UNIVERSITÄT MANNHEIM

Eine Infrastruktur zur Einschätzung des aktuellen Gefährdungslevels durch Malware

Diplomarbeit

Laura Anna Itzel

November 2007

Erstkorrektor:	Prof. Dr. Felix C. Freiling
Zweitkorrektor:	Prof. Dr. Christian Becker
Betreuer:	Thorsten Holz

Universität Mannheim
Lehrstuhl für Praktische Informatik I
Professor Dr. Felix C. Freiling
68131 Mannheim

Abstract

Heutzutage stellt die Verbreitung automatisierter Malware eine große Gefahr für Systeme im Internet dar. Diese Art von Malware besteht aus einer Verbreitungsroutine und Schadfunktionen. So können sich beispielsweise Würmer und Bots völlig autonom ohne menschliche Interaktion über Schwachstellen im Betriebssystem oder anderer Software verbreiten.

In dieser Arbeit wird ein System vorgestellt, mit dem automatisiert Daten über Angriffe solcher Malware gesammelt, analysiert und bewertet werden können. Das Sammeln der Daten ist über ein Sensorsystem realisiert, das mit Hilfe eines elektronischen Köders (*Honeypot*) Angreifer anlockt und die so registrierten Informationen über eingehende Angriffe abspeichert. Die so gewonnenen Daten werden mit verschiedenen Methoden und Werkzeugen analysiert. Die Ergebnisse können jeweils aktuell über ein Webinterface abgerufen werden. Mit Hilfe von Metriken, die das System anhand der gesammelten Daten berechnet, kann zusammenfassend das aktuelle Gefährdungslevel durch automatisierte Malware im Internet eingeschätzt werden.

Nowadays autonomous spreading malware is a huge thread for systems on the Internet. This type of malware consists of propagation routines and compromising functions. For example bots and worms can propagate fully autonomous without human interaction by exploiting vulnerabilities in the operating system or other software.

In this work we present a system that automatically collects, analyzes, and evaluates data about attacks of such malware. Collecting the data is realized with a sensor system, that tries to lure attackers by means of a honeypot system and stores the registered information about incoming attackers. The resulting data will be analyzed by various methods and tools. The results can currently be retrieved by a webinterface. By means of security metrics, which the system calculates with the collected data, the current level of threat by autonomous spreading malware on the Internet can be estimated.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Das Internet ist heutzutage einer der wichtigsten Kanäle zur Kommunikation und zur Beschaffung von Informationen. Nicht nur im privaten Bereich, sondern auch in der Wirtschaft und in staatlichen Einrichtungen hängt sehr viel von der Zuverlässigkeit der Medien Computer und Internet ab. Eine Störung könnte unter anderem dramatische wirtschaftliche Folgen haben. Daher ist es von zentraler Bedeutung, Wege zu finden, das Internet für seine Nutzer so sicher wie möglich zu gestalten.

Derzeit vergeht kaum ein Tag, an dem in der Fachpresse nicht von einer neuen Sicherheitslücke in einer Software berichtet wird. Die Gefahr, die von diesen Sicherheitslücken ausgeht, ist vielen Anwendern kaum bewusst. Über solche Löcher im System kann die Funktionalität vieler Computer, die mit dem Internet verbunden sind und auch die Funktionalität des Internets an sich, empfindlich gestört werden.

Viren, *Würmer*, *Trojaner* und ähnliche schädliche Software können sich über solche Löcher verbreiten, wenn nicht rechtzeitig *Patches* zur Verfügung stehen, über die die Nutzer ihre Systeme aktualisieren und somit schützen können. Aber nicht nur die Verfügbarkeit solcher Sicherheitsaktualisierungen ist ein Thema, auch sind die Nutzer häufig nachlässig mit der Wartung ihres Systems.

Diese Tatsachen machen es wichtig, Gefahren im Internet frühzeitig zu erkennen und dementsprechende Sicherheitsmaßnahmen einzuleiten. Einen Beitrag zur Möglichkeit der frühen Erkennung und Warnung leistet diese Arbeit.

1.1 Gefährdung durch Malware

Schon in einer frühen Phase des Internets, zu der es noch kaum durch Privatanwender genutzt wurde, gab es erste *Computer-Viren*, die Schaden auf dem betroffenen System anrichten sollten. Seitdem ist die Sicherheit von Informationstechnik ein wichtiges Thema. Dabei stellt sich zunächst die Frage, was überhaupt die Sicherheit der IT bedroht. In diesem Abschnitt wird erläutert, wodurch Systeme im Internet bedroht sind und welche Facetten von Gefährdung in diesem Zusammenhang existieren.

Im November 1988 wurde das erste Programm gestartet, das sich automatisch über eine Sicherheitslücke über das Internet verbreitete. Später bekam dieser erste *Internet-Wurm* den Namen *Morris Worm* nach seinem Programmierer Robert Morris [Rey07]. Dieses Programm enthielt zwar keine Routinen, um das angegriffene System zu schädigen, jedoch begann damit eine Entwicklung, die bis heute anhält. Der *Morris Worm* befiel

damals über 6.000 Systeme, was zu dieser Zeit etwa 10% der gesamten mit dem Internet verbundenen Systeme entsprach.

In den letzten Jahren hat sich der Begriff *Malware* als Kurzform für *malicious Software* etabliert, was soviel bedeutet wie bösartiges Computerprogramm. Er beschreibt Software, deren Ziel es ist, auf einem System unerwünschte Funktionen auszuführen und damit das System zu schädigen. Unter *Malware* fallen unter anderem *Computerviren*, *Würmer*, *Bots* und *Trojanische Pferde*. Daneben existieren zahlreiche weitere Unterarten von Malware, die hier nicht separat betrachtet werden.

Unter dem Begriff *Computervirus* oder auch kurz *Virus* versteht man Schadsoftware, die sich durch Kopie verbreitet und Systeme infizieren kann. Dazu kopiert sich der Virus in andere Programme, Datenträger oder Dokumente. Ein *Computerwurm* oder kurz *Wurm* ist eine sich selbst verbreitende Schadsoftware. Im Gegensatz zu Viren verteilen sich Würmer aktiv über Netzwerke wie das Internet und versuchen, über bestimmte Sicherheitslücken in andere Systeme einzudringen. Ein *Trojanisches Pferd* oder kurz *Trojaner* ist ein meist nützliches Programm, in dem sich für den Benutzer nicht sichtbar bösartige Funktionalität verbirgt, wie beispielsweise *Spyware*, um das Nutzerverhalten auszuspionieren. Trojaner verbreiten sich in der Regel nicht selbst, sondern müssen vom Benutzer oder anderer Malware, wie beispielsweise Bots, installiert werden.

In dieser Arbeit wird lediglich so genannte *automatisierte Malware* betrachtet. Dies ist Malware, die sich selbst über das Internet verbreitet und für ihre Angriffe keine menschliche Interaktion benötigt. Sie besteht im Allgemeinen aus einer *Verbreitungsroutine* und einer *Schadfunktionen*. Die Verbreitungsroutine ist darauf ausgelegt, Sicherheitslücken auf entfernten Systemen, die über das Internet erreichbar sind, auszunutzen, um die Schadfunktionen auf diesen Systemen auszuführen. Unter einem *Angreifer*, der mit automatisierter Malware angreift, wird in dieser Arbeit daher kein Mensch verstanden, sondern das Computersystem, von dem aus die Verbreitungsroutine ausgeführt wird.

Meist ist automatisierte Malware so aufgebaut, dass Verbreitungsroutine und Schadfunktionen getrennt sind. Das bedeutet, es wird zunächst ein so genannter Exploit – das Ausnutzen einer Sicherheitslücke – ausgeführt, um Zugriff auf das entfernte System zu erlangen. Danach wird ein so genanntes *Malware Binary* – also eine ausführbare Binärdatei – nachgeladen und seine Funktionalität gestartet. Das Binary kann entweder auf dem gleichen System, von dem auch der Angriff ausging oder auf einem weiteren System im Internet, beispielsweise einem zentralen Server, liegen. Es enthält die eigentlichen Schadfunktionen.

Die Schadensroutinen automatisierter Malware enthalten häufig Funktionen, um das kompromittierte System an ein so genanntes *Botnetz* anzuschließen. Ein Botnetz ist ein Zusammenschluss von mehreren Systemen, die mit einem Bot (auch *Zombie* oder *Drone* genannt) infiziert sind, es kann Größenordnungen von mehreren hunderttausend Systemen erreichen. Ein Bot installiert auf dem angegriffenen System die Möglichkeit des entfernten Zugriffs für den Angreifer. Dadurch kann dieser die Systeme im Botnetz fernsteuern und für seine Zwecke nutzen [Hol05, HW06].

Botnetze werden über so genannte *Command- and Control Server (C&C-Server)* gesteuert. Dazu bauen die mit dem Bot infizierten Systeme – kurz Bots – autonom eine Verbindung zu diesem Server auf. Der C&C-Server gibt dann weitere Anweisungen an die Bots weiter [fSidI07].

Über Botnetze können beispielsweise *Distributed Denial of Service (DDoS)*-Angriffe durchgeführt werden. Dies sind Angriffe, bei denen durch eine Überlastung von bestimmten Computersystemen oder Netzwerken Ausfälle generiert werden, typischerweise bei großen Firmen oder Einrichtungen. Andere Zwecke, zu denen Botnetze genutzt werden, sind das Versenden von Spam-Emails, das Ausspionieren von Daten oder das Verbreiten von neuer Malware [Pro05].

Häufig wissen die Benutzer der infizierten Systeme nicht, dass ihr Computer Teil eines Botnetzes ist. Bereits Angreifer mit wenig Erfahrung können mit Hilfe von Botnetzen großen Schaden anrichten. Daher stellen Botnetze eine große Bedrohung im Internet dar.

1.2 Projektüberblick

Die vorliegende Diplomarbeit ist Teil des Projekts *Internet Malware Analyse System (InMAS)* für das *Bundesamt für Sicherheit in der Informationstechnik (BSI)*. Im Rahmen dieses Projekts soll durch die Sammlung und Analyse jeglicher Form von Malware ein nationales Frühwarnsystem gespeist werden, das frühzeitig vor Gefahren im Internet warnen kann. Neben der Universität Mannheim sind die Universität Dortmund und die Fachhochschule Gelsenkirchen an der Entwicklung beteiligt.

In Abbildung 1.1 ist ein schematischer Überblick über das Gesamtprojekt und die darin enthaltenen Teilprojekte dargestellt. Zunächst wird mit Hilfe des *Honeypots nepenthes* Malware aus dem Internet gesammelt und in dem so genannten *Raw Malware Repository*, einer Datenbank, abgelegt. Das Konzept eines Honeypots wird in Kapitel 2.1 erläutert. Die so gesammelten Malware Binaries werden dann mit verschiedenen Methoden statisch und dynamisch analysiert.

Die Ergebnisse der Analysen werden im *Malware Analysis Repository*, ebenfalls einer Datenbank, abgelegt. Über diese Datenbank wird dann eine Kopplung des InMAS mit dem *IAS* der FH Gelsenkirchen und mit einem System zur Signaturgenerierung der Universität Dortmund realisiert. Die Daten aus allen drei Systemen – InMAS, IAS und das System der Universität Dortmund – gehen im Lagezentrum des BSI ein, um das Frühwarnsystem zu speisen. Informationen zum Projekt InMAS mit allen Teilprojekten können der Projektskizze entnommen werden [InM07].

Im Rahmen der vorliegenden Diplomarbeit wurden verschiedene Teilprojekte von InMAS berührt. Das in dieser Arbeit entwickelte System heißt *BEAN – Bedrohungserkennung und -analyse Netzwerk*. Seine Infrastruktur ist schematisch in Abbildung 1.2 dargestellt.

Zunächst wurde für das Teilprojekt *Nepenthes (NE)* ein Sensorsystem entwickelt, das mit *nepenthes* als Grundlage auf verschiedenen Systemen zum Sammeln von Malwaredaten

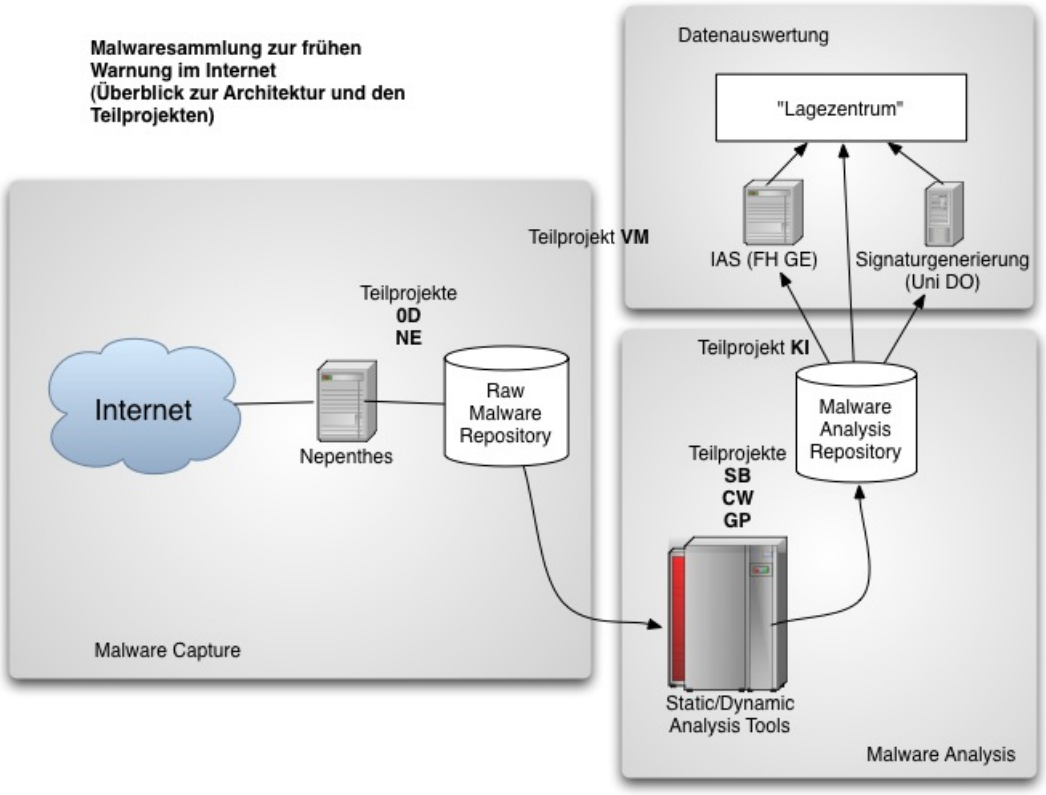


Abbildung 1.1: Schematische Übersicht über das gesamte Projekt

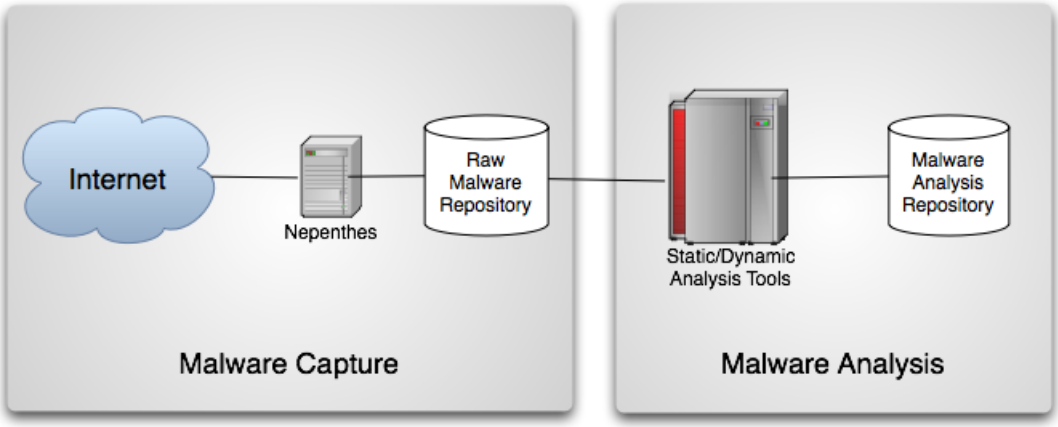


Abbildung 1.2: Schematische Übersicht über die Diplomarbeit

eingesetzt werden kann. Zur Speicherung der Daten wurde das *Raw Malware Repository* und Teile des *Malware Analysis Repositories* entwickelt.

Zur Darstellung der Daten wurde eine interaktive Benutzerschnittstelle in Form eines Webinterface entwickelt. Darin können sowohl die Daten aus den beiden Datenbanken als auch grafische Auswertungen der Daten abgerufen werden.

In *BEAN* wurden verschiedene statische Analysen der gesammelten Daten integriert, wie beispielsweise die Untersuchung mit verschiedenen Virenscannern. Zudem wurde *BEAN* an die *CWSandbox* angebunden, um die gesammelten Malware Binaries dynamisch zu analysieren.

Zuletzt wurde für das System eine erste Umsetzung des Teilprojekts *Statistik-Backend (SB)* erstellt. Dazu wurden für die gesammelten Daten automatisiert ablaufende statistische Analysen erstellt und eine Reihe Metriken zur Einschätzung des Gefährdungslevels durch automatisierte Malware entwickelt, die sowohl mit aktuellen als auch mit historischen Daten im Webinterface abgerufen werden können. Die weiteren Teilprojekte von InMAS aus Abbildung 1.1 werden in dieser Arbeit nicht behandelt.

1.3 Verwandte Arbeiten

Das in dieser Arbeit bearbeitete Gebiet ist bisher in der wissenschaftlichen Forschung noch wenig untersucht. Die einzige uns bekannte wissenschaftliche Bearbeitung auf dem Gebiet der Sammlung und späteren Analyse von Malware, die mit dieser Arbeit vergleichbar ist, haben bisher Goebel et al. präsentiert [GHW07]. Dort wurde ein sehr ähnliches Setup wie in dieser Arbeit verwendet, um im universitären Umfeld Malware zu sammeln und zu analysieren.

Goebel et al. haben *nepenthes* auf ungefähr 16.000 IP-Adressen im Netzwerk der Universität Aachen zum Sammeln eingesetzt. In einem Zeitraum von acht Wochen konnten sie damit über 13,4 Millionen Angriffe registrieren und mehr als 2.000 Malware Binaries herunterladen. Die gesammelten Daten wurden mit der *CWSandbox*, verschiedenen Virenscannern und *botspy*, einem Tool zum Aufspüren der Fernsteuerungsfunktion von Bots, untersucht.

In Form eines *Intrusion Detection Systems (IDS)*, ein System zum Erkennen von Angriffen auf Netzwerke, hat *Surfnet* mit *SurfIDS* [bv07] im *Open Source*-Bereich ebenfalls bereits ein Setup aufgebaut, das mit dem in dieser Arbeit entwickelten vergleichbar ist. Dort wurde *nepenthes* in ähnlicher Konfiguration wie für *BEAN* installiert. Der Umfang der statistischen Aussagen ist jedoch sehr begrenzt.

Ein weiterer Bereich, in dem bereits Arbeiten vorliegen, ist der der Sicherheitsmetriken. Zum einen wurden in den Arbeiten von Swanson et al. und Payne Definitionen und Leitlinien zur Entwicklung von Sicherheitsmetriken erarbeitet [SBS+03, Pay07]. Zum zweiten hat Jaquith grundlegende Richtlinien für Metriken in der IT-Sicherheit entwickelt [Jaq07].

1.4 Ziele und Aufbau der Arbeit

Die vorliegende Arbeit hatte zum Ziel, ein System zur voll automatisierten Sammlung und Analyse von automatisierter Malware im Internet zu entwickeln. Dieses System soll zum Einschätzen des aktuellen Gefährdungslevels durch automatisierte Malware im Internet eingesetzt werden.

Die Hauptherausforderungen der Arbeit lagen damit zum einen in der Entwicklung eines gut skalierbaren Sensorsystems zum Sammeln von Daten über Angriffe durch automatisierte Malware. Zum anderen sollen alle Daten zentral gespeichert werden, um sie für ebenfalls voll automatisierte Analysen zugreifbar zu machen.

Für alle Daten, die mit Hilfe des Sensorsystems gesammelt werden, sollen diese automatisierten Analysen direkt im System integriert werden. Dabei lag eine weitere Herausforderung darin, Auswertungen zu entwickeln, mit deren Hilfe aus den gesammelten Daten und Ergebnissen der integrierten Analysewerkzeuge Zusammenhänge sichtbar gemacht werden können, die Einfluss auf das aktuelle Gefährdungslevel haben können.

Um das Ziel der Einschätzung des Gefährdungslevels zu erreichen, soll eine interaktive Benutzerschnittstelle implementiert werden, die ebenfalls automatisiert grafische Aufbereitungen der Daten und Analyseergebnisse bietet. Vor allem soll diese Benutzerschnittstelle Kennzahlen anbieten, anhand derer eine erste Einschätzung zur aktuellen Gefährdung möglich ist.

Die vorliegende Arbeit ist wie folgt gegliedert: Nach der Einleitung wird in *Kapitel 2* erläutert, wie mit *BEAN* die Rohdaten über Malware gesammelt werden. Dazu wird zunächst das allgemeine Konzept eines *Honeypots* erläutert. Danach werden die einzelnen Komponenten von *BEAN* vorgestellt, die für das Sammeln der Rohdaten benötigt werden und an einem Beispiel erläutert. Zuletzt wird die für diese Arbeit verwendete Datengrundlage vorgestellt. Dieses Kapitel beschreibt, wie die Herausforderung der Entwicklung eines skalierbaren Sensorsystems und der zentralen Speicherung der gesammelten Daten gelöst wurde.

Im Anschluss daran beschreibt *Kapitel 3* mit welchen Methoden und Werkzeugen die gesammelten Daten in *BEAN* analysiert werden. Dazu werden zunächst einige Werkzeuge vorgestellt. Im Anschluss daran wird die Integration dieser Werkzeuge in *BEAN* erläutert, sodass das Ziel der automatisierten Analyse der Daten erreicht wird. Zudem wird auf die Vorgehensweise bei der Datenanalyse und auf die genutzten Möglichkeiten der Visualisierung der Ergebnisse eingegangen.

In *Kapitel 4* werden die Ergebnisse der in Kapitel 3 vorgestellten Analysemethoden beschrieben. Zudem werden hier die Auswertungen präsentiert, die wie weiter oben beschrieben Zusammenhänge aufzeigen, die Einfluss auf die aktuelle Gefährdung haben können. Dabei werden die Ergebnisse nach verschiedenen Kategorien untergliedert. Zunächst werden Erkenntnisse zu den Angriffen selbst, danach zu den Daten über die Angreifer und zuletzt über die gesammelten Malware Binaries präsentiert.

Anhand der in Kapitel 4 dargestellten Ergebnisse werden in *Kapitel 5* Metriken zur Einschätzung des aktuellen Gefährdungslevels durch Malware im Internet erarbeitet. Es wird zunächst allgemein erläutert, was unter einer Metrik zu verstehen ist und welchen Kriterien eine gute Metrik genügen muss. Daraufhin werden sechs konkrete Metriken beschrieben, die im Rahmen dieser Arbeit entwickelt wurden. Sie werden zu einer Gesamtmetrik zusammengefasst und beispielhaft für die Daten aus der Datengrundlage dargestellt. *Kapitel 6* schließlich gibt eine Zusammenfassung der gesamten Arbeit und beschreibt Möglichkeiten zur zukünftigen Erweiterung von *BEAN*.

1.5 Ergebnisse der Arbeit

In der vorliegenden Arbeit wird die Entwicklung des Systems *BEAN* beschrieben. Zunächst wurde ein Sensorsystem entwickelt, auf dem der Honeypot *nepenthes* und andere Tools laufen, um Daten über automatisierte Malware im Internet zu sammeln. Neben diesem Sensorsystem wurde eine Serverumgebung entworfen und realisiert, in der die gesammelten Daten zentral gespeichert und mit verschiedenen Werkzeugen automatisiert ausgewertet werden können. Zur Aufbereitung der Daten wurde ein Webinterface entwickelt, in dem sowohl die Daten, als auch die Ergebnisse der Analysen strukturiert abrufbar sind.

Es wurde für diese Arbeit eine Datengrundlage aufgebaut, indem das entwickelte Sensorsystem in verschiedenen Umgebungen installiert wurde. Die einzelnen Sensorsysteme waren über unterschiedliche deutsche *Internet Service Providern (ISP)* mit dem Internet verbunden, um eine möglichst breite Datenbasis zu erhalten.

Die Analysen der Datengrundlage ergaben, dass es bezüglich verschiedenster Datendimensionen zahlreiche Zusammenhänge gibt. Beispielweise wurden zeitliche und geographische Strukturen bei den Angriffsdaten festgestellt. Ebenso wurden Zusammenhänge zwischen dem ISP des Angreifers und dem des Angegriffenen gefunden.

Zuletzt wurden zur Einschätzung der IT-Sicherheitslage in Bezug auf automatisierte Malware Metriken definiert. Anhand der jeweils aktuellen Werte dieser Metriken können Trends erkannt werden und das Risiko eines Systems in Bezug auf die Kompromittierung durch automatisierte Malware im Internet beurteilt werden.

1.6 Danksagungen

Zunächst möchte ich mich bei Prof. Dr. Felix Freiling bedanken, der mir diese interessante Arbeit ermöglicht hat und mich während der letzten sechs Monate unterstützt hat. Mein Dank geht ebenso an Thorsten Holz für die Betreuung dieser Arbeit. Prof. Dr. Christian Becker möchte ich dafür danken, dass er die Zweitkorrektur übernommen hat.

Ein besonderer Dank geht an Ben Stock, der mich während der ganze Arbeit in vielerlei Hinsicht unterstützt hat. Ebenfalls bedanken möchte ich mich bei Christoph Klasik für das Korrekturlesen und die hilfreichen Inputs.

Zudem geht ein großer Dank an Marion Bieger-Itzel, Lennart Itzel, Markus Krammer, Florian Müller, Alexander Pfister und Johannes Stüttgen dafür, dass sie ihren Computer für mich über viele Wochen Daten haben sammeln lassen. Jürgen Jaap danke ich dafür, dass er bei auftretenden technischen Problemen immer hilfsbereit war.

Meiner Mutter, meinem Bruder, Martin Diers, Hans Verbeek und allen anderen, die immer ein offenes Ohr für mich hatten, möchte ich danken, dass sie mir geholfen haben in den arbeitsreichen letzten Monaten niemals aufzugeben.

Für seine Geduld, viele motivierende Worte und interessante und inspirierende Gespräche möchte ich mich ganz besonders bei Christoph Krammer bedanken.

2 Sammeln der Rohdaten

Der erste Schritt zur Einschätzung der Gefahren im Internet ist es, eine Datengrundlage aufzubauen. Das bedeutet konkret für dieses Projekt, dass voll automatisiert Daten über Angriffe gesammelt werden müssen. Diese Daten setzen sich zusammen aus Daten über den Angriff und den Angreifer wie zum Beispiel die IP-Adresse, von der der Angriff ausging und aus den Malware Binaries, die während dem Angriff heruntergeladen werden. Eine detaillierte Erläuterung zu Malware findet sich in Kapitel 1.1.

Im Folgenden werden zunächst generelle Konzepte von Honeypots erläutert und der in *BEAN* eingesetzte Honeypot *nepenthes* beschrieben. Daraufhin wird der Aufbau der im Rahmen dieser Arbeit entwickelten Sensor- und Serversysteme erklärt. Zur Veranschaulichung der Funktionalität der Komponenten von *BEAN* wird der Ablauf eines Angriffs beispielhaft beschrieben. Abschließend wird die Datengrundlage, die für diese Arbeit zugrunde gelegt wurde dargestellt.

2.1 Der Honeypot - nepenthes

Ein Honeypot – zu Deutsch *elektronischer Köder* – lockt durch das Bereitstellen von ungepatchten Sicherheitslücken Angreifer an. So kann der Honeypot Informationen über Angriffe, Angreiferverhalten oder die nachgeladenen Malware Binaries erkennen und speichern.

Zum Sammeln der Daten über Angriffe, Angreifer und der Malware selbst wird in *BEAN* die *Honeypot-Software nepenthes* verwendet. Im folgenden Abschnitt wird zunächst eine Definition eines Honeypots gegeben. Daraufhin werden verschiedene Arten von Honeypots voneinander abgegrenzt. Zuletzt wird der Aufbau, die Vorzüge und die Grenzen der Möglichkeiten von *nepenthes* dargestellt.

2.1.1 Honeypots

Honeypots sind nach Spitzner informationstechnische Sicherheitsressourcen, deren Wert darin liegt, unautorisiert beziehungsweise unerlaubt genutzt zu werden [Spi02]. Mit anderen Worten, der Nutzen eines Honeypots ist es, untersucht, angegriffen oder kompromittiert zu werden. Das generelle Konzept eines Honeypots ist es, ein System zu schaffen, bei dem niemand einen Grund hat, es zu nutzen oder mit ihm zu interagieren. Somit sind alle Verbindungen auf einen Honeypot per Definition unerlaubt und damit verdächtig.

Dadurch sind Honeypots sehr verlässliche Sicherheitssysteme, da sie nahezu keine *false positives*, also als bösartig erkannte gutartige Verbindungen produzieren [Spi04a].

Um Honeypots besser differenzieren zu können, werden sie in der Literatur nach dem *Level der Interaktion* in zwei Klassen eingeteilt, *low-interaction* und *high-interaction*. Das Level der Interaktion bezeichnet dabei die Möglichkeiten, die der Honeypot dem Angreifer bietet. Je mehr Interaktion der Honeypot erlaubt, desto mehr Aktivitäten kann der Angreifer auf dem Honeypot ausführen und desto mehr Informationen kann man über den Angriff gewinnen [Spi02].

Low-interaction Honeypots

Low-interaction Honeypots sind die einfachere Form von Honeypots. Sie *emulieren* Services lediglich, die Interaktion mit dem Angreifer ist sehr eingeschränkt. Bei dieser Art Honeypots wird nicht die komplette Funktionalität der emulierten Services implementiert. Meist sind lediglich ein paar wenige Kommandos möglich.

Dadurch sind die gewonnenen Informationen verhältnismäßig begrenzt auf zum Beispiel Datum und Zeit des Angriffs, die IP-Adresse des Angreifers oder den Port, auf den der Angriff zielte. Informationen über das Verhalten oder das Vorgehen des Angreifers können nicht gesammelt werden. Allerdings zeichnen sich low-interaction Honeypots durch geringen Implementierungsaufwand und ein geringes Risiko aus, denn der Angreifer hat durch die fehlende Interaktion nahezu keine Möglichkeit, den Honeypot tatsächlich zu kompromittieren oder andere Systeme über den Honeypot anzugreifen.

High-interaction Honeypots

High-interaction Honeypots stellen die komplexere Form von Honeypots dar. Sie stellen dem Angreifer ein komplettes Betriebssystem und vollständige Anwendungen zur Verfügung [Spi04b]. Die Dienste werden nicht emuliert, sie sind uneingeschränkt zugreifbar. Ein Angreifer kann tatsächlich in das System einbrechen und reale Dienste kompromittieren. Dadurch können mit einem high-interaction Honeypot viele Informationen gewonnen werden. Beispielsweise können die Tastenanschläge des Angreifers mitgeschrieben werden, es können Informationen über die Fähigkeiten der Angreifer und deren genaues Vorgehen gewonnen werden.

Diese Art Honeypots bietet außerdem die Möglichkeit, unbekannte Schwachstellen zu erkennen und bisher unbekannte Verhaltensmuster von Angreifern zu analysieren. Allerdings ist dieser hohe Grad an Interaktion mit großen Risiken verbunden, der Angreifer kann über die Dienste andere Dienste oder andere Systeme angreifen und kompromittieren. Der Aufwand, einen solchen Honeypot einzurichten und zu administrieren, ist dadurch sehr hoch.

2.1.2 Nepenthes-Plattform

Der Fokus in dieser Arbeit liegt, wie bereits in Kapitel 1.1 erläutert, auf Malware, die sich selbst verbreitet. Die Verbreitung läuft also automatisch. Solche Malware nutzt so genannte *Exploits*, also bestimmte Fehlfunktionen eines Dienstes. Durch ihre autonome Verbreitung stellt sie ein hohes Sicherheitsrisiko dar, da der Aufwand, viele Systeme zu infizieren, gering ist.

Die *nepenthes-Plattform* ist eine modulare Lösung zum Sammeln solcher Malware. Auf ihr werden verschiedene Netzwerkdienste emuliert. Dem Angreifer wird darüber ein recht hoher Grad an Interaktion mit dem Honeypot ermöglicht, da die komplette für den Angreifer relevante Funktionalität der Dienste abgebildet wird. Nach obiger Definition im Punkt Emulation fällt *nepenthes* trotzdem in die Kategorie der low-interaction Honeypots.

Allerdings sticht *nepenthes* durch den hohen Grad an Interaktion aus der Gruppe der klassischen *low-interaction Honeypots* heraus, bietet aber gleichzeitig im Vergleich zu high-interaction Honeypots eine gute Skalierbarkeit. Es gab daher Ansätze, eine dritte Klasse von Honeypots, die *medium-interaction Honeypots* zu etablieren, die mit emulierten Diensten eine hohe Interaktionsmöglichkeit bieten [Spi04b], jedoch hat sich diese Einteilung nicht durchgesetzt. Daher gilt *nepenthes* als low-interaction Honeypot.

Im Folgenden wird die Architektur der *nepenthes-Plattform* beschrieben. Dabei wird insbesondere auf die verschiedenen Komponenten dieses Honeypots eingegangen.

Nepenthes arbeitet mit verschiedenen Arten von Modulen: *Schwachstellen Module*, *Shellcode Handler*, *Download Module*, *Submit Module* und *Logging Module*. Zudem gibt es einige andere Komponenten, die für die Funktionalität von *nepenthes* zentral sind. Alle Module registrieren sich im Kern der Anwendung, dem so genannten *nepenthes core*, der die Koordination der Module wie auch der Netzwerkinterfaces übernimmt. Abbildung 2.1 zeigt einen schematischen Überblick über die einzelnen Komponenten. Im Folgenden werden diese näher erläutert.

Schwachstellen Module. Für jeden Dienst, der emuliert werden soll, muss in *nepenthes* ein *Schwachstellenmodul* implementiert sein. Diese Module bieten die notwendige Funktionalität des Dienstes, um dem Angreifer vorzutäuschen, er könne die entsprechende Schwachstelle im Dienst tatsächlich ausnutzen. Dadurch, dass nicht die volle Funktionalität des Dienstes implementiert wird, braucht *nepenthes* verhältnismäßig wenig Ressourcen.

Shellcode Handler Module. Die Schwachstellenmodule übergeben den *Shellcode Handler Modulen* die Payloads – also die eigentlichen Schadfunktionen – der Angriffe. Die Shellcode Handler Module versuchen dann diese zu analysieren. Gelingt dies, werden relevante Informationen über den Angriff extrahiert, wie zum Beispiel die Download URL, von der ein Malware Binary nachgeladen werden soll. Es gibt verschiedenste Shellcode

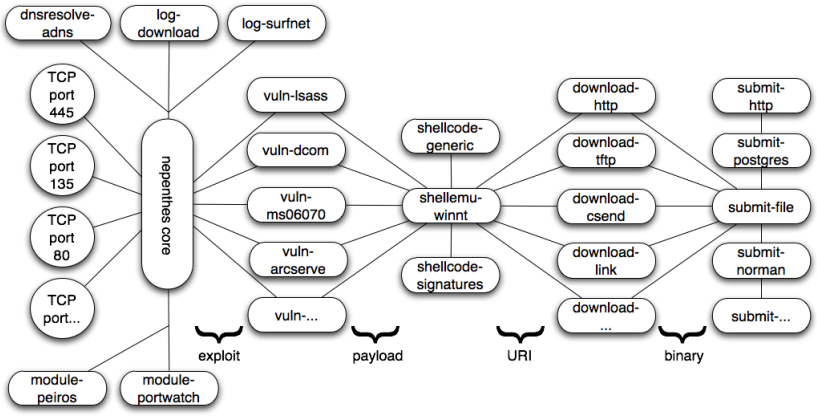


Abbildung 2.1: Aufbau von Nepenthes (aus [BKH+06])

Handler Module, da die Payload zum Beispiel auf eine bestimmte Art verschlüsselt sein kann und der entsprechende Shellcode Handler diese dann zuerst entschlüsseln muss, bevor er an den eigentlichen Schadcode gelangt.

Download Module. Um nicht nur Informationen über den Ablauf eines Angriffs, sondern auch über die verwendete Schadsoftware zu gewinnen, muss diese aus dem Internet heruntergeladen werden. Die *Download Module* übernehmen den Download von Dateien – also der Malware Binaries – aus dem Internet. Für jedes verfügbare Protokoll, zum Beispiel FTP, HTTP oder botspezifische Protokolle wie *csend*, ist jeweils ein Download-Handler implementiert.

Submit Module. Sobald die Dateien erfolgreich heruntergeladen wurden, werden sie an *Submit Module* übergeben. Diese übernehmen dann zum Beispiel das Speichern der Dateien auf die Festplatte oder über HTTP auf einen Server. Oder sie übergeben die Dateien an Antivirenprogramme oder andere externe Analysewerkzeuge.

Logging Module. Diese Module speichern Informationen über Ereignisse aus dem *nepenthes core*. Zum Beispiel gibt es ein Modul, das alle Informationen über Angriffe und den Download von Dateien in eine *PostgreSQL*-Datenbank protokolliert.

Weitere Komponenten. Manche Angreifer arbeiten nicht über Shellcodes, sondern über das Ausführen von einzelnen Shellbefehlen. Um auch diese Angriffe verarbeiten zu können, kann *nepenthes* in diesem Fall eine *Windows Shell* emulieren. Es reicht dabei die Emulation einiger weniger Kommandos und die Möglichkeit, Kommandodateien auszuführen, um automatische Angreifer zu täuschen.

Ebenso wie bei Shellcode Handler Modulen wird hierbei versucht, Informationen wie die URL, über die die Malware nachgeladen werden soll, zu extrahieren und an die Download Module zu übergeben. Diese Komponente arbeitet *nepenthes*-intern wie ein Shellcode Handler. Zudem bietet *nepenthes* ein *virtuelles Dateisystem* an, damit solche Shell-Angreifer erfolgreich getäuscht werden können, indem er beispielsweise während des Angriffs temporäre Dateien anlegen kann [BKH+06].

Abschließend werden die Grenzen in den Möglichkeiten von *nepenthes* erläutert. Wie bereits beschrieben wurde, kann *nepenthes* lediglich Daten über automatisierte Malware sammeln. Dies ist eine Beschränkung, die viele low-interaction Honeypots aufweisen.

Da *nepenthes* durch die Emulation vieler Dienste unterschiedlicher Betriebssysteme auf viele verschiedene TCP-Pakete antwortet, ist es denkbar dass Angreifer erkennen, wenn sich hinter einer IP-Adresse ein *nepenthes*-System befindet [PH07]. Außerdem ist es möglich, *nepenthes* dadurch zu erkennen, dass lokale IP-Adressen in Download-URLs durch die IP-Adresse des Angreifers ersetzt werden [mwc07].

Über diese Wege wäre es möglich, Malware so zu konfigurieren, dass sie Systeme, auf denen von *nepenthes* Schwachstellen emuliert werden, nicht angreift. Bisher gibt es keine Anzeichen dafür, dass aktuelle Malware solche Tests implementiert hat. Trotz dieser Einschränkungen ist *nepenthes* ein leistungsfähiges Werkzeug zum Sammeln von Daten über automatisierte Malware.

2.2 Sensoren

Zum Sammeln der Daten wurde im Rahmen dieser Arbeit eine verteilte Umgebung aus mehreren so genannten *Sensoren* und einem Server aufgebaut. Ein Sensor ist ein System, auf dem *nepenthes* und andere Software läuft, um Angreifer anzulocken. Alle gesammelten Daten werden zentral auf dem Server gespeichert und weiter verarbeitet. Die serverseitigen Komponenten von *BEAN* werden in Kapitel 2.3 näher beschrieben.

Zielsetzung für das Sensorsystem war es, ein gut skalierbares System aufzusetzen, das alle relevanten Daten über Angriffe zur Beurteilung des Gefährdungslevels automatisch sammelt. Zudem sollte erreicht werden, dass das System möglichst nicht selbst durch automatisierte Malware kompromittiert werden kann.

Als Grundlage für das Sensorsystem kommt ein *Ubuntu-Linux* [Ltd07] ohne grafische Oberfläche zum Einsatz, da ein solches System verhältnismäßig wenig Ressourcen benötigt bei gleichzeitig einfacher Installation und Wartung. Zur besseren Handhabung wurde das System als Image für *VMware* [Inc07] aufgesetzt, sodass es in nahezu jeder Systemumgebung ohne großen Aufwand eingerichtet werden kann. Außerdem wurde durch die Kapselung in *VMware* das Risiko minimiert, dass das System durch unbekannte Komponenten auf dem Gastsystem beeinflusst werden kann.

Den Kern des Sensors bildet *nepenthes* in der zum Zeitpunkt der Installation (Juli 2007) neuesten Version im *SVN-Repository* von mwcollect.org [BKWW07]. *nepenthes* wurde

mit 34 Schwachstellenmodulen installiert, die meisten emulieren Schwachstellen von *Microsoft Windows*. Ein Teil der Schwachstellenmodule ist in der Standardinstallation von *nepenthes* enthalten, einige wurden für das InMAS-Projekt hinzugefügt. Eine Übersicht über alle Schwachstellenmodule bietet Tabelle 2.1.

Schwachstellenmodul	Port	emulierte Schwachstelle/Dienst
vuln-apache2058	80	Apache Mod_Rewrite Off-by-one Remote Overflow Exploit
vuln-arcserve	41523	CA BrightStor ARCserve Backup Buffer Overflow
vuln-acrservesql	6070	CA BrightStor ARCserve Backup Agent for SQL
vuln-asn1	445, 80	Microsoft Windows (MS03-007)
vuln-axigen2b	110	Axigen eMail Server 2.0.0b2
vuln-bagle	2745	Backdoor des Bagle Wurm
vuln-brightstor11520	111	CA BrightStor Backup 11.5.2.0
vuln-chimaeraftp	21	WFTPD v3.23 und FreeFTPD bis v1.0.8
vuln-com3tftp	69	3Com TFTP Service v2.0.1
vuln-dameware	6129	DameWare Mini Remote Control Username Remote Overflow
vuln-dcom	135, 445, 1025	Microsoft Windows (MS03-039, MS04-012)
vuln-ftpd	21	freeFTPd 1.0, warFTPd 1.65
vuln-iis	443	Microsoft Windows (MS03-007, MS03-051, MS04-011)
vuln-imail2006	25 und 587	IMail 2006 and 8.x
vuln-kuang2	17300	Backdoor des Kuang2 Wurm
vuln-lsass	445	Microsoft Windows (MS04-011)
vuln-mailenable1x	143	Mail Enable Professional/ Enterprise v1.04-54
vuln-mailenable234	143	Mail Enable Professional/ Enterprise v2.32-4 Buffer Overflow
vuln-ms06040	139	Microsoft Windows
vuln-ms06070	445	Microsoft Windows
vuln-msdtc	1025, 3372	Schwachstellen in MSDTC (MS05-051)
vuln-msmq	2103, 2105, 2107	Microsoft Windows (MS05-017)
vuln-mssql	1434	Microsoft SQL Server (MS02-039)
vuln-mydoom	3127	MyDoom-Wurm
vuln-netbiosname	139	gibt einen angefragten Netbios Hostname zurück

Fortsetzung nächste Seite

Schwachstellenmodul	Port	emulierte Schwachstelle/Dienst
vuln-netdde	139	Microsoft Windows (MS04-031)
vuln-optix	3140	Backdoor des Optix Pro Trojaners
vuln-pnp	445	Schwachstelle in Plug and Play MS Windows (MS05-039)
vuln-sasserftpd	5554, 1023	FTP Schwachstelle im Sasser Virus
vuln-sub7	27347	Backdoor des Sub7 Trojaners
vuln-upnp	5000	Microsoft Windows (MS01-059)
vuln-veritas	10000	VERITAS Backup Exec Remote Agent (VX05-002)
vuln-wftpd323	21	WFTPD server 3.23, Buffer Overflow
vuln-wins	42	Microsoft Windows (MS04-006, MS04-045)

Tabelle 2.1: Emulierte Schwachstellen des Sensors

Um die über die Schwachstellenmodule gesammelten Angriffsdaten in eine Datenbank zu speichern, kommt das Modul *log-surfnet* zum Einsatz. Es wurde gemeinsam von den Entwicklern von *nepenthes* und dem Intrusion Detection System (IDS) *SurfIDS* [bv07] von *surfnet* entwickelt. Das Modul protokolliert Daten über Angriffe auf das *nepenthes*-System umfassend in eine *PostgreSQL*-Datenbank. Dabei arbeitet es asynchron. Dadurch liefert es konsistente Daten, verwirft also keine Ereignisse. Durch die asynchrone Verarbeitung der verschiedenen Ereignisse bietet das Modul auch auf langsamen Maschinen eine gute Performanz.

Eine wichtige Funktionalität, die durch das *log-surfnet Modul* nicht abgedeckt ist, ist ein zentrales Abspeichern der heruntergeladenen Binaries. Damit die Binaries nicht lokal auf den Sensoren gesammelt werden müssen, sondern zentral auf dem Server gespeichert werden, um sie dort direkt weiter zu verarbeiten, kommt das *submit-http Modul* von *nepenthes* zum Einsatz. Es ermöglicht das Senden der Binaries vom Sensor zum Server über einen *HTTP-Request*. Dies ist die einfachste und effizienteste Möglichkeit, alle eingehenden Binaries der Sensoren zentral zu speichern.

Neben *nepenthes* läuft mit *p0f-db* [Kru07] noch ein weiteres Tool zum Sammeln von Daten über Angreifer auf den Sensoren. Dabei handelt es sich um ein so genanntes passives *OS fingerprinting Tool*, das anhand der Datenpakete, die bei Angriffen zwischen Sensor und Angreifer hin- und hergehen, zu ermitteln versucht, unter welchem Betriebssystem der Angreifer arbeitet. *p0f-db* speichert die gesammelten Informationen direkt in die *PostgreSQL*-Datenbank. Auf dem Sensorsystem wurde *p0f-db* geringfügig angepasst, da es in der Originalversion für Angreifer, bei denen kein Betriebssystem ermittelt werden kann, keinen Eintrag in der Datenbank erzeugt. In der angepassten Version wird in diesem Fall *Unknown* gespeichert, um spätere Auswertungen zu erleichtern.

Zusätzlich zu den eben beschriebenen Tools laufen auf den Sensoren noch einige selbst entwickelte Skripte. Diese gewährleisten, dass *nepenthes* und *p0f-db* zu jeder Zeit laufen und bei einem Absturz automatisch wieder gestartet werden. Zudem wird regelmäßig die aktuelle öffentliche IP-Adresse jedes Sensorsystems zum Server übermittelt und dort in die Datenbank gespeichert. Das ist notwendig, damit auch Sensoren, die über eine dynamisch vergebene IP-Adresse an das Internet angebunden sind, erfasst werden können. Somit können zum einen nützliche Informationen gewonnen werden, zum Beispiel darüber, ob mehr Angriffe aus der aktuell eigenen *IP-Range* des Sensors kommen als aus fremden IP-Ranges. Zum anderen kann auf das Sensorsystem so jederzeit, zum Beispiel für Wartungsarbeiten, über SSH zugegriffen werden. Auf dem Sensorsystem läuft der *SSH-Deamon* nicht auf dem Standardport, um durch *nepenthes* nicht kontrollierte Angriffe auf den Sensor zu minimieren.

2.3 Server

In dem entwickelten System gibt es verschiedene serverseitige Komponenten, die entweder verteilt auf verschiedenen Servern laufen können oder gebündelt auf einem leistungsfähigen Server. Die Komponenten teilen sich auf in speichernde Komponenten, auswertende Komponenten und aufbereitende Komponenten. Somit ist eine *Three-Tier-Architecture* umgesetzt, die sich aus einer Datenschicht – den speichernden Komponenten – einer Logikschicht – den auswertenden Komponenenten – und einer Präsentationsschicht, den aufbereitenden Komponenenten zusammensetzt. Durch diese Trennung wird ein hohes Maß an Skalierbarkeit ermöglicht. Da für diese Arbeit ein leistungsfähiger Rechner zur Verfügung stand, liegen im hier aufgesetzten System alle Komponenten zentral auf diesem Rechner.

Speichernde Komponenten sind die Datenbank, in der alle gesammelten Daten und Auswertungen gespeichert werden und die Routine, die die gesammelten Malware Binaries abspeichert. Zu den auswertenden Komponenten gehören Tools zur Auswertung und Weiterverarbeitung der Daten wie zum Beispiel *packerid* oder *hexdump*. Außerdem gehören zu diesen Komponenten die Virenscanner, mit denen die gesammelten Malware Binaries in regelmäßigen Abständen untersucht werden. Die aufbereitende Komponente bildet ein Webinterface, in dem alle gesammelten Rohdaten sowie die entpsrechenden Resultate der Auswertungen angezeigt und sinnvoll aufbereitet werden.

2.3.1 Speichernde Komponenten

Zur Speicherung der Daten kommt eine *PostgreSQL*-Datenbank zum Einsatz. Dies ist dadurch bedingt, dass das verwendete *nepenthes*-Modul *log-surfnet* mit einer asynchronen *PostgreSQL*-Schnittstelle implementiert ist. *PostgreSQL* ist eine *OpenSource* Software und gilt als sicher und performant und ist dadurch für das System gut geeignet [PGDG].

In der Datenbank werden Daten über die Angriffe, über Angreifer und deren Verhalten, über die Malware Binaries sowie Konfigurationsdaten für das System gespeichert.

Tabelle	Herkunft	Tabellentyp	Anzahl Datensätze
attacks	log-surfnet	Angriffe	1.010.310
cws_analyses	BEAN	Binaries	729
details	log-surfnet	Angriffe	488.629
ipdata	BEAN	Angreifer	1.010.920
metrics_values	BEAN	Metriken	396
remote_ips	BEAN	Konfiguration	610
system	p0f	Angreifer	73.995
system_details	p0f	Angreifer	73.995
uniq_binaries	log-surfnet	Binaries	729
virus_scans	BEAN	Binaries	431.719
geoipasn	MaxMind	Konfiguration	129.316
geoipcountry	MaxMind	Konfiguration	96.459
geoipisp	MaxMind	Konfiguration	111.718
graphics	BEAN	Konfiguration	36
login	BEAN	Konfiguration	7
metrics_definition	BEAN	Konfiguration	6
scanners	log-surfnet/BEAN	Konfiguration	4
sensors	log-surfnet/BEAN	Konfiguration	10
severity	log-surfnet	Konfiguration	4
shell_analyser	BEAN	Konfiguration	4

Tabelle 2.2: Datenbanktabellen, die von BEAN verwendet werden

Hinzu kommt eine Tabelle, in der die Ergebnisse der Metriken gespeichert werden. Eine Übersicht über alle Datenbanktabellen bietet Tabelle 2.2. Ob die Tabellen bereits im *log-surfnet*-Modul enthalten sind oder für *BEAN* entwickelt wurden, welche Art Daten in der Tabelle gespeichert werden und wie viele Datensätze in jede Tabelle während des in Kapitel 2.5 beschriebenen Betrachtungszeitraums gespeichert wurden ist ebenfalls dieser Tabelle zu entnehmen.

Konfigurationsdaten sind solche Daten, die zur Funktionsfähigkeit des Systems notwendig sind. In der Tabelle *sensors* werden Informationen über die einzelnen Sensoren hinterlegt. Die zentralen Informationen hierbei sind ein Name zur Identifikation des Sensors und die lokale IP-Adresse des Sensors. Im Falle einer festen öffentlichen IP-Adresse wird auch diese in die Tabelle *sensors* eingetragen. Diese Tabellen sind Bestandteil des *log-surfnet*-Moduls von *nepenthes*.

Viele Internetzugänge sind über eine dynamische öffentliche IP-Adresse mit dem Internet verbunden, die sich häufig ändert. Daher wurde eine weitere Tabelle hinzugefügt, *remote_ips*, in der über ein *PHP*-Skript und eine *SQL Stored Procedure* in der Datenbank die jeweils öffentliche IP-Adresse der Sensoren gespeichert wird.

Die verschiedenen Angriffsarten legt *log-surfnet* in *severity* ab. Es gibt vier verschiedene:

- 0 – Verbindung zum Sensor
- 1 – bösartige Verbindung zum Sensor
- 16 – Versuch, ein Malware Binary herunterzuladen
- 32 – erfolgreicher Download eines Malware Binaries

Neben der Tabelle *remote_ips* wurde in dieser Arbeit die Datenbank um weitere Tabellen für Konfigurationsdaten erweitert. In *scanners* stehen die einzelnen Virenscanner, die eingesetzt werden können mit ihren Befehlen zum Scannen und Updaten und einem Statusflag, das anzeigt, ob der Scanner einsatzfähig ist. Damit das Scanergebnis leichter weiterverarbeitet werden kann, ist zu jedem Scanner zusätzlich ein regulärer Ausdruck hinterlegt, der aus der Ergebnisausgabe des Scanners nur den eigentlichen Befund extrahiert. Die Tabelle *login* speichert die Login-Daten der Benutzer für das Webinterface. In der Tabelle *metrics_definition* werden die Namen der einzelnen Metriken sowie ihre Grenzen für niedrige, mittlere und hohe Gefährdung hinterlegt.

Die Daten über Angriffe und Angreifer unterscheiden sich in *direkte* und *indirekte* Daten. Direkte Angriffsdaten beziehen sich auf den Angriff selbst. Indirekte Angriffsdaten sind Daten, die aus dem Angriff resultieren wie zum Beispiel Informationen über den Angreifer.

Die direkten Angriffsdaten verteilen sich auf die *log-surfnet*-Tabellen *attacks* und *details*. In *attacks* werden die grundlegenden Daten über einen Angriff gespeichert. Dazu gehören der Zeitstempel des Angriffs, die IP-Adresse des Angreifers, auf welchen Sensor der Angriff ging, die Art des Angriffs, von welchem Port der Angriff ausging und auf welchen Port er zielte. Aufgrund der spezifischen Implementierung des *log-surfnet*-Moduls kann es pro Angriff maximal drei Einträge in dieser Tabelle geben, einen zur Verbindung und jeweils einen zum Downloadversuch und zum erfolgreichen Download des Malware Binaries. Identifiziert *nepenthes* den Inhalt der Pakete, die über eine Verbindung geschickt wurden als bösartig, wird kein neuer Eintrag in der Datenbank generiert, sondern der Eintrag zum entsprechenden Verbindungsversuch wird entsprechend aktualisiert, indem die *Severity* auf 1 erhöht wird.

Die Tabelle *details* enthält noch einige weitere Details zu den Angriffen. Dies sind konkret die Schwachstelle, über die der Angriff einging, der Shellcode Handler, den der Angriff ansprach, die URL, von der ein Malware Binary nachgeladen werden soll und bei erfolgreichem Download der *MD5 Hash* des Malware Binaries.

In den Tabellen *uniq_binaries*, *system* und *system_details* stehen die indirekten Angriffsdaten. Wird bei einem Angriff ein Malware Binary zum ersten Mal heruntergeladen, wird dessen MD5 Hash, der Zeitpunkt des Downloads und dessen Dateigröße in die Tabelle *uniq_binaries* gespeichert. Über die Einzigartigkeit eines Binaries entscheidet dabei

dessen MD5 Hash. Diese Tabelle ist ursprünglicher Bestandteil von *log-surfnet*. Für diese Arbeit wurden ihr Felder für die Dateigröße und das Datum des Erstauftretens der Malware hinzugefügt.

Zum Abspeichern der Ergebnisse der Virenscans wurde die Tabelle *virus_scans* angelegt. Sie enthält die Befunde einzelner Malware Binaries. Es wird hierbei der Virenscanner, mit dem geprüft wurde, der Zeitpunkt des Scans, der MD5 Hash des geprüften Binaries und natürlich der Befund erfasst.

Die Ergebnisse, die *p0f-db* liefert, werden in *system* und *system_details* gespeichert. Dort werden das ermittelte Betriebssystem, der Zeitstempel des ersten und letzten Auftretens dieser Angreifer-IP-Adresse und Daten über die Internetverbindung des Angreifersystems hinterlegt.

Damit die Malware Binaries weiterverarbeitet werden können, werden sie von den Sensorsystemen über das *nepenthes*-Modul *submit-http* über einen HTTP-Request zentral auf der Festplatte des Servers abgespeichert. Dazu rufen die Sensoren beim Download eines Binaries ein *PHP*-Skript auf dem Server auf. Dieses Skript speichert das Binary aus dem Request in eine vorgegebene Verzeichnisstruktur, wobei die ersten zwei Stellen des MD5 Hashs des Binaries als Unterordner verwendet werden. Dies ermöglicht die effiziente Verarbeitung von großen Datenmengen.

Um den historischen Verlauf der Metriken darstellen zu können, wird für jede Metrik täglich ihr aktueller Wert berechnet und in der Tabelle *metrics_values* gespeichert. Eine Berechnung der historischen Daten bei Bedarf würde sonst ab einem Zeitraum von einigen Tagen eine inakzeptabel lange Laufzeit nach sich ziehen.

Zur effizienten Verarbeitung der grafischen Auswertungen wurde eine Tabelle *graphics* angelegt, in der Daten über die Grafiken, die im Webinterface abrufbar sein sollen, gespeichert werden. Diese Tabelle enthält neben dem Namen der Grafik eine Beschreibung dessen, was sie darstellt und einige Parameter beispielsweise ob für die Grafik eine Filterung nach Sensor möglich ist.

2.3.2 Auswertende Komponenten

Zusätzlich zu der Datenbank laufen auf dem Server einige Tools zum Auswerten der gesammelten Daten. *objdump*, *hexdump* und *packerid* sind Werkzeuge zur technischen Analyse der gesammelten Malware Binaries. Ihre Funktionalität wird in Kapitel 3.2.3 näher beschrieben.

Zudem sind auf dem Server vier Virenscanner installiert, auf die in Kapitel 3.2.1 detailliert eingegangen wird. Alle heruntergeladenen Malware Binaries, die sich in der entsprechenden Verzeichnisstruktur auf der Festplatte des Servers befinden, werden in regelmäßigen Abständen mit diesen Virenscannern überprüft. Die Ergebnisse dieser Untersuchungen werden in die Tabelle *virus_scans* in die Datenbank gespeichert. Implementiert ist dies in einem *Python*-Skript, das alle vier Stunden per *Cronjob* auf dem Server ausgeführt wird. Es überprüft jedes Mal das gesamte Binary-Verzeichnis.

Das *Python*-Skript bietet drei verschiedene Scanoptionen. Zum einen den vollen Scan eines Verzeichnisses, wobei ausnahmslos alle Binaries im entsprechenden Verzeichnis überprüft werden und der Befund unabhängig von vorhergehenden Befunden in die Datenbank gespeichert wird. Wird ein Binary von einem Scanner nicht als Malware eingestuft, wird kein Eintrag in die Datenbank erzeugt. Daneben gibt es die Option, nur Binaries, zu denen es noch kein Ergebnis in der Datenbank gibt, zu überprüfen – also entweder neue Binaries, die noch nicht überprüft wurden oder alte, die bisher nicht als Malware eingestuft wurden. Es gibt zusätzlich die Möglichkeit, ein einzelnes Binary mit dem Skript zu überprüfen und den Befund dazu in die Datenbank zu speichern. Bei allen Optionen werden jeweils alle einsatzbereiten Scanner verwendet.

2.3.3 Aufbereitende Komponente

Ein Bestandteil der vorliegenden Arbeit war die Entwicklung eines Webinterfaces zur Darstellung und Aufbereitung der gesammelten Daten. Das Webinterface ist in *PHP 5* implementiert. Es bietet Informationen über die Sensoren, über Angriffe, Angreifer und über gesammelte Malware Binaries. Alle Informationen können nur über einen gültigen Benutzeraccount abgerufen werden, sodass eine angemessene Sicherheit für die Daten gewährleistet ist.

In einer Übersicht über alle eingegangenen Angriffe werden die zentralen Informationen über die Angriffe bereitgestellt wie beispielsweise die IP-Adresse des Angreifers und die lokale IP-Adresse des angegriffenen Sensors. Des Weiteren gibt es Details zu einem bestimmten Angriff auf einer speziellen Detailseite. Dies ist beispielsweise eine URL, von der im Rahmen dieses Angriffs ein Malware Binary angeboten wurde.

Ebenso gibt es eine Übersicht über alle heruntergeladenen Malware Binaries wie in Abbildung 2.2 dargestellt sowie eine Seite mit Details zu einem bestimmten Malware Binary. Dort werden Informationen wie die Dateigröße des Binaries und die Befunde der Virenscanner für dieses Binary angezeigt. Zudem gibt es die Möglichkeit, sich das Binary vom Server herunterzuladen, um weitere Analysen damit durchzuführen. Sowohl für die Angriffsübersicht als auch für die Übersicht der heruntergeladenen Binaries stehen entsprechende Filterfunktionen zur Zusammenstellung bestimmter Datensätze zur Verfügung.

Da das System mit verschiedenen Sensoren aufgesetzt wurde, bietet das Webinterface eine Übersicht über alle in der Datenbank registrierten Sensoren. Zu jedem Sensor werden seine lokale IP-Adresse und seine aktuell registrierte öffentliche IP-Adresse mit dem Zeitpunkt der Registrierung angezeigt. Außerdem wird mit Hilfe von *netstat* überprüft, ob die Sensoren aktuell mit der Datenbank verbunden sind. Dies ermöglicht einen schnellen Überblick über die aktuelle Funktion der Sensoren.

Neben den reinen Datensammlungen bietet das Webinterface eine Reihe von Auswertungen in Form von Grafiken zum Beispiel über die Anzahl der heruntergeladenen Malware Binaries pro Tag. Die Ergebnisse dieser Auswertungen für die Datengrundlage dieser

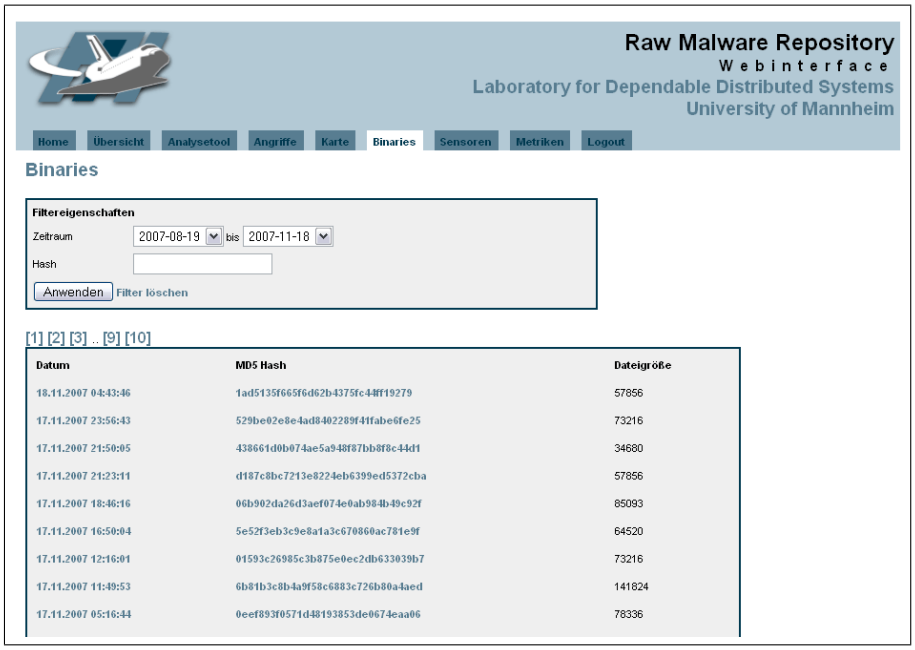


Abbildung 2.2: Liste von heruntergeladenen Binaries aus dem *BEAN*-Webinterface

Arbeit (siehe Kapitel 2.5) werden in Kapitel 4 interpretiert. Für die grafischen Auswertungen gibt es eine Reihe von Filterfunktionen, die es dem Benutzer beispielsweise erlauben, den Zeitraum der Auswertung einzuschränken.

Zudem bietet das Webinterface die Möglichkeit, die in Kapitel 5 beschriebenen Metriken abzurufen. Es werden die jeweils tagesaktuellen Daten aller Einzelmetriken und der Gesamtmetrik angezeigt. Zusätzlich dazu ist es möglich, die historische Entwicklung der einzelnen Metriken grafisch darstellen zu lassen.

Das gesamte Webinterface steht in englischer und in deutscher Sprache zur Verfügung, damit ein möglichst großer Personenkreis mit den gesammelten Informationen und den entwickelten Metriken arbeiten kann. Es ist sehr leicht möglich über eine Sprachendatei weitere Sprachen zu integrieren. Eine detaillierte Anleitung dazu findet sich in der Dokumentation des Webinterface.

2.4 Beispiel eines Angriffs

Zum besseren Verständnis der Abläufe im vorgestellten System wird in diesem Kapitel ein Angriff beispielhaft durchgespielt. Abbildung 2.3 zeigt schematisch die beschriebenen Abläufe. Die grau unterlegten Elemente stellen dabei Komponenten von *nepenthes* dar. Beschrieben wird ein realer Angriff, der auf einer der installierten Sensoren des in Kapitel 2.5 beschriebenen Datensets einging. Der Angriff war innerhalb weniger Sekunden

erfolgt und es wurde ein Malware Binary heruntergeladen, das die Virenscanner als die ursprüngliche Form des bekannten Wurms *W32.Blaster-A* identifizierten. Aus Datenschutzgründen wurden die Daten von Angreifer und Opfer anonymisiert. Zum besseren Verständnis sind die einzelnen Schritte sowohl im Text als auch in der Abbildung nummeriert.

Zu Beginn jeden Angriffs muss der Angreifer eine Verbindung zum Opfer auf dem Port aufbauen, auf dem der Dienst lauscht, dessen Schwachstelle ausgenutzt werden soll. Im Beispiel *W32.Blaster* ist dies der TCP-Port 135, der auf die *DCOM RPC* Schwachstelle zielt. Zum Aufbau der Verbindung (1) schickt der Angreifer ein TCP-Paket mit SYN-Flag an das Opfer. Dieses antwortet, sofern der angefragte Port offen ist, mit einem TCP-Paket mit SYN/ACK-Flag und bekommt vom Angreifer im Anschluss ein Paket mit ACK-Flag. Danach ist die Verbindung aufgebaut. Dieses Verhalten nennt man *Three-Ways-Handshake*.

Der erfolgreiche Verbindungsaufbau auf Port 135 löst in *nepenthes* ein entsprechende Event aus. Daraufhin wird über das *log-surfnet Modul* die *Stored Procedure surfnet_attack_add* (2a) aufgerufen, die einen Eintrag in der Tabelle *attacks* mit der Severity 0 – also möglicherweise bösartiger Angriff – erzeugt. Als nächstes versucht *nepenthes*, die Verbindung einem Schwachstellenmodul zuzuordnen. Gelingt dies, ruft *log-surfnet* die Stored Procedure *surfnet_detail_add* (2b) auf, die in der Datenbanktabelle *details* einen Eintrag mit dem Dialog-Namen des Schwachstellenmoduls erzeugt, in diesem Fall *DCOMDialogue*. Zudem erneuert *log-surfnet* durch Aufruf der Stored Procedure *surfnet_attack_update_severity* den vorherigen Eintrag in der Tabelle *attacks* auf Severity 1, also definitiv bösartige Verbindung.

Nachdem *W32.Blaster* erfolgreich eine Verbindung auf den TCP-Port 135 zum Sensor aufgebaut hat, öffnet er darüber eine Shell auf dem Sensor (3), die auf dem TCP-Port 4444 lauscht. Daraufhin schickt er die Payload an den Sensor, der veranlassen soll, dass der Wurm per *Trivial File Transfer Protocol (TFTP)* heruntergeladen und installiert wird. Bis zu diesem Zeitpunkt hat *nepenthes* dem Angreifer vorgetäuscht, das System wäre angreifbar und hat entsprechend reagiert. Der Shellcode in der Payload wird nun jedoch nicht ausgeführt, sondern intern von *nepenthes* weiterverarbeitet. Dazu wird die Payload den Shellcode Handlern übergeben (4), die sie versuchen zu analysieren.

Im konkreten Fall des *W32.Blaster* erzielt der Shellcode Handler *bindshell::adenau*, der die *Windows Emulation* erkennt, einen Treffer, da *W32.Blaster* ein Kommando schickt, das auf dem Opfer ausgeführt werden soll. Die Analyse des Shellcodes löst ein *nepenthes*-Event aus, auf das *log-surfnet* mit einem erneuten Aufruf von *surfnet_detail_add* zum Eintragen des benutzten Shellcode Handlers in die Datenbank reagiert.

Kann der Shellcode Handler aus dem Shellcode eine URL zum Download des Malware Binaries extrahieren (5), gibt er diese an das Download-Modul für das entsprechende Protokoll, also in diesem Fall *download-tftp*, weiter. Daraus resultiert zudem ein *nepenthes*-Event, das zur Folge hat, dass *log-surfnet* die Stored Procedure *surfnet_detail_add_offer* aufruft, die je einen Eintrag in den Tabellen *attacks* und *details* generiert. In *attacks* wird ein neuer Eintrag zu dem Angriff mit Severity 16 – also dem Downloadangebot – ange-

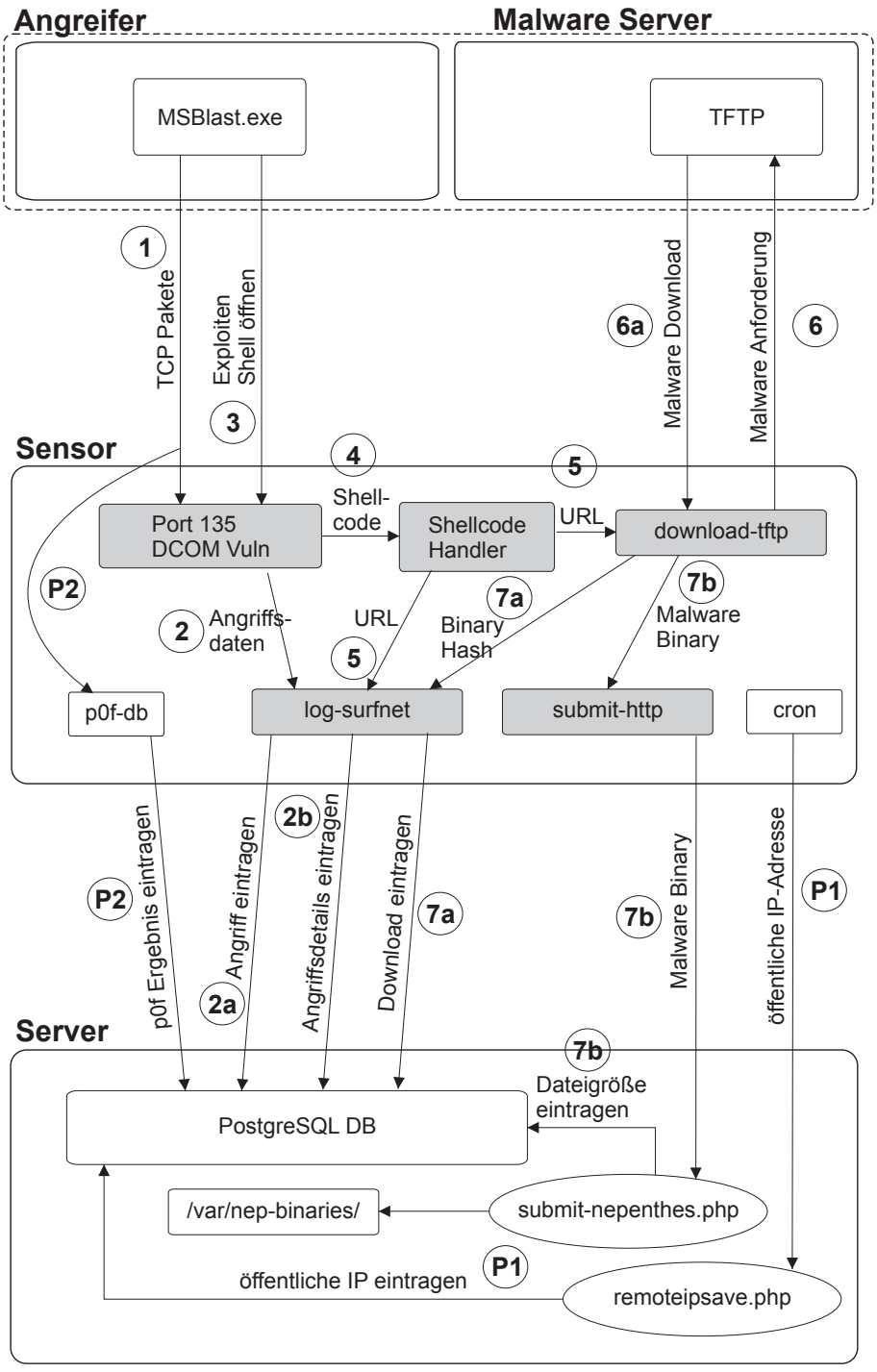


Abbildung 2.3: Angriffsablauf W32.Blaster

legt, in *details* wird mit der eben generierten *attackid* die Download-URL eingetragen, im konkreten Fall `tftp://1.2.3.4/msblast.exe`.

Der Download Handler *download-tftp* versucht nun, das Malware Binary von der angegebenen URL herunterzuladen (6). Das Malware Binary kann hierbei sowohl auf dem Angreifersystem als auch auf einem weiteren System liegen. Ist der Download erfolgreich (6a), wird das Binary lokal auf der Festplatte des Sensors zwischengespeichert. Das durch den erfolgreichen Download ausgelöste *nepenthes*-Event stößt zwei parallele Prozesse an.

Zum einen (7a) ruft *log-surfnet* die Stored Procedure *surfnet_add_detail_download* auf. Diese generiert als erstes einen Eintrag in der Tabelle *attacks* mit Severity 32 – also dem erfolgreichen Download. Danach prüft diese Prozedur, ob der MD5 Hash des heruntergeladenen Binaries bereits in der Tabelle *uniq_binaries* vorhanden ist und falls nicht, erstellt sie den entsprechenden Eintrag. Zuletzt wird in dieser Prozedur in die Tabelle *details* zum Eintrag in *attacks* die Download-URL, `tftp://1.2.3.4/msblast.exe` und der MD5-Hash des heruntergeladenen Malware Binaries (`5ae700c1dffb00cef492844a4db6cd69`) eingetragen.

Parallel dazu wird durch das Event das Modul *submit-http* angestoßen, das das Malware Binary auf den Server speichern soll. Dazu ruft das Modul (7b) eine *PHP*-Seite auf dem Server auf und schickt per HTTP-Request das Binary mit. Das Binary wird dann in der im Skript definierten Verzeichnisstruktur abgelegt. In diesem Fall kommt das Binary in das Verzeichnis `/var/nep-binaries/5/a/`, da die Binaries nach den ersten beiden Stellen ihres MD5 Hashs sortiert werden.

Damit sind die von *nepenthes* ausgeführten Prozesse und damit auch der Angriff beendet. Parallel zum eigentlichen Angriff laufen auf Sensor und Server noch weitere Prozesse ab, um Informationen über Angriff und Angreifer zu sammeln. Der Sensor führt alle fünf Minuten ein *PHP*-Skript mit seiner privaten IP-Adresse und seinem Sensornamen als Parameter auf dem Server aus. Das Skript speichert (P1) die öffentliche IP-Adresse des Sensors in die Tabelle *remote_ips*, falls dort von den letzten 24 Stunden von diesem Sensor nicht bereits ein Eintrag mit dieser öffentlichen IP-Adresse existiert.

Die eingehenden TCP-Pakete werden von *p0f-db* untersucht, um das Betriebssystem und die Internetanbindung des Angreifers zu ermitteln. Das Ergebnis dieser Analyse speichert *p0f-db* (P2) in die Tabelle *system* und *system_details*. Zu dem beschriebenen Angreifer hat *p0f-db* als Betriebssystem `Windows XP SP1+, 2000 SP3` ermitteln können. Außerdem konnte das Tool ermitteln, dass der Angreifer direkt über DSL angebunden war und dass er 13 Hops, also Zwischenstationen wie zum Beispiel Router, vom Sensor entfernt war.

Ein dritter paralleler Prozess ist die Übermittlung des Binaries an die *CWSandbox*. Dieser geschieht ebenfalls über ein *PHP*-Skript. Da die *CWSandbox* im für diese Arbeit genutzten Setup auf einem externen Server lief, ist dieser Prozess nicht in der Abbildung dargestellt.

Sensor	ISP	24h-Disconnect	feste/dynamische IP	Standort
1	Arcor	ja	dynamisch	Mannheim
2	Uni Mannheim	nein	fest	Mannheim
3	Telekom	ja	dynamisch	Speyer
4	1&1	ja	dynamisch	Mannheim
5	Telekom	nein	fest	Speyer
6	QSC	nein	dynamisch	Mannheim
7	Telekom	ja	dynamisch	Grünstadt
8	KabelBW	ja	dynamisch	Mannheim
9	Telekom	ja	dynamisch	Böblingen
10	Arcor	ja	dynamisch	Mannheim

Tabelle 2.3: Sensoren im Datenset

Damit sind alle Daten, die mit Hilfe von *BEAN* über den Angriff erfasst werden können gespeichert. Sie werden nun mit Hilfe der in Kapitel 3 beschriebenen Methoden analysiert. Sowohl die Rohdaten als auch die Analyseergebnisse können im Webinterface abgerufen werden.

2.5 Datengrundlage

Über einen Zeitraum von insgesamt zehn Wochen liefen eine Reihe von Sensoren in verschiedenen Umgebungen. Die dadurch gesammelten Daten dienten dieser Arbeit als Datengrundlage. Um erste Aussagen über die Sicherheitslage in Deutschland in Bezug auf automatisierte Malware machen zu können, waren die Sensoren über möglichst unterschiedliche *Internet Service Provider* (ISP) mit dem Internet verbunden.

Einige der Sensoren konnten nicht über den gesamten Betrachtungszeitraum eingesetzt werden. Sofern die Einsatzdauer vom Betrachtungszeitraum abweicht, wird die konkrete Abweichung bei der Beschreibung des jeweiligen Sensors entsprechend spezifiziert. Die Daten sind trotz dieser Unterschiede konsistent, da alle Sensoren in ihrem jeweiligen Einsatzzeitraum 24 Stunden am Tag liefen.

Das Datenset setzt sich aus insgesamt zehn Sensoren zusammen, die ab dem 20. August 2007 bis zum 29. Oktober 2007 liefen. Alle Sensoren waren mit der gleichen Software ausgestattet wie in Kapitel 2.2 beschrieben. Dadurch sind die Daten vergleichbar. Durch Systemabstürze oder Providerausfälle fielen einige Sensoren für kurze Zeitspannen von wenigen Stunden aus, was die Vergleichbarkeit nicht vermindert.

Im Folgenden werden die einzelnen Sensoren näher beschrieben. Dabei wird der ISP und eventuelle Besonderheiten des Sensors genannt. Tabelle 2.3 zeigt eine Übersicht über die Sensoren des Datensets, ihren ISP, ihren Standort und Informationen zur Vergabe ihrer öffentlichen IP-Adresse.

Sensor 1 war über den ISP *Arcor* an das Internet angeschlossen und bezog seine öffentliche IP-Adresse dynamisch. Die IP-Adresse änderte sich spätestens alle 24 Stunden, da der Provider danach automatisch die Verbindung kurz trennt und neu aufbaut.

Sensor 2 lief im Netz der Universität Mannheim und hatte somit eine feste IP-Adresse aus dem IP-Bereich der Universität Mannheim.

Sensor 3 wurde über die *Telekom* mit dem Internet verbunden und bezog seine IP-Adresse dynamisch. Ebenso wie bei *Arcor* trennt die *Telekom* die Verbindung zum Internet alle 24 Stunden und vergibt eine neue IP-Adresse.

Sensor 4 lief im Netz von *1und1* und bezog seine IP-Adresse dynamisch mit einer Trennung alle 24 Stunden. Eine Besonderheit dieses Sensors war, dass der in diesem lokalen Netz installierte Router einige der relevanten Ports, wie den TCP-Port 135, sperrt. Die eingesetzte Konfiguration des Routers lässt eine Abschaltung dieser Sperrung nicht zu. Details dazu liefert Kapitel 4.1.

Sensor 5 war über die *Telekom* mit dem Internet verbunden, jedoch über eine feste IP-Adresse. Der Sensor stand in einem mittelständischen Unternehmen, jedoch liefen auf der entsprechenden IP-Adresse keine weiteren Dienste neben dem Sensor.

Sensor 6 hatte *QSC* als ISP. Die Besonderheit dieses Providers ist es, dass es keine automatische Verbindungstrennung nach 24 Stunden gibt. Somit wechselte der Sensor seine öffentliche IP-Adresse über den Sammelzeitraum nie. Dieser Sensor lief ab dem 23. August 2007 bis zum Ende des Betrachtungszeitraums und war etwa sieben Tage nicht erreichbar.

Sensor 7 lief bei der *Telekom* über einen herkömmlichen DSL-Privatanschluss, bezog somit also seine IP-Adresse dynamisch und wurde nach 24 Stunden automatisch neu mit dem Internet verbunden. Dieser Sensor lief von Beginn des Betrachtungszeitraums bis zum 11. Oktober 2007 und war etwa zehn Tage nicht erreichbar.

Sensor 8 war der einzige Sensor im Datenset, der über einen Kabelnetzbetreiber angebunden war. Er bezog sein Internet von *KabelBW*. Dieser Provider filtert von vornherein in seinem Netz die relevanten Ports, sodass kaum ein Angriff auf diesem Sensor ankam. Hierzu finden sich detaillierte Informationen in Kapitel 4.1.

Sensor 9 war über die *Telekom* an das Internet angeschlossen und bezog seine öffentliche IP-Adresse dynamisch. Er lief vom 2. September 2007 bis zum Ende des Betrachtungszeitraums. Ihm standen aufgrund technischer Einschränkungen nur die Ports 21, 25, 42, 69, 80, 110, 111, 135, 139, 143, 220, 443, 445 und 465 zur Verfügung.

Sensor 10 lief bei *Arcor* und bezog seine öffentliche IP-Adresse dynamisch. Er lief vom 6. Oktober 2007 bis zum Ende des Betrachtungszeitraums.

2.6 Zusammenfassung

In diesem Kapitel wurde die Infrastruktur beschrieben, die für *BEAN* entwickelt wurde, um Rohdaten über automatisierte Malware zu sammeln. Es wurde ein Sensorsystem entwickelt, das die Zielsetzung der Skalierbarkeit und der vollen Automatisierung des Sammelns von automatisierter Malware erfüllt. Die Grundlage des Sensorsystems ist der Honeypot *nepenthes*, der auf dem Sensorsystem Schwachstellen emuliert. Die eingehenden Angriffe auf diesen Schwachstellen werden registriert und es werden verschiedenste Daten über Angriffe, Angreifer und Malware Binaries gespeichert.

Das Sensorsystem wurde so entwickelt, dass es gut skalierbar ist und so auf vielen verschiedenen Rechnern installiert werden kann. Damit die gesammelten Daten zentral verfügbar sind, wurde eine Serverumgebung entwickelt, die ebenso dem Anspruch der Skalierbarkeit gerecht wird. Auf dieser Serverumgebung läuft ein Datenbanksystem, um die Daten aller registrierten Sensorsysteme zu speichern. Zudem laufen in der Serverumgebung Werkzeuge zur Analyse der eingehenden Daten, wie beispielsweise die *CWSandbox* oder verschiedene Virenscanner.

Um die Daten und deren Auswertungen zentral abrufbar zu machen, wurde ein Webinterface für das *BEAN*-System implementiert. Es enthält neben Darstellungen der Rohdaten grafische Darstellungen der Analyseergebnisse sowohl von aktuellen als auch von historischen Datensätzen. Die Berechnung der grafischen Darstellungen erfolgt voll automatisiert. Es bietet dem Benutzer zahlreiche Möglichkeiten der Filterung der Daten und Auswertungen nach beispielsweise zeitlichen Kriterien.

Über zehn Wochen – zwischen August und Oktober 2007 – wurde mit Hilfe von zehn Sensoren eine Datengrundlage aufgebaut. Dazu wurde das Sensorsystem auf verschiedenen Systemen installiert. Die einzelnen Sensoren waren über verschiedene ISP mit dem Internet verbunden, um eventuelle Unterschiede erkennen zu können. Die Daten in dieser Datengrundlage werden mit allen im nächsten Kapitel beschriebenen Analysemethoden untersucht und die Ergebnisse in Kapitel 4 interpretiert.

3 Analysemethoden

Zur Einschätzung von Gefahren durch Malware im Internet reicht das bloße Sammeln von Rohdaten nicht aus, erst die Analyse auf verschiedene Weise bringt die Möglichkeit, die Daten zu bewerten. Daher wurden für *BEAN* verschiedene Methoden zur Analyse der Rohdaten eingesetzt. Dazu wurden eigene Methodiken implementiert oder bereits existierende Tools eingesetzt. Ebenso wie für das Sammeln der Rohdaten war die Zielsetzung auch für die Analysen der Daten, dass sie voll automatisiert zur Verfügung stehen.

Im folgenden Kapitel wird die generelle Struktur der Analysen beschrieben. Dazu wird zunächst ein Überblick über die verschiedenen Dimensionen der eingesetzten Analysemethoden gegeben. Dabei wird außerdem auf die technische Integration der verschiedenen Methoden eingegangen.

Im nächsten Schritt werden dann die einzelnen Werkzeuge beschrieben, die zur Analyse der Daten zum Einsatz kommen. Hier wird außerdem konkret erläutert, wie die Werkzeuge eingesetzt werden und wie die technische Umsetzung der angestrebten Automatisierung geschieht. Der letzte Abschnitt gibt einen Überblick über die statistischen Auswertungen der Daten. Abschließend wird beschrieben wie die Automatisierung der grafischen Darstellung der Ergebnisse erreicht wurde.

3.1 Struktur

Es gibt verschiedene Klassen von Analysen, die aus der Art der mit *BEAN* gesammelten Daten resultieren. Zum einen ist dies die Klasse der Analysen auf den Malware Binaries, zum anderen die der Analysen über die Daten über Angriffe und über Angreifer. Für jede dieser Klassen wurden in *BEAN* verschiedene Analysemethoden verwendet.

Aufgrund der Ähnlichkeit der Struktur der Rohdaten über Angriffe und Angreifer wurden für beide Datentypen ähnliche Analysemethoden eingesetzt. Sie bestehen hauptsächlich aus statistischen Auswertungen, mit deren Hilfe beispielsweise nicht intuitive Zusammenhänge zwischen verschiedenen Informationen erkannt werden können. In Kombination mit den Erkenntnissen über ein Malware Binary können mit Hilfe der Angriffsdaten genauere Aussagen zur Gefährlichkeit des Angriffs getroffen werden. Die Daten über Angreifer bieten die Möglichkeit, genauer zu spezifizieren, welche Gefährdung von automatisierter Malware ausgeht.

Die Malware Binaries können im Hinblick auf die Schadfunktionen oder die Verbreitungsroutine, die sie enthalten untersucht werden. Zum einen kann dies statisch über

eine Codeanalyse beispielsweise mit Hilfe eines Virenscannerns geschehen. Zum anderen sind dynamische Verhaltensanalysen der Malware möglich. Das bedeutet, die Malware Binaries werden im Hinblick darauf analysiert, wie sie sich auf einem System verhalten, um es zu kompromittieren. Diese Methoden sind um einiges komplexer und es sind spezifischere Tools notwendig als bei der Analyse der Angriffs- und Angreiferdaten. Die Informationen, die aus diesen Analysen gewonnen werden können, sind ein wichtiger Bestandteil zur Einschätzung der Gefahr, die von der mit *BEAN* gesammelten Malware ausgeht.

Die technische Struktur der Analysen ist an die Struktur zum Sammeln der Rohdaten angelehnt. Der Ablauf der Analysen ist nahezu immer gleich. Es werden zuerst die benötigten Rohdaten aus der *PostgreSQL*-Datenbank oder aus dem Dateisystem abgerufen, mit denen dann die Analysen durchgeführt werden. Im Anschluss werden die Analyseergebnisse wiederum in eine Datenbank gespeichert oder direkt als Grafik dargestellt.

Ein zentraler Aspekt bei der Beurteilung von Datenanalysen ist die Aufbereitung der Ergebnisse. Bei *BEAN* wurden zu jeder Analyse geeignete Darstellungsformen, beispielsweise grafische oder tabellarische, herausgearbeitet, die automatisiert generiert werden und in das *BEAN*-Webinterface eingebunden sind.

Im Folgenden werden zunächst die Tools vorgestellt, mit deren Hilfe zusätzliche Informationen über die gesammelten Malware Binaries gewonnen werden. Danach wird erläutert, wie die Analyse der daraufhin zur Verfügung stehenden Daten erfolgt. Dabei wird auch darauf eingegangen wie die Daten angemessen dargestellt werden können und wie sowohl ihre Analyse als auch ihre Darstellung automatisiert werden kann.

3.2 Werkzeuge

Bei automatisch ablaufenden Angriffen ist ein zentraler Bestandteil das nachgeladene Malware Binary, das die eigentlichen Schadfunktionen enthält. Daher ist zur Einschätzung des Gefährdungslevels durch solche automatisierte Malware die Analyse der Binaries von großem Interesse. Dabei gibt es verschiedene Aspekte, die mit Hilfe von verschiedenen Tools untersucht werden können. Zum einen ist von Interesse, was für eine Art Schadfunktion das Binary enthält. Um dies zu erkennen, werden die Binaries mit verschiedenen Virenscannern untersucht. Zum anderen kann das Verhalten der Malware auf dem System, das kompromittiert werden soll, untersucht werden. Dazu wird das Binary in einer kontrollierten Umgebung ausgeführt.

Neben diesen beiden Hauptanalyseblöcken zu Art und Verhalten der Malware gibt es weitere Aspekte, die *BEAN* analysiert. Einige dieser Untersuchungen dienen in erster Linie dazu, technische Eigenschaften über die Malware Binaries zu erhalten, beispielsweise zur Entwicklung von effektiven Gegenmaßnahmen. In den folgenden Abschnitten werden die verschiedenen Tools, die zur Analyse der Malware Binaries in *BEAN* zum Einsatz kommen, vorgestellt.

3.2.1 Virenscanner

Heutzutage sollte jeder Internetnutzer seinen Rechner durch einen der zahlreichen auf dem Markt angebotenen Antivirenlösungen schützen. Um Malware ihrer Gefährlichkeit nach einzuordnen ist es daher von zentraler Bedeutung, ob verbreitete Virenscanner die Malware als solche erkennen. Daher ist eine Analyse der Malware Binaries mit möglichst vielen verschiedenen Virenscannern von großem Interesse. Virenscanner untersuchen Dateien auf bekannte oder verdächtige Muster – so genannte Signaturen – die darauf hindeuten, dass es sich bei der Datei um Malware handelt. Anhand entdeckter Signaturen ermittelt der Scanner dann die Art der Malware. Somit bietet die Analyse mit Antivirenlösungen neben der bloßen Erkennung als Malware zusätzlich die Möglichkeit, die gesammelte Malware nach ihrer Art zu klassifizieren. Von unterschiedlichen Arten von Malware gehen in der Regel auch unterschiedliche Gefahren aus.

BEAN ermöglicht die Untersuchung der eingehenden Malware Binaries mit einer Reihe verschiedener Antivirenlösungen. Dabei gibt es zwei verschiedene Stufen von Analysen. Zum einen stehen lokale Virenscanner zur Verfügung, mit denen die Malware Binaries in regelmäßigen Abständen untersucht werden. Zum anderen steht mit *VirusTotal* ein externer Dienst zur Verfügung, der jedes Malware Binary mit einer Vielzahl von aktuellen *Virenscan-Engines* analysiert. Da lokal aus technischen Gründen nur vier Scanner zum Einsatz kommen, bietet *VirusTotal* eine sinnvolle Ergänzung der Analysen. Im Folgenden wird auf die konkrete Umsetzung beider Analysemöglichkeiten eingegangen und ihre Vor- und Nachteile erläutert.

Lokale Virenscans

Es stehen lokal vier Antivirenlösungen zur Verfügung – *Antivir* [Gmb07a], *Norman Antivirus* [Nor07], *F-Prot* [INT07] und *ClamAV* [Sou07]. Alle Lösungen sind auf dem Server als Kommandozeilenscanner installiert und werden lediglich bei Bedarf gestartet, damit sich die einzelnen Engines nicht gegenseitig stören. Die derzeitige Konfiguration sieht vor, dass zu jeder Zeit jeweils nur ein Scanner läuft, um Kollisionen zu verhindern. Alternativ ist auch eine Lösung möglich, in der jeder Scanner in einer eigenen virtuellen Maschine installiert wird. Dies hat den Vorteil, dass das System leichter skalierbar ist, da zur Auslagerung einzelner Virenscanner auf weitere Server nur die *VMware* auf dem neuen System gestartet werden muss.

Über ein *Python*-Skript wird die Untersuchung der vorhandenen Malware Binaries gesteuert. Es bietet drei verschiedene Optionen, die Untersuchung eines einzelnen Binaries, der Binaries, die bisher noch nicht erkannt wurden oder aller Binaries. Detaillierte Informationen zu den Optionen liefert Kapitel 2.3.2. In der Standardeinstellung von *BEAN* werden alle vier Stunden alle an der vorgesehenen Stelle im Dateisystem vorhandenen Binaries analysiert. Dazu werden zunächst alle Virenscanner aktualisiert, sodass sie mit aktuellen Virensignaturen arbeiten. Daraufhin werden nacheinander mit jedem Scanner alle Binaries untersucht und das aufbereitete Ergebnis in die Datenbank gespeichert.

Die Ergebnisse werden so aufbereitet, dass nur der Name des Befundes, also beispielsweise *WORM/RBot* gespeichert wird und nicht die gesamte Scanner-Ausgabe. Zu jedem untersuchten Binary wird für jeden Scanner, der das Binary als Malware erkannt hat, ein Eintrag in der Datenbank generiert, in dem der MD5-Hash des Binaries, der Scanner, das Datum der Untersuchung und der Befund gespeichert werden. Somit ist es auch möglich, einen Ergebnisverlauf aufzuzeigen, falls mit aktualisierten Virensignaturen ein Scanner andere (genauere) Ergebnisse liefert. Zudem wird damit protokolliert, wieviel Zeit zwischen dem ersten Auftreten des Binaries und dem ersten Erkennen durch die Virenscanner liegt. Auswertungen auf diesen Daten werden in Kapitel 4.3.3 beschrieben.

Eine Hinzunahme von weiteren Scannern ist sehr einfach möglich, da für alle Scanner die Befehle zum Starten und Aktualisieren sowie der reguläre Ausdruck zum Auswerten der Ergebnisse in der Datenbank gespeichert sind. Somit müssen für einen weiteren Scanner nach der Installation lediglich die entsprechenden Befehle in der Datenbank hinzugefügt werden. Die lokale Untersuchung der Binaries hat gegenüber externen Lösungen wie *VirusTotal* den Vorteil, dass keine langen Wege zwischen dem Binary und den Scannern liegen und die Untersuchung somit verhältnismäßig schnell prozessiert werden kann. Hinzu kommt, dass lokal genügend Ressourcen zur Verfügung standen, um alle Binaries in regelmäßigen Abständen erneut zu untersuchen und somit die Analyse von Verlaufsdaten möglich ist.

VirusTotal

Als Ergänzung zu den lokalen Antivirenlösungen werden über die Anbindung an die CWSandbox (siehe Kapitel 3.2.2) die von *BEAN* gesammelten Binaries zusätzlich von *Virus Total* als externem Dienst einmalig untersucht. *VirusTotal* untersucht jedes übermittelte Binary mit einer ganzen Reihe – derzeit 32 – aktuellen Antivirenlösungen. Dies bietet die Möglichkeit, weitaus breitere Ergebnsise zu bekommen als durch die begrenzte Zahl an lokalen Lösungen, die in *BEAN* laufen.

Aufgrund der begrenzten Ressourcen von *VirusTotal* wird jedes Binary dort lediglich einmal untersucht, wodurch keine Verlaufsdaten zur Verfügung stehen. Zudem resultiert aus den begrenzten Ressourcen häufig eine Verzögerung zwischen Eintreffen des Binaries, der Untersuchung und der Übermittlung des Ergebnisses. Somit ist der exakte Zeitpunkt der Untersuchung des Binaries nicht bekannt.

Die Binaries werden zu *.zip*-Archiven gepackt und bei entsprechender Anzahl per Email an die Server von *VirusTotal* übermittelt. Das Ergebnis wird wiederum per Email versendet und als *XML*-Datei in der *CWSandbox*-Datenbank gespeichert, um Auswertungen über die Daten erstellen zu können. Dieses Prozedere passiert auf Seiten der *CWSandbox*, da die Untersuchungsergebnisse in beiden Systemen benötigt werden und so eine geringere Ressourcenbelastung entsteht.

3.2.2 CWSandbox

Die Untersuchung der Malware Binaries mit Antivirenlösungen bringt nur Informationen über die Art der Malware. Die Gefährlichkeit der Malware hängt jedoch zu einem großen Teil auch davon ab, wie sie sich auf dem zu kompromittierenden System verhält. Um diese Informationen zu gewinnen, stehen so genannte *Sandbox-Technologien* zur Verfügung, mit Hilfe derer die Malware in einer abgeschirmten Umgebung beobachtet werden kann. In *BEAN* kommt die *CWSandbox* zur dynamischen Malwareanalyse von *Win32*-Anwendungen zum Einsatz. Sie führt die Malware Binaries in einer simulierten Umgebung aus, beobachtet und protokolliert alle Systemaufrufe und generiert daraus automatisch einen detaillierten Bericht [WHF07, PH07].

Die Verhaltensanalyse in der *CWSandbox* benutzt hauptsächlich zwei Techniken – *API Hooking* und *DLL injection*. Unter API Hooking versteht man das Abfangen von Aufrufen der *Windows API* und darauffolgendes Umleiten auf die so genannten *Hook Funktionen*. Dies sind eigene Funktionen, die zum Beispiel den API-Aufruf protokollieren. Im Anschluss an die Ausführung der Hook Funktion wird wieder zurück in die ursprüngliche *API Funktion* gesprungen.

Zur Umsetzung des API Hookings in der *CWSandbox* kommt DLL Injection zum Einsatz. DLL Injection bedeutet, dass eigener Code in eine *Dynamic Link Library (DLL)*, also eine Programmbibliothek, geschrieben wird und die Hook Funktion dann den Prozess, der das Malware Binary ausführt diese DLL in den Speicher laden lässt. Somit wird erreicht, dass die Hook Funktionen von der Malware selbst unerkannt aufgerufen werden.

Durch diese Funktionalitäten ist es möglich, mit der *CWSandbox* automatisierte Verhaltensanalysen von Malware Binaries zu erstellen. Der generierte Bericht enthält detaillierte Informationen über die analysierten Prozesse. Dazu gehören Informationen über Änderungen am Dateisystem und der *Windows Registry*. Zudem umfasst der Bericht die Daten, die über das Netzwerk gesendet worden sind.

Bei der Analyse der Malware Binaries durch die *CWSandbox* wird im Gegensatz zu *nepenthes* nicht verhindert, dass die Malware ihre Schadfunktionen ausführt und sich so beispielsweise auf das Dateisystem kopiert. Daher wird nach jeder Analyse eines Binaries die Simulationsumgebung der *CWSandbox* in einen *sauberen* Zustand zurückversetzt, bevor das nächste Binary untersucht wird.

Zur *CWSandbox* gibt es ein umfangreiches Webinterface [Wil07], über das man zu der jeweils darunter liegenden *CWSandbox*-Installation Daten übermitteln kann. Über diesen Weg ist die Anbindung von *BEAN* an die *CWSandbox* realisiert. Jedes neue Binary wird automatisch an die *CWSandbox* übermittelt und die URL, unter der die Analyseergebnisse für das Binary abzurufen sind, wird in die Datenbank gespeichert und als Link in das *BEAN*-Webinterface eingebunden.

3.2.3 Weitere Werkzeuge

Neben den Ergebnissen der zwei Hauptanalysen zu Art und Verhalten des Malware Binaries gibt es noch andere interessante Informationen, die aus dem Binary extrahiert werden können. Diese Informationen beziehen sich zum Großteil auf die technische Umsetzung des Binaries. Dazu gehören beispielsweise ein Speicherauszug oder *Header*-Informationen des Binaries. Sie sind vor allem für Analysen interessant, die zur Entwicklung von effektiven Gegenmaßnahmen technische Details der Malware benötigen.

Alle hier beschriebenen Tools sind auf die gleiche Weise in *BEAN* integriert. Es wurde der gleiche Ansatz verwendet wie bereits bei den lokalen Virenscannern beschrieben, um eine einfache Erweiterbarkeit zu garantieren. Die Befehle zum Aufruf der Tools sind in der Datenbank in der Tabelle *shell_analyser* hinterlegt. Die Ausgabe der Tools wird im *BEAN*-Webinterface dargestellt. Da die Ergebnisse dieser Werkzeuge entweder keine direkten Rückschlüsse auf die Eigenschaften der Malware zulassen oder keine über die Analysen der *CWSandbox* hinausgehenden Ergebnisse bringen, wurde auf eine weitere Auswertung im Folgenden verzichtet. Die Einbindung der Werkzeuge dient vielmehr der Demonstration der guten Erweiterbarkeit von *BEAN*.

objdump

Das Tool *objdump* [TGP] aus den *GNU binutils* bietet die Möglichkeit, Informationen über Binaries in Form von *dumps*, also Speicherauszügen, darzustellen. Beispielsweise kann *objdump* das Binary disassemblieren und den resultierenden *Assembler*-Code anzeigen. In diesem Zusammenhang außerdem von Interesse ist die Option, mit *objdump* alle *Header*-Informationen des Binaries ermitteln zu können.

hexdump

Ein weiteres sehr verbreitetes Tool ist *hexdump*, das einen Speicherauszug des Binaries erstellt und diesen formatiert ausgibt. Die meistverwendete Formatierung hierbei ist die hexadezimale Darstellung der Zeichen. Diese Darstellung wird auch in *BEAN* verwendet.

packerid

Immer mehr Malware wird mit so genannten *Packern* komprimiert, um sie an Sicherheitsmaßnahmen wie beispielsweise Virenscannern vorbeizuschleusen. Ein *Packer* komprimiert den Schadcode der Malware mit seinem spezifischen Algorithmus. Damit die Malware später ausgeführt werden kann, wird der Code zum Entpacken vor den komprimierten Schadcode im Binary gestellt. Mit *packerid* [Cla] ist in *BEAN* ein *Linux*-Tool integriert, das versucht, bekannte *Packer* zu erkennen.

3.3 Datenanalyse

Dieser Abschnitt befasst sich damit, wie die Rohdaten und die Metadaten, die aus den Analysen mit den im vorangegangenen Abschnitt vorgestellten Methoden resultieren, in *BEAN* statistisch analysiert werden. Zudem wird vorgestellt, welche Darstellungsformen für die Analyseergebnisse gewählt wurden und wie die Darstellung realisiert wurde.

Als Rohdaten gelten alle Daten, die von den Sensorsystemen über Angriffe und Angreifer aufgezeichnet wurden. Zudem fallen die gesammelten Malware Binaries selbst unter die Rohdaten. Die Resultate, die die einzelnen Tools liefern, sind Metadaten, sie bieten Informationen über die eigentlichen Rohdaten.

Zunächst wird im Folgenden die Vorgehensweise bei der Analyse dieser Daten beschrieben. Die generelle Methodik lehnt sich an Methoden des *Data Mining* an. Darunter versteht man die Anwendung statistischer Methoden auf die Daten zur Erkennung von intuitiv nicht ersichtlichen Zusammenhängen oder Mustern. Daraufhin werden die Formen der Visualisierung, die für die Darstellung der Rohdaten und der Analyseergebnisse zur Verfügung stehen, erläutert.

3.3.1 Vorgehensweise

Die Zielsetzung aller Analysen war es, Zusammenhänge zwischen Daten aufzudecken, die zur Einschätzung des Gefährdungslevels durch Malware hilfreich sind. Dazu war es nötig, die über zwei Millionen Datensätze, die im Laufe des Betrachtungszeitraums in die Datenbank geschrieben wurden, zu verdichten, um Wissen daraus zu gewinnen.

Im ersten Schritt wurden dazu die Daten in verschiedene Kategorien gruppiert. Die Grobstruktur der Daten wurde wie bereits in Kapitel 2.3 durch die Einteilung in direkte und indirekte Angriffsdaten und in Daten über die Malware Binaries vorgenommen. Jeder dieser drei Cluster wurde weiterhin nach verschiedenen Aspekten aufgeteilt. Dabei gibt es Daten, die mehreren Kategorien zuzuordnen sind, da sie grundlegende Informationen enthalten, die mehrere Kategorien miteinander verbinden.

Die direkten Angriffsdaten wurden in zwei Kategorien unterteilt. In die erste Kategorie fallen alle Daten, die Aussagen über die zeitliche Struktur der Angriffe zulassen. Die wichtigste Information hierbei ist der Zeitstempel der registrierten Angriffe. In die zweite Kategorie fallen Daten, die Informationen darüber geben, was die Verbreitungswege der Angriffe sind, also wie die Angreifer ins System gelangten. Dazu gehören vor allem der eingehende Port auf dem angegriffenen Sensorsystem, über den der Angreifer den Angriff ausführte.

Als nächstes wurden die indirekten Angriffsdaten untersucht. Sie wurden in drei Kategorien gruppiert. Die erste Kategorie umfasst alle Daten, die Informationen über die vom Angreifer genutzte Netztopologie enthalten. Dazu gehören neben dem *Internet Service Provider* (ISP) und der *AS Nummer* des Angreifers auch die Entfernung zwischen Angreifer und angegriffenem Sensorsystem in *Hops*. Erläuterungen zu diesen Daten sind in

Kapitel 4.2 zu finden. Zur Zuordnung der gesammelten Daten zu ISP und AS Nummern wurden Datenbanken von *MaxMind* [LLC07] genutzt.

Daten über die geographische Verteilung der Angreifer sind in der nächsten Kategorie zusammengefasst. Auch hier wurde zur Identifikation der geographischen Herkunft der Angreifer eine Datenbank von *MaxMind* verwendet. Die technologische Struktur stellt die dritte Kategorie für indirekte Angriffsdaten dar. Die Daten, die hierunter fallen, sind Ergebnisse von *p0f* zum Betriebssystem des Angreifers.

Der letzte Cluster, der in weitere Kategorien unterteilt wurde, sind Daten zu den Malware Binaries. Hier wurden vier Kategorien definiert, nach denen diese Daten eingeteilt werden. Die ersten beiden Kategorien enthalten die gleiche Struktur an Daten. Ihr Unterscheidungskriterium liegt darin, dass zur ersten Kategorie nur Daten über das Erstauftreten jedes gesammelten Malware Binaries zählen, zur zweiten dann Daten über jedes Auftreten des Binaries. Die Daten zum Erstauftreten setzen sich jeweils aus den Datensätzen zu einem Malware Binary, die bei dem ersten Download dieses Binaries aufgezeichnet wurden, zusammen. Die dritte Kategorie, die für die Binarydaten identifiziert wurde, sind die Ergebnisse der lokalen Virenscanner. Die Ergebnisse der *CWSandbox* wurden als vierte Kategorie zusammengefasst.

Nachdem die Daten in verschiedene Kategorien unterteilt wurden, wurden in allen Kategorien erste Auswertungen auf den Daten identifiziert. Zunächst wurden dabei Auswertungen zur Verteilung der einzelnen Daten beziehungsweise zu Häufungen bestimmter Ausprägungen erstellt. Bei den direkten Angriffsdaten wurden dazu beispielsweise die am häufigsten ausgenutzten Schwachstellen ermittelt.

Aufgrund der Ergebnisse dieser ersten Auswertungen wurden weitere Analysen identifiziert, die Zusammenhänge zwischen bestimmten Daten aufdecken sollen. Aufgrund der auffällig hohen Werte einzelner Provider bei den Angreifern wurde beispielsweise untersucht, wieviel Prozent der Angriffe von Angreifern kamen, die beim gleichen Provider wie das Opfersystem ihr Internet bezogen. Alle Auswertungen und deren Ergebnisse werden ausführlich in Kapitel 4 beschrieben und interpretiert.

3.3.2 Visualisierung

Die überwiegende Zahl an Methoden der Datenanalyse brachten Ergebnisse, die am besten durch eine grafische Darstellung veranschaulicht werden können. Daher wurden sowohl für die Daten aus dem *Raw Malware Repository* als auch für die Daten aus dem *Malware Analysis Repository* geeignete Visualisierungen erarbeitet, die im *BEAN*-Webinterface automatisiert mit jeweils aktuellen sowie mit historischen Daten dargestellt werden können. Im Folgenden werden diese Visualisierungen und ihre Umsetzung beschrieben.

Zunächst wurden die Rohdaten mit Hilfe von Tabellen und Detailseiten so dargestellt, dass alle relevanten Informationen zu bestimmten Datensätzen übersichtlich abrufbar sind. Ein Beispiel für eine solche Seite ist in Abbildung 3.1 dargestellt. Die Abbildung

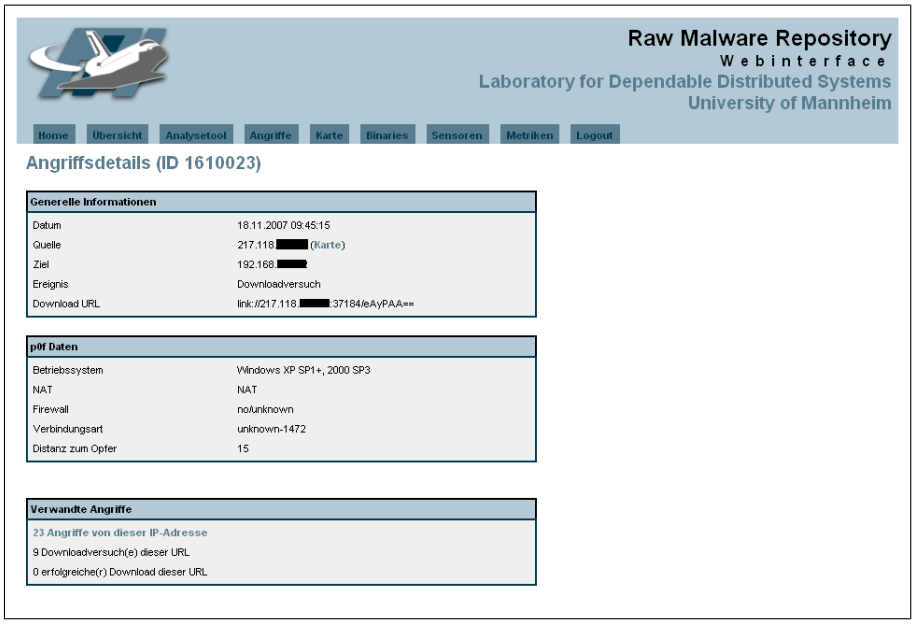


Abbildung 3.1: Detailseite zu einem Angriff aus dem *BEAN*-Webinterface

zeigt die Übersicht zu einem Angriff mit erfolgreichem Download eines Malware Binaries und zusätzlich die Daten, die *p0f* über den Angreifer lieferte. Abbildung 2.2 ist ein weiteres Beispiel. Sie zeigt eine Übersichtseite in Form einer Tabelle mit verschiedenen heruntergeladenen Malware Binaries.

Für die Darstellung der Analyseergebnisse wurden grafische Visualisierungsmöglichkeiten benötigt. Es wurden dazu in dieser Arbeit vier verschiedene Formen von Diagrammen verwendet, *Kuchendiagramme*, *Balkendiagramme*, *Liniendiagramme* und Landkarten. Da das *BEAN*-Webinterface in *PHP* implementiert ist, wurden die ersten drei Diagrammtypen mit *JPGraph* [Con07] erstellt. Bei *JPGraph* handelt es sich um eine *PHP*-Bibliothek zum Erstellen von Diagrammen. Für die Landkarten wurde eine *PHP*-Implementierung [Kis07] zur Verwendung der *Google Maps API* [Goo07] genutzt.

Das Erstellen von Diagrammen mit Hilfe von *JPGraph* folgt immer einem bestimmten Schema. Zunächst werden über ein *SQL-Query* die Datensätze, die dargestellt werden sollen, aus der Datenbank abgefragt. Zur Weiterverarbeitung dieser Datensätze muss entschieden werden, welche Art von Diagramm dafür geeignet ist. Die Datensätze werden an die entsprechenden Routinen von *JPGraph* übergeben, um die Grafik zu erstellen. Die Darstellung im Webinterface findet dann in Form von *Portable Network Graphics* (PNG) statt.

Für Kuchendiagramme gibt es eine Besonderheit, da für diese Diagramme Datensätze, die nur kleine Anteile der Gesamtdaten ausmachen, zu einem Wert *andere* zusammengefasst werden. Je nach Verteilung der gewählten Datensätze setzt sich *andere* aus allen

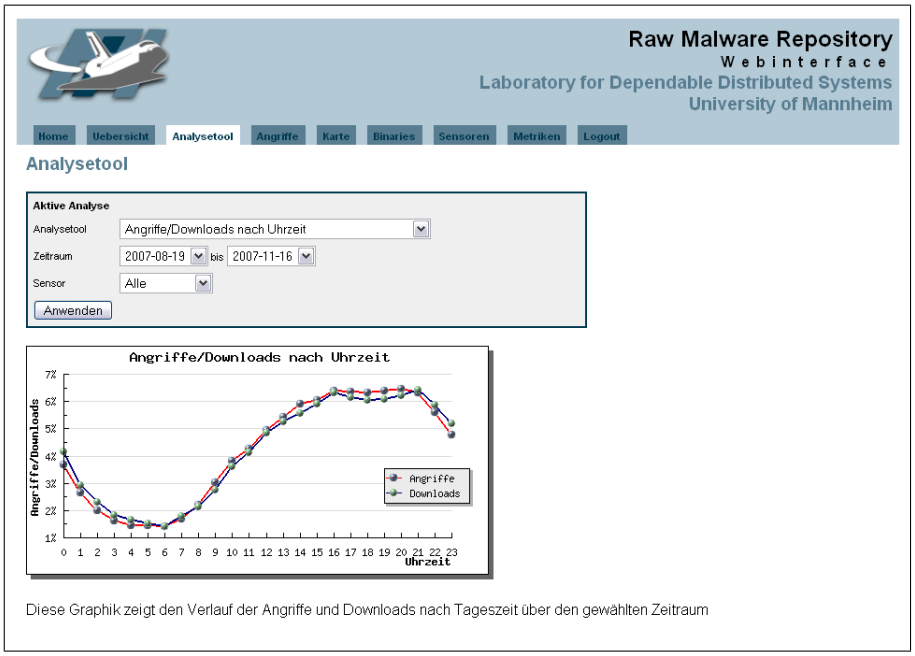


Abbildung 3.2: Analysetool aus dem *BEAN*-Webinterface

Datensätzen zusammen, die weniger als 0,5% bis 2% der gesamten Daten ausmachen.

Die Grundlage für die Landkarten sind IP-Adressen. Mit ihnen können Verteilungen der Angreifer über Deutschland beziehungsweise über die ganze Welt dargestellt werden wie auf den Abbildungen 4.18 und 4.19 zu sehen. Außerdem kann eine einzelne IP-Adresse auf einer Landkarte abgebildet werden.

Alle Grafiken können mit den jeweils aktuellen ebenso wie mit historischen Datensätzen im *BEAN*-Webinterface im so genannten *Analysetool* abgerufen werden, das in Abbildung 3.2 dargestellt ist. Da in vielen Situationen nur einzelne Sensoren oder bestimmte Zeiträume von Interesse sind, gibt es über Filter die Möglichkeit, die Grafiken jeweils für einen bestimmten Zeitraum anzeigen zu lassen. Einige der Grafiken bieten ebenso die Möglichkeit der Filterung der Daten nach einem einzelnen Sensor.

Zur Visualisierung der Metriken, die in Kapitel 5 vorgestellt werden, wurden zwei verschiedene Darstellungen verwendet. Für die jeweils aktuellen Tageswerte der Metriken kann im Webinterface eine Tabelle mit den Werten und der zugehörigen Kategorie abgerufen werden. Zu jeder Metrik gibt es zudem die Möglichkeit, den historischen Verlauf ihrer Werte grafisch darstellen zu lassen. Dies geschieht auf die gleiche Weise wie die Darstellung der Analysegrafiken, sodass auch für die Metriken Filter für den anzuzeigenden Zeitraum einstellbar sind. Alle Grafiken in den Kapiteln 4 und 5 sind aus dem Webinterface entnommen und mit den hier beschriebenen Methoden erstellt worden.

3.4 Zusammenfassung

Im diesem Kapitel wurden die Methoden und Werkzeuge vorgestellt, mit Hilfe derer die Rohdaten, die das Sensorsystem sammelt, analysiert werden. Dazu wurde zunächst die Struktur und die technische Umsetzung der Analysen kurz dargestellt. Es wurden die Werkzeuge vorgestellt, die in *BEAN* zur Analyse zum Einsatz kommen. Dabei wurde insbesondere herausgestellt, wie das Hauptziel für diesen Teil der Arbeit, die Automatisierung aller Analysen und die volle Integration der Werkzeuge in das *BEAN*-System, erreicht wurde.

Jedes der Malware Binaries wird in regelmäßigen Abständen durch eine Reihe lokaler Virenscanner untersucht. Ebenso wird jedes Binary über eine automatisierte Anbindung zur *CWSandbox* zur dynamischen Verhaltensanalyse gesendet. Über diese Anbindung werden die Binaries einmalig zu einem externen Dienst zum Überprüfen mit 32 Virenscan-Engines geschickt. Die Ergebnisse dieser Werkzeuge sind im Webinterface abrufbar.

Zusätzlich wurde in *BEAN* die Möglichkeit implementiert, die Malware Binaries mit verschiedenen kleineren Analysewerkzeugen bei Bedarf auf technische Details zu untersuchen. Dies sind *objdump* und *hexdump* zu Darstellung von Speicherauszügen des Binaries und *packerid* zur Erkennung von *Packern* in den Binaries. Die Ausgaben dieser Werkzeuge werden ebenfalls im Webinterface dargestellt.

Abschließend wurden in diesem Kapitel die statistischen Auswertungen auf den gewonnenen Daten beschrieben. Dazu wurde zunächst die generelle Vorgehensweise erläutert. Im Anschluss daran wurden die Methoden der Visualisierung, die zur Darstellung der Ergebnisse im *BEAN*-Webinterface verwendet wurden, beschrieben.

4 Analyseergebnisse

Dieses Kapitel beschreibt und bewertet die Ergebnisse, die die Anwendung der in Kapitel 3 beschriebenen Analysemethoden auf die Datengrundlage ergaben. Wie bereits in Kapitel 3 werden auch in diesem Kapitel die Analysen in die drei Kategorien Daten zu Angriffen, Angreifern und Malware Binaries unterteilt.

Die zugrundeliegenden Daten für die Analysen lieferten die in Kapitel 2.5 beschriebenen Sensoren über einen Zeitraum von zehn Wochen. Detailinformationen zu den eingesetzten Sensoren bietet Tabelle 2.3. Dort ist auch die Zuordnung der Sensoren zu der jeweils in diesem Kapitel verwendeten Sensornummer zu finden. Zu Beginn werden die verschiedenen Ergebnisse der Analysen auf die Angriffsdaten erörtert, danach diejenigen zu den Daten über Angreifer und zuletzt die in Bezug auf die Malware Binaries.

In Abbildung 4.1 ist die Entwicklung der Datengrundlage über den Betrachtungszeitraum dargestellt. Die roten Datenpunkte zeigen die Zahl der Angriffe an, die blauen die Anzahl an erfolgreichen Downloads und die grünen die Anzahl an unterschiedlichen Angreifern.

Zum Verständnis der folgenden Abschnitte ist es notwendig, die Begriffe *Verbindungsversuch*, *Angriff* und *Download* voneinander abzugrenzen. Ein *Verbindungsversuch* ist eine Verbindung, die *nepenthes* als nicht eindeutig bösartig einstuft. Unter *Angriff* werden alle aufgezeichneten Verbindungen zu den Sensoren verstanden, die von *nepenthes* als bösartig identifiziert werden konnten, also sowohl Verbindungen, bei denen lediglich

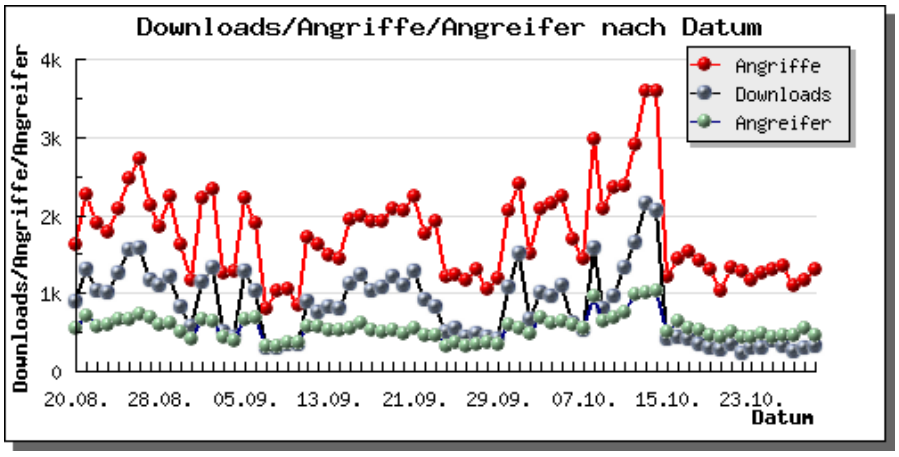


Abbildung 4.1: Entwicklung der Datengrundlage über den Betrachtungszeitraum

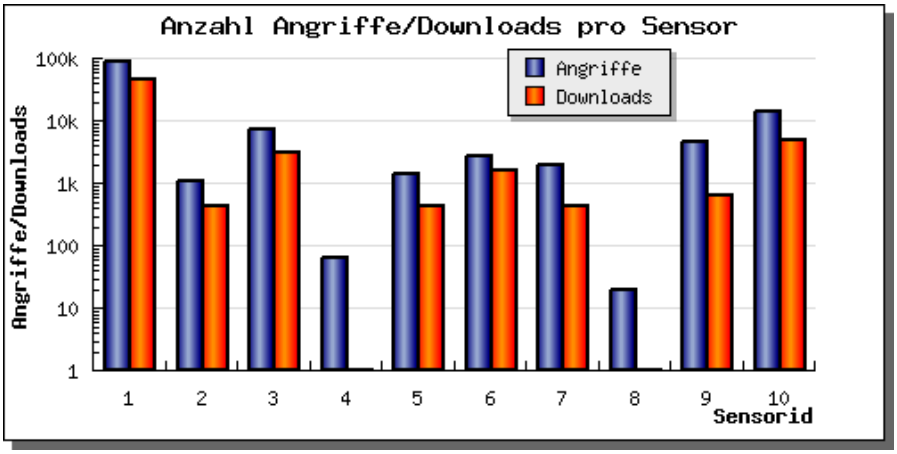


Abbildung 4.2: Absolute Anzahl an Angriffen/Downloads für alle Sensoren im Datenset

Schadcode gesendet wurde als auch Verbindungen, über die versucht wurde, ein Malware Binary herunterzuladen oder über die dies erfolgreich war. *Downloads* sind nur diejenigen Verbindungen, über die erfolgreich ein Malware Binary heruntergeladen werden konnte.

Alle in diesem Kapitel dargestellten Grafiken sind aus dem *BEAN*-Webinterface entnommen. Sie können dort immer mit den jeweils aktuellen Daten abgerufen werden, da sie dynamisch erzeugt werden.

4.1 Angriffe

In diesem Abschnitt werden die Analyseergebnisse der direkten Angriffsdaten dargestellt. Direkte Angriffsdaten sind zum Beispiel die IP-Adresse des Angreifers, eine Definition liefert Kapitel 2.3. Insgesamt werden in diesem Kapitel 705.478 Verbindungsversuche, 124.660 Angriffe und 59.141 Downloads untersucht. Die Verteilung der Daten auf die einzelnen Sensoren zeigt Abbildung 4.2.

Über alle Sensoren betrachtet, führten fast 50% der Angriffe zum Download. Zwischen den einzelnen Sensoren fallen in Bezug auf das Verhältnis zwischen Angriffen und Downloads jedoch erhebliche Unterschiede auf. Bei vier der Sensoren liegt das Verhältnis mit zwischen 40% und 58% nahe beim Gesamtdurchschnitt, während bei zwei Sensoren überhaupt keine Downloads registriert wurden.

Da auf allen Sensoren das gleiche Sensorsystem lief und die gleichen Ports offen waren, muss die Begründung für die Unterschiede außerhalb der Sensoren zu finden sein. Bei einem der beiden Sensoren wurde die Internetverbindung über eine *FritzBox* der Firma *AVM*, ein Hardwarerouter, weitergeleitet. Die Einstellungen dieses Routers lassen keine ein- und ausgehenden Verbindungen auf den TCP- und UDP-Ports 135, 139 und 445 zu [uIA07].

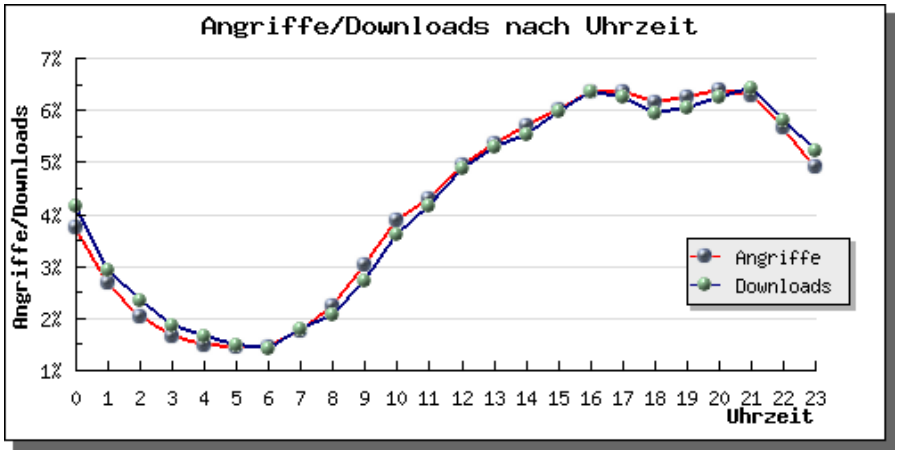


Abbildung 4.3: Verlauf der Angriffe und Downloads über die Tageszeit

Der zweite Sensor, bei dem kein einziger Download auftrat, war über den *Internet Service Provider* (ISP) *KabelBW* mit dem Internet verbunden. Dieser blockiert in seinem Netz alle Pakete auf den TCP- und UDP-Ports 137 bis 139 und 445 sowie auf den TCP-Ports 135, 593, 3127 und 4444 und den UDP-Ports 135 und 69 [uCK07]. Der Zusammenhang zwischen dem Sperren dieser Ports und dem Fehlen von Downloads wird durch die Beobachtungen in Kapitel 4.1.2 ersichtlich.

Die gesammelten Daten zu den Angriffen werden nach den zwei Gesichtspunkten *zeitliche Struktur* und *Verbreitungswege* untersucht. Bei der *zeitlichen Struktur* wird auf Aspekte der zeitlichen Verteilung der Angriffe eingegangen, aber auch darauf, wieviel Zeit durchschnittlich zwischen dem Verbinden eines Systems mit dem Internet und seiner Kompromittierung durch Malware verging. Im Rahmen der Analysen nach *Verbreitungswegen* werden beispielsweise die Hauptschwachstellen im Rahmen der vom Sensorsystem emulierten Schwachstellen herausgearbeitet, die von automatisiert verbreitender Malware ausgenutzt wurden.

4.1.1 Zeitliche Struktur

Im ersten Schritt wird die zeitliche Struktur der Angriffe untersucht. Unter diese Dimension der Analysen fallen alle Untersuchungen der Angriffsdaten, die eine zeitliche Dimension betrachten. Alle zeitlichen Analysen beziehen sich auf die lokale Zeit des Servers und aller Sensoren (mitteleuropäische Sommerzeit).

Die erste Fragestellung in diesem Bereich ist, ob zwischen der Angriffsdichte und der Tageszeit ein Zusammenhang besteht. Dazu sind in Abbildung 4.3 alle Angriffe (rote Linie) beziehungsweise Downloads (blaue Linie) über die Tageszeit abgetragen. Es ist deutlich zu erkennen, dass sowohl die Anzahl der Angriffe als auch die der Downloads im Betrachtungszeitraum tageszeitabhängig war.

Mehr als 40% der Angriffe kamen zwischen 15 Uhr und 21 Uhr. Dahingegen wurden weniger als 20% aller Angriffe zwischen 1 Uhr und 9 Uhr registriert. Zwischen 9 und 15 Uhr kann ein kontinuierlicher Anstieg der Angriffszahl beobachtet werden, zwischen 21 und 1 Uhr ein ebenso kontinuierlicher Abfall. Jeweils um 1 Uhr und um 9 Uhr war das Angriffslevel in etwa gleich hoch, ebenso um 15 Uhr und um 21 Uhr. Betrachtet man im Vergleich dazu nur erfolgreiche Downloads, unterscheiden sich die Werte kaum. Daraus lässt sich schlussfolgern, dass es keine bestimmten Tageszeiten gibt, zu denen Angriffe erfolgreicher waren als zu anderen Tageszeiten.

Da die Sensoren aufgrund unveränderter Konfiguration zu jeder Tageszeit gleich anfällig für Angriffe waren, lässt der Gesamtverlauf den Schluss zu, dass die Gefährdung nachts deutlich geringer war als in den Nachmittags- und Abendstunden. Da hier ausschließlich automatisierte Malware betrachtet wurde, lässt diese zeitliche Struktur der Angriffe die Vermutung über einen geographischen Zusammenhang zwischen Angreifer und Sensorsystem zu. Ob dieser tatsächlich besteht, wird in Kapitel 4.2.2 näher untersucht.

Der Verlauf über die Tageszeit könnte dadurch erklärt werden, dass sich die Angreifer in einer nahen oder der gleichen Zeitzone wie das angegriffene Sensorsystem befanden, da nachts viele kompromittierte Rechner, die als Teil eines Botnetzes vom Benutzer meist unbemerkt weitere Systeme angreifen, ausgeschaltet sind. Zudem fällt der Höchststand an Angriffen zeitlich mit dem Feierabend vieler Arbeitnehmer zusammen. Dies legt nahe, dass geschäftlich genutzte Systeme seltener infiziert und damit besser geschützt sind als private Systeme.

Beobachtungen dieser Art wurden bereits früher gemacht. Beispielsweise haben Dagon et al. auf Basis dieser zeitlichen Strukturen ein Modell zur Vorhersage der Verbreitung von Botnetzen entwickelt [DZL06].

Im Hinblick auf die zeitliche Struktur der Angriffe ist ein weiterer interessanter Aspekt die Verteilung der Angriffe über die Wochentage. Nach den Beobachtungen zur Tageszeit liegt die Vermutung nahe, dass bei der Verteilung über die Wochentage Unterschiede zwischen Werktagen und Wochenenden erkennbar sind. In Anknüpfung an die bei der Tageszeit gesehene Abhängigkeit zwischen der Zahl der Angriffe und dem Tagesrhytmus der Systemnutzer wäre hier zu vermuten, dass an Wochenenden deutlich mehr Angriffe und Downloads verzeichnet wurden als an Wochentagen.

Abbildung 4.4 zeigt, dass die Unterschiede zwischen den einzelnen Wochentagen gering sind. Pro Tag wurden etwa ein Siebtel aller Angriffe verzeichnet. Ebenso wie bei der Verteilung über die Tageszeit unterscheiden sich die Beobachtungen zu den Downloads nicht von denen der Angriffe.

Diese Beobachtungen decken sich mit der Studie von Eimeren et al. zur Onlinenutzung in Deutschland. Dort wurde festgestellt, dass die durchschnittliche Verweildauer im Internet bei Deutschen pro Tag bei etwa zwei Stunden liegt. In dieser Studie konnte ebenfalls ein leichter Unterschied in der Verweildauer von Werktagen und Wochenenden beobachtet werden [EF07].

Bei der Betrachtung sicherheitsrelevanter Analysen in der zeitlichen Dimension ist ein

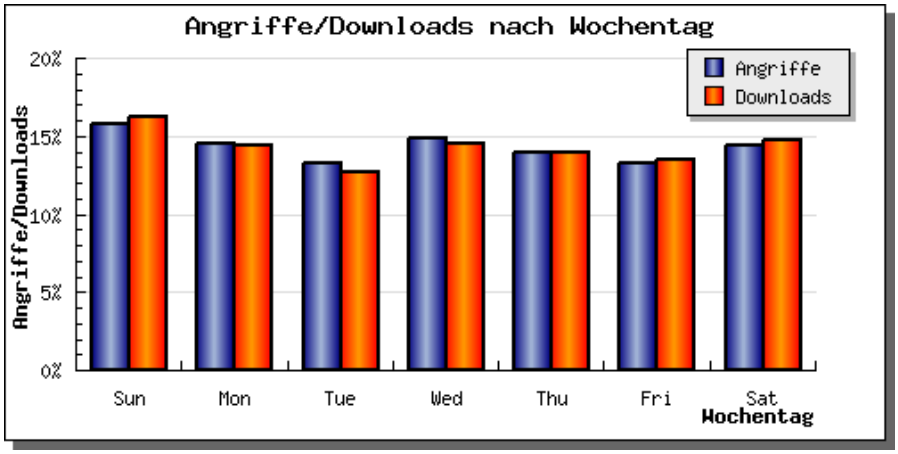


Abbildung 4.4: Verteilung der registrierten Angriffe/Downloads nach Wochentag

weiterer Aspekt von zentraler Bedeutung. Die Fragestellung, wieviel Zeit zwischen dem Verbinden eines Systems mit dem Internet und der Kompromittierung des Systems durch Malware vergeht, wird im Folgenden näher betrachtet. Die Sicherheitsrelevanz dieses Ergebnisses besteht darin, dass es aussagt, wie lange ein Benutzer Zeit hat, ein neues System über das Internet mit den nötigen *Sicherheitspatches* und *Sicherheitsvorkehrungen*, wie einem Virenscanner, auszurüsten.

Da die Sensoren nach einem verzeichneten Download nicht tatsächlich kompromittiert sind, wird ein Sensorsystem immer dann als *neu* betrachtet, wenn es eine neue öffentliche IP-Adresse zugewiesen bekommt. Das bedeutet, dass für Sensoren, die ihre öffentliche IP-Adresse dynamisch von ihrem ISP beziehen, das System bei jedem IP-Wechsel als *neu* gilt.

In Abbildung 4.5 sind die Zeitwerte vom Bezug einer neuen IP-Adresse bis zum ersten erfolgreichen Download für alle Sensoren aus den beiden Datensets in Minuten dargestellt. Da die Spanne, in der sich die Zeitwerte bewegen, verhältnismäßig groß ist, wurde eine logarithmische Y-Achse gewählt. Für die Sensoren 4 und 8, für die im Betrachtungszeitraum kein Download registriert wurde, sind in der Abbildung keine Balken dargestellt.

Die Unterscheidung zwischen fester und dynamischer IP-Vergabe ist farblich gekennzeichnet. Sensoren mit fester Adresse sind rot dargestellt, Sensoren mit dynamischer Adresse blau. Die Werte für Sensoren mit dynamisch vergebener IP-Adresse sind die Durchschnittswerte aus den Werten für alle IP-Wechsel des jeweiligen Sensors, wohingegen für Sensoren mit fester IP-Adresse jeweils nur ein Wert existiert.

Die Sensoren 2 und 4 waren beide über eine feste IP-Adresse mit dem Internet verbunden. Die Sensoren 6 und 8 konnten zwar von Seiten des Providers die IP-Adresse dynamisch wechseln, haben dies jedoch im Betrachtungszeitraum nie getan. Die restlichen Sensoren bezogen im betrachteten Zeitraum mindestens einmal pro Tag eine neue IP-Adresse von ihrem Provider. Insgesamt wurden im Betrachtungszeitraum 610 verschiedene öffentliche

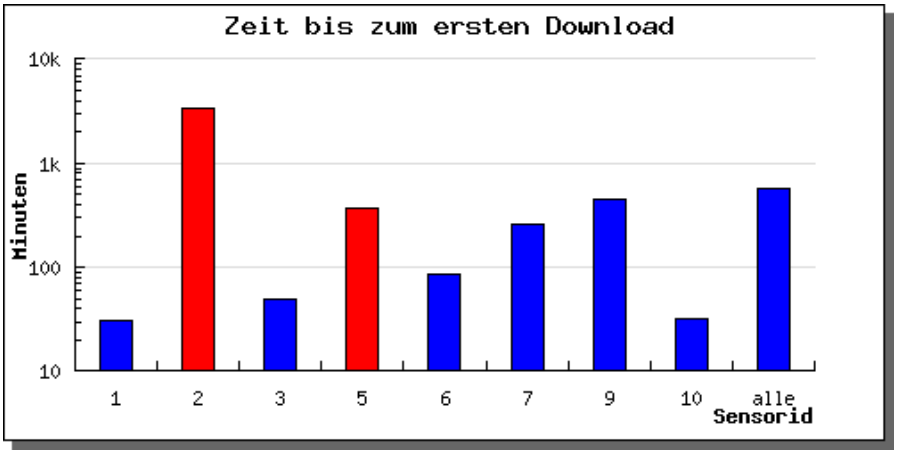


Abbildung 4.5: Zeit bis zum ersten Download eines Malware Binaries(logarithmische Y-Achse)

IP-Adressen für die Sensoren registriert. Je nach Sensor sind dabei zwischen 57 und 153 Adressen in die Betrachtung eingegangen.

Wie in der Grafik zu sehen ist, gibt es deutliche Unterschiede zwischen den Werten für einzelne Sensoren. Der Sensor, bei dem der Wert am niedrigsten ist, war im Durschnitt nach 30,6 Minuten kompromittiert. Am längsten, nämlich über 2 Tage, dauerte es bei einem Sensor mit fester öffentlicher IP-Adresse. Im Durchschnitt war ein System nach 9,5 Stunden kompromittiert. Betrachtet man nur Sensoren, die ihre IP-Adresse dynamisch beziehen, lag der Durchschnitt bei etwa 2,5 Stunden.

Aufgrund dieser Daten lässt sich vermuten, dass die Zeit bis zur Kompromittierung eines Systems abhängig von dessen Bezugsart der IP-Adresse und von dessen Standort ist. Um dies mit statistischer Signifikanz nachweisen zu können, müssten die Untersuchungen allerdings mit einer größeren Anzahl an Sensoren durchgeführt werden.

4.1.2 Verbreitungswege

Zur Absicherung eines Systems gegen automatisierte Malware ist es von zentraler Bedeutung zu wissen, über welchen Weg der Angreifer in das System kommt. Daher wird im Rahmen dieser Arbeit untersucht, welches die Verbreitungswege der automatisierten Malware sind. Die Betrachtungen beziehen sich auf bekannte Schwachstellen, für die auf den Sensoren ein *nepenthes-Schwachstellenmodul* konfiguriert wurde.

Jeder Angriff wird über eine Verbindung zu einem bestimmten Port durchgeführt. Daher ist die erste wichtige Information über Verbreitungswege der beobachteten Malware, über welche Ports die Angriffe abliefen. Abbildung 4.6 zeigt die Eingangsports, auf denen die meisten bösartigen Verbindungen von *nepenthes* registriert wurden.

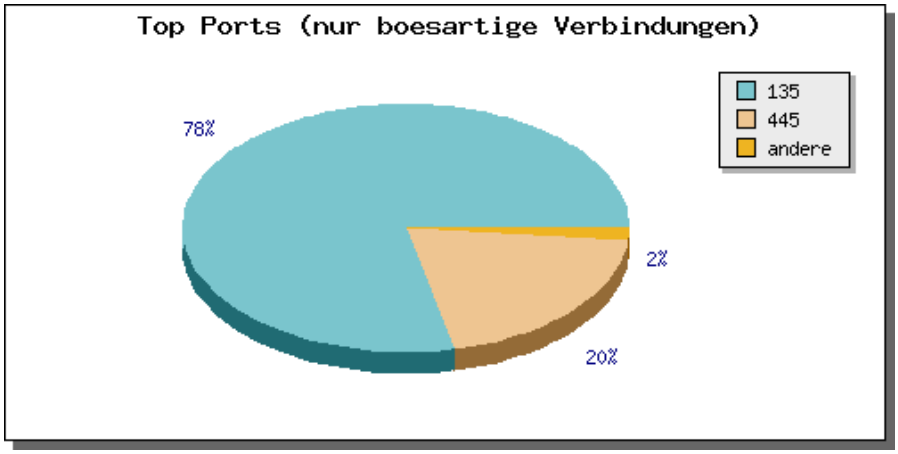


Abbildung 4.6: Die am häufigsten angegriffenen Ports

Man kann sehen, dass die Angriffe im untersuchten Datenset überwiegend über die zwei Ports 135 und 445 erfolgt sind. Auf Port 135 ist im Sensorsystem der *DCOM-Dienst* emuliert, für den es eine Reihe von Schwachstellen gibt. Einer der Würmer, die das System über diesen Dienst kompromittieren, ist der Wurm *W32.Blaster*. Für diesen Wurm ist ein beispielhafter Angriff in Kapitel 2.4 beschrieben. Auf Port 445 bietet das Sensorsystem verschiedene Schwachstellen an – beispielsweis den *LSASS-Dienst* oder eine Schwachstelle im *Plug-and-Play*-System von *Windows*.

Das Sensorsystem war insgesamt über 31 verschiedene Ports durch *Schwachstellenmodule* verwundbar. Es wurden im Betrachtungszeitraum Angriffe auf zehn verschiedenen Ports aufgezeichnet, wobei acht davon insgesamt weniger als 2% der Angriffe ausmachen. Offenbar stellen die Schwachstellen auf diesen Ports ein höheres Sicherheitsrisiko in Bezug auf das Infizieren mit automatisierter Malware dar als die auf den anderen Ports.

Die Verteilung der Angriffe auf Dienste, zu denen *nepenthes* Schwachstellen emuliert, zeigt das gleiche Bild wie die Verteilung der angegriffenen Ports. Abbildung 4.7 zeigt die am häufigsten angegriffenen Schwachstellen. Etwa 80% der Angriffe zielten auf die *DCOM-Schwachstelle* über das *nepenthes*-Schwachstellenmodul *vuln-dcom*. Etwa 15% griffen über das *SMB-Protokoll*, das *nepenthes*-Schwachstellenmodul *vuln-asn1* an und 5% der Angriffe kamen über die *LSASS-Schwachstelle* (*vuln-lsass*) in das System. *DCOM* ist über Port 135 angreifbar und sowohl *SMB* als auch *LSASS* über Port 445, was zeigt, dass sich die Verteilung der angegriffenen Ports und die der Schwachstellen decken.

Auffallend ist, dass die Verteilung bei unterschiedlichen Sensoren deutlich von der Verteilung über die gesamte Datenmenge abweichen kann. Abbildung 4.8 zeigt die Verteilung der angegriffenen Ports und Dienste von *Sensor 2*, einem Sensorsystem, das im Netzwerk der *Universität Mannheim* betrieben wurde. In über 80% aller Fälle ging der Angriff bei diesem Sensor über Port 445 und unter 10% der Angriffe kamen über Port 135. Der am

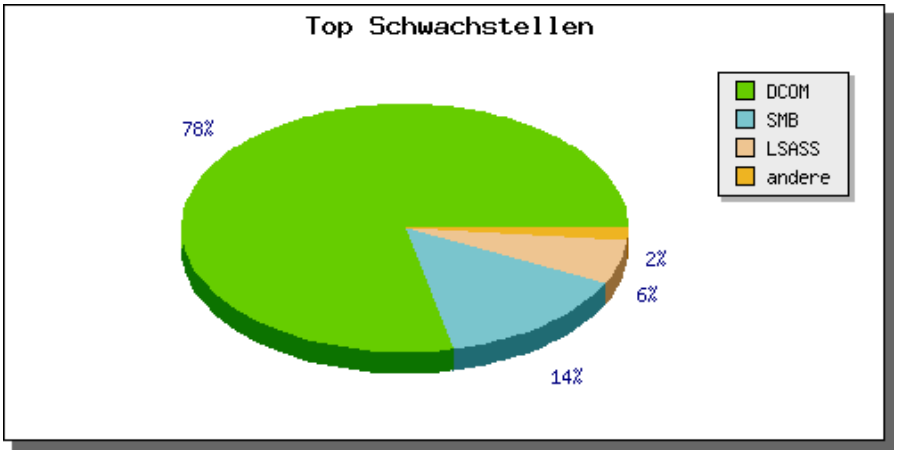
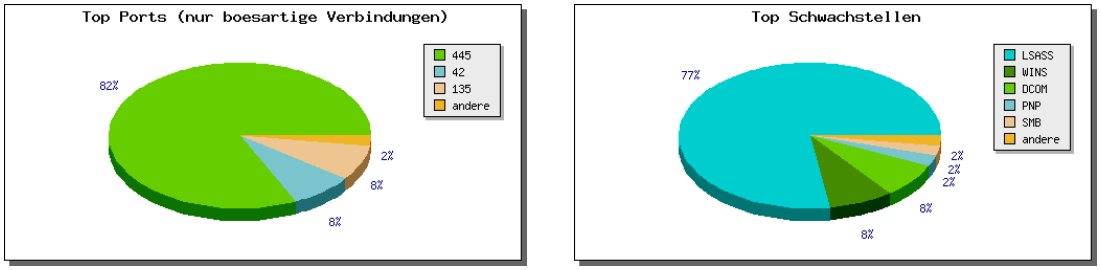


Abbildung 4.7: Die am häufigsten angegriffenen Dienste



(a) Ports

(b) Dienste

Abbildung 4.8: Die am häufigsten angegriffenen Ports/Dienste bei Sensor 2

häufigsten angegriffene Dienst war hier *LSASS* mit über drei Viertel der Angriffe. Eine mögliche Erklärung für diese Unterschiede liefert Kapitel 4.2.1, in dem unter anderem Zusammenhänge zwischen dem ISP des Sensors und dem ISP des Angreifers untersucht werden.

Sobald *nepenthes* über eine Verbindung eine Payload empfängt, wird diese wie in Kapitel 2.1.2 beschrieben versucht, einem *Shellcodehandler* zuzuordnen. In Abbildung 4.9 sind die *Shellcodehandler* dargestellt, denen die meisten Angriffe zugeordnet werden konnten. Insgesamt können die registrierten Angriffe 15 verschiedenen *Shellcodehandlern* zugeordnet werden. Es kann aus den Daten kein *Handler* identifiziert werden, der überwiegend verwendet werden konnte. Die benutzten *Handler* besitzen kaum gemeinsame Merkmale, sodass keine allgemeinen Aussagen über diese Ergebnisse getroffen werden können.

Bisher wurden in diesem Abschnitt nur Verbindungen betrachtet, die *nepenthes* tatsächlich als bösartig identifiziert hat. Da *nepenthes* ausschließlich bekannte automatisierte Malware verarbeiten kann, können auch nicht als definitiv bösartig erkannte Ver-

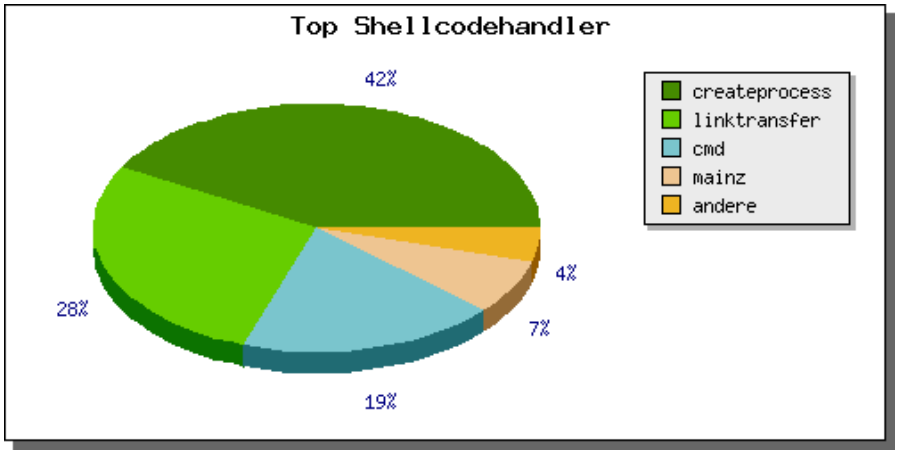


Abbildung 4.9: Die am häufigsten genutzten Nepenthes-Shellcodehandler

bindungen Angriffe darstellen. Abbildung 4.10 zeigt die Verteilung aller Verbindungen, die *nepenthes* als nicht eindeutig bösartig eingestuft hat und die somit nicht weiter verarbeitet wurden.

Die Verteilung der Ports, auf denen Verbindungen eingingen, aus denen kein automatisierter Angriff auf *nepenthes* folgte unterscheidet sich deutlich von der vorher betrachteten Abbildung. Es waren insgesamt mehr Ports betroffen, 98% der Verbindungsversuche verteilen sich auf fünf Ports. Die große Differenz lässt den Schluss zu, dass es beispielsweise auf Port 139, auf den mehr als 25% der Verbindungsversuche, aber nur weniger als 2% der Angriffe fielen, Schwachstellen gibt, die auf dem Sensorsystem nicht emuliert wurden. Die Malware versucht, über diesen Port anzugreifen, bekommt jedoch nicht die erwartete Reaktion, sodass sie gar nicht erst ihre Payload an das Sensorsystem schickt. Somit identifiziert *nepenthes* diese Verbindung nicht als bösartig.

Insgesamt sind im Datenset Verbindungsversuche auf 3728 verschiedenen Ports erfasst. Diese hohe Zahl lässt sich dadurch erklären, dass das *log-surfnet*-Modul von *nepenthes* auf den Sensorsystemen so konfiguriert ist, dass es alle eingehenden Verbindungen versucht zu verarbeiten, auch wenn *nepenthes* auf diesem Port keine Schwachstelle emuliert. Somit können alle eingehenden Netzwerkaktivitäten beobachtet und so durch manuelle Analyse womöglich neue Angriffspunkte gefunden werden, um *nepenthes* durch entsprechende Schwachstellenmodule zu erweitern.

Betrachtet man unter den Ports, auf denen Verbindungsversuche registriert wurden, nur diejenigen, zu denen keine einzige bösartige Verbindung registriert wurde, lässt sich feststellen, dass dabei einige Ports dominieren. Konkret sind dies mit etwa einem Viertel der Verbindungen Port 443 und mit etwa einem Fünftel Port 44445. Zudem traten noch mit 13% und 7% die Ports 4460 und 25 verhältnismäßig häufig in dieser Konstellation auf. Für diese Ports ist demnach anzunehmen, dass die bereits existierenden *nepenthes*-Schwachstellenmodule, die auf diesen Ports lauschen, den Angriff nicht entgegennehmen

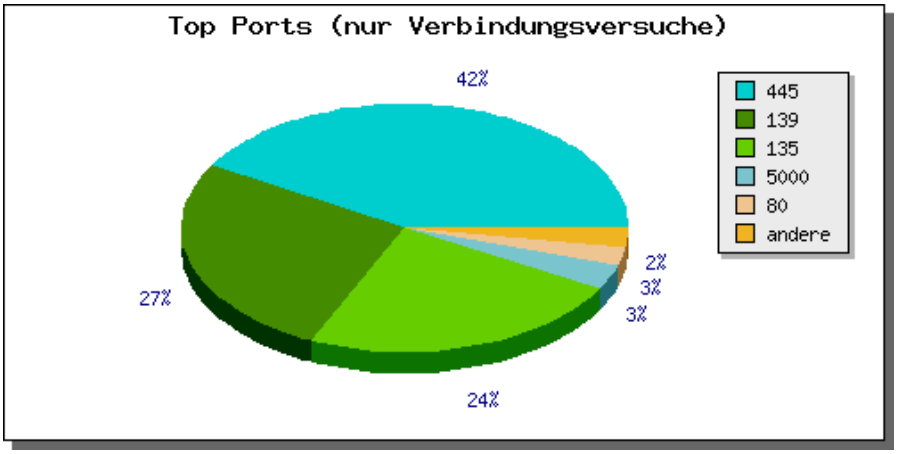


Abbildung 4.10: Die Ports mit den meisten Verbindungsversuchen

und somit in ihrer Funktionalität überarbeitet werden sollten. Für Ports wie 44445, auf dem noch kein *nepenthes*-Schwachstellenmodul läuft, wäre es ratsam, die betreffenden Angriffe näher zu analysieren, um gegebenenfalls ein Schwachstellenmodul dafür zu implementieren.

4.2 Angreifer

Dieser Abschnitt beschreibt die Ergebnisse der Analsyen von indirekten Angriffsdaten. Zu den indirekten Angriffsdaten zählen alle Daten über den Angreifer (Definition in Kapitel 2.3). Als Angreifer wird hier das System verstanden, von dem der Angriff ausgeht und keine natürliche Person. Dies resultiert daraus, dass diese Arbeit wie bereits beschrieben ausschließlich automatisierte Malware betrachtet, die keiner Interaktion einer natürlichen Person als Angreifer bedarf.

Es gibt zwei Möglichkeiten aus der gesammelten Datengrundlage die Zahl der Angreifer zu bestimmen. Beide weisen gewisse Unschärfen auf, was daraus resultiert, dass das einzige verfügbare Identifikationsmerkmal die IP-Adresse ist, von der der Angriff ausgeht. Die IP-Adresse ist jedoch nicht dauerhaft eindeutig einem bestimmten System zuzuordnen. Angreifersysteme, die automatisierte Malware verbreiten, beziehen ihre öffentliche IP-Adresse nicht selten dynamisch von ihrem Provider. Das bedeutet, dass ein Angreifersystem im Lauf von mehreren Tagen und Wochen über unterschiedliche IP-Adressen verbunden sein kann. Ebenso kann ein Angreifersystem im Laufe der Zeit über eine IP-Adresse verbunden sein, über die vorher ein anderes, unabhängiges Angreifersystem verbunden war, da jeder ISP nur eine begrenzte Anzahl IP-Adressen zur Verfügung hat.

Die erste Möglichkeit zur Bestimmung der Anzahl der Angreifer im vorliegenden Datenset ist, die Angreifer anhand ihrer IP-Adresse zu identifizieren. Das bedeutet, die Gesamt-

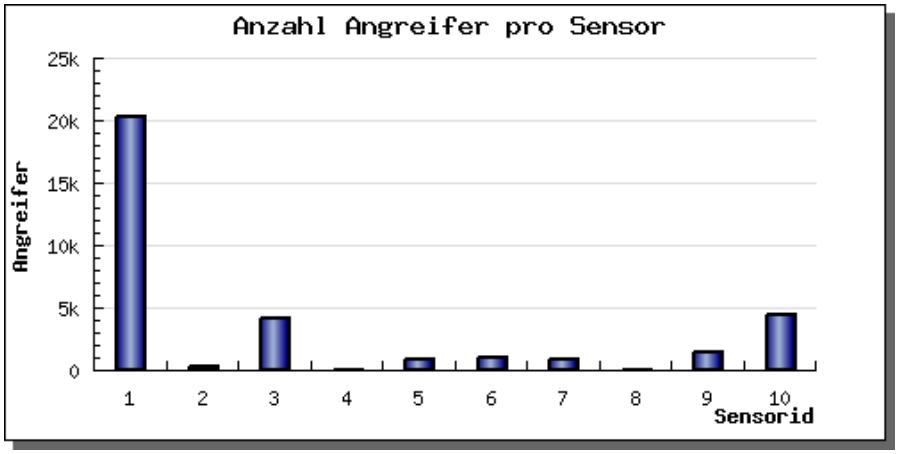


Abbildung 4.11: Absolute Anzahl an Angreifern pro Sensor im Datenset

zahl der Angreifer im Datenset wird durch die Anzahl unterschiedlicher IP-Adressen, von denen Angriffe ausgingen, bestimmt. Diese Methode hat den Vorteil, dass Angreifer, die gleichzeitig oder kurz hintereinander mehrere Angriffe auf die laufenden Sensorsysteme ausführten, nicht mehrfach gezählt werden. Andererseits wird das Ergebnis dadurch verfälscht, dass Angreifer, die zum Zeitpunkt eines Angriffs die gleiche IP-Adresse haben wie ein anderes Angreifersystem zu einem früheren Zeitpunkt, nicht als zusätzlicher Angreifer gewertet werden. Zudem werden Angreifer, die im Laufe des Betrachtungszeitraums mehrfach unter verschiedenen IP-Adressen angriffen, mehrfach einberechnet.

Um die Nachteile der ersten Identifikationsmethode zu umgehen, kann stark vereinfacht angenommen werden, dass jeder Angriff von einem gesonderten Angreifersystem ausging und die Anzahl der Angreifer gleich der Anzahl der Angriffe ist. Mit dieser Methode ist die Wahrscheinlichkeit einer deutlich zu hohen Angreiferzahl vergleichsweise hoch. Daher wird ein Angreifer in dieser Arbeit über seine öffentliche IP-Adresse definiert. Die Zahl der so identifierten Angreifer ist in Abbildung 4.11 dargestellt, insgesamt traten im Betrachtungszeitraum 32.740 unterschiedliche Angreifer auf, wobei etwa 3% aller Angreifer mehrere Sensorsysteme angegriffen haben.

Mit den derzeit genutzten Technologien ist es somit nicht ohne Weiteres möglich, ein System im Internet eindeutig zu identifizieren. Dies wäre zur eindeutigen Identifikation von Angreifersystemen jedoch notwendig.

Zu Beginn dieses Abschnitts werden Analyseergebnisse in Bezug auf die *Netztopologie* der Angreifer vorgestellt. Dabei werden hauptsächlich Unterschiede zwischen verschiedenen *Internet Service Provider (ISP)* herausgearbeitet. Im Folgenden werden dann geographische und technologische Strukturen der Angreifer dargestellt, wie zum Beispiel die Länder, aus denen die Angreifer angriffen.

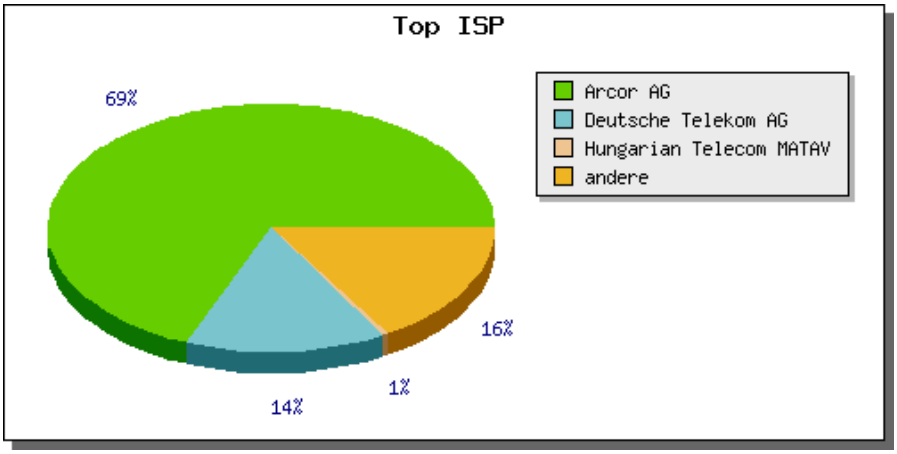


Abbildung 4.12: Am häufigsten auftretende Angreifer-ISP

4.2.1 Netztopologie

In diesem Abschnitt wird untersucht, ob bestimmte *Netzwerkeigenschaften* des Angreifers Rückschlüsse zulassen auf sein Angriffsverhalten. Netzwerkeigenschaften sind in diesem Zusammenhang beispielsweise der ISP des Angreifers oder die Distanz vom Angreifer zu dem angegriffenen Sensorsystem gemessen in so genannten *Hops*, also der Anzahl an Schritten, die die Datenpakete mit den Angriffsdaten vom Angreifer zum angegriffenen Sensorsystem zurücklegen muss. Die Zuordnung der ISP und autonomen Subnetze zu den IP-Adressen der Sensorsysteme und der Angreifer erfolgt über entsprechende Datenbanken der Firma *MaxMind* [LLC07].

Das Datenset für diese Arbeit enthielt Sensoren bei sechs verschiedenen ISP. Abbildung 4.12 zeigt, dass die meisten Angreifer von drei verschiedenen ISP kamen, *Arcor AG* mit knapp 70%, *Deutsche Telekom AG* mit etwa 15% und *Hungarian Telecom MATAV* mit nur 1% aller Angreifer. Alle anderen Angreifer-ISP machten jeweils weniger als 0,5% der Angriffe aus und sind in der Abbildung unter *andere* zusammengefasst. Insgesamt wurden Angriffe von 998 verschiedenen ISP registriert, von 115 davon kamen mindestens 10 Angriffe, von lediglich sieben Providern wurden mindestens 100 Angriffe registriert.

Abbildung 4.13 zeigt die Verteilung der Angreifer auf *autonome Subnetze* anhand von *AS Nummern*. Autonome Subnetze sind eigenständige Segemente des Internets, die über jeweils den gleichen Router mit dem restlichen Internet verbunden sind. In der Praxis kann ein Provider über mehrere autonome Subnetze verfügen, jedes autonome Subnetz ist jedoch nur genau einem Provider zugeordnet. Dies erklärt die Übereinstimmung der Verteilungen der Provider und der AS Nummern.

Im Falle der zwei Provider, die am häufigsten auftraten, besitzt jeder Provider jeweils nur eine AS Nummer. Der dritthäufigste Provider *Hungarian Telecom MATAV* hat schon 14 AS Nummern zur Verfügung, wovon jedoch nur eine die 1% der Angreifer ausmacht,

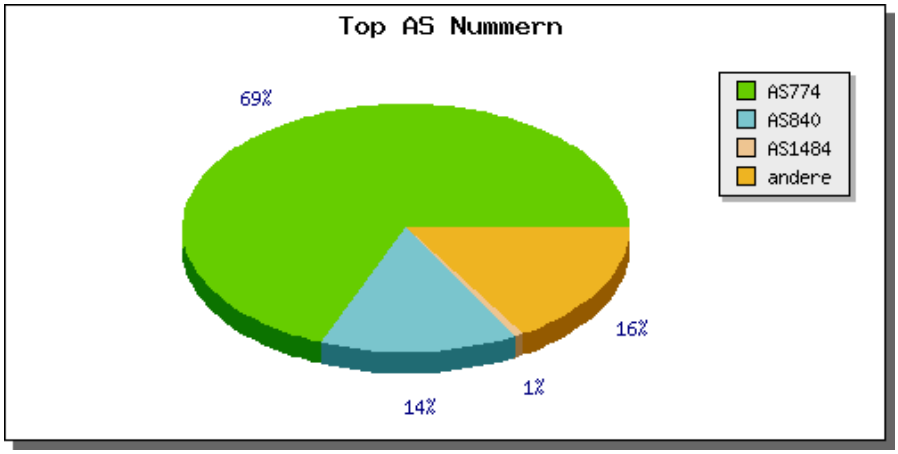


Abbildung 4.13: Am häufigsten auftretende Angreifer-AS Nummern

die über diesen Provider verbunden waren.

Bezüglich dem ISP und der AS Nummern der Angreifer zeigt Abbildung 4.14 bei einigen Sensoren einen starken Zusamenhang zwischen dem ISP beziehungsweise der AS Nummer des Angreifers und des angegriffenen Sensorsystems. Der Anteil der Angriffe, die von Angreifern beim gleichen Provider kamen, liegt insgesamt bei knapp über 80%. Bei der Betrachtung einzelner Sensoren lassen sich jedoch starke Unterschiede erkennen. Die Sensoren 2, 4, 5 und 6 wurden fast ausschließlich von Angreifern bei fremden Providern angegriffen. Diese Sensoren unterscheiden sich wie bereits in Abschnitt 4.1.1 erläutert in ihren Eigenschaften, sodass diese Beobachtungen verschiedene Schlüsse zulassen.

Sensor 2 stand im Netzwerk der Universität Mannheim. Durch die umfangreiche Systemadministration an einer Universität gibt es vermutlich innerhalb des Netzwerks der Universität verhältnismäßig wenig verseuchte Systeme, die Mitglieder in Botnetzen sind und damit potentielle Angreifersysteme für automatisierte Malware. Dies erklärt, dass dieser Sensor vorwiegend von außerhalb seines eigenen Subnetzes angegriffen wurde.

Eine ähnliche Vermutung liegt bei Sensor 4 nahe, der über *1&1* mit dem Internet verbunden war. Wie bereits erläutert, blockiert der Standardrouter, den dieser Provider mit seinen *DSL-Zugängen* ausliefert, alle ein- und ausgehenden Verbindungen auf kritischen Ports, wie beispielsweise den TCP-Ports 135 und 445. Daraus resultiert zum einen, dass dieser Sensor nur sehr wenig angegriffen wurde. Zum anderen können Systeme, die über diesen Router mit dem Internet verbunden werden, auch über die gefilterten Ports keine anderen Systeme kompromittieren, da die ausgehenden Pakete schon beim lokalen Router des Systems verworfen werden. Dies bietet offensichtlich einen umfangreichen Schutz vor der meisten automatisierten Malware.

Bei den beiden Sensoren 5 und 6 liegt keine solche Erklärung nahe. Sensor 5 war über die *Telekom* mit einer festen öffentlichen IP-Adresse angebunden, Sensor 6 bezog von

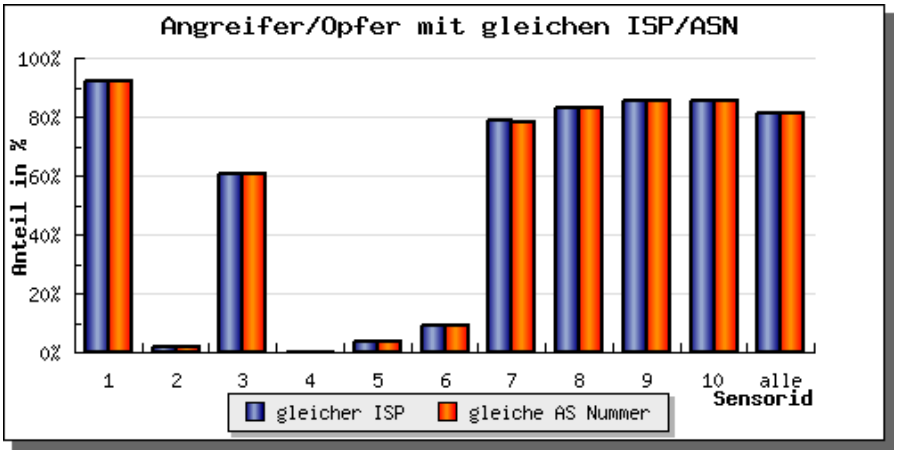


Abbildung 4.14: Anteil Angreifer mit gleichem ISP/gleicher AS Nummer wie das angegriffene Sensorsystem

QSC seine Internetverbindung. Dies passierte zwar nicht über eine tatsächlich feste IP-Adresse, jedoch gibt es bei diesem Anbieter keinen Neuaufbau der Verbindung nach 24 Stunden. Dadurch hatten diese Sensoren eine *quasi-feste* öffentliche IP-Adresse, da diese sich im Betrachtungszeitraum nie änderte. Da andere Sensoren bei der *Telekom* mit einer Anbindung über dynamische IP-Adressen mehrheitlich von Angreifern beim gleichen Anbieter angegriffen wurden, liegt die Vermutung nahe, dass die niedrige Rate von Angreifern aus dem gleichen Subnetz bei Sensor 5 und 6 mit der sich nicht ändernden öffentlichen IP-Adresse zusammenhängt.

In Abbildung 4.14 ist außerdem zu sehen, dass die Anteile von Angreifern vom gleichen ISP und von gleichen AS Nummern wie das Opfer nahezu identisch waren. Da die meisten der Provider der hier untersuchten Sensoren nur über eine AS Nummer verfügen, war diese Übereinstimmung zu erwarten.

Insgesamt kann man anhand dieser Daten vermuten, dass die Bedrohung durch automatisierte Malware von Angreifern beim gleichen Provider deutlich höher ist als von Angreifern fremder Provider. Dies bedeutet in der Praxis, dass durch einen internen Schutz jedes Providers ein großer Teil der Bedrohung ausgeschaltet werden kann und die Bedrohung aus Sicht eines Providers nicht nur *von außen* besteht.

Die Vermutung, dass automatisierte Malware so konfiguriert ist, dass sie vorwiegend ihr *nahe* Systeme im Sinne ihrer Netztopologie angreift wird auch durch Abbildung 4.15 bekräftigt. Diese Grafik zeigt, wie viele Hops die Angreifer von dem angegriffenen Sensorsystem entfernt waren. Bei etwa der Hälfte aller Angreifer konnte diese Distanz mit Hilfe von *p0f* ermittelt werden. Die mittlere gemessene Distanz zweier Sensoren beim gleichen ISP lag bei fünf Hops, bei Sensoren zweier unterschiedlicher ISP bei etwa zehn Hops. Bei etwa 85% davon war die Distanz kleiner gleich zehn Hops, bei der Hälfte war sie kleiner gleich fünf Hops.

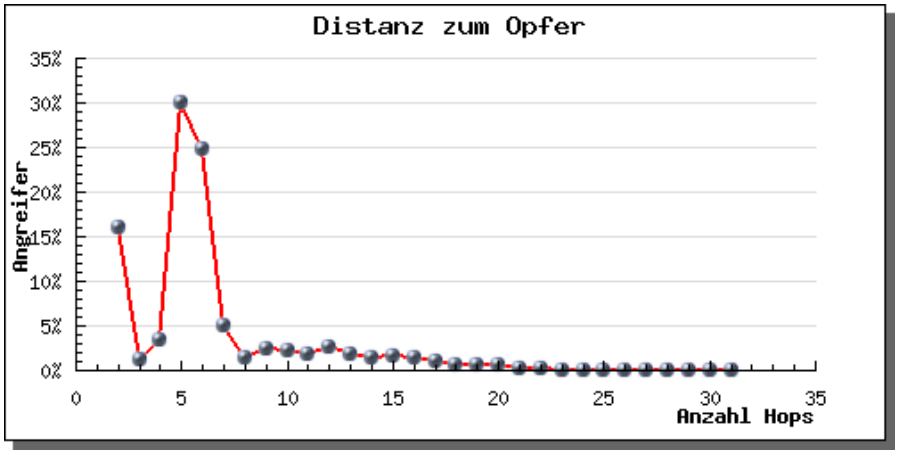


Abbildung 4.15: Distanz zwischen Angreifer und angegriffenem Sensorsystem in Hops

Je mehr Hops zwischen Angreifer und Sensorsystem liegen, desto höher ist für den Angreifer das Risiko, dass seine Datenpakete auf dem Weg zum Opfer durch eine Firewall gefiltert werden oder auf andere Art verloren gehen. Daher ist die Erfolgschance eines Angreifers bei Angriffen auf nahe Systeme deutlich besser als bei Angriffen auf weit entfernte Systeme.

Neben den Daten über die Entfernung zwischen Angreifer und angegriffenem Sensorsystem werden auch Daten darüber gesammelt, mit welcher Art von Verbindung der Angreifer mit dem Internet verbunden ist. Wie Abbildung 4.16 zeigt, waren die meisten Angreifer über *IPv6/IPIP* verbunden. Unter dieser Bezeichnung versteht man, dass der Angreifer seine Verbindung nicht direkt über ein Modem herstellt, sondern über das lokale Netzwerk über einen Router. Viele DSL-Nutzer sind auf diese Weise mit dem Internet verbunden, um über einen DSL-Anschluss mehreren Rechnern den Zugang zum Internet zu ermöglichen.

Der hohe Anteil von 72% ist hierbei jedoch überraschend, da Standardrouter für die DSL-Anbindung häufig mit *Network Address Translation (NAT)* und einer Firewall ausgestattet sind, die den Befall der Systeme mit automatisierter Malware verhindern sollte. Die vorliegenden Daten zeigen, dass dies jedoch nicht ausreichend ist, um die hinter dem Router liegenden Systeme zu schützen.

Mit insgesamt 20% der zweithäufigste Verbindungstyp der Angreifer war eine direkte DSL-Verbindung, bei der das Angreifersystem direkt mit dem Internet verbunden ist ohne einen dazwischen geschalteten Router. Hierbei werden von unerfahrenen Benutzern häufig keine Firewalls eingesetzt, was die Systeme besonders verwundbar macht. Ebenso verhält es sich mit den 3% der Angreifer, die über eine analoge Modemverbindung auf das Internet zugegriffen haben.

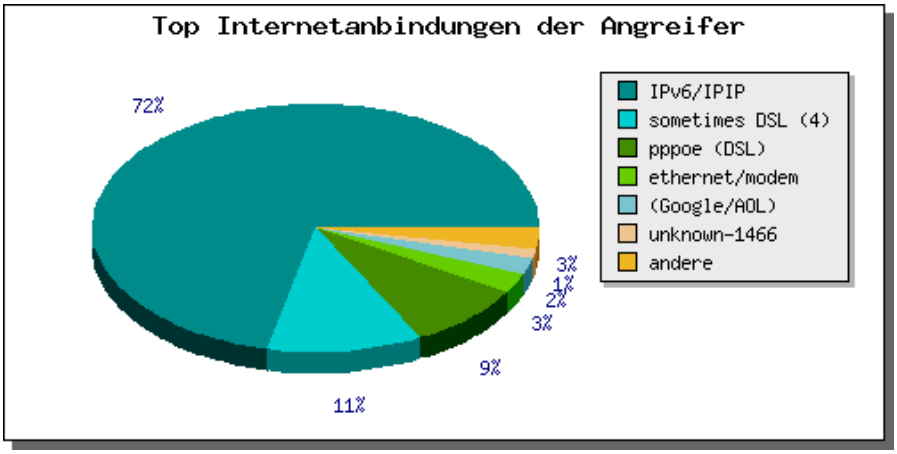


Abbildung 4.16: Verteilung der ermittelten Internetanbindung der Angreifer

4.2.2 Geographische Struktur

Zur Auswertung der geographischen Struktur der Angreifer wurde die Länderdatenbank von *MaxMind* [LLC07] zur Zuordnung von IP-Adressen zu Ländern zugrunde gelegt. Abbildung 4.17 zeigt die Herkunftsländer der registrierten Angreifer als Kuchendiagramm.

Mit etwa 85% kam die große Mehrheit aller Angreifer aus Deutschland. Dieser hohe Wert war zu erwarten, da in Kapitel 4.2.1 bereits beobachtet wurde, dass ein Großteil der Angreifer beim gleichen Provider waren wie das angegriffene Sensorsystem. Da deutsche Provider in der Regel nur Benutzer innerhalb Deutschlands beliefern, erklärt das den hohen Anteil an Angreifern aus Deutschland.

Die weiteren Angreifer bieten eine breite Spanne verschiedener Herkunftsländer. Insgesamt wurden Angreifer aus 107 verschiedenen Ländern regstriert. Dabei entspricht die Verteilung häufig beobachteten Mustern von verschiedensten Angriffsarten im Internet.

Betrachtet man nur Angreifer, die nicht über den gleichen ISP verbunden waren wie das angegriffene Sensorsystem, so reiht sich Deutschland mit etwa 1% der Angreifer im oberen Mittelfeld der Herkunftsländer ein. Dies zeigt insbesondere, dass das Risiko eines Angriffs von einem Angreifer bei einem fremden ISP innerhalb Deutschlands nicht höher war als das eines Angriffs von einem Angreifer aus dem Ausland.

Zur besseren Visualisierung der Ergebnisse wurde für das *BEAN*-Webinterface eine Landkartenansicht der geographischen Verteilung der Angreifer erstellt. Für die während des Betrachtungszeitraums dieser Arbeit registrierten Angreifer ist die Kartenansicht über die ganze Welt in Abbildung 4.18 dargestellt. Sie zeigt ebenso wie bereits das Kuchendiagramm eine Verdichtung der Angreifer in Deutschland.

Die Markierungen auf der Karte zur geografischen Lokalisierung der Angreifer sind in drei verschiedenen Farben dargestellt, um die Menge an Angriffen, die von dem jeweils

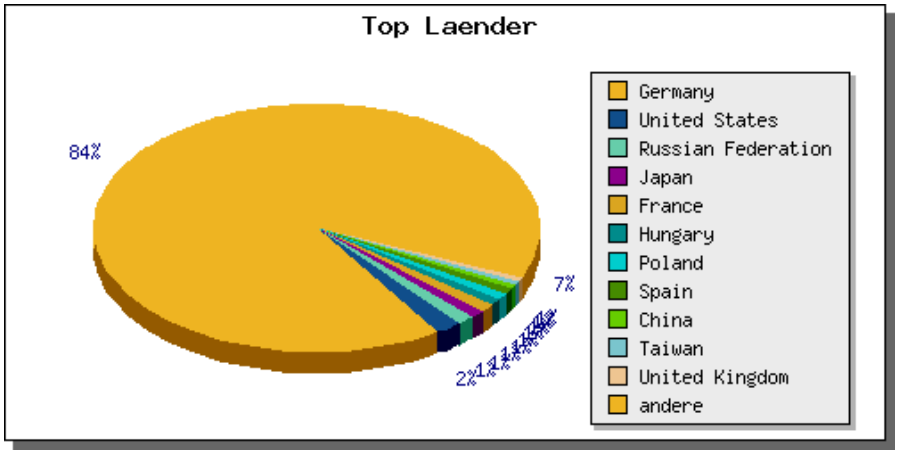


Abbildung 4.17: Verteilung der Herkunftsländer der Angreifersysteme

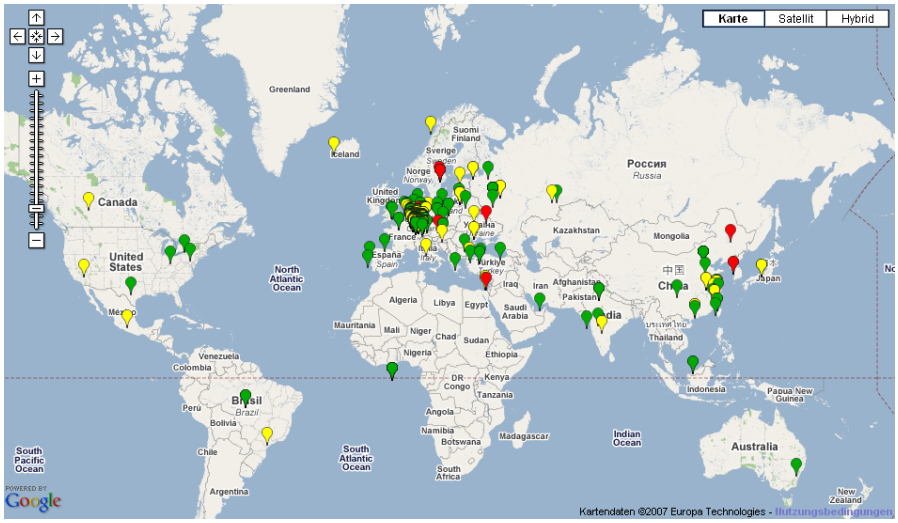


Abbildung 4.18: Google Map der Angreifer für die ganze Welt

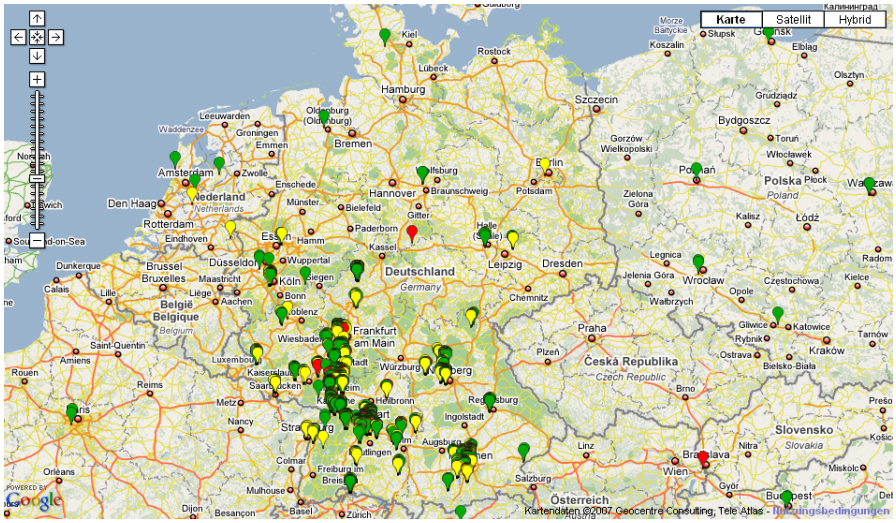


Abbildung 4.19: Google Map der Angreier für Deutschland

dargestellten Angreifer ausgingen, zu unterscheiden. Ein grüner Marker stellt Angreifer dar, von denen bis zu 15 Angriffe ausgingen. Angreifer, die als gelbe Markierung dargestellt sind, haben maximal 50 Mal ein Sensorsystem angegriffen. Rote Markierungen stellen alle Angreifer dar, von denen mehr als 50 Angriffe ausgingen.

Aufgrund der festgestellten Konzentration der Angreifer auf Deutschland ist in Abbildung 4.19 die Landkartenansicht vergrößert und auf Deutschland fokussiert. Die Mehrheit der Angreifer konnten im Südwesten Deutschlands lokalisiert werden. Dort lagen auch die Standorte der Sensorsysteme, was zudem einen regionalen Zusammenhang vermuten lässt.

4.2.3 Technologische Struktur

Eine weitere Eigenschaftsdimension, nach der die Angreifer untersucht wurden, stellt die verwendete Technologie dar. Als Daten können mit Hilfe der Sensoren die Betriebssysteme ermittelt werden, unter denen das Angreifersystem betrieben wird. Das hierzu benutzte Tool *p0f*, das in Kapitel 2.2 näher beschrieben wurde, konnte bei etwa der Hälfte aller Angreifer das genutzte Betriebssystem feststellen. Es gibt hierbei keinerlei Grund zu der Annahme, dass einzelne Betriebssysteme schlechter erkannt werden können als andere.

Abbildung 4.20 zeigt für die Angreifer, bei denen ein Betriebssystem erkannt werden konnte, die Verteilung der genutzten Betriebssysteme. Überraschend ist der mit 42% hohe Anteil an Systemen, die unter *Windows 98* betrieben wurden, da *Microsoft* für dieses Betriebssystem schon seit Mitte 2006 keinen Support mehr anbietet [Cor07].

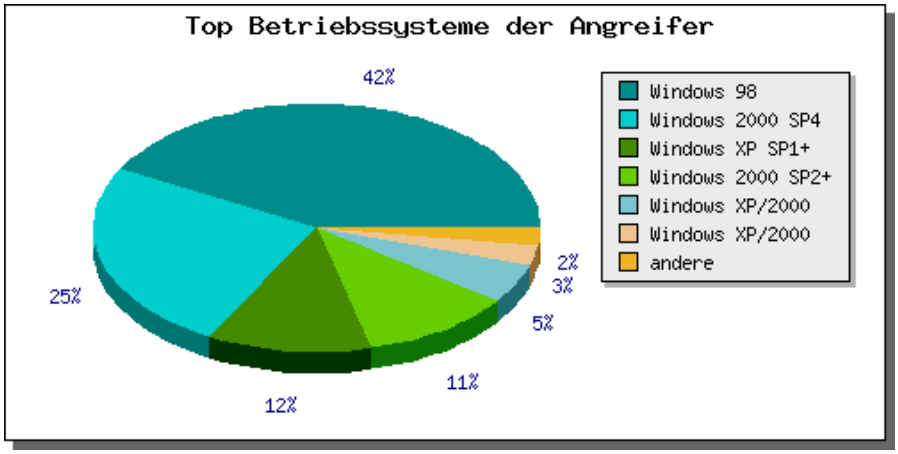


Abbildung 4.20: Betriebssysteme, unter denen die Angreifer betrieben wurden

Die Schwachstellen, die auf den Sensoren für diese Arbeit emuliert wurden, sind jedoch zum Großteil solche, die bereits vor 2006 bekannt waren und für die *Windows 98* entweder nicht verwundbar ist oder für die ein *Patch* angeboten wird. Daher liegt die Vermutung nahe, dass diese Systeme von unerfahrenen Benutzern betrieben werden, die sich der Gefahren im Internet nicht bewusst sind und dementsprechend keine Patches einspielen. Auch die Tatsache, dass *Windows 98* aufgrund der begrenzten Möglichkeiten nur noch auf alter Hardware sinnvoll eingesetzt wird, unterstützt die Vermutung des unerfahrenen Benutzers, da dieser häufig keine aktuelle Hard- und Software benötigt.

Da *Windows 98* im Gegensatz zu neueren Versionen des Betriebssystems *Windows* keine automatische Benachrichtigung bei neuen *Updates* bietet, wird der Benutzer auch nicht auf die Möglichkeiten zur Absicherung von Sicherheitslücken aufmerksam gemacht. Viele Angreifer benutzen aber auch *Windows 2000* und *Windows XP*, die beide über eine solche automatische Benachrichtigung verfügen. *Windows 2000* wurde bei 36% der Angreifersysteme als genutztes Betriebssystem erkannt.

Viele aktuelle Computersysteme werden mit *Windows XP* als Betriebssystem ausgeliefert, das somit das zur Zeit am weitesten verbreitete Betriebssystem im Privatanwenderbereich ist. Über die betrachteten Daten werden etwa 20% der Angreifersysteme mit *Windows XP* betrieben. Dies zeigt, dass offensichtlich die eingebauten Sicherheitsmechanismen bei diesem Betriebsssytem, wie beispielsweise die Möglichkeit von automatischen Sicherheitsupdates, zumindest in Teilen greifen.

Angreifer von einem *Unix*- bzw. *Linux*-basierten Betriebssystem sind keine registriert worden. Dies ist das erwartete Ergebnis, da die meiste automatisierte Malware Sicherheitslücken in *Windows*-Betriebssystemen ausnutzt, die in *Linux*-Systemen nicht vorhanden sind. Somit können Systeme unter *Linux* nicht von dieser Malware befallen und somit auch nicht unbemerkt Mitglied eines Botnetzes sein, um andere Systeme zu kompromittieren.

Linux wird noch immer im Anwenderbereich kaum genutzt. Das derzeitige Hauptverbreitungsgebiet von Linux sind Server, dort wird es häufig eingesetzt, aber auch meist gut gewartet und administriert. Dadurch ist die Attraktivität, Malware für dieses Betriebssystem zu entwickeln, nicht besonders hoch, da die Menge an potentiellen Opfern zu gering ist.

4.3 Malware Binaries

Neben den Angriffsdaten werden mit *BEAN* Daten über die Malware gesammelt, die der Angreifer durch die Angriffe verbreitet. In diesem Abschnitt werden Analyseergebnisse über die gesammelten Malware Binaries sowie über deren Metadaten, wie beispielsweise dem Zeitpunkt ihres Erstauftretens, vorgestellt.

Ein Malware Binary gilt dann als unbekannt, wenn es seit Beginn des Betrachtungszeitraums noch von keinem Sensor heruntergeladen wurde. Als Identifikationsmerkmal wird dafür der *MD5-Hash* des Binaries verwendet. Insgesamt wurden im betrachteten Zeitraum 59.141 Malware Binaries heruntergeladen, davon 729 einzigartige.

4.3.1 Unbekannte Binaries

Im ersten Schritt wird für jedes Malware Binary das erste Auftreten im System betrachtet, ungeachtet davon, wie viele weitere Male das gleiche Binary danach heruntergeladen wurde. Ein bisher für das System unbekanntes Binary enthält mit einem höheren Risiko neue Malware, für die noch wenige oder gar keine Schutzmechanismen existieren als ein bereits mehrfach aufgetretenes Binary. Daher sind solche Binaries von besonderem Interesse bei der Einschätzung der aktuellen Gefährdung.

Abbildung 4.21 zeigt im Verlauf, wie viele unbekannte Binaries das System an den Tagen des Betrachungszeitraumes registriert hat. Am ersten Tag der Aufzeichnung wurden mit 65 Binaries die meisten unbekannten Binaries heruntergeladen. Dass an den folgenden Tagen nur noch deutlich weniger Binaries unbekannt waren, war zu erwarten, da keines der Binaries zu Beginn der Aufzeichnung an einem vorherigen Tag bereits aufgetreten sein kann.

Im weiteren Verlauf bewegten sich die Zahlen zwischen keinem und 22 unbekannten Binaries pro Tag. Es traten jedoch an nahezu jedem Tag im Betrachtungszeitraum bisher unbekannte Binaries auf. Aufgrund des Betrachtungszeitraums von zehn Wochen lässt dies den Schluss zu, dass ständig eine nicht unbeträchtliche Anzahl neuer oder mutierter Malware in Umlauf gebracht wird.

Im Rahmen zeitlicher Analysen wird ebenfalls untersucht, wie viele bisher unbekannte Binaries zu welcher Tageszeit auftraten. Das Ergebnis dieser Untersuchung zeigt Abbildung 4.22. Für das Erstauftreten von Malware Binaries ist ein ähnlicher Verlauf zu

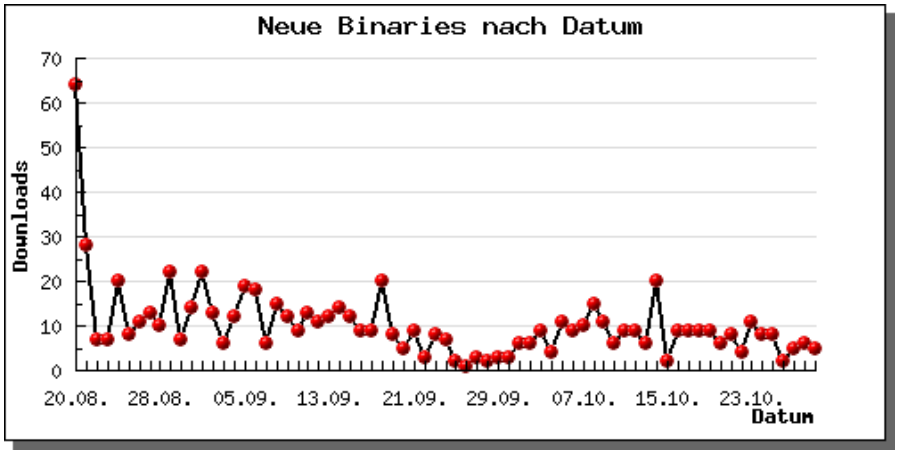


Abbildung 4.21: Anzahl neuer Binaries pro Tag im Betrachtungszeitraum

erkennen wie für alle Downloads in Abbildung 4.3. Dies legt nahe, dass die Wahrscheinlichkeit für einen Download eines neuen Binaries zu jeder Zeit genauso hoch ist wie die für den Download eines beliebigen Binaries.

Die nächste Auswertung bezieht sich auf den Verbreitungsweg der Malware Binaries. Es wird untersucht, wie viele Binaries von Angreifern beim gleichen ISP heruntergeladen wurden. Dies traf lediglich auf 38% aller bisher unbekannten Binaries zu, 62% wurden von Angreifern bei anderen ISP heruntergeladen. Betrachtet man im Vergleich Abbildung 4.14 so sieht man, dass im Gegensatz zu den Binaries über 80% aller Angreifer über den gleichen ISP wie das angegriffene Sensorsystem mit dem Internet verbunden waren. Innerhalb des Netzes eines ISP ist demnach jeweils überwiegend gleiche Malware im Umlauf.

Dies bedeutet zum einen, dass innerhalb des eigenen Providernetzes die Gefahr für ein System, mit unbekannter Malware kompromittiert zu werden, geringer war. Auf der anderen Seite wurden über die nur etwa 20% der Angriffe, die nicht vom eigenen Provider kamen, weit über die Hälfte der unbekannten Binaries verteilt. Damit war zwar die Gefahr eines Angriffs von einem fremden Provider deutlich geringer, jedoch war die Gefährdung durch diese Angriffe hoch, da sie häufig unbekannte Binaries verbreitet haben, für die eventuell noch kein Schutz auf dem System installiert wurde.

Eine weitere interessante Fragestellung in Bezug auf die Verbreitungswege der Malware ist, über welche Schwachstellen die bisher unbekannten Binaries das System befallen haben. Diese Auswertung ist mit den vorhandenen Daten jedoch nicht möglich, da das *log-surfnet*-Modul von *nepenthes* für den Download die Schwachstelle nicht abspeichert. Lediglich für den Angriff werden diese Daten registriert, jedoch ist es nicht möglich, die Datensätze zu einem Angriff und dem daraus resultierenden Download eindeutig einander zuzuordnen. Das Entwickeln einer Heuristik wäre wegen der hohen Dichte, in der Angriffe registriert wurden, nicht sehr genau und würde keine allgemeine Aussage

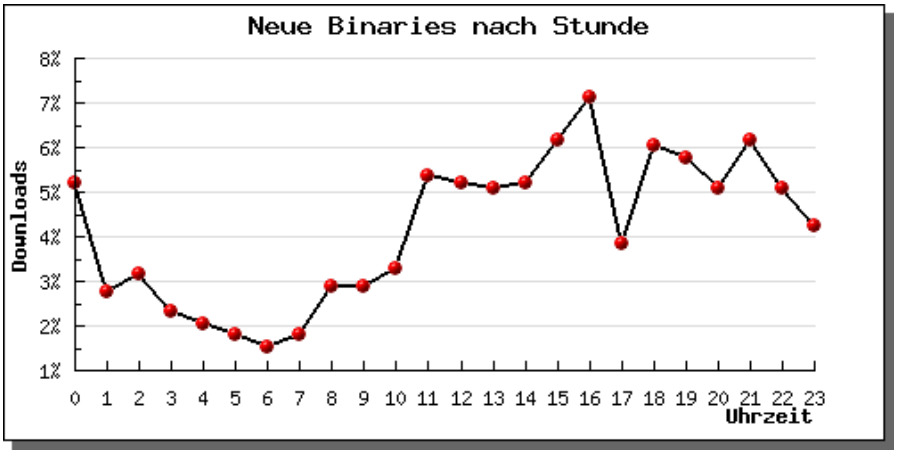


Abbildung 4.22: Anzahl neuer Binaries als Verlauf über die Uhrzeit

zulassen. Es wäre hierfür sinnvoll, das Modul zum Speichern der registrierten Angriffe in die Datenbank dahingehend anzupassen, dass eine solche Zuordnung möglich ist.

Wie bereits im Rahmen der Analysen der Angreifer, ist auch an dieser Stelle das Untersuchen der Binaries auf ihre Herkunft von Interesse. In Abbildung 4.23 ist die Verteilung der Binaries nach dem Herkunftsland bei ihrem Erstauftreten aufgezeigt. Ebenso wie bereits bei den Herkunftsländern der Angreifer, wurden die meisten Malware Binaries erstmalig von Angreifern aus Deutschland heruntergeladen. Jedoch macht dieser Teil nur 41% der Binaries aus, wohingegen 85% aller Angreifer aus Deutschland kamen wie Abbildung 4.17 gezeigt hat. Betrachtet man nur Angreifer, von denen erfolgreich ein Malware Binary heruntergeladen wurde, sind sogar 89% davon Systeme aus Deutschland.

Bei den neuen Binaries, die nicht aus Deutschland kamen, führt mit 9% Amerika die Liste an und es folgen weitere Länder, die bei der Betrachtung der Herkunft von Malware und Spam häufig als Hauptverbreiter auftauchen. Der Anteil der Länder, aus denen weniger als 2% der neuen Binaries kamen, liegt mit 25% vergleichsweise hoch. Bei den Binaries ausländischer Herkunft ist die Streuung groß, wie dies bereits bei den Angreifern zu sehen war.

Diese Beobachtungen zeigen, dass zwar die Mehrheit der Angreifer, die deutsche Systeme angriffen, aus dem Inland kamen, jedoch die Bedrohung durch neue, unbekannte Malware aus dem Ausland deutlich höher war. Die Vermutung ist daher, dass der Ursprung der Botnetze, über die solche automatisierte Malware verbreitet wird, häufig nicht in Deutschland liegt.

4.3.2 Bekannte Binaries

Im vorangegangenen Abschnitt wurde jeweils nur das Erstauftreten jedes Malware Binaries untersucht. Von den im Betrachtungszeitraum gesammelten Binaries wurden über

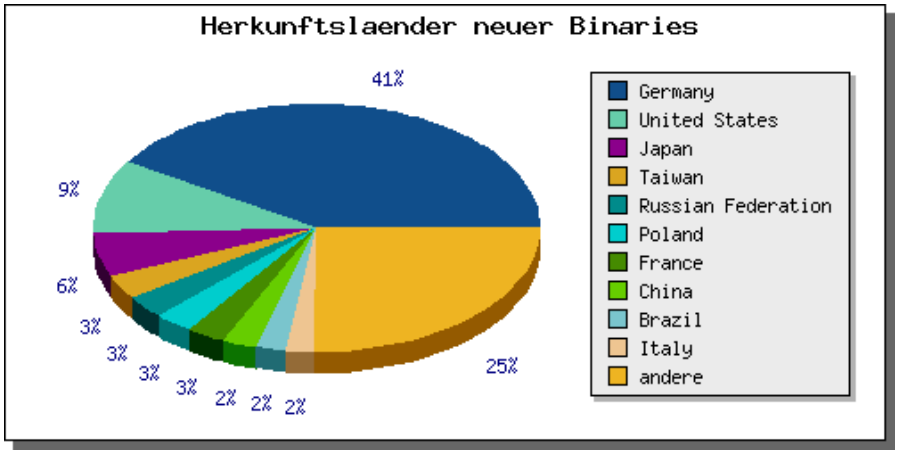
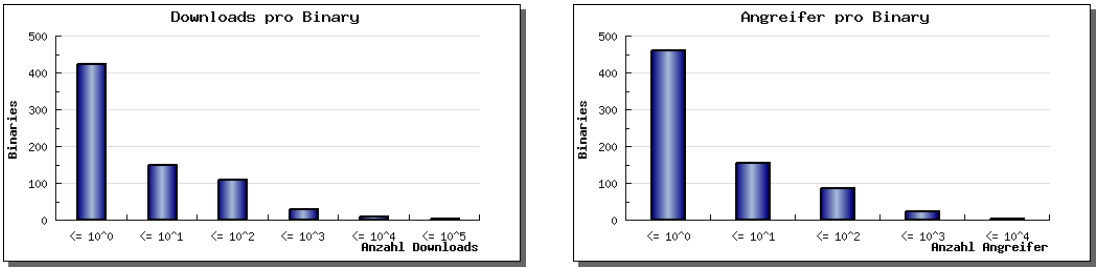


Abbildung 4.23: Verteilung der Herkunftsländer des Angreifers bei Download eines neuen Binaries



(a) Downloads

(b) Verschiedene Angreifer

Abbildung 4.24: Anzahl Downloads/verschiedener Angreifer pro Binary

40% mehr als einmal heruntergeladen. Daher werden in diesem Abschnitt Untersuchungen über alle registrierten Downloads von Malware Binaries gemacht, nicht nur über ihr Erstauftreten.

Im Durchschnitt wurde jedes Binary 81 Mal heruntergeladen. Besonders auffällig ist dabei die große Spanne bei der Anzahl der Downloads pro Binary. Während zwei Binaries über 11.000 Mal heruntergeladen wurden, gab es 429 Binaries, die nur ein einziges Mal auftraten. Abbildung 4.24 (a) zeigt die Verteilung der Downloads auf die Binaries.

Bemerkenswert sind die beiden Binaries, die mit Abstand am häufigsten heruntergeladen wurden. Sie traten beide zum ersten Mal wenige Minuten nach dem Beginn des Betrachtungszeitraumes auf. Die Virenscanner erkannten beide als *RBots*, eine Art Bot, der über *IRC* kommuniziert.

Des Weiteren wurde die Anzahl der unterschiedlichen Angreifer untersucht, von denen ein Binary heruntergeladen wurde. Das Ergebnis ist in Abbildung 4.24 (b) dargestellt. Die resultierenden Werte sind mit denen der Anzahl an Downloads pro Binary vergleichbar.

Die Mehrheit der Binaries kam nur von einem einzigen Angreifer. Über 150 Binaries wurden aber schon von zwei bis zehn Angreifern heruntergeladen, 88 Binaries von zehn bis 100 Angreifern und sogar drei von über 1.000 verschiedenen Angreifern.

Neben der reinen Anzahl an Downloads pro Binary ist auch die Frequenz, mit der die Binaries heruntergeladen wurden, von Interesse. Es gibt Binaries, wie die beiden bereits erwähnten, die über den gesamten Betrachtungszeitraum konstant etwa alle sieben Minuten auftraten. Andere Binaries wurden über etwa 24 Stunden 45 Mal mit einer Frequenz von etwa 30 Minuten heruntergeladen, danach jedoch nicht mehr. Ein weiteres Extrem sind Binaries, die etwa acht Wochen nach ihrem Erstauftreten ein zweites Mal erscheinen, jedoch dazwischen kein einziges Mal.

Diese Auswertungen zeigen, dass der Mehrfachdownload von Binaries offensichtlich keiner zeitlichen Struktur unterliegt. Es ist anzunehmen, dass die Häufigkeit und die Frequenz der Downloads von Malware Binaries hauptsächlich von der Größe des Botnetzes abhängt, über das die Malware verbreitet wird.

4.3.3 Ergebnisse der Virenscanner

Bisher wurden lediglich Analysen vorgestellt, die die Binaries nach ihren Verbreitungsstrukturen, wie beispielsweise ihrer Herkunft, untersuchen. In diesem Abschnitt werden nun die Ergebnisse der Analysen über die Malware Binaries selbst vorgestellt, die durch die Virenscanner gewonnen wurden.

Alle Untersuchungen in diesem Abschnitt beziehen sich sowohl auf die lokalen Virenscanner, wie sie in Kapitel 3.2.1 vorgestellt wurden, als auch auf die Ergebnisse von *VirusTotal*. Da bei *VirusTotal* jedoch jedes Binary nur einmalig untersucht wurde, können hierfür keine Auswertungen über Ergebnisverläufe ermittelt werden. Mit den lokalen Virenscannern wurden alle Binaries alle vier Stunden untersucht, sodass daher über jedes Binary umfangreiche Verlaufsdaten vorliegen.

Zunächst werden die Ergebnisse der lokalen Virenscanner für neue, unbekannte Binaries betrachtet. Dazu wird als erstes untersucht, wie lange es dauert, bis neue Binaries von den Virenscannern als Malware erkannt werden. Dazu werden die Binaries zunächst dahingehend analysiert, wie viele bei ihrem Erstauftreten direkt im ersten Lauf von mindestens einem der lokalen Virenscanner als Malware erkannt wurden. Da die Scanner alle vier Stunden laufen, gilt es als direktes Erkennen, wenn ein Scanner innerhalb von vier Stunden nach Erstauftreten eines Binaries ein Ergebnis liefert.

Es konnte beobachtet werden, dass mit 79% die Mehrheit der Binaries direkt von mindestens einem der lokalen Virenscanner erkannt wurde. Jedoch sind 21% der Binaries von keinem der vier Scanner direkt als Malware identifiziert worden. Das bedeutet, dass fast ein Viertel der Binaries Systeme, die mit einem dieser vier Scanner geschützt werden, ungehindert kompromittieren können.

Die beste Erkennungsrate neuer Binaries lieferte unter den lokalen Virenscannern mit 73% der Scanner *Antivir*, die schlechteste mit nur 35% *Norman*, welches von den vier

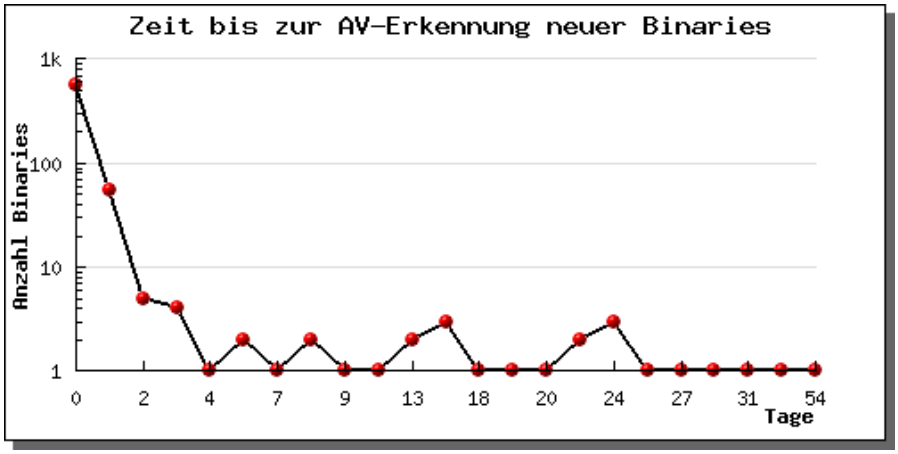


Abbildung 4.25: Zeit bis zur ersten Erkennung neuer Binaries durch mindestens einen Virenscanner

Scannern die einzige kommerzielle Lösung ist. Da es sich bei den eingesetzten Scannern um vor allem auf privaten Systemen verbreitete Lösungen handelt, zeigt diese hohe Zahl deutlichen Handlungsbedarf zur Verbesserung dieser Scanner.

Abbildung 4.25 zeigt im zeitlichen Verlauf, wie viele Tage es dauerte, bis die Binaries durch mindestens einen lokalen Virenscanner erkannt wurden. Es ist deutlich zu erkennen, dass für die meisten Binaries am Tag ihres Erstauftretens mindestens ein Scanner bereits ein Ergebnis lieferte. Eines der Binaries wurde erst nach 69 Tagen von einem Scanner als Malware erkannt. Im Schnitt dauerte es einen Tag bis zum ersten Ergebnis. 70 Binaries – also 1% – wurden jedoch bis zum Ende des Betrachtungszeitraumes von keinem der lokalen Virenscanner als Malware identifiziert.

Für die nächsten Auswertungen werden alle Vorkommen der gesammelten Binaries einbezogen. Dazu wird zunächst bewertet, wie hoch die Erkennungsraten der lokalen Virenscanner und der Scanengines bei *VirusTotal* über alle gesammelten Binaries waren.

Wie bereits beim Erstauftreten der Binaries beobachtet werden konnte, lieferte auch über alle gesammelten Binaries von den lokalen Virenscannern der Scanner *Antivir* die besten Resultate. Er erkannte 86% der Binaries als Malware. An zweiter Stelle liegt *ClamAV* mit 67% erkannten Binaries. Nur etwa die Hälfte aller Binaries erkannten die anderen beiden Scanner *F-Prot* und *Norman*.

Die Ergebnisse von *VirusTotal* bieten eine Erkennungsrate von 100% über die gesammelten 729 Binaries. Volle Erkennung bedeutet, dass jedes untersuchte Binary von mindestens einem der Scanner als Malware erkannt wurde. Schlüsselt man die Ergebnisse nach Virenscanner auf, so sieht man, dass der beste Scanner eine Erkennungsrate von etwa 98% bot, der schlechteste etwa 16%.

Beim besten Scanner handelte es sich um *Webwasher-Gateway*, was keine gewöhnliche Einzelplatzlösung darstellt, sondern eine Gatewaylösung für große Netzwerke. Der zweit-

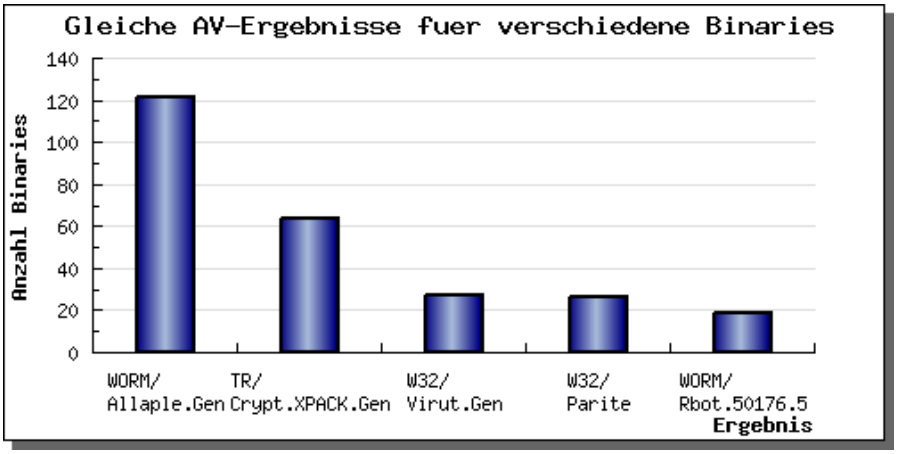


Abbildung 4.26: Unterschiedliche Binaries mit gleichem AV-Ergebnis (Antivir)

beste Scanner bei *VirusTotal* war wie bereits bei den lokalen Scannern *AntiVir*. Daher werden die Resultate dieses Scanners für die folgenden Untersuchungen, die sich auf die Ausgaben eines Virenscanners beziehen, zugrunde gelegt.

Da das einzige Unterscheidungsmerkmal für Malware Binaries deren *MD5-Hash* ist, können verschiedene Binaries den gleichen Schadcode enthalten und somit zum gleichen Antivirenergebnis führen. Abbildung 4.26 zeigt für den Scanner *Antivir* die fünf Antivirenergebnisse, für die jeweils die meisten unterschiedlichen Binaries das gleiche Ergebnis brachten. Als unterschiedlich gelten zwei Binaries, wenn sie unterschiedliche *MD5-Summen* haben.

Die ersten drei Ergebnisse *WORM/ Allaple.Gen*, *TR/ Crypt.XPACK.Gen* und *W32/ Virut.Gen* stehen alle für ein generisches Ergebnis, es handelt sich also um Erkennungsroutinen, die gemeinsame Familienmerkmale verschiedener Varianten der Malware erkennen. Daher ist es nicht verwunderlich, dass bis zu etwa 120 Binaries in diese Kategorie fallen. Mit immerhin fast 30 unterschiedlichen Binaries hat der Wurm *W32/Parite* ein breites Spektrum an verschiedenen Dateien, über die er sich verbreitet. Auch der Wurm *WORM/Rbot.50176.5* wurde noch in 20 der Binaries von *Antivir* erkannt.

Zustande kommen solche Ergebnisse dadurch, dass automatisierte Malware häufig so implementiert ist, dass sie sich beim Weiterverbreiten selbst leicht modifiziert. Diese Art von Malware wird als *polymorphe Malware* bezeichnet. Durch diese Mechanismen hoffen Programmierer solcher Malware, dass Virenscanner die Malware schlechter oder gar nicht entdecken und sie so an den Schutzmechanismen vorbei ins System gelangt.

Bei der mehrfachen Untersuchung von Binaries kann es vorkommen, dass ein Virenscanner bei einem erneuten Scan des Binaries nach einigen Stunden oder Tagen ein anderes Ergebnis liefert als bei der ersten Untersuchung. Diese Änderung von Ergebnissen ist durch das Aktualisieren der Scanner mit neuen Signaturen bedingt. Es wurde daher für

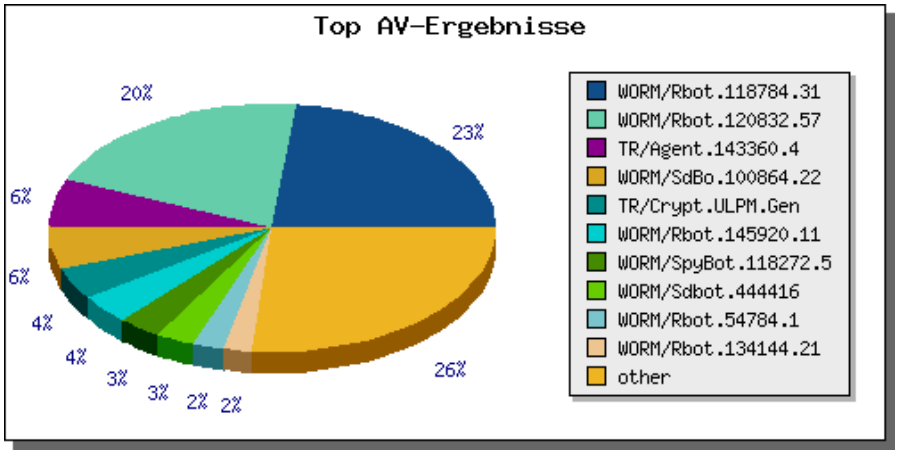


Abbildung 4.27: Verteilung der häufigsten Virenscan-Ergebnisse für Scanner 2 (Antivir)

alle gesammelten Binaries und lokalen Virenscanner untersucht, wie hoch der Anteil der Binaries ist, bei denen dieses *Ändern der Meinung* auftrat.

Bei knapp 40% der Binaries haben Scanner ihr Resultat revidiert oder konkretisiert. Beispielsweise hat der Scanner *Norman* bei einigen Binaries zunächst nur erkannt, dass es sich um Malware handelt, hat aber noch keine Aussage darüber treffen können, welcher Art die Malware ist. In einem späteren Lauf nach einem Update des Scanners lieferte dieser dann für diese Binaries häufig einen konkreteren Befund. Die Heuristiken der Virenscanner sind also offensichtlich in der Lage, beispielsweise durch den Einsatz von *Sandbox-Technologien*, anhand bestimmter Merkmale der Binaries prinzipiell die schädliche Absicht des Binaries zu erkennen, ohne genau spezifizieren zu können, um welche Art Malware es sich handelt.

Betrachtet man die Ergebnisse des Virenscanners *Antivir* ohne Einschränkung über alle gesammelten Binaries, ergibt sich die Verteilung wie in Abbildung 4.27 dargestellt. Die beiden Würmer, die am häufigsten in Binaries erkannt wurden, sind zwei verschiedene Varianten eines *RBot*, also eines *IRCBots*. Diese Art von Wurm steckt in den beiden Binaries, die über 11.000 Mal heruntergeladen wurden. Daher war zu erwarten, dass sie einen hohen Anteil der Ergebnisse der Scanner ausmachen. Etwas über ein Viertel der Ergebnisse trat bei weniger als 2% der Binaries auf. Die hohe Vielfalt der verschiedenen Resultate zeigt, wie zentral der Einsatz eines gut funktionierenden Virenscanners zur Abwehr von Malware ist.

4.3.4 Ergebnisse der CWSandbox

Im Folgenden werden die Ergebnisse, die die *CWSandbox* aus den 729 im Betrachtungszeitraum gesammelten Malware Binaries ermittelt hat, dargestellt und interpretiert. Als erstes ist festzustellen, dass 22% der zur *CWSandbox* übermittelten Binaries

Angelegter *Mutex*	Anzahl
a1c21d0e0d6af099e3b6ed38f9d85d58ced8	154
e3f44ac0094cfa2ba3b711b29a822b0ed8bc	98
jhdheddfffffhjk5trh	82
b0tss	56
jhdgcjhasgdc09890gjasgcjhg2763876uyg3fhg	46
jhdheruhfrthkgjhtjkghjk5trh	29
b1790f4c06f035c083b712e3f4f6a1a8c30c	28
dcf7d2f7071938ba83b50c70eedd5ceb8984	25
00y010x36	20
cd9f82a30d4ee5d683ae1fc7575b139ab344	16

Tabelle 4.1: Die zehn häufigsten Mutex

keine ausführbaren Programme waren. Daher konnten diese Binaries von der *CWSandbox* nicht weiter analysiert werden. Es ist davon auszugehen, dass bei diesen Binaries Übertragungsfehler beim Download auftraten.

Für die übrigen 569 Malware Binaries wurde von der *CWSandbox* eine dynamische Verhaltensanalyse erstellt. Dabei wurden Veränderungen im Betriebssystem, die die Malware vornahm ebenso wie Netzwerkverbindungen, die sie aufbaute, untersucht. Im Folgenden werden die Ergebnisse zu erstellten *Mutex*, erstellten Dateien, veränderten *Registrykeys* und aufgebauten Netzwerkverbindungen präsentiert.

Zunächst wird betrachtet, unter welchem Namen die Malware in dem Betriebssystem *Mutexes* erstellt hat. Unter einem *Mutex* versteht man ein Verfahren, das verhindern soll, dass mehrere Programme auf die gleichen Daten zugreifen. Praktisch bedeutet dies, dass unterschiedliche Malware Binaries der gleichen Familie das System nicht mehrfach befallen. Jedes Binary kann beliebig viele Mutexes auf einem System einrichten.

Die sechs häufigsten Mutexes, die 302 Mal alle unter einer Bezeichnung *CTF.[...]* abgelegt wurden, wurden alle jeweils vom gleichen Malware Binary angelegt. Bei diesen Mutexes handelt es sich um solche, die der *Microsoft Internet Explorer* anlegt. Es ist daher zu vermuten, dass die Malware, die diese Mutexes auslöste, Dateien über den *Internet Explorer* nachgeladen hat.

Tabelle 4.1 listet die zehn am häufigsten festgestellten Mutexes, die nicht vom Betriebssystem angelegt wurden, mit der jeweiligen Anzahl an Binaries, von denen sie erstellt wurden, auf. Die Mutexes, deren Name aus anscheinend zufälligen Zeichenfolgen besteht, wurden vermutlich alle von Binaries der gleichen Malware-Familie angelegt.

Der nächste Aspekt, nach dem die Ergebnisse der *CWSandbox* untersucht wurden, sind Dateien, die während der Analyse erstellt wurden. Dabei fällt auf, dass es keine Dateien gibt, die von einer Mehrzahl der Malware im gleichen Verzeichnis angelegt wurden. Das häufigste Vorkommen einer Datei lag bei 85, was etwa 8% der untersuchten Binaries entspricht.

Betrachtet man allerdings nur die Pfade, in denen Dateien abgelegt wurden, kann eine Häufung beobachtet werden. Die große Mehrheit der Dateien wurden im Ordner `C:\DOKUME~1\HANSWU~1\LOKALE~1\TEMP\`erstellt. Dort legt üblicherweise der *Internet Explorer* temporäre Dateien ab. Dies bestätigt die Vermutung, dass viele der Binaries während der Kompromittierung über den *Internet Explorer* Dateien nachgeladen hat. Verhältnismäßig häufig wurden auch Dateien direkt im Ordner `C:\WINDOWS\system32\` abgelegt, also direkt im Systemordner des Betriebssystems.

Ebenso können in der Veränderung der *Registrykeys* bestimmte Muster erkannt werden. Die große Mehrheit der Binaries änderte Einstellungen der Netzwerkverbindung, indem der *Registrykey* `HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters` verändert wurde. Zudem konnte eine häufige Änderung des *Keys* `HKLM\SOFTWARE\Microsoft \Windows\CurrentVersion\Run` beobachtet werden. Dieser ist für die Registrierung der Programme, die über die *Autostart*-Funktion des Betriebssystems beim Systemstart hochgefahren werden, zuständig. Eine große Anzahl an Malware hat so sichergestellt, dass von ihr eingeschleuste Software auch bei einem Neustart des kompromittierten Systems wieder gestartet wird. Dies ist vor allem für die Steuerung von Bots nötig.

Die *CWSandbox* hat außerdem während der Analyse Daten darüber aufgezeichnet, ob die Malware Netzwerkverbindungen nutzte. Dies war bei der überwiegenden Zahl der untersuchten Malware der Fall. Zudem versucht die *CWSandbox* zu ermitteln, welches Netzwerkprotokoll genutzt wurde. Insgesamt konnten bei den untersuchten Malware Binaries 83.953 Netzwerkverbindungen registriert werden. Bei der Mehrheit konnte jedoch lediglich festgestellt werden, dass es sich um TCP-Verbindungen handelte, jedoch nicht genau welche. Bei 332 der Verbindungen konnte das Nutzen von *IRC* als Netzwerkprotokoll beobachtet werden.

Es konnte dabei beobachtet werden, dass bestimmte *IRC-Channel* vermehrt betreten wurden. Tabelle 4.2 gibt einen Überblick über die zehn Häufigsten. Aus der Tabelle geht hervor, dass viele der Channel einen sehr ähnlichen Namen haben. Dies lässt den Schluss zu, dass die Binaries, die das Betreten dieser Channels auslösten, der gleichen Malware-Familie angehören.

4.4 Zusammenfassung

In diesem Kapitel wurden die Resultate vorgestellt und interpretiert, die die in Kapitel 3 beschriebenen Analysemethoden auf der Datengrundlage dieser Arbeit ergaben. Die Ergebnisse wurden ebenso wie die Analysen selbst in die drei Kategorien zu Angriffsdaten, Angreiferdaten und den Malware Binaries gegliedert.

Für die Angriffsdaten wurden zunächst die Analysen bezüglich zeitlicher Aspekte betrachtet. Dabei konnte festgestellt werden, dass sowohl die Anzahl der Angriffe als auch die der Downloads in einem bestimmten Muster über die Tageszeit verteilt auftraten. Außerdem wurde in diesem Anschnitt untersucht, wie lange es durchschnittlich dauert, bis ein ungeschütztes System im Internet durch automatisierte Malware verseucht

Betretene *IRC-Channel*	Anzahl
#cool	44
#dcz	28
#[r4]BïtNåt	27
#wawa	22
#BïtN[r5]åt	21
#BïtNåt[r3]	20
#Bït[r1]Nåt	17
#Bït[duo]Nåt	15
#last	13
#&virtu	11

Tabelle 4.2: Die zehn häufigsten IRC-Channel

ist. Daraufhin wurden die Ports und Schwachstellen untersucht, über die die Angriffe abliefen. Dabei konnten Unterschiede zwischen einzelnen Sensoren festegestellt werden.

Im Rahmen der Analysen der Angreiferdaten konnte beobachtet werden, dass die Angreifersysteme häufig über den gleichen ISP mit dem Internet verbunden sind wie die Sensorsysteme. Die gesammelte Malware hat also meist ihr netztopologisch nahe Systeme angegriffen. Daraus ergibt sich auch eine gewisse geographische Nähe, da deutsche ISP häufig nur deutsche Benutzer beliefern. Der Aspekt der geographischen Verteilung der Angreifer wurde ebenfalls näher untersucht. Die gesammelten Daten über die Angreifer zeigten außerdem, dass eine beträchtliche Anzahl der Angreifer mit alten, nicht mehr aktualisierten Betriebssystemen, wie beispielweise *Windows 98*, arbeitet.

In Bezug auf die gesammelten Malware Binaries konnte unter anderem festgestellt werden, dass es große Unterschiede in der Downloadhäufigkeit einzelner Binaries gab. Interessant ist zudem, dass die Malware Binaries bei ihrem Erstauftreten zu einem großen Teil nicht von Angreifern in Deutschland heruntergeladen wurden.

Abschließend wurden die Ergebnisse der Virenscanner und der *CWSandbox* näher untersucht. Dabei konnte beobachtet werden, dass die lokalen Scanner im Betrachtungszeitraum über ein Fünftel der Binaries nicht direkt nach ihrem Download als Malware erkannt haben. Sowohl die Ergebnisse der Virenscanner als auch die der *CWSandbox* legen den Schluss nahe, dass viele der gesammelten Malware Binaries gleichen Malware-Familien angehören.

5 Metriken zum Gefährdungslevel

Eine zentrale Frage bei der Einschätzung von Gefährdung ist, wie diese überhaupt gemessen werden kann. Es müssen Kennzahlen identifiziert werden, nach denen bewertet werden kann, wie hoch das Level der Gefährdung ist.

Im Rahmen dieser Arbeit wurden zunächst anhand der Ergebnisse aus Kapitel 4 einige Kennzahlen ausgearbeitet, um einzelne Dimensionen der Gefährdung durch automatisierte Malware einschätzen zu können. Im Anschluss daran wird aus diesen Einzelkennzahlen eine Gesamtmetrik bestimmt, die eine allgemeine Aussage über das Gefährdungslevel macht. Alle in diesem Kapitel entwickelten Kennzahlen sind mit dem Ziel entwickelt worden, Möglichkeiten aufzuzeigen, wie eine Beurteilung des Gefährdungslevels anhand einzelner Kennzahlen erfolgen kann.

Ein wichtiger Aspekt bei der Entwicklung der Metriken in dieser Arbeit war – wie bereits bei den anderen Komponenten der Arbeit – die Automatisierbarkeit. Alle Metriken können automatisiert mit den durch *BEAN* gesammelten Daten bestimmt und im Webinterface abgerufen werden. Dies kann, ebenso wie bei den Ergebnissen aus Kapitel 4, sowohl mit den jeweils aktuellen, als auch mit historischen Daten erfolgen.

In diesem Kapitel werden zunächst einige Grundlagen zu Metriken erläutert. Dabei werden Kriterien zur Bewertung der Güte einer Metrik herausgearbeitet. Im Anschluss daran werden sechs Kennzahlen identifiziert und deren Güte bewertet. Zuletzt wird eine Gesamtmetrik zur Einschätzung des aktuellen Gefährdungslevels erarbeitet.

5.1 Grundlagen

In diesem Abschnitt werden einige Grundlagen zu Metriken im Allgemeinen und speziell zu Sicherheitsmetriken erläutert. Dazu wird zunächst eine Definition dieser beiden Begriffe erarbeitet.

Metrik ist von dem altgriechischen Wort $\mu\varepsilon\tau\rho\iota\kappa\varepsilon\iota\nu$ abgeleitet, was mit *messen* oder *zählen* übersetzt werden kann. Eine Metrik im ursprünglichen Wortsinn ist also ein System, mit dem man etwas messen oder bewerten kann. In Naturwissenschaft und Technik handelt es sich meist um ein Kennzahlensystem, anhand dessen beispielsweise Qualität oder Verlässlichkeit gemessen wird. Nach Jacquith ist eine Metrik ein „konsistenter Standard zum Messen“ [Jaq07]. Metriken sind von Methoden des *Benchmarking* zu differenzieren. Letzteres sind Vergleichstests, die dem Vergleich von verschiedenen Organisationen oder Systemen dienen. Benchmarking kann anhand von Metriken durchgeführt werden.

In dieser Arbeit sollen *Sicherheitsmetriken* für die Einschätzung des aktuellen Gefährdungslevels durch automatisierte Malware im Internet entwickelt werden. Sicherheitsmetriken sind spezielle Metriken, die sich auf die Sicherheit von Systemen beziehen. Konkret für diese Arbeit werden Kennzahlen beschrieben, anhand derer die Gefährdung quantifiziert werden kann.

Es gibt verschiedene Ansätze, Grundeigenschaften von Metriken zu definieren, die sich jedoch nur marginal voneinander unterscheiden. Für diese Arbeit wurden die Ansätze von Swanson et al., Payne und Jaquith zugrunde gelegt [SBS+03, Pay07, Jaq07]. Alle diese Arbeiten stimmen darin überein, dass Metriken bestimmte Kriterien erfüllen müssen, um als gute Metriken bezeichnet zu werden.

Nach Jaquith müssen gute Metriken vier Grundeigenschaften erfüllen. Zunächst müssen sie *konsistent messbar* sein und dürfen nicht von subjektiven Determinanten abhängig sein. Wird die Metrik mehrfach auf der gleichen Datengrundlage berechnet, muss demnach immer das gleiche Ergebnis resultieren. Zum zweiten müssen gute Metriken *günstig zu berechnen* sein. Am besten kann dies dadurch erreicht werden, dass sie automatisiert bestimmt werden können, ohne dass eine Interaktion mit Menschen nötig ist. Dabei ist jedoch zu beachten, dass die Rechenzeit im Verhältnis zum erzielbaren Nutzen steht.

Eine gute Metrik muss des Weiteren als *Kardinal-* oder *Prozentzahl* angegeben werden. Sie muss also entweder eine absolute oder relative Zahl des *Wieviel von dem zu Messenden* darstellen. Daher sind die häufig verwendeten *Ampellichter* allein keine gute Darstellungsform für eine Metrik, da sie kein numerisches und damit gut messbares Ergebnis einer Metrik darstellen.

Als letzte Grundeigenschaft für gute Metriken führt Jaquith an, dass die Metrik mindestens eine Maßeinheit besitzt, besser jedoch zwei oder mehr. Die Maßeinheit muss zum einen ausdrücken, was gemessen wird und zum anderen ermöglichen, dass das Ergebnis mit den Ergebnissen anderer Datensätze vergleichbar ist und somit die Möglichkeit zum Benchmarking gibt. Beispielsweise kann die Maßeinheit für den Benzinverbrauch eines Autos *Liter* sein oder – wie in der Praxis üblich – *Liter pro 100 km*, was die Vergleichbarkeit mit anderen Fahrzeugen herstellt.

Neben diesen vier Grundeigenschaften nennt Jaquith noch eine Weitere, die eine gute Metrik haben sollte, nämlich sollte sie *kontextspezifisch* sein. Das bedeutet, eine Metrik sollte ihrem Betrachter eine Erkenntnis darüber ermöglichen, welche konkreten Handlungen er aus der Metrik ableiten kann. Daher ist es für die Bestimmung guter Metriken wichtig, das Ziel der Metrik genau zu definieren [Jaq07].

Swanson et al. beschreiben, dass Metriken in der IT-Sicherheit „quantifizierbare Informationen für Vergleichszwecke“ ergeben müssen. Zudem müssen die Prozesse, die durch Metriken beschrieben werden, messbar, wiederholbar und kosistent sein. Auch hier wird vorausgesetzt, dass Metriken in Bezug auf den Aufwand einfach zu erzielen sind und dass sie zielgerichtet auf die Problembereiche gestaltet werden, um dem Betrachter einen direkten Nutzen zu bringen [SBS+03].

Payne gibt ebenso wie Swanson et al. eine ähnliche Definition von guten Metriken. Gute

Metriken werden dort als *SMART* beschrieben, „*specific, measurable, attainable, repeateble and time-dependent*“, zu Deutsch spezifisch, messbar, erreichbar, wiederholbar und zeitabhängig [Pay07]. Die Mehrheit dieser Eigenschaften wurde bereits in der Definition von Jaquith verwendet. Spezifisch, messbar und erreichbar sind mit der kontextspezifischen Eigenschaft, der konsistenten Messbarkeit und der günstigen Berechnung bei Jaquith vergleichbar. Wiederholbar werden gute Metriken bei Jaquith durch die Konsistenz der Messbarkeit.

Zeitabhängigkeit ist ein Aspekt, der bei Jaquith nicht explizit genannt wird, er kann jedoch durch eine entsprechende kontextspezifische Definition der Metrik erreicht werden. Die Zeit stellt gerade bei Sicherheitsmetriken einen wichtigen Faktor dar, da sich die Gefährdungslage über die Zeit stark verändern kann, wie die Ergebnisse aus Kapitel 4 bereits gezeigt haben.

Da der Ansatz von Jaquith alle relevanten Aspekte abdeckt und von den drei Vorgestellten der Ansatz mit den konkretesten Definitionen ist, wird er für die folgenden Abschnitte zugrunde gelegt. Jede der Metriken in den folgenden Abschnitten wird anhand der von Jaquith definierten Eigenschaften gewürdigt.

Jede Metrik muss, um Handlungsempfehlungen ableiten zu können, interpretiert werden. Zusätzlich zu der numerischen Darstellung jeder Metrik werden Kategorien definiert, die das Gefährdungslevel ausdrücken. Diese sind kein Bestandteil der eigentlichen Metrik. Einen Anhaltspunkt zur Interpretation der in dieser Arbeit entwickelten Metriken geben die dazu festgelegten Kategorien.

5.2 Kennzahlen

In diesem Abschnitt werden anhand der Ergebnisse aus Kapitel 4 einzelne Metriken erarbeitet. Zunächst wird für jede dieser Kennzahlen beschrieben, was konkret mit ihr bewertet werden kann. Anhand von Daten aus dem in Kapitel 2.5 beschriebenen Datenset wird jede Kennzahl beispielhaft berechnet und ihr Verlauf über den Betrachtungszeitraum grafisch dargestellt.

Daraufhin wird kritisch betrachtet, ob die entworfenen Metriken den im vorangegangenen Abschnitt identifizierten Gütekriterien genügen. Das zusätzlich geforderte Kriterium des *Kontextspezifischen* wird dadurch bewertet, dass der Nutzen der jeweiligen Metrik in Bezug auf die Einschätzung der Sicherheitslage im Internet erläutert wird. Zudem werden mögliche Schlüsse aus der jeweiligen Kennzahl aufgeführt.

Für die Berechnung der Metriken wird jeweils der Zeitraum der letzten sieben Tage zugrunde gelegt. Dadurch wird eine Vergleichbarkeit einzelner Ausprägungen einer Metrik hergestellt. Ein kleinerer Zeitraum würde unter Umständen zu großen Schwankungen in den Ausprägungen der Metrik führen. Wäre der Zeitraum größer gewählt, beispielsweise 30 Tage, würden viele historische Daten berücksichtigt, die nicht die aktuelle Sicherheitslage widerspiegeln. Gerade im schnelllebigen Bereich der IT-Sicherheit ist die Verwendung aktueller Daten von großer Bedeutung.

Metrik	Maßeinheit	niedrig	mittel	hoch
1	Minuten	> 120	30-120	0-30
2	Anteil Betriebssyteme (in %)	$< 5\%$	5%-25%	$> 25\%$
3	Anzahl Angreifer	< 3.000	3.000-33.000	> 33.000
4	Anzahl Binaries	< 2.000	2.000-4.000	> 4.000
5	Anzahl Binaries	< 2.000	2.000-4.000	> 4.000
6	Anteil unerkannter Binaries (in %)	0%	–	$> 0\%$

Tabelle 5.1: Maßeinheiten und Kategoriengrenzen der Metriken

Allen Metriken liegt die Berechnung mittels einem *gleitenden Durchschnitt* oder einer *gleitenden Summe* über die entsprechenden Ereignisse zugrunde. Dabei werden jeweils alle für die Metrik relevanten Ereignisse – beispielsweise Angriffe – der letzten sieben Tage summiert und gegebenenfalls der Durchschnitt gebildet.

Tabelle 5.1 zeigt eine Übersicht über die Metriken, die im Folgenden definiert werden. Ebenso der Tabelle zu entnehmen sind die Maßeinheiten der jeweiligen Metriken und die Interpretation, die für die drei Kategorien *niedrige*, *mittlere* und *hohe Gefährdung*, gewählt wurden. In den folgenden Abschnitten werden die einzelnen Kennzahlen vorgestellt und ihr Nutzen zur Einschätzung des Gefährdungslevels bewertet. Die Begriffe *Kennzahl* und *Metrik* werden hierbei synonym verwendet.

Die in dieser Arbeit gewählte Kategorisierung für jede Metrik ist ein Vorschlag, wie die Werte der Metrik interpretiert werden können. Dazu wurden ein *niedriges* (grün), ein *mittleres* (gelb) und ein *hohes* (rot) Gefährdungslevel unterschieden. Es wurde versucht, aussagekräftige Vergleichswerte für jede Metrik zu finden. Die Grenzen können jedoch bei Bedarf sehr einfach in der Datenbank angepasst werden. Dies kann insbesondere dann erforderlich werden, wenn Anzahl oder Aufbau der Sensoren verändert werden.

5.2.1 Zeit bis zum ersten Malware Download

Die erste Kennzahl soll ausdrücken, wie groß die Gefahr für ein ungeschütztes System ist, zum Nachladen der nötigen Sicherheitsvorkehrungen an das Internet angeschlossen zu werden. Diese Gefahr wird anhand der bereits in Kapitel 4.1.1 beschriebenen Zeitspanne zwischen dem Bezug einer neuen IP-Adresse und dem darauffolgend ersten erfolgreichen Download eines Malware Binaries gemessen.

Die Berechnung dieser Metrik erfolgt, indem zunächst zu jeder neuen Sensoren-IP-Adresse der letzten sieben Tage die Zeit in Minuten bis zum nächstfolgenden Download eines Malware Binaries bestimmt wird. Der Wert der Metrik ist der Durchschnitt aus diesen Einzelwerten. Angegeben wird die Metrik in Minuten.

Die so definierte Kennzahl ist voll automatisiert berechenbar und objektiv messbar. Damit genügt sie dem Kriterium der konsistenten Messbarkeit. Die Laufzeit der Berechnung ist mit etwa zwei Minuten nicht gering. Da die Metrik jedoch eine wichtige Information

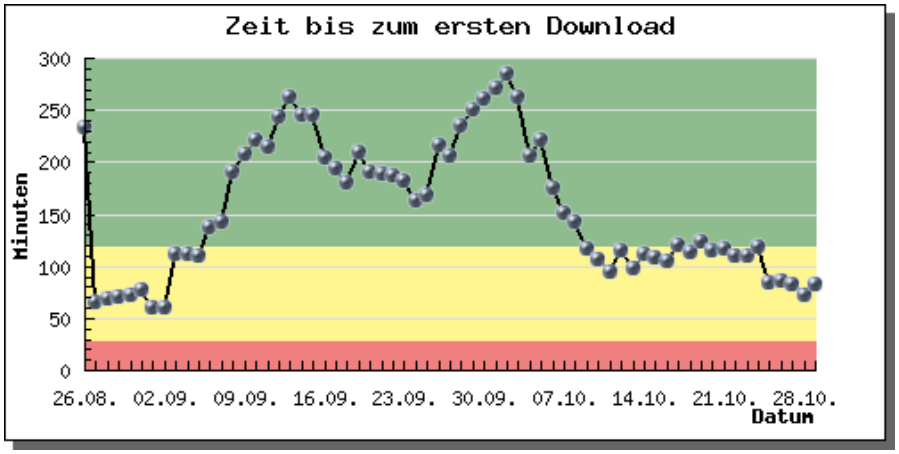


Abbildung 5.1: Kennzahl zur Zeit bis zum ersten Malware Download im Verlauf

zur Beurteilung des aktuellen Gefährdungslevels durch Malware im Internet liefert, ist die Laufzeit tolerierbar. Die Kosten für die Metrik stehen somit im Verhältnis zu ihrem Nutzen. Die Ausprägungen der Kennzahl werden anhand einer einzelnen Maßeinheit, nämlich Minuten, gemessen. Dadurch wird sowohl dem Kriterium der numerischen Darstellung als auch dem der Vergleichbarkeit genügt. Somit kann diese Metrik nach den Kriterien von Jaquith als gute Metrik bezeichnet werden.

Zur Interpretation der Ausprägungen dieser Metrik in Kategorien wird die Zeit zum Aktualisieren eines *Windows XP*-Systems auf *Service Pack 2* als Referenzpunkt angenommen. Dies ist sinnvoll, da mit der Installation dieses Service Packs das Sicherheitsrisiko durch eine automatisch aktivierte *Firewall* minimiert wird. Die Installationsdatei dieses Service Packs hat eine Größe von etwa 250MB, mit einem *DSL 1000*-Anschluss dauert der Download etwa 30 Minuten.

Zur Installation nach erfolgreichem Download kann die Internetverbindung wieder getrennt werden. Damit benötigt man durchschnittlich eine halbe Stunde, um das System gegen die größten Gefahren durch automatisierte Malware abzusichern. Ist der Wert der Metrik also kleiner als eine halbe Stunde, so wird die Gefährdung als hoch eingeschätzt. In mehr als zwei Stunden sollten auch Systeme mit einer langsamen Internetanbindung abzusichern sein. Damit ist die Grenze zwischen einer mittleren und einer niedrigen Gefährdung bei zwei Stunden zu ziehen.

Abbildung 5.1 zeigt den Verlauf der Metrik über den Betrachtungszeitraum. Der hohe Wert am ersten Tag ist damit zu erklären, dass der Wert der Sensoren im Datenset mit fester öffentlicher IP-Adresse nur an diesem Tag in die Metrik einfloss, da diese Sensoren für *BEAN* lediglich einmal als *neu* galten. Details zu dem Unterschied zwischen Sensoren mit fester öffentlicher IP-Adresse und Sensoren mit dynamischer IP-Adresse sind in Kapitel 4.1.1 erläutert.

Im weiteren Verlauf bewegten sich die Ausprägungen der Kennzahl meist im grünen

Bereich, was bedeutet, dass die Gefährdung die meiste Zeit niedrig war. Der rote Bereich, also ein hohes Gefährdungslevel wurde während dem Betrachtungszeitraum nie erreicht. Im Kontext der IT-Sicherheit ist mit dieser Kennzahl eine Aussage möglich, wie *sicher* es ist, ein neues System mit den nötigen Sicherheitsvorkehrungen über die Bezugsquelle Internet auszustatten.

Die Aussagekraft der Kennzahl ist sehr stark von der Anzahl der verfügbaren Daten, die der Berechnung zugrunde gelegt werden, abhängig. Optimal für diese Metrik ist ein Datenset mit einer großen Anzahl Sensorsystemen, die ihre IP-Adresse dynamisch beziehen. Im Allgemeinen ist der Wert dieser Kennzahl sehr hoch, da sie ein Indikator für die generelle Situation von automatisierter Malware im Internet ist.

5.2.2 Anteil out-of-date Betriebssysteme

Anhand der zweiten identifizierten Kennzahl kann bewertet werden, wie stark die Gefährdung im Internet durch Betriebssysteme ist, für die vom Hersteller kein Support mehr anboten wird. Dies ist im Zusammenhang automatisierter Malware sehr interessant, da für neu entdeckte Sicherheitslücken für diese *Out-of-date*-Systeme keine Sicherheitsmechanismen mehr angeboten werden. Je mehr dieser Systeme mit dem Internet verbunden sind, desto höher ist das Risiko für andere Systeme im Internet, von automatisierter Malware infiziert zu werden.

Derzeit werden für die Betriebssysteme *Windows NT 4.0 und frühere Versionen*, *Windows 95*, *Windows 98* und *Windows ME* keine sicherheitsrelevanten Aktualisierungen mehr angeboten. Noch ältere Betriebssyteme wie beispielsweise *Windows 3.11* sind nur noch höchst selten im Einsatz, da die meiste Hardware von ihnen nicht mehr unterstützt wird, weshalb sie in dieser Betrachtung zu vernachlässigen sind.

Die Berechnung der Kennzahl basiert auf den Ergebnissen, die *p0f* liefert. Es werden jeweils alle Daten berücksichtigt, bei denen für einen Angreifer ein Betriebssystem ermittelt werden konnte. Auch hier beträgt der Berechnungszeitraum sieben Tage. Das Ergebnis der Metrik wird in Prozent angegeben. Dabei wird die Gesamtmenge der Angreifer, für die ein Betriebssytem erkannt wurde in Relation zu der Menge der Angreifer, die ein *Windows NT 4.0 und frühere Versionen*, *Windows 95*, *Windows 98* oder *Windows ME*-System verwendeten gesetzt.

Die so definierte Kennzahl enthält keine subjektiven Berechnungskriterien und ist durch eine verhältnismäßig schnelle Laufzeit bei der automatisierten Berechnung günstig zu bestimmen. Die Angabe der Werte der Metrik in Prozent stellt eine relative Zahl zur Bestimmung des *Wieviel* dar. Ihre Maßeinheit ist *Anteil Out-of-date Betriebssysteme*, bei mehrfacher Berechnung auf Grundlage der selben Daten berechnet sie sich immer zum gleichen Wert. Daher ist diese Metrik nach Jaquith als gute Metrik zu bezeichnen.

Bei dieser Kennzahl gelten alle Werte über 25% als hohes und alle Werte unter 5% als niedriges Gefährdungslevel. Ein großer Teil der automatisierten Malware, die verbreitet wird, basiert auf Sicherheitslücken, die bereits seit einiger Zeit (bis zu einigen Jahren)

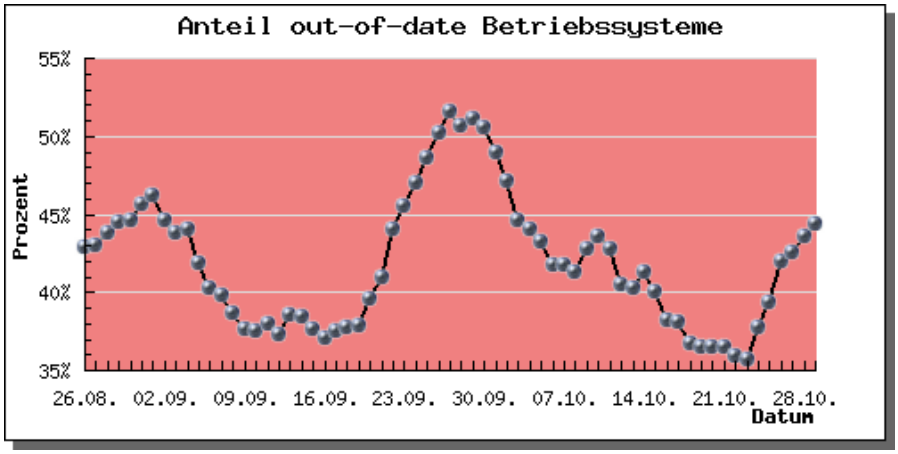


Abbildung 5.2: Anteil der *Out-of-date*-Betriebssysteme im Verlauf

bekannt sind und somit auch in den oben genannten Betriebssysteme durch Patches abgesichert sind. Daher kann die Grenze für eine hohe Gefährdung mit 25% recht hoch gewählt werden.

Der Verlauf der Metrik über den Betrachtungszeitraum ist in Abbildung 5.2 dargestellt. Die Kennzahl bewegte sich im kompletten Betrachtungszeitraum deutlich im roten Bereich. Sie überstieg sogar an einigen Tagen 50%. Ein sehr hoher Anteil an Angreifersystemen im Datenset wurde demnach mit einem der oben genannten *Out-of-date*-Betriebssystemen betrieben.

Da *p0f* in *BEAN* nur Datenpakete von Angreifern und Sensorsystemen zur Verfügung stehen, ist ein Vergleich mit Daten, die nicht über ein *nepenthes*-System gesammelt wurden jedoch schwierig. Daher wäre es sinnvoll, in die Berechnung dieser Kennzahl die jeweils aktuelle Anzahl an Systemen im Internet, auf denen solche *Out-of-date*-Betriebssysteme laufen, einzubeziehen. Da der automatisierte Bezug dieser Daten nicht ohne weiteres möglich ist, standen sie für diese Arbeit nicht zur Verfügung.

Dennoch kann die Kennzahl zur Einschätzung von Gefährdung verwendet werden. Solange eine große Anzahl der registrierten Angreifersysteme mit einem *out-of-date*-Betriebssystem arbeiten, ist davon auszugehen, dass diese Systeme auch mit neuen Arten von automatisierter Malware infiziert werden. Damit stellen sie eine potentielle Gefahr für alle anderen Systeme im Internet dar.

5.2.3 Anzahl unterschiedlicher Angreifer

Ein weiterer Indikator für das aktuelle Gefährdungslevel ist die Anzahl an unterschiedlichen Angreifern, die registriert werden. Die dritte Kennzahl beschreibt diesen Indikator. Die Anzahl an verschiedenen Angreifern spiegelt die Anzahl an potentiell gefährlichen Systemen im Internet wieder.

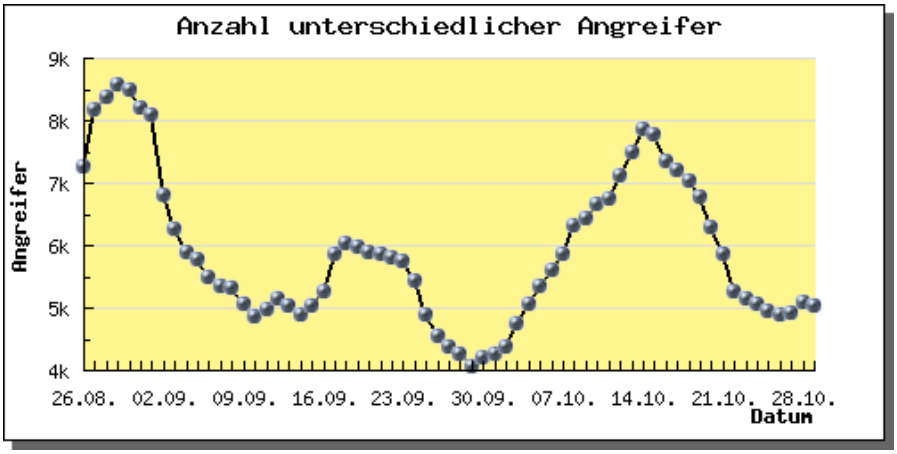


Abbildung 5.3: Anzahl unterschiedlicher Angreifer im Verlauf

Ein Angreifer wird wie in Kapitel 4.2 bereits beschrieben anhand seiner öffentlichen IP-Adresse identifiziert. Die Berechnung der Kennzahl geschieht, indem die Anzahl aller in der jeweiligen Periode von sieben Tagen registrierten Angreifer bestimmt wird.

Aufgrund der gegebenen einheitlichen Definition eines Angreifers ist die Metrik konsistent messbar, sie unterliegt keinen subjektiven Kriterien. Auch diese Metrik wird von *BEAN* automatisiert berechnet. Die Laufzeit für diese Berechnung ist gering, daher sind die Kosten der Metrik gering. Die Werte der Metrik werden numerisch als absolute Anzahl der unterschiedlichen Angreifer angegeben. Durch die Absolutzahl über sieben Tage ist die Metrik mit den Ergebnissen gleicher Systemumgebungen vergleichbar. Die Erfüllung dieser Kriterien zeigt die Güte der Metrik.

Die Einteilung der Ausprägungen der Kennzahl in Kategorien stützt sich auf die Studie von Eimeren et al. [EF07] zur Internetnutzung in Deutschland. Nach dieser Studie nutzen im Jahr 2007 etwa 40 Millionen Deutsche das Internet, etwa zwei Stunden pro Tag. Durchschnittlich sind also etwa 3,3 Millionen Deutsche gleichzeitig mit dem Internet verbunden. Diese Zahl wird als Basis für die Bestimmung der Grenzen verwendet.

Die Anzahl der registrierten Angreifer wird für diese Bestimmung in Relation gesetzt zu der Zahl der gleichzeitigen Internetnutzer in Deutschland. Das Gefährdungslevel wird als niedrig interpretiert, wenn die Ausprägung der Kennzahl weniger als 0,1% von 3,3 Millionen, also etwa 3.000 beträgt. Hoch ist die Gefährdung, wenn der Wert der Kennzahl 1%, also 33.000 übersteigt.

Abbildung 5.3 zeigt den Verlauf der Kennzahl über den Betrachtungszeitraum. Die Werte lagen durchgängig im mittleren Gefährdungsbereich. Die zu erkennenden Schwankungen verdeutlichen, dass aktuelle Daten für die Aussagekraft einer IT-Sicherheitsmetrik von zentraler Bedeutung sind und ein größerer Berechnungszeitraum das Ergebnis verfälschen würde.

Mit Hilfe dieser Metrik kann beurteilt werden, ob die Gefährdung durch automatisierte Malware im Internet von vielen unterschiedlichen Systemen ausgeht oder ob nur überschaubar wenige Angreifersysteme für die Angriffe verantwortlich sind. Dies ist vor allem dazu von Nutzen, Trends frühzeitig zu erkennen. Ändert sich der Wert der Metrik signifikant, sollte nach der Ursache geforscht werden. Steigt der Wert der Kennzahl über einen längeren Zeitraum kontinuierlich an, ist zu vermuten, dass mehr Systeme als vorher als Angreifer fungieren, also demnach selbst mit automatisierter Malware infiziert sind. Bei einem solchen Trend sollte die konkrete Ursache gefunden werden, um gegebenenfalls frühzeitig neue Sicherheitsmaßnahmen zu entwickeln.

5.2.4 Anzahl einzigartiger Binaries

Die vierte Kennzahl bezieht sich auf die heruntergeladenen Malware Binaries. Sie zeigt, wie viele einzigartige Binaries im jeweiligen Zeitraum der letzten sieben Tage mit *BEAN* heruntergeladen wurden. Damit kann ausgedrückt werden, wie viele Binaries tatsächlich ein System im Internet bedrohen. Eine hohe Gefahr für ein System im Internet geht von einem Malware Binary häufig nur zum Zeitpunkt seines ersten Downloads auf das System aus. Ist ein System bereits mit einer bestimmten Art Malware infiziert, so ist es nicht mehr relevant, wie oft nach der Infektion die gleiche Malware das System wiederholt zu kompromittieren versucht.

Die Berechnung dieser Metrik erfolgt, indem die gleitende Summe aller Erstauftritte von Malware Binaries in den jeweils letzten sieben Tagen gebildet wird. Die Ausprägungen der Metrik werden demnach als Absolutwerte in der Maßeinheit *Malware Binaries pro sieben Tage* ausgedrückt.

Auch für diese Metrik wird bewertet, ob sie den Kriterien einer guten Metrik nach Jaquith genügt. Die Metrik unterliegt keinen subjektiven Kriterien, das Erstauftreten eines Binaries in einem bestimmten Zeitraum ist ein objektiv definiertes Ereignis. Bei wiederholter Berechnung für den gleichen Zeitraum mit den gleichen Daten ergibt sie immer den gleichen Wert. Daher kann die Kennzahl als konsistent messbar bezeichnet werden. Dadurch, dass auch diese Kennzahl automatisiert berechnet wird und die Laufzeit gering ist, sind auch ihre Kosten gering. Die Angabe der Metrik erfolgt als numerischer Wert und es gibt eine klar definierte Maßeinheit, *Malware Binaries pro sieben Tage*, die eine gute Vergleichbarkeit herstellt. Somit erfüllt diese Metrik die Kriterien der Güte.

Zur Bestimmung der drei Gefährdungslevel für diese Metrik wurden Zahlen über die Aktualisierungen von Antivirenlösungen als Grundlage verwendet. Dazu wurde der Scanner *AntiVir* von *Avira* zugrundegelegt, da er wie in Kapitel 4.3.3 beschrieben die besten Ergebnisse lieferte. Dieser Scanner wird innerhalb von sieben Tagen durchschnittlich um 8.000 neue Signaturen erweitert [Gmb07b]. Dadurch kann der Virenscanner jede Woche etwa 8.000 verschiedene Arten von Malware neu erkennen oder genauer erkennen als vorher. Auf Basis dessen wird davon ausgegangen, dass für die Metrik Werte über 4.000 Malware Binaries pro sieben Tage hoch sind und unter 2.000 Malware Binaries pro sieben Tage niedrig.

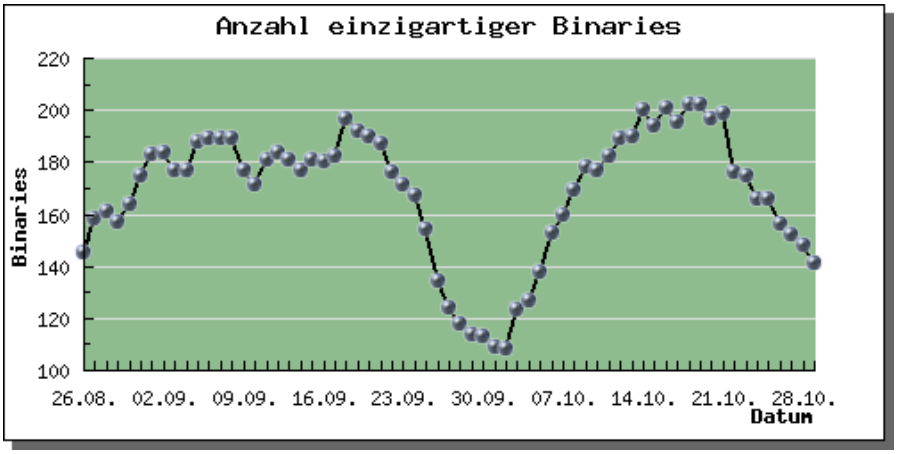


Abbildung 5.4: Verlauf der Kennzahl zu einzigartigen Binaries

Abbildung 5.4 zeigt den Verlauf der Metrik über den Betrachtungszeitraum. Im kompletten Zeitraum erreichte die Metrik nie einen kritischen Wert, sie bewegte sich immer deutlich im grünen Bereich, also dem Bereich, der ein niedriges Gefährdungslevel darstellt. Auch bei dieser Metrik können, wie bereits bei der zuletzt vorgestellten Metrik über die Angreifer, Schwankungen erkannt werden, die nochmals zeigen, wie wichtig zur Einschätzung einer Gefährdung im Bereich IT-Sicherheit tagesaktuelle Zahlen als Grundlage sind.

Anhand der so definierten Kennzahl kann beobachtet werden, wieviel unterschiedliche Malware in dem Zeitraum von sieben Tagen bei *BEAN* ankommt. Diese Zahl ist abhängig davon, wie das Verhältnis zwischen sich automatisch verändernder Malware (*polymorphe Malware*) und tatsächlich neuer Malware ist. Trotzdem gibt diese Zahl einen groben Indikator, durch wie viel unterschiedliche Malware die Bedrohung im Internet verursacht wird. Eine speziellere Aussage über die aktuelle Bedrohung durch die heruntergeladenen Malware Binaries ermöglicht die nächste Kennzahl.

5.2.5 Anzahl unbekannter Binaries

Aufbauend auf der im letzten Abschnitt beschriebenen Metrik zu einzigartigen Binaries wird nun eine weitere Kennzahl definiert. Sie stellt die Anzahl der bisher unbekannten Binaries über den Zeitraum der jeweils letzten sieben Tage dar. Dabei liegt der Unterschied beider Kennzahlen darin, dass bei der ersten lediglich die Einzigartigkeit des Binaries während des Berechnungszeitraums von sieben Tagen betrachtet wird, während bei der zweiten die Einzigartigkeit seit Einschalten des *BEAN*-Systems ausschlaggebend ist.

Dieser Bezug auf die komplette Zeit, in der das System lief, bedingt im Vergleich zum Bezug auf den Berechnungszeitraum der Kennzahl eine höhere Aussagekraft der Metrik

bezogen auf die Gefahr durch neue Malware. Die Anzahl der bisher unbekannten neuen Binaries ist ein genauerer Indikator dafür, ob es sich tatsächlich um neue Malware handelt oder um Malware, die im System schon lange bekannt ist. Ist die Malware schon längere Zeit bekannt, so ist die Wahrscheinlichkeit, dass bereits wirksame Gegenmaßnahmen existieren wesentlich höher als wenn die Malware zum ersten Mal im System auftritt.

Ebenso wie bereits die Metrik in Bezug auf die einzigartigen Binaries ist auch diese Metrik frei von subjektiven Kriterien und ergibt bei wiederholter Berechnung mit den selben Daten gleiche Ergebnisse. Damit ist sie konsistent messbar. Auch die Kosten ihrer Bestimmung sind gering, da ihre automatisierte Berechnung eine kurze Laufzeit benötigt.

Die Kennzahl wird in der Maßeinheit *Binaries pro sieben Tage* angegeben. Ihre Ausprägungen sind absolute numerische Werte. Damit erfüllt sie auch die Kriterien bezüglich einer vergleichbaren Maßeinheit und bezüglich der klaren numerischen Darstellung. Sie ist demnach als gute Metrik zu bezeichnen.

Die Kategoriesierung dieser Metrik erfolgt auf der gleichen Basis wie bei der vorangegangenen Kennzahl, auch die Grenzen wurden gleich gewählt. Eine hohe Gefährdung liegt bei einem Wert größer 4.000 Malware Binaries vor, eine niedrige ab einem Wert unter 2.000 Binaries. Es wird die durchschnittliche Anzahl an neuen Signaturen, um die *Avira* ihren Virenscanner *AntiVir* pro Woche erweitert, als Grundlage für die Interpretation genommen. Bei bisher unbekannten Binaries ist die Wahrscheinlichkeit, dass für diese Binaries noch keine Signaturen für Antivirenlösungen existieren, verhältnismäßig hoch.

Abbildung 5.5 zeigt den Verlauf dieser Kennzahl über den Betrachtungszeitraum dieser Arbeit. Ihre Ausprägungen liegen über den kompletten Zeitraum deutlich im niedrigen Gefährdungsbereich. Die anfängliche Spitze bei den ersten beiden Werten ist dadurch zu erklären, dass der Beginn des Betrachtungszeitraums dieser Arbeit mit dem Einschalten des Gesamtsystems zusammenfällt. Somit waren zu Beginn alle eingehenden Malware Binaries für das System unbekannt.

Anhand dieser Metrik kann beurteilt werden, wie groß die aktuelle Gefährdung durch neue, also bisher im System nicht bekannte, Malware Binaries ist. Je mehr Binaries auftreten, die dem System nicht bekannt sind, desto größer ist die Gefahr, dass sich unter diesen neuen Binaries solche befinden, gegen die die Systeme im Internet noch nicht geschützt sind.

5.2.6 Anteil unerkannter neuer Binaries

In Bezug auf die Gefahr, die von bisher unbekannten Malware Binaries ausgeht, wird im Folgenden noch eine weitere Metrik definiert. Sie sagt aus, wie viele der bisher unbekannten Binaries, die im *BEAN*-System eingehen, zum Zeitpunkt ihres Erstauftretens von keinem der lokalen Virenscanner als Malware identifiziert werden.

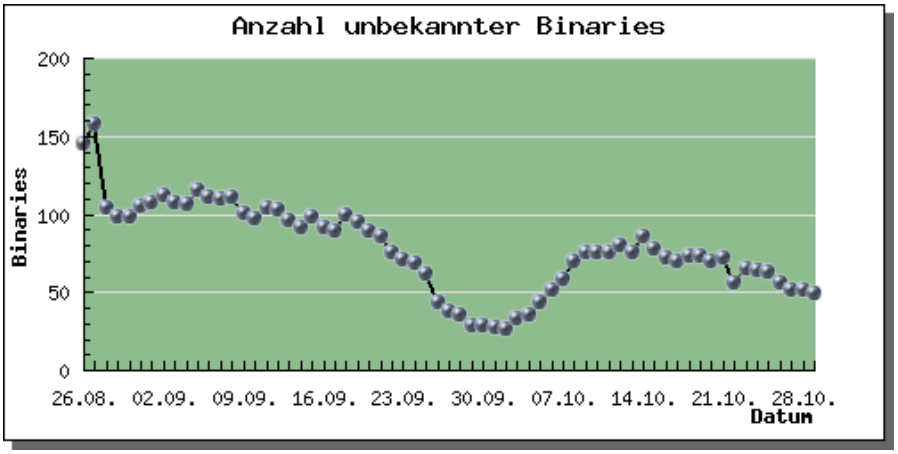


Abbildung 5.5: Metrik zur Anzahl unbekannter Binaries im Verlauf

Zur Berechnung dieser Metrik wird das Ergebnis der im vorangegangenen Abschnitt in Bezug auf die bisher unbekannten Malware Binaries definierten Metrik zu der Anzahl der Binaries, die von keinem der Virenscanner erkannt wurden, ins Verhältnis gesetzt. Dazu wird jeweils nur die erste Untersuchung jedes Binaries von den Virenscannern betrachtet. Das Ergebnis dieser Metrik wird in Prozent angegeben. Die Maßeinheit ist somit: *von Virenscannern unerkannte neuen Binaries in Prozent.*

Auch diese Metrik wird anhand der Kriterien von Jaquith bewertet. Die Laufzeit zur automatisierten Berechnung der Metrik ist gering. Dadurch, dass bereits die Kennzahl, die dieser Metrik zugrunde liegt, als konsistent messbar bewertet wurde und zur Berechnung dieser Metrik lediglich objektive Ergebnisse hinzugenommen werden, ist auch diese Metrik konsistent messbar. Die Maßeinheit ist klar definiert und wird in Prozent angegeben. Sie ist somit mit anderen Systemen vergleichbar. Zusammenfassend kann diese Kennzahl also ebenso wie die vorangegangene als gute Metrik bezeichnet werden.

Da jedes einzelne Malware Binary, das bei seinem ersten Auftreten von keinem der Virenscanner erkannt wird, das System empfindlich schädigen kann, sind die Grenzen zur Interpretation dieser Metrik recht restriktiv gewählt. Ein niedriges Gefährdungslevel liegt nur dann vor, wenn 0% der Binaries unerkannt sind. Das bedeutet, nur wenn die eingesetzten Virenscanner alle in den jeweils letzten sieben Tagen aufgetretenen Binaries als Malware identifizieren konnten, liegt eine niedrige Gefährdung für die Systeme im Internet vor.

Eine Besonderheit dieser Metrik ist, dass für sie nur zwei Kategorien festgelegt wurden. Da, wie bereits beschrieben, jedes Malware Binary, das nicht von den Virenscannern erkannt wird, eine große Gefahr für Systeme im Internet darstellt, wird für diese Metrik keine mittlere Bedrohungsstufe definiert. Alle Werte der Metrik, die über 0% liegen, gelten als ein hohes Gefährdungslevel.

In Abbildung 5.6 kann man den Verlauf der Metrik über den Betrachtungszeitraum

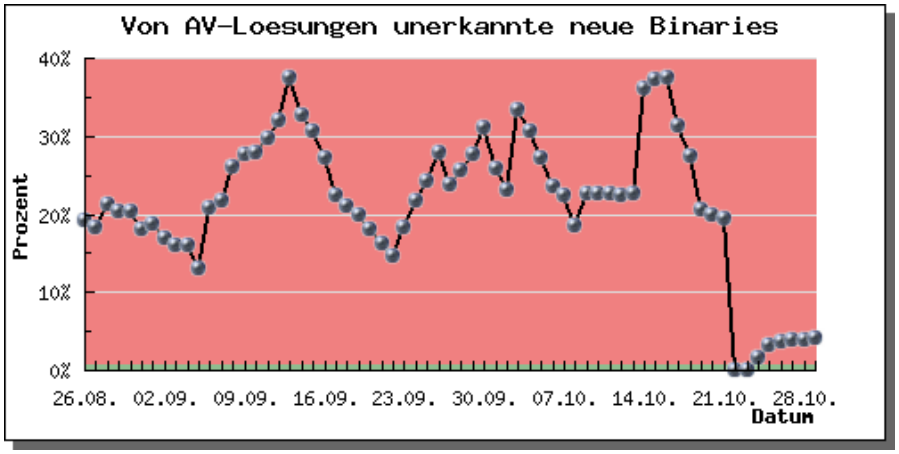


Abbildung 5.6: Verlauf der Metrik zu von Virenscanner nicht erkannten Binaries

sehen. An zwei Tagen im Betrachtungszeitraum erreichte die Metrik den grünen Bereich, es wurden also im Zeitraum von sieben Tagen alle Binaries durch die Virenscanner erkannt. Davor erreichte die Kennzahl Werte von bis zu 38%. Dies zeigt, dass bekannte Virenscanner keinen optimalen Schutz vor automatisierter Malware bieten.

Die Aussage dieser Metrik ist für den Anwender von großem Nutzen, da sie zeigt, dass man sich nicht ausschließlich auf Virenscanner verlassen sollte. Das ständige Aktualisieren von Systemen, die mit dem Internet verbunden sind, ist grundlegend, um die Systeme gegen automatisierte Malware abzusichern. Auch für Entwickler von Sicherheitsmaßnahmen ist der jeweilige Wert dieser Metrik interessant, da sie die Effizienz der vorhandenen Virenscanner abzeichnet und somit den Bedarf an neuen Sicherheitsmaßnahmen aufzeigten.

5.3 Gesamtmetrik

Zur Einschätzung des aktuellen Gefährdungslevels durch automatisierte Malware im Internet wäre es wünschenswert, die gesamte Gefährdung kompakt abzubilden. Dazu wird in diesem Abschnitt eine Gesamtmetrik definiert, die diese kompakte Abbildung möglich macht. Zur Bestimmung der Gesamtmetrik werden zunächst die in Kapitel 5.2 erläuterten Kennzahlen zugrunde gelegt.

Diese sechs Kennzahlen sind in ihrer Art sehr verschieden, sie bilden sehr unterschiedliche Bereiche der Sicherheitsbetrachtung ab und haben verschiedene, zum Teil unvereinbare Maßeinheiten. Daher ist es nicht möglich, sie mathematisch zu verknüpfen, um eine Gesamtmetrik zu berechnen. Es können demnach nur die Interpretationen der einzelnen Metriken verknüpft werden.

Die Bestimmung der Gesamtmetrik erfolgt über die Ausprägungen der Einzelmetriken

gemessen in Kategorien ihrer Interpretation. Jede dieser Einzelmetriken betrachtet einen spezifischen Bereich der Sicherheitslage im Internet in Bezug auf automatisierte Malware. Daher kann keine Priorisierung unter den einzelnen Kennzahlen vorgenommen werden. Sobald eine der Metriken einen hohen Wert annimmt, ist eine Dimensionen der Gefährdung hoch.

Um dieser Tatsache gerecht zu werden, wird die Gesamtgefährdung bestimmt als das jeweilige Maximum aller Kategorienwerte der Einzelmetriken. Ist also eine der Kennzahlen im roten Bereich, ist auch die Gesamtmetrik hoch. Nur, wenn in allen Dimensionen ein niedriges Gefährdungslevel vorliegt, ist auch die Gesamtgefährdung niedrig. Während des Betrachtungszeitraums war die Gesamtgefährdung somit immer hoch, da die Kennzahl zum Anteil der *Out-of-date* Betriebssysteme immer im roten Bereich lag und ebenso die Kennzahl zum Anteil der von den Virenscannern nicht erkannten neuen Binaries fast durchgängig hohe Werte annahm.

Die Gesamtmetrik wurde bewusst sehr allgemein gehalten, da jede Einzelmetrik aus Kapitel 5.2 spezielle Schlüsse zulässt und eine Gesamtgefährdung nur einen generellen Trend signalisieren kann. Da sich die Gesamtmetrik aus den Werten der Einzelmetrik zusammensetzt, kann der Wert dieser Metrik ein Indikator für das generelle Gefährdungslevel sein. Sie liefert somit als Erweiterung zu den Einzelmetriken einen zusammenfassende Information über die Gefährdung durch automatisierte Malware.

5.4 Zusammenfassung

Dieses Kapitel hat gezeigt, dass es möglich ist, über einzelne Metriken das Level der Gefährdung durch automatisierte Malware in den Dimensionen Angriffe, Angreifer und Malware Binaries einzuschätzen. Dazu wurde zunächst definiert, was eine Metrik ist. Es wurden Kriterien identifiziert, die eine gute Metrik auszeichnen.

Danach wurden sechs verschiedene Kennzahlen definiert. Die erste ermöglicht eine Aussage über die Zeit, die vergeht, bis ein neues System über das Internet mit Malware kompromittiert ist. Den Anteil der Angreifersysteme, auf denen *Out-of-date*-Betriebssysteme laufen bildet die zweite Kennzahl ab. Eine der Kennzahlen beschreibt die Anzahl verschiedener Angreifer, die das *BEAN*-System angreifen. Es wurde außerdem ein Block von drei Kennzahlen definiert, die Aussagen über die Gefährlichkeit der heruntergeladenen Malware Binaries machen.

Zu jeder Kennzahl wurden Interpretationsmöglichkeiten in Form von Kategorien erarbeitet, die signalisieren, welches Level der Gefährdung der jeweils aktuelle Wert der Metrik widerspiegelt. Die drei Kategorien sind niedrig, mittel und hoch und werden durch die Ampelfarben Grün, Gelb und Rot repräsentiert.

Im Anschluss an die Definition der Einzelkennzahlen wurde eine Gesamtmetrik entwickelt, anhand derer zusammenfassend gesehen werden kann, wie hoch das aktuelle Gefährdungslevel ist. Die Gesamtmetrik ergibt sich aus den Interpretationskategorien

der sechs Einzelmetriken, indem das Maximum dieser sechs Interpretationskategorien ermittelt wird.

Zur Veranschaulichung der definierten Kennzahlen wurden deren Werte über den Betrachtungszeitraum dieser Arbeit grafisch dargestellt. Die Ergebnisse aller Metriken können mit jeweils aktuellen Werten und im historischen Verlauf im *BEAN*-Webinterface abgerufen werden.

6 Zusammenfassung und Ausblick

Nachdem die Entwicklungen und Analysen dieser Arbeit im Detail beschrieben wurden, wird in diesem Kapitel abschließend eine Zusammenfassung der Resultate der Arbeit gegeben. Zunächst wird noch einmal kurz die Infrastruktur von *BEAN* erläutert. Im Anschluss werden die Ergebnisse der Analysen, die anhand der gesammelten Datengrundlage erstellt wurden, dargestellt. Dabei wird auch auf die entwickelten Metriken zur Einschätzung des aktuellen Gefährdungslevels durch Malware im Internet eingegangen. Abschließend werden einige Ideen für die zukünftige Arbeit in diesen Gebieten gegeben.

6.1 Das BEAN-System

Zunächst wurde eine Infrastruktur – *Bedrohungserkennung und -analyse Netzwerk (BEAN)* – entwickelt, die das voll automatisierte Sammeln und Analysieren von sich automatisch verbreitender Malware im Internet ermöglicht. Diese Infrastruktur besteht aus einer Komponente zum Sammeln der Rohdaten – dem Sensorsystem – und einer Komponente zum Speichern und Auswerten der gesammelten Daten – dem Serversystem. *BEAN* beschränkt sich auf das Sammeln von Malware, die sich autonom im Internet verbreitet und so für ihre Verbreitung keiner menschlichen Interaktion bedarf.

Die Herausforderungen bei der Entwicklung von *BEAN* waren vor allem, das Sensorsystem gut skalierbar und mit geringem Ressourcenbedarf zu gestalten, um eine möglichst hohe Verbreitung zu ermöglichen. Außerdem war ein Hauptziel, dass alle Mechanismen in *BEAN* zum Sammeln und Analysieren der Daten voll automatisiert ausgeführt werden. Dazu mussten eine Reihe von Technologien miteinander kombiniert und ihre jeweilige Funktionalität durch Skripte automatisiert werden.

Zur Erreichung der Zielsetzungen für das Sensorsystem wurde es als *VMware-Image* mit einem *Ubuntu-Linux* ohne grafische Oberfläche entwickelt. Als Grundlage für das Sammeln automatisierter Malware wurde der low-interaction Honeypot *nepenthes* verwendet. Er emuliert Schwachstellen, über die Angreifer angelockt werden. Für *BEAN* wurden 34 Schwachstellenmodule verwendet. Registriert *nepenthes* einen Angriff, extrahiert er eine Reihe von Informationen über diesen Angriff wie beispielsweise die IP-Adresse des Angreifersystems.

Zur Speicherung der gesammelten Daten wurde für *nepenthes* das Modul *log-surfnet* konfiguriert. Es bietet eine asynchrone Anbindung an eine *PostgreSQL*-Datenbank zur Speicherung. So können alle gewonnenen Daten über den Angriff in einer Datenbank

gespeichert werden. Neben der Gewinnung von Daten über den Angriff bietet *nepenthes* zusätzlich die Möglichkeit, die beim Angriff nachgeladenen Malware Binaries zentral abzuspeichern.

Zusätzlich zu *nepenthes* wurde in das Sensorsystem das Tool *p0f-DB* integriert, ein *passives OS-Fingerprinting Tool*, das anhand der eingehenden TCP-Pakete versucht festzustellen, welches Betriebssystem auf dem Angreifersystem läuft. Die Ergebnisse speichert das Tool ebenfalls in eine *PostgreSQL*-Datenbank. Das Tool wurde zur vollständigen Erfassung der Daten so angepasst, dass es für Angreifer, für die es kein Betriebssystem erkennen kann, trotzdem einen Eintrag in der Datenbank generiert.

Neben dem Sensorsystem wurde ein Serversystem entwickelt, das als zentrale Komponente in *BEAN* fungiert. Auf dem Serversystem werden die Rohdaten abgespeichert und die Analysen durchgeführt. Der Server kann die Daten zahlreicher Sensorsysteme verwalten. Die einzelnen Komponenten des Serversystems können auf einem einzelnen Server liegen oder auf verschiedene Server verteilt werden.

Die erste Serverkomponente ist die *PostgreSQL*-Datenbank. In ihr ist das *Raw Malware Repository* und Teile des *Malware Analysis Repository* realisiert. Es werden sowohl die Rohdaten aller angeschlossenen Sensoren gespeichert, als auch die Ergebnisse der Analysen auf diesen Daten. Zudem werden im Dateisystem des Servers die von den Sensoren heruntergeladenen Malware Binaries abgelegt.

Neben der Datenbank sind auf dem Serversystem einige Werkzeuge zur Analyse der Malware Binaries installiert. Konkret sind das die Werkzeuge *objdump*, *hexdump* und *packerid* zur technischen Auswertung der Malware Binaries und vier Virenscanner zur regelmäßigen Untersuchung aller heruntergeladenen Malware Binaries. Der Server verfügt über ein Skript, das alle vier Stunden eine vollständige Untersuchung aller Malware Binaries im Dateisystem veranlasst. Die Ergebnisse dieser Analysen werden in die Datenbank gespeichert.

Eine weitere Komponente auf dem Serversystem ist das *BEAN*-Webinterface. Es ermöglicht das Anzeigen aller Rohdaten in der Datenbank sowie das grafische Aufbereiten der Analyseergebnisse. Alle mit *BEAN* gesammelten Daten gehen dazu in voll automatisierte Auswertungen und Darstellungen ein. Alle Auswertungen sind sowohl mit aktuellen, als auch mit historischen Daten möglich. Zur Erstellung der Grafiken im Webinterface wurden die Bibliotheken *JPGraph* und *GMapper* verwendet.

Zusätzlich zu den auf dem Serversystem installierten Analysewerkzeugen wurden zwei externe Tools in *BEAN* integriert. Zum einen wurde eine Anbindung an die *CWSandbox* realisiert, die eine dynamische Verhaltensanalyse der Malware Binaries macht. Zudem wurde über diese Anbindung der Dienst *VirusTotal* integriert, der zum Untersuchen von Malware Binaries 32 Virenscanner-Engines zur Verfügung stellt. Lädt einer der Sensoren also ein neues Malware Binary herunter, wird es an die *CWSandbox* übermittelt, die die Analyse des Binaries und seine Übermittlung an *VirusTotal* anstößt. Die Ergebnisse dieser beiden Analysen sind als externer Link auf das Webinterface der *CWSandbox* im *BEAN*-Webinterface abrufbar.

6.2 Auswertungen

Zum Erstellen einer Datengrundlage wurden für diese Arbeit auf zehn verschiedenen Systemen die im letzten Abschnitt beschriebenen Sensorsysteme installiert und über zehn Wochen zwischen August und Oktober 2007 betrieben. Die Sensoren waren über verschiedene *Internet Service Provider (ISP)* mit dem Internet verbunden.

Die so gesammelten Daten wurden mit den in *BEAN* integrierten Analysemethoden untersucht und die Ergebnisse interpretiert. Die Ergebnisse dieser Analysen und ihre Bewertung werden im Folgenden zusammengefasst. Die Herausforderung dabei war es aus der Vielzahl von verschiedenen Rohdaten Zusammenhänge herauszuarbeiten, die zur Einschätzung der aktuellen Gefährdungslage durch automatisierte Malware im Internet beitragen können.

Zunächst wurden die Daten in drei verschiedene Kategorien geclustert. Die erste Kategorie umfasst alle Daten, die direkt den Angriff betreffen, wie beispielsweise die IP-Adresse, von der der Angriff ausging. Zur zweiten Kategorie zählen alle Daten, die Informationen über den Angreifer geben. Dies sind beispielsweise die Ergebnisse von *p0f* über das vom Angreifer genutzte Betriebssystem. In der dritten Kategorie wurden alle Daten über die gesammelten Malware Binaries zusammengefasst. Dies umfasst beispielsweise die Ergebnisse der Virenscanner.

Die direkten Angriffsdaten wurden nach zwei verschiedenen Dimensionen untersucht – einer zeitlichen Struktur und der Wege, über die die Sensorsysteme angegriffen wurden. Im Bereich der zeitlichen Dimension konnte im Datenset ein Zusammenhang zwischen der Tageszeit und der Anzahl an Angriffen festgestellt werden. In den Nachmittags- und frühen Abendstunden wurden deutlich mehr Angriffe beobachtet als nachts und am frühen Morgen. Zwischen einem Wochentag und dem Wochenende lagen jedoch keine gravierenden Unterschiede in der Anzahl an Angriffen vor. Ein weiterer Aspekt, der in Bezug auf die zeitliche Dimension der Angriffe betrachtet wurde ist, wie lange es dauert, bis ein ungeschütztes System, das mit dem Internet verbunden wird, mit automatisierter Malware infiziert ist. Dabei wurden zwischen einzelnen Sensoren gravierende Unterschiede festgestellt.

In Bezug auf die Verbreitungswege der Malware konnte beobachtet werden, dass einige Ports auf den Sensorsystemen besonders häufig angegriffen wurden. Welche Ports dies sind, variiert zum Teil je nach Sensor. Ebenso konnte ein Unterschied bei der Verteilung der Ports festgestellt werden zwischen der Betrachtung der Verbindungen, die *nepenthes* als bösartig identifiziert hat und denen, für die dies nicht zutrifft. Solche Verbindungen können harmlose *Portscans* sein, es könnte jedoch ebenso ein Hinweis sein auf Angriffsversuche auf Schwachstellen, für die *nepenthes* keine Emulation anbietet.

Im nächsten Schritt wurden die indirekten Angriffsdaten, also die Daten, die über die Angreifersysteme gesammelt werden konnten, ausgewertet. Die Unterscheidung von Angreifersystemen erfolgt über die öffentliche IP-Adresse, von der der Angriff ausging. Zunächst wurden die Angreiferdaten im Hinblick auf die Netztopologie, in der sich die

Angreifersysteme befanden untersucht. Dabei konnte festgestellt werden, dass im Durchschnitt etwa 80% aller Angreifer vom gleichen ISP angriffen. Das bedeutet, die Mehrheit der Angriffe kam aus dem gleichen Subnetz, in dem sich auch das jeweilige Sensorsystem befand.

Bei der geographischen Auswertung der Herkunftsländer der Angreifersysteme konnte dementsprechend beobachtet werden, dass die große Mehrheit der Angriffe aus Deutschland kam. Die Herkunft des restlichen Teils der Angreifer verteilte sich breit auf verschiedene Staaten wie beispielsweise die USA oder Russland und Japan.

Im Bereich der Technologie wurde untersucht, unter welchem Betriebssystem die Angreifersysteme betrieben wurden. Es konnte lediglich bei etwa der Hälfte der Angreifer ein Betriebssystem ermittelt werden. Insgesamt konnte jedoch festgestellt werden, dass etwa ein Viertel aller Angreifer unter *Windows 98* betrieben wurde.

Die heruntergeladenen Malware Binaries wurden ebenfalls nach verschiedenen Dimensionen untersucht. Zunächst wurde jeweils nur das Erstauftreten eines Malware Binaries während des Betrachtungszeitraums analysiert. Dabei konnte beobachtet werden, dass nahezu an jedem Tag neue, bisher unbekannte Binaries auftraten. In Bezug auf die Herkunft der Malware konnte festgestellt werden, dass nur etwa 40% der Binaries von Angreifern heruntergeladen wurden, die über den gleichen ISP mit dem Internet verbunden waren, wie das jeweilige Sensorsystem. Verglichen mit dem Anteil von Angriffen, die vom gleichen ISP kamen, ist dieser Anteil deutlich geringer.

Auch bei der Verteilung der Herkunftsländer der Malware Binaries konnte ein Unterschied festgestellt werden zu den Angriffen im Allgemeinen. Nur weniger als die Hälfte der Malware Binaries kamen bei ihrem Erstauftreten von Angreifern aus Deutschland.

Betrachtet man die gesamten Downloads aller Binaries und nicht nur ihr jeweiliges Erstauftreten, so kann beobachtet werden, dass es große Unterschiede zwischen einzelnen Binaries gab. Über die Hälfte der Binaries wurden lediglich ein einziges Mal während des Betrachtungszeitraums heruntergeladen, während einige wenige mehrere tausend Male heruntergeladen wurden. Im Durchschnitt wurde jedes Malware Binary 81 Mal heruntergeladen.

Als nächstes wurden die Resultate, die die Untersuchungen der Malware Binaries mit den Virenscannern lieferten, betrachtet. Dabei wurde festgestellt, dass sich die Virenscanner in ihrer Erekennungsrate der Binaries zum Teil gravierend unterscheiden. Insgesamt wurden gut ein Fünftel der heruntergeladenen Malware Binaries innerhalb der ersten vier Stunden von keinem der lokalen Virenscanner als Malware erkannt. Durchschnittlich wurden neue Binaries nach einem Tag von mindestens einem Virenscanner als Malware erkannt.

Es wurde außerdem untersucht, welches die häufigsten Ergebnisse des Scanners mit der besten Erkennungsrate (*AntiVir*) waren. Dabei konnte festgestellt werden, dass die meisten Malware Binaries Varianten des so genannten *RBot*, einem *IRCBot* waren. Es konnte eine große Vielfalt an unterschiedlichen Ergebnissen festgestellt werden.

Die Ergebnisse der *CWSandbox* ließen erkennen, dass jeweils eine Reihe von Malware Binaries offensichtlich zur gleichen Malware-Familie gehörten. Außerdem konnte beobachtet werden, dass viele der Binaries während der Kompromittierung über den *Microsoft Internet Explorer* Dateien nachgeladen haben. Dies konnte anhand der von der *CWSandbox* ermittelten angelegten *Mutexes* und Dateien erkannt werden.

6.3 Einschätzung der Sicherheitslage

Als Ergänzung zu den statistischen Analysen wurde in dieser Arbeit untersucht, mit welchen Mitteln es möglich ist, anhand der mit *BEAN* gesammelten Daten, eine Aussage über die aktuelle Gefährdungslage durch automatisierte Malware im Internet zu treffen. Dazu wurden sechs Metriken zu einzelnen Dimensionen der Sicherheitslage entwickelt.

Zunächst wurden einige Kriterien für gute Metriken herausgearbeitet. Die sechs entwickelten Metriken wurden anhand dieser Kriterien einzeln bewertet. Die Hauptziele in Bezug auf die Metriken waren, eine automatisierte Berechnung zu implementieren und mögliche Interpretationen und Handlungskonsequenzen herauszuarbeiten. Zudem war es wichtig, die Berechnung der Metriken so zu gestalten, dass sie die jeweils aktuelle Lage widerspiegeln. Dazu werden die Metriken jeden Tag neu über die jeweils letzten sieben Tage berechnet und in der Datenbank gespeichert.

Die erste der sechs Metriken misst, wie lange es dauert, bis ein ungeschütztes System, das mit dem Internet verbunden wird, mit automatisierter Malware kompromittiert wird. Sie stellt einen Indikator dafür dar, wie die allgemeine Situation von automatisierter Malware im Internet aktuell ist.

Die bereits in Kapitel 4.2.3 betrachteten *out-of-date*-Betriebssysteme dienen als Grundlage für die zweite Kennzahl. Sie sagt aus, wie hoch der Anteil unter den Angreifern ist, die unter einem solchen *out-of-date*-Betriebssystem arbeiten. Diese Systeme stellen eine generelle Gefahr für andere Systeme im Internet dar, denn es werden keine neuen sicherheitsrelevanten Aktualisierungen mehr für diese Systeme angeboten.

Zur frühzeitigen Erkennung von Trends in Bezug auf automatisierte Malware kann die dritte Kennzahl verwendet werden. Sie stellt die Anzahl an unterschiedlichen Angreifern dar, von denen Sensorsysteme angegriffen werden. Können gravierende Veränderungen dieser Kennzahl beobachtet werden, sollte nach der Ursache geforscht und gegebenenfalls neue Sicherheitsmaßnahmen ergriffen werden.

Die beiden folgenden Metriken beschreiben die Anzahl an Malware Binaries, die heruntergeladen werden konnten. Die erste der beiden Kennzahlen misst die Anzahl an einzigartigen Binaries innerhalb des Berechnungszeitraums von sieben Tagen. Die andere Metrik betrachtet lediglich die bisher unbekannten Binaries, also diejenigen Binaries, die innerhalb des Berechnungszeitraums zum ersten Mal im System registriert werden. Beide Metriken lassen Aussagen darüber zu, wie hoch die Gefahr für ein System im Internet ist, von einem Malware Binary infiziert zu werden.

Die letzte Metrik zeigt auf, wie viele der heruntergeladenen Malware Binaries von keinem der lokalen Virenscanner als Malware erkannt wurden. Diese Kennzahl macht eine zentrale Aussage über die Sicherheitslage für Systeme im Internet, da ein Malware Binary, das von dem installierten Virenscanner nicht erkannt wird, mit sehr hoher Wahrscheinlichkeit das betroffene System infiziert.

Für jede dieser sechs Metriken wurden als Interpretationsmöglichkeit drei Kategorien gebildet. Diese Kategorien stellen jeweils ein aktuell niedriges, mittleres oder hohes Gefährdungslevel dar. Es wurde versucht, jeweils sinnvolle Grenzen für die Kategorien zur Interpretation zu finden.

Diese Interpretationskategorien wurden abschließend zusammengefasst zu einer Gesamtmetrik, die zusammenfassend einen generellen Anhaltspunkt für das aktuelle Gefährdungslevel durch automatisierte Malware gibt. Sie wird durch das Maximum der Interpretationskategorien der sechs Einzelmetriken bestimmt. Über die so definierten Kennzahlen wurde somit eine Möglichkeit gegeben, die in dieser Arbeit gesammelten Daten in Bezug auf die Sicherheitslage zu interpretieren.

6.4 Ausblick

Zum Abschluss dieser Arbeit werden noch einige Möglichkeiten aufgezeigt zur zukünftigen Arbeit auf dem hier bearbeiteten Gebiet. Dazu werden sowohl mögliche Erweiterungen speziell für das Sensorsystem genannt, als auch generelle Erweiterungen für das *BEAN*-System.

In dieser Arbeit wurde mit zehn Sensoren über einen Zeitraum von zehn Wochen bisher nur eine verhältnismäßig kleine Datenbasis betrachtet. Daher ist es sinnvoll, *BEAN* mit einer weitaus größeren Anzahl an Sensoren über einen deutlich längeren Zeitraum laufen zu lassen. Die in dieser Arbeit vorgestellten Analyseergebnisse sollten dann mit dieser größeren Datenbasis verifiziert werden.

Es steht bisher technisch keine Lösung zur Verfügung, mit der Angreifersysteme eindeutig unterschieden werden können. Um zuverlässige Aussagen über die Menge an verschiedenen Angreifersystemen mach zu können, wäre es nötig, dass Merkmale entwickelt werden, anhand derer eine eindeutige Identifikation eines Systems im Internet möglich ist.

In Bezug auf das Sammeln der Rohdaten wäre eine Weiterentwicklung des Sensorsystems sinvoll, die eine voll automatisierte Konfiguration des Systems beinhaltet. Dadurch würde die Installation eines Sensors deutlich vereinfacht. Die Verbreitung des Sensorsystems könnte somit deutlich erhöht werden. Eine denkbare Lösung wäre, eine CD zu entwickeln, die die Installation und Konfiguration des Sensorsystems automatisch erledigt. Dabei könnte das Sensorsystem außerdem um ein grafisches Konfigurationsmenü erweitert werden, das die Bedienung erleichtert, da die Konfiguration bisher ausschließlich über die Kommandozeile erledigt wird. Dieser Ansatz wird derzeit in einer weiteren Diplomarbeit bearbeitet.

Ebenfalls sinnvoll wäre eine Erweiterung des *log-surfnet*-Moduls zum Speichern der Daten über die von *nepenthes* registrierten Angriffe. Dabei wäre es sinnvoll, dass das Modul eine eindeutige Zuordnung der Verbindungen und der daraus resultierenden Downlaods ermöglicht. Dadurch könnten beispielsweise Analysen im Hinblick auf die verwendeten Schwachstellen der Angriffe, die mit einem erfolgreichen Malware Download endeten, gemacht werden.

Im Bereich der Analysemethoden, die in *BEAN* integriert sind, wäre eine Hinzunahme einiger weiterer lokaler Virenscanner denkbar. Sie könnte bei entsprechender Anzahl an Scannern die Nutzung des externen Dienstes *VirusTotal* auf lange Sicht ersetzen. Es könnten somit über alle Binaries mit vielen verschiedenen Virenscannern Verlaufsdaten ermittelt werden. Zudem würde dadurch die Unsicherheit über den Zeitpunkt der Untersuchung, die bei *VirusTotal* besteht, entfallen.

Das *BEAN*-Webinterface bietet ebenfalls eine Reihe von Möglichkeiten. Es wäre beispielsweise denkbar, durch die Einbindung von *AJAX* die Benutzerfreundlichkeit zu erhöhen. Ebenso wäre es hilfreich, eine zentrale Möglichkeit zur Konfiguration der Sensoren über das Webinterface zu implementieren. Beispielsweise könnte eine Funktion angeboten werden, um neue *nepenthes*-Schwachstellenmodule zentral über das Webinterface gleichzeitig auf allen Sensoren einzubauen.

Zur Beurteilung der aktuellen Gefährdung sollten zusätzlich zu den bereis entwickelten Metriken weitere Kennzahlen erarbeitet werden. Zudem sollten die vorhandenen Metriken dahingehend überprüft werden, ob sie mit Hilfe formaler Methoden verfeinert werden können.

Zur umfassenden Einschätzung der aktuellen Sicherheitslage im Internet wäre es allgemein sinnvoll, das gesamte *BEAN*-System dahingehend zu erweitern, dass es außer automatisierter Malware auch andere Gefahren im Internet erkennen kann. Dazu könnte dem Sensorsystem ein high-interaction Honeypot hinzugefügt werden oder ein so genannter *Client-side Honeypot*, der beispielsweise versucht, Malware, die sich über die Nutzung von bestimmten Webbrowsern verbreitet, zu identifizieren. Zusätzlich sollten Funktionalitäten entwickelt werden, mit Hilfe derer *BEAN* bisher unbekannte, also so genannte *Zero-Day*, Schwachstellen erkennen kann. Dadurch könnte eine bessere Repräsentation aller Gefahren im Internet ermittelt werden.

Literaturverzeichnis

[BKH+06] Paul Baecher, Markus Koetter, Thorsten Holz, Maximilian Dornseif, and Felix Freiling. The Nepenthes Platform: An Efficient Approach to Collect Malware. In *Recent Advances in Intrusion Detection: 9th International Symposium, Raid 2006*, pages 165–184, Berlin Heidelberg, 2006. Springer Verlag.

[BKWW07] Paul Bächer, Markus Kötter, Georg Wicherski, and Tillmann Werner. mwcollect, Malware Collection Made Easy. www.mwcollect.org, 1.11.2007.

[bv07] SURFnet bv. SURFids. http://ids.surfnet.nl/wiki/doku.php, 2.11.2007.

[Cla] Jim Clausing. packerid. http://isc.sans.org/diary.html?storyid=3432.

[Con07] Aditus Consulting. JPGraph – PHP Graph Creating Library. http://www.aditus.nu/jpgraph/, 4.11.2007.

[Cor07] Microsoft Corporation. Windows 98 LifeCicle Informationen. http://support.microsoft.com/default.aspx?scid=fh;DE;lifewin98, 1.11.2007.

[DZL06] David Dagon, Cliff Zou, and Wenke Lee. Modeling Botnet Propagation Using Time Zones. In *Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS '06)*, 2006.

[EF07] Birgit van Eimeren and Beate Frees. Internetnutzung zwischen Pragmatismus und YouTube-Euphorie – ARD/ZDF-Online-Studie 2007. *Media Perspektiven*, pages 362–378, 08/2007.

[fSidI07] Bundesamt für Sicherheit in der Informationstechnik. Die Lage der IT-Sicherheit in Deutschland 2007. http://www.bsi.de/literat/lagebericht/lagebericht2007.pdf, 2007.

[GHW07] Jan Goebel, Thorsten Holz, and Carsten Willems. Measurement and Analysis of Autonomous Spreading Malware in a University Environment. In *Detection of Intrusions and Malware, and Vulnerability Assessment: Proceedings of the 4th International Conference DIMVA 2007*, pages 109–128. Springer, July 2007.

[Gmb07a] Avira GmbH. Avira AntiVir® PersonalEdition Classic. http://www.free-av.de/, 2.11.2007.

[Gmb07b] Avira GmbH. AntiVir Virus Definition File History. http://www.avira.com/de/threats/section/vdfhistory/index.html, 8.11.2007.

[Goo07] Google. Google Maps API. http://www.google.com/apis/maps/, 4.11.2007.

[Hol05] Thorsten Holz. A short visit to the bot zoo. *Security and Privacy Magazine, IEEE*, pages 76–79, 2005.

[HW06] Thorsten Holz and Georg Wicherski. Effektives Sammeln von Malware mit Honeypots. *13th DFN-CERT Workshop „Sicherheit in vernetzten Systemen“*, 03/2006.

[Inc07] VMware Inc. VMware. http://www.vmware.com/de/, 1.11.2007.

[InM07] Projektteam InMAS. Das Internet-Malware-Analyse-System (InMAS) – Aktualisierte Projektskizze und Angebot, 8.10.2007.

[INT07] FRISK SOFTWARE INTERNATIONAL. F-PROT. http://www.fprot.org/, 2.11.2007.

[Jaq07] Andrew Jaquith. *Security Metrics*. Addison-Wesley, 2007.

[Kis07] Philipp Kiszka. GMapper – Google Maps in PHP. http://gmapper.ajax-info.de/, 4.11.2007.

[Kru07] Nerijus Krukauskas. p0f-db. http://freshmeat.net/projects/p0f-db/, 2.11.2007.

[LLC07] MaxMind LLC. MaxMind GeoIP. http://www.maxmind.com/app/ip-location, 1.11.2007.

[Ltd07] Canonical Ltd. Ubuntu Linux. www.ubuntu.com, 1.11.2007.

[mwc07] mwcollect. The beginning of a honeypot dnsbl? http://nepenthes.mwcollect.org/, 11.11.2007.

[Nor07] Norman. Norman Antivirus. http://www.norman.com/de, 2.11.2007.

[Pay07] Shirley C. Payne. A Guide to Security Metrics. https://www2.sans.org/reading_room/whitepapers/auditing/55.php ?portal=158328adc61c8f1cc8ffe8439ffb1ac1, 2007.

[PGDG] PostgreSQL Global Development Group. Postgresql 8.2. http://www.postgresql.org.

[PH07] Niels Provos and Thorsten Holz. *Virtual Honeypots*. Addison-Wesley, 2007.

[Pro05] The Honeynet Project. Know your enemy: Tracking Botnets. http://www.honeynet.org/papers/bots/, 13.3.2005.

[Rey07] J. Reynolds. The Helminthiasis of the Internet. http://tools.ietf.org/html/rfc1135, 10.11.2007.

[SBS+03] Marianne Swanson, Nadya Bartol, John Sabato, Joan Hash, and Laurie Graffo. Security Metrics Guide for Information Technology Systems. http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf, 2003.

[Sou07] Inc. Sourcefire. ClamAV. http://www.clamav.net/, 2.11.2007.

[Spi02] Lance Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2002.

[Spi04a] Lance Spitzner. *Know your enemy: learning about security threats*, chapter 3. Addison-Wesley, 2004.

[Spi04b] Lance Spitzner. *Know your enemy: learning about security threats*, chapter 2. Addison-Wesley, 2004.

[TGP] The GNU Project. objdump. http://www.gnu.org/software/binutils/manual/html_chapter/binutils_4.html.

[uCK07] KabelBW GmbH und Co KG. KabelBW Internet FAQ. http://kabelbw.de/kabelbw/services/faq.do, 31.10.2007.

[uIA07] 1und1 Internet AG. 1und1 Internet-Zugang FAQ. http://faq.1und1.de/access/dsl/11_dsl_e_mail__unified_messaging/8.html, 31.10.2007.

[WHF07] Carsten Willems, Thorsten Holz, and Felix Freiling. Towards Automated Dynamic Malware Analysis using CWSandbox. *Security and Privacy Magazine, IEEE*, 2007.

[Wil07] Carsten Willems. CWSandbox. http://cwsandbox.org/, 2.11.2007.

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertig zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mannheim, 19. November 2007

Laura Anna Itzel