# Sammeln von Würmern und Bots mit Honeypots

Sebastian Eckl
Betreuer: Lothar Braun
Seminar Future Internet WS09/10
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: sebastian.eckl@mytum.de

***Kurzfassung*—Eine stetig steigende Zahl von Internetangriffen und die Verbreitung immer ausgeklügelterer Schadsoftware bedingt die Entwicklung und den Einsatz neuer Verfahren und Möglichkeiten bezüglich Schutz und Abwehr. Honeypots sind reale oder virtuelle Rechnerumgebungen mit Netzwerkanbindung, deren Zweck einzig und allein darin besteht, gezielt Opfer manueller oder automatischer Angriffe und Übernahmen zu werden, um somit unerkannt Informationen über den Angreifer zu erlangen. Da nicht jeder Honeypot-Ansatz in jedem Einsatzszenario Erfolg verspricht, soll die folgende Arbeit einen Überblick über die verschiedenen Honeypot Arten und ihren jeweiligen Einsatzzweck geben.**

***Schlüsselworte*—Honeypot, Low-Interaction Honeypot, High-Interaction Honeypot, Client Honeypot**

## I. EINLEITUNG

Seit Februar 2000 [4], [7], als die ersten Distributed Denial of Service (DDoS) Attacken auf große E-Commerce Unternehmen und News Seiten gesichtet wurden, hat sich das Angriffspotential im World Wide Web zu einem gewichtigen Problem entwickelt. Spielte zu Beginn eher die blinde Kraft der Zerstörung eine Rolle, so wandelte sich die Intention im Laufe der Zeit in Richtung realer Internetkriminalität – das Erpressen von Unternehmen und das Ausspähen von Daten im Rahmen des Kreditkartenbetrugs seien hier exemplarisch genannt. Die bislang zugrundeliegende Infrastruktur bilden dabei Botnetze, also Netzwerke - oftmals sogar tausender - kompromittierter Rechner, die von einem Angreifer ferngesteuert werden [6]. Um die dabei eingesetzte Schadsoftware aufzuspüren und zu analysieren, wurde das Konzept der Honeypots entwickelt und zum Einsatz gebracht. Im Folgenden sollen nun die drei Honeypot Varianten Low-Interaction, High-Interaction und Client Honeypot vorgestellt, Einsatzbereiche, Vor- und Nachteile herausgearbeitet sowie eine abschließende Bewertung vorgenommen werden.

## II. DEFINITION VON HONEYPOTS

Ein Honeypot bezeichnet eine reale oder virtuelle Rechnerumgebung, deren Zweck darin besteht, gezielt Angreifer sowie automatisch verbreitende Schadsoftware (sog. Malware) anzulocken, um die dabei verwendeten Strategien und Techniken anschließend genauer analysieren zu können. Dem potentiellen Angreifer wird dabei eine scheinbar ungesicherte bzw. schlecht gesicherte Rechnerkonfiguration vorgetäuscht. Dieser Rechner erfüllt die Voraussetzung, keiner weiteren Benutzung zugeordnet zu sein, womit sich theoretisch gesehen keine ein- bzw. ausgehenden Verbindungen beobachten lassen dürften. Somit lässt sich jeder Verbindungsversuch entweder als Unfall oder, viel wahrscheinlicher, als Angriffsversuch werten. „False Positives“, also die irrtümliche Einstufung an sich harmlosen Datenverkehrs als Gefahr, sind hiermit minimiert bzw. faktisch ausgeschlossen - ein Hauptvorteil der Honeypots.

Der Wert eines Honeypots liegt in Umfang und Art der Informationen, die er generiert. So lassen sich hierbei auch verschlüsselte Daten (z.B. Tastatureingaben des Angreifers) mitschneiden, auf die gängige „Network Intrusion Detection“ Systeme (NDIS) nicht ansprechen. Der signifikanteste Vorteil von Honeypots liegt in der Tatsache, dass bestimmte Konzepte auch mit noch unbekannten Angriffssignaturen zurechtkommen. Letztere ermöglichen die Entdeckung bis dato unbekannter Schwachstellen [8] durch die Analyse ein- und ausgehender Netzwerkverbindungen sowie der Rechneraktivitäten.

Die Funktionsweise des Honeypots erklärt Aufgabe und Einsatzbereich. Ziel ist nicht die direkte Abwehr, sondern das Sammeln von Informationen über Angreifer und Schadsoftware zur Erforschung und Entwicklung langfristig angelegter Abwehrstrategien auf Basis bestehender Schutzsoftware.

Unterteilen lassen sich Honeypots allgemein in die beiden Oberkategorien Server- und Client-Honeypots. Zu den Server-Honeypots zählen sowohl Low-Interaction als auch High-Interaction Honeypots. Sie zeichnen sich im Wesentlichen dadurch aus, dass sie passiv auf Angreifer, in Form von Clients, warten. Client Honeypots versuchen dagegen aktiv, indem sie z.B. einschlägige Webseiten abgrasen, Schadsoftware zu sammeln.

Einzelne Honeypots lassen sich auch zu einem gesamten Netz zusammenschließen und bilden dann ein sogenanntes Honeynet bzw. eine Honeyfarm [10].

Eingesetzt werden Honeypots u.a. von Herstellern von Sicherheitssoftware (z.B. von Antiviren-Software) zur

stetigen Verbesserung der eigenen Produkte bzw. von Universitäten zu Forschungszwecken und Überwachung des eigenen Netzwerkes.

## III. LOW-INTERACTION HONEYPOTS

### A. Konzept

Low-Interaction Honeypots stellen die einfachste Honeypot Variante dar. Installation und Betrieb erfordern wenig Aufwand. Low-Interaction Honeypots offerieren einem Angreifer nur eine eingeschränkte Bandbreite an Möglichkeiten, indem sie entweder ein Betriebssystem oder nur ausgewählte Systemdienste, Schwachstellen, etc. emulieren. Sie versuchen damit, einem Angreifer in gewisser Weise ein reales System vorzutäuschen, bieten jedoch in Wirklichkeit kein vollwertiges System an. Oft enthält ein Low-Interaction Honeypot beispielsweise nur eine eingeschränkte Anzahl an Netzwerkdiensten und bietet gerade eben so viele Internetprotokolle (im Wesentlichen TCP und IP), wie zur Interaktion mit dem Angreifer notwendig sind, um ihm Glauben zu machen, er verbinde sich mit einem realen System. Der Grad der Interaktionsmöglichkeiten sollte dabei jedoch hoch genug sein, einen potentiellen Angreifer bzw. automatische Malware auszutricksen und nicht von vornherein zu verlieren.

Low-Interaction Honeypots dienen primär dazu, statistische Daten (z.B. Anzahl der Gesamtangriffe) und grundlegende Informationen über bestimmte Angriffsmuster zu sammeln. Aufgrund des einfach gehaltenen Betriebssystems fehlen komplexe Rechneraktivitäten, weshalb tiefergehende Analysen, wie die Erforschung gespeicherter Schadsoftware, an andere Systeme ausgelagert werden müssen.

Bekannte Low-Interaction Honeypot Distributionen sind honeyd und die Nepenthes Plattform. Per honeyd [11], [12] lassen sich komplexe Netzwerke und der TCP/IP Stack beliebiger Betriebssysteme simulieren. Kleine Scripte emulieren dabei reale Systemdienste und bieten dem Angreifer eingeschränkte Interaktionsmöglichkeiten.

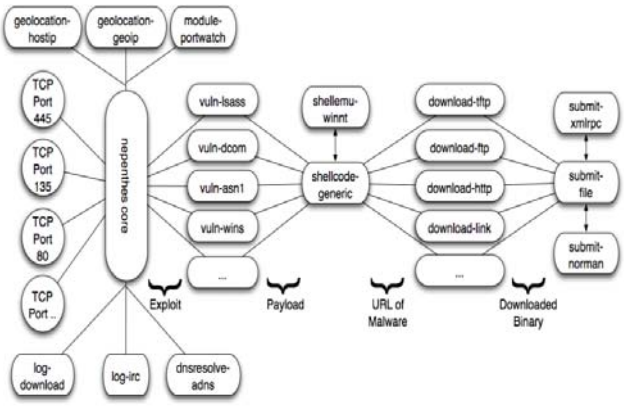


Abbildung. 1. Konzept der Nepenthes Plattform [14]

Die Nepenthes Plattform [14] (siehe Abbildung 1) bietet ein Framework um sich selbst verbreitende Malware im großen Stil zu sammeln. Nepenthes versteht sich dabei nicht als Honeypot per se, sondern vielmehr als Plattform, bestehend aus flexibel austausch- und erweiterbaren Honeypot Modulen. Vorgesehen sind dabei Module zur Emulation von Sicherheitslücken, zum Parsen auszuführender Shellcode Befehle, zum Download der per Shellcode angeforderten Malware und schlussendlich zum Mitschneiden und Speichern sämtlicher Aktivitäten auf dem Honeypot. Eine Erweiterung dieses Konzepts bieten Leita und Dacier mit SGNET [3] (siehe Abbildung 2). Sie verknüpfen Nepenthes mit dem Konzept automatisch dazu lernender Sensoren, die auf verschiedene IP-Adressbereiche weltweit verteilt sind. Sobald ein Angreifer einen Sensor attackiert, werden sämtliche Interaktionen über ein speziell entwickeltes Protokoll an ein zentrales Gateway gesandt und von dortigen Modulen zur Emulation von Systemdiensten und Ausführung von Shellcode abgearbeitet.

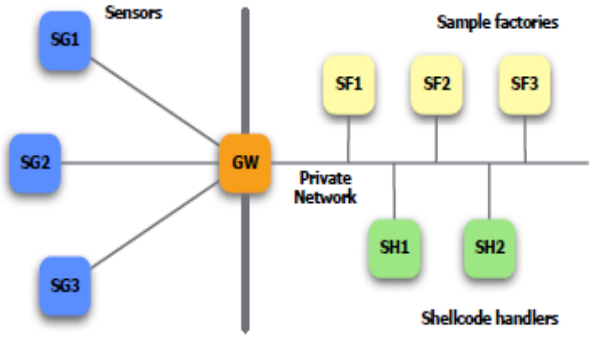


Abbildung. 2. Architektur der SGNET Plattform [3]

### B. Abgrenzung

Low-Interaction Honeypots basieren im Gegensatz zu High-Interaction Honeypots auf keinem vollständig ausgestatteten Betriebssystem, sondern emulieren lediglich gewisse Komponenten einer realen Rechnerumgebung. Einem Angreifer ist es somit nicht möglich, ein komplettes System zu kompromittieren und zu übernehmen.

Im Vergleich zu Client Honeypots verhalten sich Low-Interaction Honeypots passiv, d.h. sie warten darauf, angegriffen zu werden. Bestimmte Arten moderner Schadsoftware können somit nicht erfasst werden.

### C. Bewertung

Der klare Vorteil von Low-Interaction Honeypots liegt in ihrer Einfachheit. Installation und Inbetriebnahme gestalten sich problemlos. Es werden keine großen Anforderungen an zugrundeliegende Hardware Ressourcen gestellt, weshalb auch betagtere Rechner als Low-Interaction Honeypot in Frage kommen können.

Durch die Simulation bekannter Schwachstellen eignen sie sich hervorragend dafür, die Ausbeutung bereits bekannter Sicherheitslücken zu dokumentieren. Da es einem Angreifer aufgrund der kontrollierten und limitierten Umgebung nicht möglich ist, ein System komplett zu übernehmen, besteht wenig Risiko, dass der Honeypot für weitere kriminelle Zwecke missbraucht werden kann.

In der Einfachheit liegen allerdings auch die größten Nachteile von Low-Interaction Honeypots.

Sie erfordern einen hohen Arbeitsaufwand an Entwicklung und Aktualisierung, denn die zu emulierenden Sicherheitslücken müssen ständig von Hand programmiert und implementiert werden.

Aufgrund ihrer Konzeption und der durch die Emulation einzelner Schnittstellen eingeschränkten Möglichkeiten sind sie eher weniger geeignet für das Sammeln sogenannter Zero-Day Exploits, also noch unbekannter bzw. erst seit kurzer Zeit bekannter Sicherheitslücken, für die noch kein Patch existiert.

Low-Interaction Honeypots geben lediglich vor, ein reales System darzustellen, verweigern dem Angreifer jedoch die mächtigen Optionen einer realen Root-Shell. Experten könnten den Täuschungsversuch durchschauen und frühzeitig abgeschreckt werden.

Aufgrund der Warteposition von Low-Interaction Honeypots müssen die Angreifer sowie mögliche Schadsoftware zwingend aktiv zum Honeypot gelangen. Da gewisse IP-Adressbereiche aber mehr oder weniger zufällig von Angreifern abgegrast werden, kann es passieren, dass der Honeypot gar nicht bzw. nur sehr selten attackiert wird.

Netzwerkes.

## IV. HIGH-INTERACTION HONEYPOTS

### *A. Konzept*

Der High-Interaction Honeypot repräsentiert ein vollwertiges System, z.B. in Form eines gewöhnlichen Standardrechners, Routers oder Switches. Er simuliert also nicht entsprechende Schnittstellen bzw. Sicherheitslücken, sondern präsentiert sich in Form eines realen, wenn auch sehr schwach bis ungeschützten Betriebssystems, welches sich so auch im täglichen Gebrauch finden lässt. High-Interaction Honeypots bieten Angreifern damit die gesamte Bandbreite an Interaktionsmöglichkeiten, also die Möglichkeit, sämtliche bekannte, aber auch noch unbekannte Sicherheitslücken ausbeuten zu lassen. Sie liefern ein breiteres Datenspektrum und ermöglichen somit tiefere Erkenntnisse bezüglich Vorgehensweise, verwendeter Programme und Absichten der Eindringlinge.

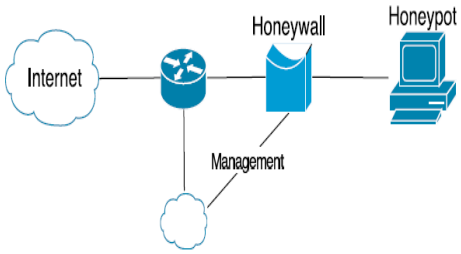


Abbildung 3: High-Interaction Honeypot geschützt durch Honeywall [7]

High-Interaction Honeypots müssen sehr stark überwacht werden, um Angreifer frühzeitig erkennen zu können und eventuellem Missbrauch gezielt vorzubeugen. Hierbei bietet sich das Konzept der sogenannten „Honeywalls“ an [13] (siehe Abbildung 3). Ein prominentes Beispiel bildet die Honeywall des Honeynet Projects, welche als Teil von GenIII und GenIV Honeynets konzipiert wurde. Die Honeywall bildet dabei eine Art zentrales Gateway - einem oder mehreren Honeypots vorgeschalten. Sie erlaubt das Sammeln von Daten, also sämtlicher Aktivitäten innerhalb des Honeynets bzw. eingehender und ausgehender Kommunikation, ohne Kenntnis des Angreifers. Darüberhinaus enthält sie auch Konzepte eines Network Intrusion Detection Systems, indem sie auffälligen ein- oder ausgehenden Traffic analysiert und eventuell kompromittierte Honeypots blockiert. Um das Risiko von DoS Attacken gegen andere Rechner zu minimieren, lassen sich per Netfilter Firewall auch gezielt spezielle Ports erlauben bzw. sperren (z.B. Port 22 für ssh-Verbindungen) sowie die Anzahl ausgehender Verbindungen limitieren. Hierbei muss jedoch die Balance gewahrt werden. Der Angreifer soll in der Lage sein, sich mit anderen Systemen im Internet zu verbinden, allerdings in eingeschränktem Rahmen, sodass von dem kompromittierten Rechner kein Bedrohungspotential ausgehen kann. Normalerweise wird die Honeywall als transparente Bridge realisiert, welche auf der Datenschicht fungiert. Ihr ist auf beiden Interfaces keine IP-Adresse zugewiesen, weshalb ein potentieller Angreifer nicht so leicht erkennen kann, dass ihm ein weiteres Netzwerkgerät vorgeschalten wird.

### *B. Abgrenzung*

High-Interaction Honeypots unterscheiden sich von Low-Interaction Honeypots primär im bereitgestellten Funktionsumfang. Sie bieten, auf Basis alltäglicher Betriebssysteme, im Wesentlichen den gleichen Funktionsumfang, der Angreifern auch bei anderen ungeschützten Rechnern im Internet zur Verfügung stünde. Auf diese Weise lieferten sie beispielsweise bereits Erkenntnisse über strategisches Verhalten der Angreifer. In ihren Experimenten haben Alata et. al. [5] herausgefunden, dass die Vorgehensweise zumeist in zwei Schritten erfolgt. Zu Beginn wird, meist automatisch per Skript, per „dictionary attack“ versucht, anhand eines Wörterbuchs,

schwache Passwörter ausfindig zu machen. Ist ein solches Passwort gefunden, verbindet sich der Angreifer einige Tage später manuell, um entsprechende Kommandos auszuführen.

Die High-Interaction Honeypots sind ebenfalls passiv konzipiert und warten auf potentielle Angreifer. Bestimmte Arten moderner Schadsoftware bleiben wiederum außen vor.

### *C. Bewertung*

Mit Hilfe eines High-Interaction Honeypots lässt sich prinzipiell der gesamte Kompromittierungsprozess, von der anfänglichen Auskundschaftung bis hin zur eigentlichen Übernahme, verfolgen:

- Wie geht der Angreifer bei der Auswahl seiner Ziele vor?
- Welche Techniken verwendet er, um mehr über ein bestimmtes System in Erfahrung zu bringen?
- Wie attackiert er den ausgewählten Zielrechner und welche Programme verwendet er?
- Welche Sicherheitslücken nützt er aus, um das System zu übernehmen?

Die sich ergebenden Erkenntnisse sowie die Vorgehensweise auf dem Rechner selbst erlauben eine Analyse der Methoden und Motive eines Hackers. Beispielsweise konnte so einiges über die typische Vorgehensweise bei Phishing Attacken und weiteren Formen des Identitätsdiebstahls herausgefunden werden. Der Prozess der Analyse ist allerdings zeitaufwendig und im Vergleich zu Low-Interaction Honeypots steigt der Ressourcen Verbrauch.

Der größte Nachteil liegt in der Gefahr der vollständigen Kompromittierung des Honeypots. Nachlässige Überwachung kann einem Angreifer Möglichkeit zu weiterem Missbrauch bieten. Der Honeypot mutiert zum Bot oder zur Schadsoftware- Schleuder und gefährdet nun weitere Rechner innerhalb oder außerhalb des eigenen Netzwerkes. Hier läuft der Honeypot Betreiber Gefahr, in rechtliche Schwierigkeiten zu geraten.

## V. CLIENT HONEYPOTS

### *A. Konzept*

Mit der steigenden Attraktivität und Benutzung des Internets, steigt auch die Anzahl von Sicherheitslücken in Internet Anwendungen, z.B. in Internet Browsern wie dem Microsoft Internet Explorer. Der typische Angreifer wird über eine Sicherheitslücke versuchen, eine Art Malware auf dem kompromittierten Rechner zu installieren. Diese Attacken erfordern keinerlei Aktionen von Seiten der Benutzer, lediglich der Besuch der Webseite reicht für eine Infektion aus. Unter Umständen erlangt der Angreifer somit die volle Kontrolle über das infizierte System und kann nach Belieben damit verfahren. Mögliche Optionen wären die Installation eines IRC-Bots, um den Rechner in ein bestehendes Botnetz einzureihen bzw. die Installation von Spyware oder einem Keylogger, um die Daten des Opfers (insbesondere Kreditkarteninformation, Bankdaten) auszuspähen.

Um sich vor diesen Gefahren zu schützen, bedarf es zwangsweise einer Ausweitung des klassischen Honeypot Prinzips: kein passives Warten, bis der Honeypot attackiert wird, sondern aktive Suche nach entsprechenden Schadsoftware-Distributionsstätten. Gemäß diesem Konzept arbeiten Client Honeypots, je nach Verwendungszweck auf Basis emulierter Schnittstellen (siehe Low-Interaction Honeypot) oder realem Betriebssystem (siehe High-Interaction Honeypot).

Die Aufgabe eines Client Honeypots ist das Auffinden, das Abspeichern und das Analysieren von Webseiten, die speziell von Angreifern mit Schadsoftware präpariert wurden, um den Internet Browser zu infizieren.

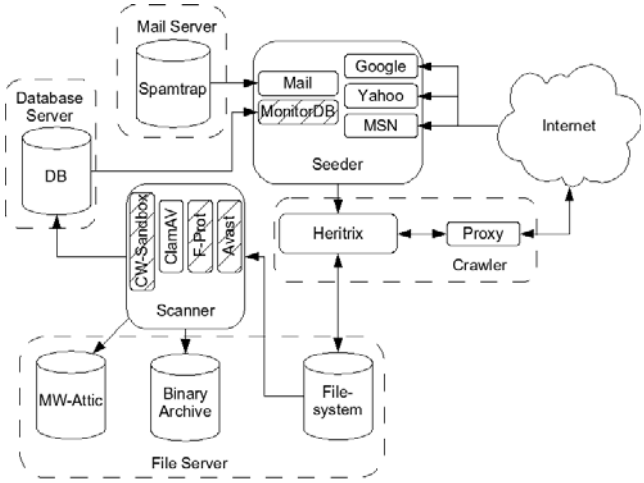


Abbildung 4: Schematischer Überblick des Monkey-Spider Konzepts [1]

Ikinci et. al. [1] beschreiben in ihrem Artikel „Monkey-Spider“ einen Client Honeypot Ansatz auf Basis des Low-Interaction Konzepts (siehe Abbildung 4). Sogenannte Web-Seeder durchforsten dabei entweder gängige Suchmaschinen bzw. extrahieren einschlägige URLs aus Spam-Mails. Eine Alternative wäre die Benutzung spezieller Blacklists, die eine Übersicht entsprechend einschlägiger Internetadressen enthalten. Gemäß Ikinci et. al. kommt dann für den Besuch der jeweiligen extrahierten URLs meist eine angepasste Version eines Web-Crawlers zum Einsatz, wie er in abgewandelter Form auch Suchmaschinenbetreibern zur Webindexierung dient. Der Crawler speichert die Webseite dann in einer Datei auf dem Server, die anschließend zur weiteren Analyse bereitsteht. Diese kann sowohl zentral als auch dezentral, weltweit verteilt, erfolgen. Ziel ist es, eine umfangreiche Datenbank über gefundene Schadsoftware zu erstellen.

Wang et. al. [15] beschreiben in „Automated Web Patrol with Strider HoneyMonkeys” einen Client Honeypot Ansatz basierend auf dem High-Interaction Prinzip in virtueller Umgebung. Hauptziel ist das Aufzeichnen sämtlicher Veränderungen, die nach erfolgreichem

Ausbeuten einer Sicherheitslücke, auf dem Honeypot stattfinden. Analog zu Ikinci et. al. müssen dabei zunächst entsprechende Website URLs extrahiert werden, ebenfalls unter Zuhilfenahme von Suchmaschinen, Spam-Mails oder Blacklists. Anschließend werden die Webseiten, im Gegensatz zur Low-Interaction Variante, mit vollständigen Browsern (je nach Konzept un- bzw. gepatcht) angesteuert. Hierbei gilt es, menschliches Surfverhalten so gut wie möglich nachzuempfinden. Bei erfolgreicher Infektion wird dem Angreifer eine gewisse Zeitspanne gewährt, in der er auf dem System Veränderungen vornehmen kann. Anschließend generiert HoneyMonkey eine XML-Datei mit einer Gesamtübersicht über sämtliche Veränderungen am vormaligen System. Enthalten sind darin Informationen über neu erstellte bzw. modifizierte ausführbare Dateien, über gestartete Prozesse, über neu angelegte oder modifizierte Registry-Einträge (im Falle von Windows), über die ausgebeutete Sicherheitslücke sowie über die vom Browser eventuell besuchten Redirect-URLs. Anschließend wird das infizierte System automatisch komplett neu aufgesetzt um somit für weitere Angreifer erneut in unberührtem Zustand zur Verfügung zu stehen.

### *B. Abgrenzung*

Von den serverseitigen Honeypots, also Low- und High-Interaction Honeypots, unterscheiden sich die Client Honeypots vor allem in ihrer Konzeption. Sie warten nicht darauf, angegriffen zu werden, sondern suchen aktiv nach Angreifern. Trotzdem basieren Client Honeypots auf den Grundkonzepten der Low- und High-Interaction Honeypots, fangen per emulierter Umgebung Schadsoftware ein oder lassen eine erfolgreiche Infektion zu.

### *C. Bewertung*

Client Honeypots reagieren auf die sich langsam wandelnde Distribution von Schadsoftware. Sie erlauben es, gezielt selbst nach Angreifern zu suchen und verzichten dabei auf das u.U. langwierige Warten serverseitiger Honeypots. Die Ausbeute an Schadsoftware ist so in der Regel um einiges größer.

Hauptproblem der Server Honeypots bleibt die Tatsache, dass sie mit bestimmten Varianten moderner Schädlinge nicht in Berührung kommen, da sich diese vom Konzept her anders verbreiten. Client Honeypots füllen diese Lücke, indem sie aktiv (einschlägige) Webseiten abgrasen und dort nach potentieller Schadsoftware fahnden.

High-Interaction Client Honeypots eignen sich aufgrund ihrer Konzeption in Form eines vollständigen Betriebssystems auch sehr gut dafür, sogenannte Zero-Day Exploits aufzudecken.

Basierend auf den Grundkonzepten von Low- und High-Interaction Honeypots übernehmen Client Honeypots allerdings auch deren jeweilige Schwachstellen. Low-Interaction Client Honeypots sind aufgrund emulierter Systemumgebung und Sicherheitslücken ebenfalls weniger geeignet, Zero-Day Exploits zu sammeln. High-Interaction Client Honeypots benötigen eine stärkere Überwachung.

## VI. AKTUELLE IMPLEMENTIERUNGEN

Im Folgenden sollen kurz aktuelle Konzepte vorgestellt werden.

Der Nepenthes Plattform [14], aus dem Bereich der Low-Interaction Honeypots, liegt ein Framework zugrunde, dessen Hauptziel es ist, sich selbst verbreitende Schadsoftware im großen Stil zu sammeln. Nepenthes besteht dabei aus austausch- und erweiterbaren Honeypot Modulen. Der Hauptvorteil liegt somit in der flexiblen Erweiterbarkeit und damit einhergehenden Zukunftsfähigkeit. Allerdings kann Nepenthes nicht das gesamte Ausmaß eines Angriffs bestimmen und es ist nur Kontakt mit sich automatisch verbreitender Schadsoftware bzw. aktiven Angreifern möglich.

SGNET [3] gelingt es, die vielleicht wichtigste Einschränkung von Nepenthes - den Prozess der manuellen Anpassung und Erweiterung emulierter Schnittstellen - aufzuheben, durch automatisch dazu lernende Sensoren, die auf verschiedene IP-Adressbereiche weltweit verteilt sind.

Alata et. al. [5] experimentierten mit einem High-Interaction Honeypot auf virtueller Basis. Ihr Ziel: die Analyse der Aktivitäten eines Angreifers, sobald dieser sich erfolgreich Zugang zum Honeypot verschafft hatte. Angriffspunkt waren hierbei schwache ssh-Passwörter. Dabei konnte einiges über die allgemeine Vorgehensweise, den Angreifertypus (z.B. Mensch oder Maschine) sowie die generellen Fähigkeiten der Angreifer in Erfahrung gebracht werden. Die Ergebnisse sind zu deuten unter Berücksichtigung der bewussten Fokussierung auf die ssh-Schwachstelle. Andere Sicherheitslücken wurden ausgeblendet und nicht mit einbezogen.

Freiling et. al. [7] verwendeten ein Honeynet auf Basis ungepatchter, virtueller Windows 2000 und XP Versionen. Zur Aufzeichnung des Netzwerkverkehrs und zum Schutz vor vollständiger Kompromittierung befanden sich sämtliche Honeypots hinter einer Honeywall. Alle 24 Stunden wurden die Honeypots zudem komplett neu aufgesetzt.

Im Bereich der Low-Interaction Client Honeypots agieren Ikinci et. al. mit dem MonkeySpider Projekt sowie Seifert et. al. mit HoneyC.

MonkeySpider [1] konzentriert sich auf das Sammeln von Schadsoftware, die sich per Webseiten verbreitet. Der Such- und Sammelprozess verläuft dabei recht zügig, neue Bedrohungen lassen sich aber nicht in vollem Umfang ausfindig machen. Prinzipiell wird zwar der Gesamtprozess beschleunigt, die Analyse der gespeicherten Schadsoftware erfordert jedoch hohen Zeitaufwand, da diese nicht auf dem jeweiligen Honeypot selbst sondern mithilfe weiterer Tools auf anderen Rechnern ausgeführt werden muss. Aufgrund der sich ständig wandelnden Internetumgebung bleibt es, trotz aktiver Suche, unmöglich sämtliche Bedrohungen zu erfassen.

HoneyC [2] besteht im Wesentlichen aus drei modular aufgebauten Bausteinen, welche zum Datenaustausch untereinander auf das XML-Format (in Form von HTTP

Requests und Responses) setzen. Die „Queuer“-Komponente liefert die zu besuchenden URLs. Die „Visitor“-Komponente arbeitet diese ab und leitet Informationen an die „Analysis Engine“ (basierend auf Snort Rules). Letztere untersucht anschließend, inwieweit die betroffenen Webseiten Bedrohungspotenzial enthalten. Die flexible Erweiterbarkeit und damit einhergehende Zukunftsfähigkeit machen die Stärke von HoneyC aus. Allerdings ist HoneyC nicht gegen „False Positives“ gefeit.

Der von Wang et. al. [15] beschriebene Client Honeypot Ansatz ermöglicht das Aufzeichnen sämtlicher Veränderungen, die nach erfolgtem Ausbeuten einer Sicherheitslücke, auf dem Honeypot stattfinden, sowie die Möglichkeit, das Ausbeuten neuer, unbekannter Sicherheitslücken festzustellen. Den Autoren gelang es unter anderem, den ersten Zero-Day Exploit auf Basis der javaprxy.dll Schwachstelle ausfindig zu machen.

## VII. ZUSAMMENFASSUNF

Aufbau und theoretische Funktionsweise von Low-Interaction Honeypots, High-Interaction Honeypots und Client Honeypots wurden in der vorliegenden Arbeit erläutert. Die unterschiedlichen Vor- und Nachteile der einzelnen Konzepte belegen deren individuelle Daseinsberechtigung. Trotzdem lässt sich wohl prognostizieren, dass die Zukunft den aktiveren und flexibleren Konzepten aus dem Bereich der Client Honeypots gehören wird.

Denkbar sind zudem kreative Kombinationen – basierend auf den Grundkonzepten der Low- und High-Interaction Honeypots – zu sogenannten hybriden Honeypot Architekturen [9].

Als wichtigste Herausforderung zeichnet sich ab, neuen Methoden der Internetkriminalität schnell wirksam begegnen zu können. An diesem Kriterium werden bestehende wie zukünftige Honeypot Konzepte gemessen werden.

## LITERATUR


[1] Ali Ikinci, Thorsten Holz, Felix Freiling (2008), ‘Monkey-Spider: Detecting Malicious Websites with Low-Interaction Honeyclients‘, in Proceedings of Sicherheit 2008, Gesellschaft für Informatik.

[2] Christian Seifert, Ian Welch, Peter Komisarczuk (2006), ‘HoneyC - The Low-Interaction Client Honeypot’, http://homepages.mcs.vuw.ac.nz/~cseifert/blog/images/seifert-honeyc.pdf, zugegriffen: 11.09.2009.

[3] Corrado Leita, Marc Dacier (2008), ‘SGNET: A Worldwide Deployable Framework to Support the Analysis of Malware Threat Models’, in Proceedings of the Seventh European Dependable Computing Conference (EDCC’07), IEEE Computer Society.

[4] E. Alata, Marc Dacier, Y. Deswarte, M. Kaâniche, K. Kortchinsky, V. Nicomette, Van H. Pham, Fabien Pouget (2005), ‘CADHo: Collection and Analysis of Data from Honeypots’, in Proceedings of the 5th European Dependable Computing Conference (EDDC'05).

[5] E. Alata, V. Nicomette, M. Kaâniche, M. Dacier, M. Herrb, (2006), ’Lessons learned from the deployment of a high-interaction honeypot’, in Proceedings of the Sixth European Dependable Computing Conference (EDCC'06), IEEE Computer Society.

[6] Evan Cooke, Farnam Jahanian, Danny Mcpherson (2005), ‘The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets’, in Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI).

[7] Felix C. Freiling, Thorsten Holz, Georg Wicherski (2005), ‘Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks’, in Proceedings of 10th European Symposium on Research in Computer Security, (ESORICS).

[8] Marc Dacier, Fabien Pouget, Hervé Debar (2004), **‘**Honeypots: Practical Means to Validate Malicious Fault Assumptions’, in Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'04), IEEE Computer Society.

[9] Michael Bailey, Evan Cooke, David Watson, Farnam Jahanian (2004), ‘A hybrid honeypot architecture for scalable network monitoring’, in Technical Report CSE-TR-499-04.

[10] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage (2005), **‘**Scalability, fidelity, and containment in the potemkin virtual honeyfarm’, Source ACM SIGOPS Operating Systems Review, vol. 39, no. 5, Dec. 2005

[11] Mohamed Kaâniche, Y. Deswarte, Eric Alata, Marc Dacier, Vincent Nicomette (2005), ‘Empirical analysis and statistical modeling of attack processes based on honeypots’, in Workshop on empirical evaluation of dependability and security (WEEDS).

[12] Niels Provos (2003), ‘A Virtual Honeypot Framework’, in Proceedings of the 13th USENIX Security Symposium.

[13] Niels Provos, Thorsten Holz (2007), ‘Virtual Honeypots: From Botnet Tracking to Intrusion Detection’, Addison Wesley Professional.

[14] Paul Baecher, Markus Koetter, Maximillian Dornseif, Felix Freiling (2006), ‘The nepenthes platform: An efficient approach to collect malware’, in Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID), Springer.

[15] Yi-Min Wang, Doug Beck, Xuxian Jiang, Roussi Roussev, Chad Verbowski, Shuo Chen, and Sam King (2006), ‘Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites that Exploit Browser Vulnerabilities’, in Proceedings of NDSS 2006.