchen Unternehmen Angriffe auch vorhersagen zu können und die Vorgehensweise der Angreifer besser zu verstehen. Mit den gesammelten Daten können auch neue Tools, mit denen vorallem Skript-Kiddies arbeiten, aufgedeckt werden.

Um die Daten besser auswerten zu können teilen verschiedene Organisationen ihre gesammelten Daten und Erkenntnisse mit anderen Unternehmen. Die große Anzahl der Daten ermöglicht eine bessere Analyse sowie eine genauere Trend-Erkennung.

Research Honeypots dienen somit nicht direkt zur Risiko-Minimierung. Vielmehr helfen die Erkenntnisse, die Angriffs-Prävention und Erkennung zu verbessern.

2.2.2 Production

Im Gegensatz zu einem Research Honeypot wird ein Production Honeypot, wie der Name schon sagt, in einem produktiven Umfeld eingesetzt. Eine mögliche Implementierung ist in Abb. 2.3 zu sehen. Ziel eines Production Honeypots ist es, das Netzwerk sicherer zu machen. Er soll dafür sorgen die Infrastruktur vor Angriffen zu schützen und die Aufmerksamkeit der Angreifer auf sich zu lenken.

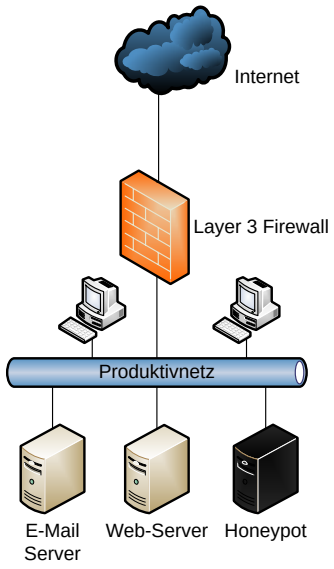

Abbildung 2.3: Production Honeypot

Um die Wirkung des Honeypots genauer zu beschreiben, werden im folgenden die drei Kategorien von „Sicherheit“, welche von —Verweiß— definiert wurden, aus Sicht des Honeypots erläutert.

- **Verhütung (Prevention)**

  Um ein Netzwerk vor einem Angriff zu schützen werden häufig Firewalls und andere Sicherheits-Systeme eingesetzt. Ein Honeypot wird jedoch selten eingesetzt um sich vorbeugend vor Angriffen zu schützen. Sobald ein Honeypot aktiv wird, ist der Angreifer

bereits in das Netzwerk bzw. System eingedrungen. Wenn der Honeypot falsch konfiguriert ist, wird er sogar zu einer Gefahr für das eigene Netzwerk und zu einer Eingangstür für Angreifer. Trotz alle dem gibt es auch Punkte, die für einen Honeypot als vorbeugende Sicherheitsmaßnahme sprechen.

Zum einen werden die Angreifer verunsichert, da sie über die Existenz der Honeypots wissen. Bricht ein Angreifer in ein System ein, muss er sicher immer bewusst sein, dass er möglicherweise gerade in eine Falle gelockt wurde. Diese Tatsache könnte Angreifer verunsichern und von größeren Netzwerken fernhalten.

Des Weiteren kann argumentiert werden, dass ein Angriff auf einen Honeypot den Schaden von produktiven Systemen vorbeugend abwehrt. Der Honeypot beschäftigt den Angreifer und führt ihn meist für eine längere Zeit in die Irre. In dieser Zeit kann der Angreifer an keine wichtigen Daten gelangen und der eigentliche Angriff auf die produktiven Systeme wird dadurch verhindert. Dieser Aspekt zur Sicherung produktiver Daten kann ebenfalls als ein vorbeugender Schutz gesehen werden.

- **Erkennung (Detection)**

  Im Gegensatz zu der Angriffs-Verhütung ist der Honeypot für die Erkennung von Angriffen besser geeignet. In einer Demilitarisierten Zone (DMZ) eines Netzwerks sind Systeme, welche meist mit dem Internet verbunden sind. Durch diese Verbindung ist das Risiko eines Angriffs an dieser Stelle erhöht. Wird hier ein Honeypot, wie es in Abb. 2.3 gezeigt, implementiert, kann dieser als eine Art Alarmanlage dienen. Da ein Honeypot nicht aktiv im Netzwerk arbeitet und somit auch keine Systeme eine Verbindung mit dem Honeypot benötigen, sollte es keine Verbindungsversuche geben. Bei Verbindungsversuchen auf Ports welche typischerweise für Internetdienste verwendet wird, handelt es sich meist um einen Port-Scan eines potenziellen Angreifers. Dieser versucht damit herauszufinden, um welches System es sich handelt und welche Dienste es zur Verfügung stellt. Oftmals werden dabei die Ports 25 für E-Mail Dienst und auch Port 80 für HTTP überprüft.

  Alarmierend sind Zugriffe von Systemen, welche sich in der selben DMZ befinden wie der Honeypot. Ein solcher Zugriff bedeutet meist, dass ein produktiv System von einem Angreifer kompromittiert wurde. In einem solchen Fall sollten zwingend alle Systeme überprüft werden, da der Angreifer möglicherweise weitere Server der DMZ kompromittiert hat.

- **Reaktion (Response)**

  Wenn ein System erfolgreich angegriffen wurde, ist es wichtig herauszufinden, wie der Angreifer vorgegangen ist. Hierfür muss das kompromittierte System untersucht werden. Wichtig sind dabei die MAC (Modify, Access, Change) Zeiten der Dateien. Mit den Zeitstempeln lässt sich nachvollziehen, welche Dateien der Angreifer manipuliert hat. Auf einem produktiven Webserver wäre eine solche Untersuchen sehr schwierig, da hier meist ein großes Datenvolumen anfällt und dementsprechend auch viele schreib- und lese-Zugriffe auf Dateien erfolgen. Der Honyepot hat den klaren Vorteil, dass Veränderungen einzelner Dateien auf den Angreifer zurückzuführen sind. Das macht eine Untersuchung

des Angriffs wesentlich leichter und schneller. Bei einer genauen Untersuchung kann festgestellt werden, wie der Hacker das System kompromittiert hat und womöglich auch welche Ziele er damit verfolgt hat. Mit diesen Informationen hat der System-Administrator die Möglichkeit, weitere Systeme in diesem Netz vor solchen Angriffen zu schützen. Zudem können mit diesen Informationen die übrigen Systeme auf eine Kompromittierung untersucht werden.

Im Bereich der Reaktion auf einen Angriff, kann ein Honeypot sehr nützlich sein. Mit Hilfe eines Honeypots können andere Systeme gegen Angriffe geschützt werden und bereits kompromittierte Systeme aufgedeckt werden. Dabei ist jedoch immer zu beachten, sollte ein Angreifer den Honeypot auslassen und nur produktive Systeme attackieren, ist der Honeypot keine Unterstützung. Der Administrator ist darauf angewiesen, dass der Honeypot auch in den Angriff mit einbezogen wird.

2.3 Rechtliche Grundlage

Der Betreiber eines Honeypots stellt bewusst ein Angriffsziel für blackhats zur Verfügung. Nutzt ein blackhat den Zugriff auf einen Honeypot, um von dort aus weitere Straftaten zu begehen, kommen dabei auch Rechtliche Fragen auf. Kann dem Betreiber des Honeypots fahrlässiges handeln oder gar eine Beihilfe vorgeworfen werden?

Hinzu kommt die Frage, mit der sich auch schon [1] beschäftigt hat, dürfen die Daten der blackhats verwendet werden, ohne das Wissen gerade an einem Experiment teil zu nehmen?

In den folgenden Kapiteln werden diese Fragestellungen kurz beleuchtet.

2.3.1 Zivil- und Strafrecht

Sollte der Honeypot, von einem blackhat, für einen Übergriff auf einen unbeteiligten Dritten genutzt werden, könnte ihm Strafrechtlich eine Beihilfe vorgeworfen werden. Um eine Beihilfe einer Straftat zu leisten, erklärt [1], muss eine aktive Hilfeleistung seitens des Honeypts-Betreibers nachgewiesen werden.

Im Fall der Honeypots ist dieser Nachweis nicht möglich, da es gezielt Maßnahmen zum Schutz vor Übergriffen durch den Honeypot auf fremde Systeme gibt. Der Sicherheitsstandard eines Honeypots ist oft nicht geringer als der anderer Systeme. Zudem kann auch nicht behauptet werden, dass sich ein Honeypot einem blackhat explizit aufdrängt oder ihn zu einer Straftat verleitet.

Zivilrechtlich stellt sich nun die Frage, ob der Betreiber für einen aufgekommen Schaden Haftbar gemacht werden kann. [1] beantwortet diese Frage ebenfalls mit nein. Um den Betreiber Haftbar zu machen, muss der Honeypot aktiv anderen Systemen schaden zufügen. Da ein solcher Sachverhalt nicht vorliegt, bleibt die Möglichkeit einer Unterlassung der Absicherung. Diese Anschuldigung stützt sich auf der sogenannten Verkehrssicherungspflicht. Hierbei geht es im groben darum, dass der Betreiber die Gefahren, die bewusst durch den Honeypot geschaffen werden, mit angemessenen Vorkehrungen gering hält. Somit soll der schaden Dritter vermieden werden.

Auch hier ist zu sagen, dass Honeypots ausreichend überwacht und gesichert werden, wodurch eine Unterlassung der Absicherung ausgeschlossen werden kann.

2.3.2 Datenschutz

Ist ein Angreifer auf einem Honeypot zu Gange, wird er oftmals von Tools überwacht und aufgezeichnet. Diese Aufzeichnungen stehen dem Betreiber zur Verfügung und dass ohne das Wissen des Angreifers. Dornseif zweifelt dabei die Unwissenheit des Angreifers an und argumentiert damit, dass die blackhat-Community über die Existenz von Honeypots informiert sind. Somit nehmen sie bei ihren Angriffen eine Aufzeichnung und Überwachung in Kauf.

Hinzu kommt, dass bei den Angriffen von blackhats keine personenbezogenen Daten in die Hände des Betreibers fallen. Mit dieser Argumentation kann auch der Datenschutzrechtliche Hintergrund entkräftet werden.

Kapitel 3

Honeynets

Honeynets bestehen aus mehreren virtuellen oder physikalischen Honeypots, die ein komplettes Netzwerk darstellen. Jede Netzwerkkomponente und jeder Server kann dabei als Honeypot angesehen werden. So besteht die Möglichkeit ein komplettes Produktivnetz nachzustellen, welches dem Hacker den Eindruck vermittelt, in ein produktiv verwendetes Firmennetz eingedrungen zu sein.

3.1 Anforderungen eines Honeynets

Die Anforderungen eines Honeynets kann grob ich drei Kategorien unterteilt werden:

Data Control: Datenkontrolle bedeutet, dass der Betreiber eines Honeypots oder Honeynets die Kontrolle der ein- und ausgehenden Datenpakete behält. Gelingt es dem Hacker dem Honeynetbetreiber diese Kontrolle zu entreißen, besteht die Möglichkeit, dass der Hacker einen Angriff auf das Produktivnetz startet. Die Datenkontrolle muss für den Angreifer unsichtbar sein. Zu strenge Sicherheitsvorkehrungen können den Hacker jedoch verunsichern und ihn auf den Honeypot aufmerksam machen (z.B. ausgehende Verbindungen blockieren). Jedoch kann z.B. eine Limitierung der ausgehenden Verbindungen ein gutes Mittel gegen den Verlust den Datenkontrolle sein.

Data Capture: Die Informationen, die ein Hacker während eines Angriffes hinterlässt, müssen möglichst reichhaltig und unauffällig dokumentiert werden. Für die Datensammlung gibt es verschiedene Möglichkeiten die Aktionen eine Hackers aufzuzeichnen.

- Packet Sniffing: Zum aufzeichnen des kompletten Netzverkehrs (ein- und ausgehende Pakete).
- Keystroke Logging: Das aufzeichnen der vom Hacker ausgeführten Tastenanschläge.
- Snapshot Software: Vergleicht Betriebssystem vor- und nach der Kompromittierung und hält die Änderungen fest.
- Log-Dateien wie z.B. Log-Daten eines Netzwerkgerätes wie Switch und Router

Wichtig bei diesen Programmen ist es, dass sie für den Hacker nicht ersichtlich sein dürfen. Findet der Angreifer eines dieser Programme ist dieser alarmiert und wird die Flucht ergreifen.

Data Collection: Bei einem verteilten System wie z.B. bei einem Honeynet muss es eine zentrale Stelle geben in der die Informationen gesammelt und gespeichert werden. Daten werden nie direkt auf einem Honeypot protokolliert, sondern an ein zentrales System übertragen. Wichtig hierbei ist es, dass die Daten sicher und unverändert an das System übertragen werden. Der Angreifer soll nicht die Möglichkeit haben einen Angriff zu vertuschen, oder gar das System selbst anzugreifen.

Data Analysis: Die gewonnenen Informationen müssen dem Zweck entsprechend ausgewertet werden. Je nach Ziel des Honeypots müssen z.B. Gegenmaßnahmen getroffen, oder aus dem gelernten Wissen Schlüsse gezogen werden, um in Zukunft eine Kompromittierung eines Produktivsystems zu verhindern.

3.2 Architektur eines Honeynets

Es gibt zwei verschieden Arten von Architekturen die sich im Laufe der Zeit durchgesetzt haben. Diese werden in Generation I und Generation II Honeynets (Abk. GenI und GenII) unterteilt.

3.2.1 GenI Honeynet

Bei einem GenI Honeypot wird das gesamte Netz durch eine Firewall in drei Teile unterteilt. Der erste Teil ist das Produktivnetz indem sich das zentrale Management System befindet. Der zweite Teil ist das Internet, welcher das Zugangsmedium des Angreifers darstellt. Der dritte und letzte Teil ist das Honeynet.

Der Prozess der Datensammlung beginnt bereits mit passieren der Firewall. Dort können Informationen wie die verwendeten Protokolle, Zeitstempel,IP-Adressen und Ports gesammelt werden. Außerdem wird hier kontrolliert wie oft der Angreifer eine Verbindung eingehen kann (Data Control). Wie viele Versuche zugelassen werden hängt vom Verwendungszweck des Honeynets ab. Der Router zwischen Honeynet und Firewall unterstützt diese auf zwei verschiedene weisen. Zum Einem versteckt er die Firewall vor dem Hacker. Der Angreifer denkt, er greift auf einen produktiven Router zu. Zum Anderen unterstützt er die Firewall in Sachen Zugriffskontrolle. So kann ein Single-Point-of-Failure vermieden werden.

Ein IDS-System steht nun noch zwischen dem Angreifer und den Honeypots. Dieses ist meist über einen Switch (oder wie in Abb. 3.1 mit einem Router) mit dem gesamten Honeynet verbunden. Dort werden alle Netzwerkaktivitäten protokolliert und bei bestimmten Angriffsmustern gegebenenfalls ein Alarm ausgelöst.

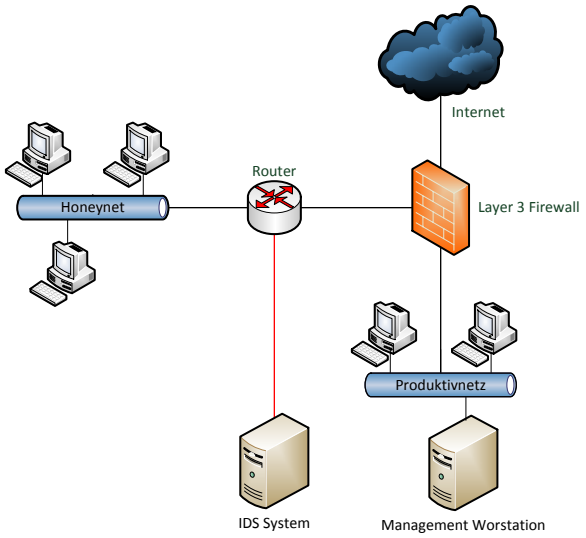

Abbildung 3.1: GenI Honeynet

3.2.2 GenII und GenIII Honeynets

Die zweite Generation von Honeynets verwendet ein Gateway, welches die Funktionen der in GenI verwendetet Komponenten enthält, und diese noch weiter ergänzt. Dieses Gateway (meist Honeywall genannt) besteht nicht wie in GenI aus einem Layer-3 Router sondern aus einer Layer-2 Bridge. Dies verhindert z.B. dass der Angreifer über die TTL-Zähler eines IP-Paketes dn Router erkennen würde. Alle zuvor genannten Anforderungen werden in der Honeywall erfüllt. Für die Datenkontrolle werden hier wie in GenI die Verbindungen limitiert (oft mit dem Programm IPTables), um so DOS-Angriffe zu vermeiden und die Kontrolle über die Verbindungen zu erhalten. Zusätzlich bietet sich nun auch die Möglichkeit Zugriffe die über bestimmte Protokolle einzeln zu limitieren.

Ein IDS oder IPS verhindert weiterhin das der Hacker vom Honeynet aus einen Angriff auf das Produktivsystem oder in das Internet starten kann. Die Zugriffskontrolle und das IDS sammeln wie beim GenI Honeypot die Daten. In Abb. 3.2 befindet sich ein Beispiel einer GenII Honeynet Architektur.

Ende 2004 wurde die vorerst letzte Generation von Honeynets vorgestellt. Ein GenIII Honeynet besitzt die selbe Netzwerkarchitektur wie dessen Vorgänger, behebt jedoch einig dessen Schwachstellen. Bei dem Versuch einen Honeynet Standard und eine Möglichkeit zu finden, ein Honeynet leichter zu Erstellen, veröffentlichte das Honeynet-Project (www.honeynet-project.org) eine CD, die alle Anforderungen eines Honeynets beinhaltet. Diese CD (*Roo* genannt) wird als dritte Generation angesehen. Die aktuelle Version (1.4, Stand: 2014) bietet neben der verbesserten Datensammlung ,eine grafische Web-Oberfläche zur Datenanalyse, und unterstützt weiter Tools wie Sebek und Hflow2 (Siehe Kapitel Tools).

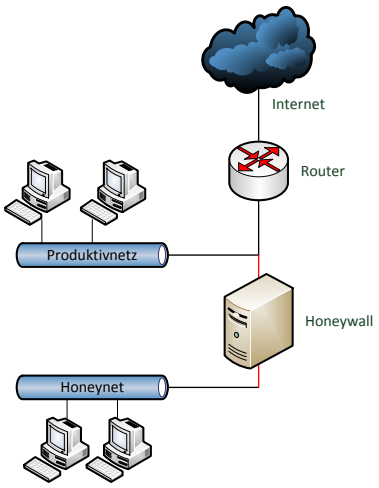

Abbildung 3.2: GenII Honeynet

Kapitel 4

Implementierungsarten

Wie viele Informationen ein Honeypot sammeln kann, hängt von seine Implementierungsart ab. Je nach Zweck und Aufwand kann zwischen den in diesem Kapitel genannten Arten unterschieden werden.

4.1 Typen von Honeypots

Die Funktionsweise von Honeypots ist von der Implementierungsart abhängig. Dabei spielt der gewünschte Grad der Interaktion, sowie der Zweck des Honeypots eine Rolle (Research oder Production). Je nach Anwendungsgebiet wird die dafür nützlichste Implementierung verwendet. Grundlegend wird dabei zwischen folgenden Implementierungsmöglichkeiten unterschieden:

4.1.1 High-Interactive

Bei High-Interactive Honeypots werden alle Dienste und Software "real"verwendet. Alle Anwendungen, Ports und alle Schichten des OSI-Modells werden dem Angreifer zur Verfügung gestellt. Diese Variante wird häufig dazu verwendet, um manuellen Hackern das Gefühl zu geben, in einen echten Server oder Computer eingedrungen zu sein. Die Überwachungsroutinen befinden sich dabei meist im Betriebssystem Kernel, um ein Entdecken des Honeypots zu erschweren. Der Vorteil dieser Methode ist, dass der Hacker alle Möglichkeiten besitzt, sein Ziel zu erreichen. Somit kann das meiste forensische Material gesammelt werden. Jedoch hat diese vollkommene Freiheit des Hackers den Nachteil, dass es für ihn leichter ist, in das Produktivsystem einzudringen.

Diese Art der Implementierung findet sich meist bei der Anwendung von Honeynets (vlg. Kapitel 2). Der Implementierungsaufwand bei High-Interaction Honeypots ist wesentlich höher als bei anderen Implementierungsarten, da nach einer erfolgreichen Kompromittierung das komplette System neu aufgesetzt werden muss.

4.1.2 Low-Interactive

Ein Low-Interaction Honeypot bietet im Vergleich zu einem High-Interactive Honeypot keine "realeÜmgebung. Meist emuliert ein Programm (z.B. honeyd) einen oder mehrere Dienste die dem Hacker einen eher eingeschränkten Zugang auf ein System gewähren. So wird meist

auch nur ein Zugriff auf die Netzwerkschicht des OSI-Modells und auf gewünschte Ports zugelassen. Da es sich bei dieser Implementierungsart um eine Emulation bestimmter Angriffsschnittstellen handelt muss nach einer versuchten Kompromittierung auch das Betriebssystem nicht vollständig neu aufgesetzt werden. Dadurch wird der Wartungsaufwand im Vergleich zu High-Interactive Honeypots erheblich vermindert.

Bei einem versuchten Angriff durch einen Wurm oder einem Hacker wird z.B. nur eine Verbindungsanfrage zugelassen, jedoch nicht akzeptiert. Dadurch kann z.B. über einem Portscanner die Herkunft der Verbindungsanfrage ermittelt werden. Das forensische Material das bei Low-Interactive Honeypots gesammelt werden kann ist deswegen im Vergleich zu High-Interactive Honeypots eingeschränkt.

4.1.3 Virtualisierung

Honeypots können wie normale Betriebssysteme über eine Virtualisierungssoftware auch virtualisiert werden. Dabei werden alle Dienste, Ports und OSI-Schichten emuliert. Zur Realisierung dieser Implementierungsart gibt es verschieden Möglichkeiten.

Virtual Machine Honeypots
Hierbei werden reale Betriebssysteme über eine Virtualisierungssoftware (z.B. VMWare oder VirtualBox) auf einem Rechner installiert. So ist es Möglich, auf einem physikalischen Rechner oder Server mehrere Honeypots in Betrieb zu halten. Der Vorteil dieser Methode ist, dass dem Hacker ein vollständiges OS mit allen Diensten, Netzwerkschichten und Anwendungen zur Verfügung gestellt wird, und trotzdem sich der Implementierungsaufwand in grenzen hält. Wird ein System kompromittiert, so kann es ohne viel Aufwand wiederhergestellt werden. Durch die Möglichkeit mehrerer Virtuelle Honeypots auf einem physikalischen Server zu betreiben ist es Möglich, ein komplettes virtuelles Honeynet auf einem Server zu erstellen.

Emulated Honeypots
Um einen Honeypot zu emulieren wird meist eine Software verwendet, die die grundlegenden Funktionen eines Betriebssystems oder Dienstes emuliert. Je nach Konfiguration kann ein emulierter Honeypot auch alle Dienste, OSI-Netzwerkschichten und Applikationen zur Verfügung stellen. Die Wiederherstellung des Systems ist dabei ähnlich einfach wie die der Virtuellen Honeypots. Durch die eingeschränkten Möglichkeiten der Emulation von verschiedenen Diensten oder Systemen werden emulierte Honeypots als Low-Ineractive eingestuft. Emulated Honeypots eigenen sich besonders für Honeypot Einsteiger, da sie meist einfach zu Konfigurieren sind und so schnell erste, zumeist automatisierter, Kompromitierungsversuche entdecken.

Viele Tools wie z.B. honeyd arbeiten mit diesem Prinzip.

4.2 Risiken

Da ein Honeypot oder Honeynet von echten Angreifern attackiert wird, muss damit gerechnet werden das der Angreifer eventuell versucht über den Honeypot in das Produktive Netz zu gelangen. Es muss dringend vermieden werden das eine solche Möglichkeit besteht, indem das/die Hostsystem/e genügend abgesichert wird. Außerdem muss vermieden werden, dass der Angreifer weiter Angriffe über das Internet vom Kompromittierte Netz aus startet (z.B. durch limitierte Verbindungsversuche aus dem Netz hinaus).

Neben den Risiken die eine Honeypot für das eigene Netz bringt, können auch unbewusst einige rechtliche Gesetzte verletzt werden (laut Strafgesetzbuch):

§ 26. Anstiftung. Als Anstifter wird gleich einem Täter bestraft, wer vorsätzlich einen anderen zu dessen vorsätzlich begangener rechtswidriger Tat bestimmt hat.

§27. Beihilfe. (1) Als Gehilfe wird bestraft, wer vorsätzlich einem anderen zu dessen vorsätzlich begangener rechtswidriger Tat Hilfe geleistet hat. (2) Die Strafe für den Gehilfen richtet sich nach der Strafdrohung für den Täter. Sie ist nach § 49 Abs. 1 zu mildern.

Über einen Emulierten Honeypot wäre eine Straftat schwer zu verwirklichen, da jedoch in Deutschland schon der versuch einer Straftat als Verbrechen gilt könnten sich dies in diesem Fall als problematisch erweisen. Des weiteren könnte ein absichtlich schlecht gewartetes Betriebssystem (z.B. aus Research-Gründen), das von einem Angreifer kompromittiert wurde, als Plattform für weitere Angriffe genutzt werden. Da das Betriebssystem vorsätzlich ungesichert ist, würde dies als Beihilfe für eine Straftat gelten.

Kapitel 5

Hacking

5.1 Syn-Flooding

5.2 Erkennungs- und Schutzmaßnahmen

Kapitel 6

Tools und Anwendungen

In diesem Kapitel werde die von Honeypots und Honeynets verwendeten Tools zum Sammeln von Daten und der Datenaufbereitung genauer betrachtet.

6.1 Honeypot und Honeynet Tools

Snort Snort_inline Hflow2(Gen3) Tcpdump Sebek

6.1.1 Sebek

6.1.2 ...

6.2 Honeypot Software

6.2.1 Honeyd

Honeyd ist ein von Niels Provos entwickelter Honeypot Deamon, der verschiedene virtuelle Host Systeme wie Workstations oder Netzwerkgeräte emulieren kann. Somit ist es möglich, eine komplette Netzstruktur zu emulieren. Die Open Source Sofware wird unter GPL veröffentlicht, und ist somit ohne Lizenzkosten benutzbar. Die aktuelle Version 1.5c wurde am 27.05.2007 veröffentlicht. Über die Benutzung von Konfigurationsdateien kann eine vielzahl von Betriebssystemen gewählt werden. Dabei werden s.g. Templates erstellt, für die zuerst eine Identität gewählt werden muss. Danach können verschiedene Parameter das simulierte Verhalten des virtuellen Clients festgelegt werden. Dazu gehören z.B. das eine emulierte Bandbreite des Übertragungskanals, Default-Aktionen verschiedener Ports und die zu aktivierenden Scripts beim Zugriff auf verschiedenen Ports. Zuletzt wird dem Template die gewünschte IP zugewiesen. In einer Honeyd Konfigurationsdatei können mehrere Clients eingetragen werden werden. Ein minimales Beispiel einer Honeyd-Konfigurationsdatei befindet sich in Abb. 6.1. Die Skripte, die bei einem versuchten Zugriff auf einem Port gestartet werden sollen können beliebig erstellt oder manipuliert werden.

```
create windows
set windows personality "Microsoft Windows 2000 SP2"
set windows default tcp action block
set windows default udp action block
set windows default icmp action block

add windows tcp port 23 "sh /usr/share/honeyd/scripts/win32/win2k/exchance-telnet.sh $ipsrc $sport $ipdst $dport"
add windows tcp port 25 "sh /usr/share/honeyd/scripts/win32/win2k/exchange-smtp.sh $ipsrc $sport $ipdst $dport"
add windows tcp port 110 "sh /usr/share/honeyd/scripts/win32/win2k/exchange-pop3.sh $ipsrc $sport $ipdst $dport"

bind 193.196.7.11 windows
```

Abbildung 6.1: Beispielkonfiguration einer Honeyd Config-Datei

Honeyd wird unter Unix Systemen in der Kommandozeile ausgeführt. Dabei stellt das Programm verschiedene Konfigurationsmöglichkeiten zur Verfügung. Über den Parameter honeyd -d wird das Programm nicht als Deamon gestartet. So kann man in der Konsole nach starten des Programmes die erfolgten Zugriffe beobachten. Diese möglicherweise versuchten Angriffe (je nach Netz werden auch alle Pakete die an die Broadcast Adresse gesendet werden empfangen) müssen danach analysiert werden. Um ein möglichst leicht auswertbares Ergebnis zu erhalten kann eine feste Angabe des abzuhörenden Netzbereichs sowie die Erstellung einer Log-Datei erfolgen. Der folgende Befehl kann so auf das zuvor genannte Beispiel angewandt werden:

honeyd -f honeyd.conf 193.196.7.11 -l honeyd.log

Hierbei werden nur die Pakete die für die Zieladresse 193.196.7.11 bestimmt sind für eine Auswertung in Betracht gezogen. Die erfolgten Zugriffe werden in der Datei honeyd.log gespeichert. Die Log-Dateien speichern folgende Datensätze:

2014-02-22-15:50:59.5296 tcp(6) S 109.193.95.133 21473 193.196.7.11 23 [Windows 2000 RFC1323]

Zunächst wird der Zeitpunkt des Angriffes festgestellt. Danach ob der Zugriff über TCP, UDP oder ICMP stattgefunden hat. Das S kennzeichnet den Start einer Verbindung. Bei UDP oder Paketen, die das Ende einer Verbindung kennzeichnen befindet sich an dieser Stelle ein E. Die nächsten beiden Werte sind die Quelle und Ziel IP-Adressen mit dem dazugehörigen Ports. Zuletzt befindet sich ein Versuch über Passive Fingerprinting das Betriebssystem des Angreifers zu erraten. Bei diesem Beispiel handelt es sich um einen Versuch über Telnet auf den Honeypot zuzugreifen. Da Honeyd seit 2007 nicht mehr weiterentwickelt wird verwendet es in diesem Beispiel eine veraltete Fingerprint Datenbank und gibt statt des verwendeten Windows 8.1 einen Windows 2000 Rechner an.

6.2.2 Kommerziell

6.2.3 Roo

6.2.4 Virtuelle Honeypots

6.3 Anwendungen zur Datenanalyse

6.3.1 Honey View

6.3.2 ...

Kapitel 7

Implementierung des Honeypots

7.1 Verwendete Software

7.2 Honeyd Implementierung

7.2.1 Installation

7.2.2 Konfiguration

7.2.3 Datenanalyse

7.3 Kommerzielle Implementierung

7.3.1 Installation

7.3.2 Konfiguration

7.3.3 Datenanalyse

7.4 Open Source Implementierung

7.4.1 Installation

7.4.2 Konfiguration

7.4.3 Datenanalyse

Kapitel 8

Auswertung der gesammelten Daten

Kapitel 9

Fazit

Anhang

Literaturverzeichnis

[1] Dornseif, M.: *Ermittlung von Verwundbarkeiten mit elektronischen Koedern*. ., RWTH Aachen, 2012.