SpamTrap

Projektdokumentation

Softwareprojekt im Sommersemester 2006

Matthias Schmidt
Matr.Nr.: 13884
ms096@hdm-stuttgart.de

Matthias Herbert
Matr.Nr.: 13873
mh062@hdm-stuttgart.de

Hochschule der Medien Stuttgart
Softwareprojekt Sommersemester 06

Betreut von: Prof. Walter Kriha

Inhaltsverzeichnis

1 Einleitung

Das Thema Spam beschäftigt uns alle. Sobald man an E-Mails oder Internet denkt, wird man automatisch mit der Problematik des SPAMs konfrontiert. Dieser riesigen Flut an unnützen Nachrichten treten inzwischen eine große Anzahl an AntiSpam - Programmen, Spamfilter und E-Mail-Verschlüsselungsprogrammen entgegen. Leider setzen alle diese Lösungen am falschen Punkt an, da durch sie nur versucht wird den Anwender mit weniger Spam zu belasten. Keiner dieser Ansätze packt das Problem an der Wurzel an. Im Gegensatz dazu wird in diesem Projekt durch das Aufstellen von so genannten 'Spammerfallen' versucht, die Absender IP - Adressen der Versender von Spam oder Phishing - Mails aus den Nachrichten zu filtern um so deren Identität durchsichtiger zu machen.

2 Einführung

2.1 Anti - Spam - Maßnahmen

Bei der Bekämpfung von Spam - Mails kommen verschiedene Lösungen zum Einsatz, die versuchen der Spamflut Herr zu werden. Auf dem Markt buhlen eine Vielzahl von Anti-Spam-Software Anbietern um Kunden und stehen zusätzlich noch in Konkurrenz zu den OpenSource Programmen. Die Anti-Spam-Programme werden auf den Rechnern oder in einem Netzwerk installiert und filtern, mit Hilfe verschiedenster Methoden, Spam-Mails heraus. Dafür schaltet sich die Software zwischen das Internet und die E-Mail-Software wie Outlook Der Spam wird dann entweder in einem extra dafür vorgesehenen Ordner abgelegt oder sofort gelöscht.

2.1.0.1 Techniken zur Filterung

- Überprüfung von E-Mails mit kostenlos zu nutzenden DNSBL-Listen (Listen, die Verweise auf Spammer enthalten). Automatische Aktualisierung dieser DNSBL - Listen vom Server des Anbieters.
- Hinzufügen von bekannten und vertrauenwürdigen E-Mail-Adressen zu den White - Listen (da White - Listen immer Vorrang haben, werden mit deren Hilfe Fehleinträge in den DNSBL - Listen ausgeglichen)
- Sperrung ganze IP -Blöcke
- Bei regelmäßigem Kontakt werden Empfänger/Absender automatisch auf eine White - Liste gesetzt
- Markierung der Spam Nachricht im Betreff der E-Mail mit z.B. ***SPAM*** zu Beginn des Betreffs
- Beliebige Erweiterung der Software durch Programmierung eingener Plugins

- Filterung duch lernfähige Filter (z.B Bayesian Filter - Errechnung der Spam - Wahrscheinlichkeit)
- Filterung mit Hilfe von Wortfiltern (Erkennung bestimmter oft genutzer Worte in Spam-Mails). Eigene Worte nach denen gefiltert werden soll können eingegeben werden.
- Imagefilterung (Bilder von Nachrichten werden nicht angezeigt siehe Phishing)
- Inhaltsfilterung (nach bestimmten Inhalten wird gefiltert)
- Sprachfilterung (E-Mails in bestimmten Sprachen werden gefiltert)

Ein vollkommen anderer Ansatz ist die Idee der Filterung der Nachrichten bevor sie den Empfänger erreichen. Dies geschieht zum Einen durch die Provider im Internet, die standardmäßig Spam - Filterung betreiben und zum Anderen durch Anbieter wie www.spam-schutz.net. Bei diesem Betreiber werden alle E-Mail-Konten des Kunden auf ein spezielles Kundenkonto beim Betreiber umgeleitet. Alle E-Mails, die in diesem Posteingang landen, müssen durch den Absender bestätigt werden. Hierfür werden Bestätigungsemails mit einem Link an den Absender versendet. Nur wenn die Nachricht mit diesem Bestätigungslink als gültig markiert wurde, wird sie an den eigentlichen Empfänger weitergeleitet. Da die Spammer fast nie solche Bestätigungsmails bedienen, landen dann die Spam-Mails in einem Ordner für verdächtige E-Mails. So wird sicher gestellt, dass der Kunde keine unerwünschten E-Mails mehr bekommt.

2.2 Auswirkungen von Spam

2.2.1 Finanzielle Auswirkungen

Auf Grund der großen Anzahl sind die durch Spam - Mails entstandenen Schäden enorm. Auch wenn man davon ausgeht, dass ein Prozent der Spam - Mails wahrgenommen werden und wiederum nur ein Prozent davon Umsatz verzeichnen kann, entsteht durch diese effiziente Werbeform ein gewaltiger betriebswirtschaftlicher Schaden. Hierzu gab es in der Vergangenheit mehrere Studien, die versuchten das Ausmaß des Schadens zu schätzen. Die Studie von Nucleus Research[1] von 2004 ergab, dass Spam ein Unternehmen aus den USA durchschnittlich 1.934 $ pro Mitarbeiter und Jahr kostet. Der Schaden in Europa wurde damals unter Berücksichtigung des gesamten Mailaufkommens und der darin enthaltenen Spamanteile[2] auf ca. 3 Mrd. $ geschätzt.

Spammer-Profit versus betriebswirtschaftlicher und persönlicher Schaden bei den Empfänger

[1] http://www.nucleusresearch.com/research/e50.pd
[2] Studien errechneten Spamanteile von 60% bis 90 %

	Spammer	**Spam-Auftraggeber**	**Empänger - Kosten (ohne Filter)**
20.000.000 Spams versenden bzw. empfangen	Kosten: 2000 Euro	-	2.800.000 Euro (nur Arbeitszeit, bei 10 s Bearbeitungszeit pro Spam-E-Mail und 1000 Euro Arbeitskosten pro Stunde)
Lesen von 1,00% der Spam	-	-	160.000 Euro (nur Arbeitszeit, bei weiteren 60 s für Lesen + WWW)
Kundengewinnung durch 0,01 % der Spam	-	-	8.000 Euro (nur Arbeitszeit, bei weiteren 300 s für die Bestellung)
Umsatz bei 2000 erfolgreichen Spams und 50 Euro pro Bestellung	Einnahmen: 30.000 Euro (30% Provision)	30.000 Euro - 100.000 Euro Provision an den Spammer	Ausgaben: 100.000 Euro für ein oft illegales oder nutzloses Produkt
Ergebnis	28.000 Euro	70.000 Euro	**- 3.068.000 Euro Schaden**

2.2.1.1 Durch Spam verursachte Kosten Kosten entstehen nicht nur beim Löschen während der Arbeitszeit sondern auch durch die Gefährdung der Netzinfrastruktur des Unternehmens und des ISP (Internet Service Provider). Des Weiteren entstehen unmittelbare Kosten durch Traffic, Mailserver- und Storage-Infrastruktur-Nutzung plus die Bezahlung des zusätzlichen Personals, dass dür die Bearbeitung der Spamaufkommens eingestellt werden muss. Der Zeitaufwand der Mailempfänger, der durch den Umgang mit Spam ensteht, führt zu einem Produktivitätsverlust. Dieser Verlust, die Nichterreichbarkeit und Verfügbarkeit und die Reparatur von beschädigten Systemen stellen die mittelbaren Kosten des Spamming dar. Sonstige Kosten entstehen aus dem eventuellen Image-Verlust, der durch ungewolltes Versenden von Spam-Mails enstehen kann.

2.2.1.2 Kosten von Anitspam - Maßnahmen Zur Spam-Abwehr benötigen die Unternehmen weitere Hardware und Software. Für letzteres fallen Lizensierungskosten an oder im Falle der Eigenentwicklung steigen die Kosten durch zusätzliches Personal und die Entwicklungszeit der eigenen Anti-Spam-Software. Hinzu kommen Schulungen der Mitarbeiter, Pflege, Wartung und Ad-

ministrierung des Spamschutzes. Alle Maßnahmen zum Schutz vor Spam können die durch Spamming entstehenden Schäden nur zum Teil verringern.

2.2.1.3 Beispiel eines Großunternehmens Das BSI[3] hat in einem Fallbeispiel die durch Spam entstehenden Kosten eines Großunternehmens berechnet. Hierbei wurden zuerst die Kosten von 1.300 Euro / Jahr, die durch den gesteigerten Traffic enstehen, berücksichtigt. Die mittelbaren Kosten des Produktivitätsausfalls belaufen sich auf 1,6 Millionen Euro pro Jahr bei einem Personentagesatz von 500 Euro und einer Arbeitsverzögerung von im Durchschnitt 10 Sekunden pro Spam-Mail. Durch Image-Schaden des Betriebs wurden nochmals 100.000 Euro aufsummiert. Hierbei wurde die durchschnittliche Größe einer Spam-Mail auf 25 KByte festgelegt.

Die Zahlen

25.000 Spam- und Viren-Mails x 365 Tage x 25 KByte durchschnittl. Größe x 6 Euro pro GByte = **1.305 Euro**

((25.000 Spam-und Viren-Mails x 365 Tage x 10 Sek.) / (8 Std. x 60 Min. x 60 Sek.)) x 500 Euro = **ca. 1.600.000 Euro**

1,6 Millionen + 100.000 = **1,7 Millionen Euro Schaden pro Jahr**

Hieraus ergeben sich ca. **0,18 Euro** Schaden pro Spam - Mail! Der Schaden pro Spam - Mail mit Schutzmaßnahmen würde sich auf 0,06 Euro belaufen, was eine Einsparung von ca. 1,15 Millionen Euro bedeuten würde.

2.2.2 Persönliche Auswirkungen

Viele Empfänger von Spam-Mails fühlen sich persönlich beleidigt oder sind irritiert. Dies liegt daran, dass viele Spam-Mails nicht als solche erkannt werden. Deshalb gehen viele Opfer von Spam davon aus, dass sie selbst die Verantwortung für die oft unmoralischen Angebote in den E-Mail Nachrichten zu tragen hätten. Durch vermehrtes Empfangen von Spam-Mails mit pornografischen Inhalten oder Viagra-Angeboten entwickeln manche Menschen regelrechte Ängste und Schuldgefühle, obwohl sie keinerlei Schuld trifft. Solche Fälle treten natürlich immer weniger auf, da die Internetnutzer inzwischen wesentlich besser informiert sind und ein immer besseres Gefühl entwickeln wann es sich um eine Spam-Mail handelt und wann nicht.

[3] http://www.bsi.de

2.3 Beschwerden und Rechtsweg

Eine Beschwerde beim Provider des Spammers sollte in jedem Fall eingereicht werden. Sollte die gewünschte Wirkung nicht eintreten kann man den Rechtsweg einschlagen. Die dadurch entstehenden Kosten mindern die Lukrativität des Spamming. Manche Provider beachten nicht einmal die Beschwerden der Internetbenutzer, während Andere Beschwerden dankbar aufnehmen und den Täter relativ schnell blocken. Die IP-Adressen der Provider findet man im Header einer Email. Den Provider der Adresse ermittelt man mit dem UNIX-Befehl 'whois' und dem Whois-Server der zuständigen Registry.

Inzwischen haben die meisten Provider eine Beschwerde E-Mail-Adresse eingerichtet, an die man seinen Beschwerdetext plus eine Kopie der Spam Nachricht schicken kann. Dieser Prozess wird auch von Automatisierungsprogrammen übernommen, die Spam-Mails erkennen und nur eine Bestätigung vom Benutzer benötigen, dass es sich bei der Nachricht wirklich um Spam handelt.

2.4 Rechtslage

2.4.1 Rechtslage in Deutschland

In Deutschland ist die Rechtslage noch nicht eindeutig definiert worden, jedoch läßt sich in den letzten Jahren eine klare Tendenz erkennen. Durch die immer weiter steigende Belastung durch Spam, die inzwischen nahezu jeden Internetbenutzer betrifft, beschäftigen sich mit dieser Thematik immer mehr Menschen. Allein durch die immensen Kosten[4], die durch Spam entstehen, sieht sich die Poltik mehr und mehr in die Verantwortung genommen, etwas gegen die Spamflut zu unternehmen. Am 17. Februar 2005 hat sich der Deutsche Bundestag über den Entwurf eines Anti - Spam - Gesetzes beraten.

Auszug aus dem Anti-Spam-Gesetz

> "Werden kommerzielle Kommunikationen per elektronischer Post (E-Mail) versandt, darf in der Kopf- und Betreffzeile weder der Absender noch der kommerzielle Charakter der Nachricht verschleiert oder verheimlicht werden. Ein Verschleiern oder Verheimlichen liegt insbesondere dann vor, wenn die Kopf- oder Betreffzeile absichtlich so gestaltet ist, dass der Empfänger vor Einsichtnahme in den Inhalt der Kommunikation keine oder irreführende Informationen über die tatsächliche Identität des Absenders oder den kommerziellen Charakter der Nachricht erhält."

Trotzdem blieb die Haftungsfrage für den Versand von Spam-E-Mails von Privatpersonen weiterhin umstritten, da ein Verstoß einer Einzelperson gegen das

[4] Die Europäische Kommission schätzt allein die Produktivitätsverluste bei europäischen Unternehmen durch die Beseitigung von Spam-Mails im Jahr 2002 auf 2,5 Mrd. Euro.

Wettbewerbsrecht schwer nachzuweisen ist. Nur Unternehmen, die unerwünschte E-Mail Werbung versenden, können haftbar gemacht werden und strafrechtlich belangt werden. Allgemein verbietet der Rechtsrahmen das Versenden von Werbe - Mails ohne Einwilligung des Empfängers und stellt eine unzumutbare Belästigung für die Empfänger dar.

Auszug aus dem Rechtsrahmen[5]

> "Bereits nach derzeitiger Rechtslage ist die Versendung von Spam-Mails unzulässig. Das unaufgeforderte Versenden von elektronischen Werbe-Nachrichten stellt einen Verstoß gegen das Wettbewerbsrecht dar. Dies ist im Rahmen der Änderung des Gesetzes gegen den unlauteren Wettbewerb (UWG) in § 7 ausdrücklich klargestellt worden. Danach liegt in jeder Versendung kommerzieller Mails ohne Einwilligung des Adressaten eine unzumutbare Belästigung der Marktteilnehmer. Unzulässig ist des Weiteren die Werbung mit Nachrichten, bei denen die Identität des Absenders ver- schleiert oder verheimlicht wird. Im Falle eines Verstoßes gegen § 7 UWG können Wettbewerber und anerkannte Klageverbände vom Versender gerichtlich Unterlassung und Schadensersatz verlangen. Zudem besteht ein Gewinnabschöpfungsanspruch. Die nicht nach dem UWG klageberech- tigten Empfänger von Spam-Mails können daneben auf dem Zivilrechtsweg Schadensersatz- und Unterlassungsansprüche aus unerlaubter Handlung geltend machen (vgl. §§ 823, 1004 BGB)"

Das Landesgericht Hannover (Urteil vom 11. Mai 2006, Az 21 O 153/04) kam der Unterlassungsklage des Heise Verlags[6] nach und zwang ein Unternehmen, dass für die firmeneigene Webseite Werbe-E-Mails versendete, die Versendung von Spam-E-Mails einzustellen. Der Betreiber war nicht in der Lage die Einwilligungen der Werbe-Mail Empfänger nachzuweisen und verstieß somit gegen die Paragraphen 7 und 8 des Gesetzes gegen unlauteren Wettbewerb (UWG).

2.4.2 Rechtslage Phishing

Die strafrechtliche Relevanz spielt bei Phishing Attacken im Gegensatz zu der Spam - Rechtslage eine wesentlich wichtigerere Rolle. Hierbei handelt es sich nicht mehr nur um unlautere Werbung wie bei den Spammern sondern um handfeste Verstöße gegen das Strafgesetz. Die Rechtslage verhät sich wie folgt:

Versenden von Phishing E-Mails

- versuchter Betrug §263,22 StGB - als schadensgleiche Vermögensgefährdung vertretbar

[5] **http://www.computerundrecht.de/docs/entwurf_anti_spam_gesetz_15_2_2005.pdf**
[6] Artikel: http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/73403&words=Spam

- strafbare Vorbereitung eines Computerbetrugs[7] §263a Abs. 3 StGB
- Fälschung beweiserheblicher Daten §269 StGB - Täuschung über den Urheber der Nachricht

Einrichten der Phishing Seite

- beweiserhebliche Daten werden gefälscht §269 StGB, hierbei ist die Beweiserheblichkeit einer Internetseite problematisch

Erlangen der Zugangsdaten

- Ausspähen von Daten §202a StGB, hierbei liegt das Problem, dass das Opfer die Daten freiwillig angibt und daher kein 'Verschaffen' unter Überwindung von Zugangssicherungen vorliegt.

Verwendung der Zugangsdaten

- Computerbertrug § 263a StGB, unbefugtes Verwenden von Daten

Gehilfen der Phisher (Phisherman's Friend)

Solche Strohmänner sind von Phishern per E Mail angeworbene Kuriere, die das illegal gewonnene Geld über ihre Konten zum eigentlichen Phisher transferieren. Diese Kuriere haben in den häufigsten Fällen keinerlei Kenntnis über die wahre Bedeutung ihrer Aufgabe, da sie oft unter dem Vorwand für einen großen ostasiatischen Konzern zu arbeiten angestellt werden. Da diese Zwischenmänner über ihre Kontodaten leicht auszumachen sind, werden sie zuerst gefasst und der Beihilfe zum Computerbetrug angezeigt (Strafbar gem. §§263a, 27 StGB).

Aus dem Gesetzesrahmen lässt sich allgemein ableiten, dass das Versenden von Phishing E-Mails und die Verwendung von gesammelten Daten wie PIN und TAN - Nummern laut Strafgesetzbuch strafbar sind. Jedoch ist das reine Phishing z.B. über Webseiten (Datenbeschaffung) nicht strafbar.

Haftungsrecht

Da hier jährlich Schäden in Milliardenhöhe enstehen ist es für die Geschädigten von hohem Interesse wer für den enstandenen Schaden haftet. Grundsätzlich lässt sich sagen, dass die Haftung bei der Bank liegt, da diese einem Computerbetrug unterlegen ist. Jedoch stehen in den Allgemeinen Geschäftsbedingungen der Banken Klauseln, die die Banken von der Haftung befreit, sofern dem Kunden bei klassischen Phishing - Attacken grobe Fahrlässigkeit nachzuweisen ist. Des Weiteren haftet der Kunde 'bei Untätigkeit nach Bekanntwerden einer Phishing - Attacke' ohne Haftungsbegrenzung. (nach Art. 50 Abs. 2 StGB)

[7] Vorbereitung eines Computerbetrugs gilt nur für lauffähige Anwendungen (funktionsfähige Phishing Seiten)

Strafrecht

Jeden Tag wächst die Zahl der versendeten Spam - E-Mails um ein vielfaches. Dadurch verschärft sich die Situation zunehmend und die Problematik rückt immer mehr in den Focus der Öffentlichkeit. Deshalb wird die Debatte über die strafrechtliche Verfolgung von nicht erwünschten Werbe-E-Mail Absendern immer hitziger. Jedoch ist die Rechtslage in Deutschland noch nicht einheitlich genug um gegen die Spammer strafrechtlich vorzugehen.

Des Weiteren sehen die Staatsanwaltschaften noch immer keinen triftigen Grund und Handlungsbedarf rechtliche Schritte gegen die Spamflut zu unternehmen. Der Grund für die scheinbare Untätigkeit der Staatsanwaltschaft in dieser Thematik lässt sich einfach nachvollziehen. Da im Strafrecht ein Verfolgungs- und Anklagezwang vorliegt, wären die Staatsanwaltschaften sobald eindeutig strafrechtliche Gestzte zur Verfolgung von Absendern von Werbe-E-Mails verabscheidet würden verpflichtet, bei Vorliegen hinreichender Hinweise ein Ermittlungsverfahren von Amts wegen einzuleiten. Dies hätte zum Einen zur Folge, dass jeder Staatsanwalt allen Spam - Mails, die an seine Dienst - E-Mail-Adresse geschickt werden, von Amts wegen strafrechtlich nachgehen müsste. Zum Anderen könnte jeder Bürger der eine Spam - Mail ohne seine Einwilligung empfangen hat Strafanzeige erstatten. Jede dieser Strafanzeigen müsste dann laut Strafrecht verfolgt werden, was bei den gewaltigen Massen an Spam unweigerlich zu einer nicht tragbaren Belastung der Staatsanwaltschaften führen würde.

Ein weiterer Ansatz beschäftigt sich mit der Idee die Unternehmen, die mit Spam - Mails für ihre Produkte werben lassen in die Verantwortung zu nehmen. Dieser Schritt würde jedoch dem Konkurrenzkampf zwischen den Firmen unterstützen, da eine manipulierte Spam - Mail (E-Mail Header Manipulation) zu strafrechtlicher Verfolgung führen würde und ein seriösen Unternehmen in große Schwierigkeiten bringen könnte. Die betroffene Firma müsste dann nachweisen, dass diese Spam - Mail nicht von ihr versendet wurde.

Ausblick

In Anlehnung an die Schwierigkeiten, die mit der Einführung der strafrechtlichen Verfolgung von Spamern einhergehen, wäre noch zu bedenken, wie nach der Einführung z.B. mit Spam - Filtern umgegangen wird. Da das Versenden von Spam - Mails strafrechtlich wäre, würden die Filter zur Vertuschung und Unterschlagung von strafrechtlichen Beweismaterial beitragen.

Abschließend lässt sich sagen, dass es Zeit ist die Spammer rechtlich in ihre Schranken zu weisen. Jedoch sollte die Umsetzung und Formulierung eines wirksamen Anti - Spam - Gesetzes wohl überlegt sein, da sonst vielleicht ein höherer Schaden als bislang entstehen könnte.

2.4.3 Rechtslage in anderen Ländern

Allgemein ist die Rechtslage in Europa durch die Richtlinie des Europäischen Parlaments und eines Rates[8] festgelegt. Hierbei lassen sich keine großen Unterschiede zur deutschen Rechtslage feststellen. Der Kern der europäischen Rechtslage legt wie in Deuschland fest, dass das Zusenden von E-Mail-Werbung nur dann rechtens ist, wenn der Empfänger vorher eingewilligt hat. Auf dieser Basis bauen die jeweiligen Umsetzungen in die nationalen Rechte auf. Speziell in den USA gibt es seit 2003 ein Anti - Spam - Gesetz[9] und seit 2004 ein Anti - Phishing - Gesetz[10]. Diese Gesetze werden seit ihrer Verabschiedung erfolgreich angewandt und können bis zum heutigen Tag mehrere Verhaftungen und Verurteilungen vorweisen.

2.5 Tricks der Spammer

2.5.1 Harvester

Ein Harvester (engl. 'harvest', zu deutsch 'ernten') ist eine kleine Software, die die Aufgabe hat E-Mail Adressen im Internet zu sammeln. Der Vorgang des virtuellen 'Erntens' der Harvester ist vergleichbar mit der Software der größeren Suchdienstanbieter. Die von den Suchmaschinen eingesetzte Software nennt sich 'Crawler' oder 'Spider' und ist zuständig für das Indizieren der Webseiten im Internet. Genau wie die Suchroboter der Suchmaschinen hangelt sich ein Harvester durch die Webseiten und folgt allen Verlinkungen auf dieser Seite. Durch diese Technik steht dem Harvester in kürzester Zeit ein großer Pool an Webseiten zur Verfügung, die er auf E-Mail Adressen untersuchen kann, um diese abzuspeichern.

Im Gegensatz zu den Suchrobotern, die mit einer individuellen Kennung identifizierbar sind, tarnen sich Harvester z.B. mit der offiziellen Kennung eines verbreiteten Browsers. Mit dieser Technik hebeln die Harvester die Gegenmaßnahmen der Webmaster, z.B. den Zugriff solcher Programme durch einen Eintrag in den Meta Daten einer Webseite zu verhindern, aus.

Meta-Angaben Beispiel : *<meta name="robots" content="noindex">*
Die Anbieter solcher Harvester Programmen sehen es natürlich nicht gerne wenn ihre Tools als E-Mail Harvester tituliert werden und nennen ihre Software auf Grund dessen 'Mail-Spider' oder sogar 'Online Marketing Tool'.

Da jedoch das Sammeln von E-Mail Adressen durch einen Harvester im Normalfall schnell von statten gehen muss, arbeiten diese Programme meist zu ungenau. Diese Ungenauigkeit führt dazu, dass alle Zeichenfolgen in denen ein @ Zeichen vorkommt als E-Mail Adresse identifiziert werden und in Folge dessen die Datenqualität leidet. Diese Schwäche der zumeist illegal genutzten Software kann nun dazu verwendet werden, dem Harvester möglichst viele falsche Adresse anzubieten und somit seine E-Mail - Listen zu vergiften. Das so genannte

[8] Rate über die Verarbeitung personenbezogener Daten und den Schutz der Privatsphäre in der elektronischen Kommunikation

[9] Anti-Spam Act of 2003

[10] Anti-Phishing Act of 2004

'vergiften' (Poisoning3.3) des gesammelten Datenbestands ist eine beliebte Methode, um die Effektivität dieser Tools zu mindern.

Eine weitere Methode (siehe 3.2) befasst sich mit der Identifikation von Benutzern, die mit Hilfe von Harvester Programmen mögliche Spam Empfänger Adressen herausfinden.

2.5.2 Phishing Mail

Diese Art von Spam Mails hat sich in den letzten Jahren zu einem eigenen Zweig der Internetkriminalität entwickelt. Hierbei werden dem Opfer in der Nachricht gefälschte und erfundene Tatsachen näher gebracht, die ihn dazu veranlassen sollen, geheime Daten, Login-Informationen abzugeben oder direkt Geld zu überweisen. Phishing - Mails tarnen sich oft als offizielle E-Mails von seriösen Internet Providern oder Bankinstituten. Der Kreativität der Absender von Phishing Mails ist hier keine Grenze gesetzt.
Es gibt inzwischen einige Arten von Phishing Angriffen, die im Folgenden kurz erwähnt werden sollen.

2.5.2.1 Geld überweisen In der Phishing Mail steht die Aufforderung eine Rechnung zu begleichen.

Der Betroffene wird darüber informiert, dass er noch eine Rechnung zu begleichen hätte und wird aufgefordert den ausstehenden Betrag unverzüglich auf ein bestimmtes Konto zu überweisen. Dies ist meist das Konto eines Mittelsmannes.

2.5.2.2 Login - Informationen Eine Aufforderung zum Login auf einer gefälschten Webseite um die Logindaten des Benutzers in Erfahrung zu bringen.

Diese Seiten sind den Orginalseiten nachempfunden, so dass der Benutzer möglichst keinen Verdacht schöpfen kann und ohne sein Wissen seine persönlichen Zugangsinformationen angibt. Manchmal wird auch direkt in der HTML - E-Mail ein Formular eingebunden, das nach dem Eintragen der gewünschten Informationen die Daten unmittelbar an den Phisher sendet. Eine gefälschte Fehlermeldung oder Bestätigung sollen in diesem Fall eventuell auftretendes Misstrauen des Opfers zerstreuen. Diese Informationen werden dann auf Kosten des Benutzers kriminell missbraucht. Ein beliebtes Ziel solcher Attacken ist das Auktionshaus ebay.de, da hier nur ein Benutzername und ein Passwort nötig sind um die volle Kontrolle über den gesamten Benutzer - Account zu erlangen.

2.5.2.3 Kreditkarten Bei dieser Thematik verhält es sich sehr ähnlich wie bei dem Beispiel 2.5.2.2 zuvor. Hier wird der Empfänger aufgefordert auf einer dafür präparierten Webseite seine Kreditkarten - Informationen anzugeben oder sie direkt als Antwort an den Versender zurückzusenden. Kontaktdaten des Benutzers, dessen Kreditkartennummer und dessen Geheimnummer in den falschen Händen können sehr viel Schaden anrichten, für den sich am Schluss keiner verantwortlich fühlt.

2.5.2.4 Bank Phishing Die wohl bekannteste Art des Phishing Mail Prinzips ist das Online - Banking Phishing. Die E-Mail ist hier als eine Benachrichtigung eines als seriös bekannten Geldinstituts getarnt. Diese Tarnung wird meistens durch die Verwendung des Firmenlogos und des Corporate Designs[11] der Bank umgesetzt.

Die Phishing Mail besteht aus nur einem Bild auf dem die Informationen stehen, die den Benutzer dazu bringen sollen, auf den Betrug einzugehen. Des Weiteren befindet sich auf dem Bild ein Link, der angeblich auf die richtige Webseite eines Bankinstituts weiterleitet. Jedoch ist dieser Link völlig nutzlos, da er nur Teil eines Bildes ist. Nicht der zu sehende Link sondern die Verlinkung des Bildes ist für die Weiterleitung verantwortlich. Dies wird raffiniert im Quellcode der E-Mail versteckt.

```
<a hREf=http://www.volksbank.de.vrnetworld.c10133577.box4ks.
info/r1/vr/><img src="http://www.E-Mailprovider.net/de/cgi
/msgpart?m=89...&PARTNO=1 &LANG=de" border=0></a>
```

Sobald der Benutzer auf das Bild in der Nachricht klickt wird er automatisch auf die falsche Webseite (`www.volksbank.de.vrnetworld.c10133577.box4ks.info/r1/vr/`) geleitet anstatt auf die im Link angegebene URL (`http://www.volksbank.de/vrnetworld/c101...`).

Sehr geehrter Kunde, sehr geehrte Kundin,

Die Technische Abteilung der Volksbanken Raiffeisenbanken führt zur Zeit eine vorgesehene Software-Aktualisierung durch, um die Qualität des Online-Banking-Service zu verbessern.

Wir möchten Sie bitten, unten auf den Link zu klicken und Ihre Kundendaten zu bestätigen.

http://www.volksbank.de/vrnetworld/c101335777777.nsf/XC701133.asp

Wir bitten Sie, eventuelle Unannehmlichkeiten zu entschuldigen, und danken Ihnen für Ihre Mithilfe.

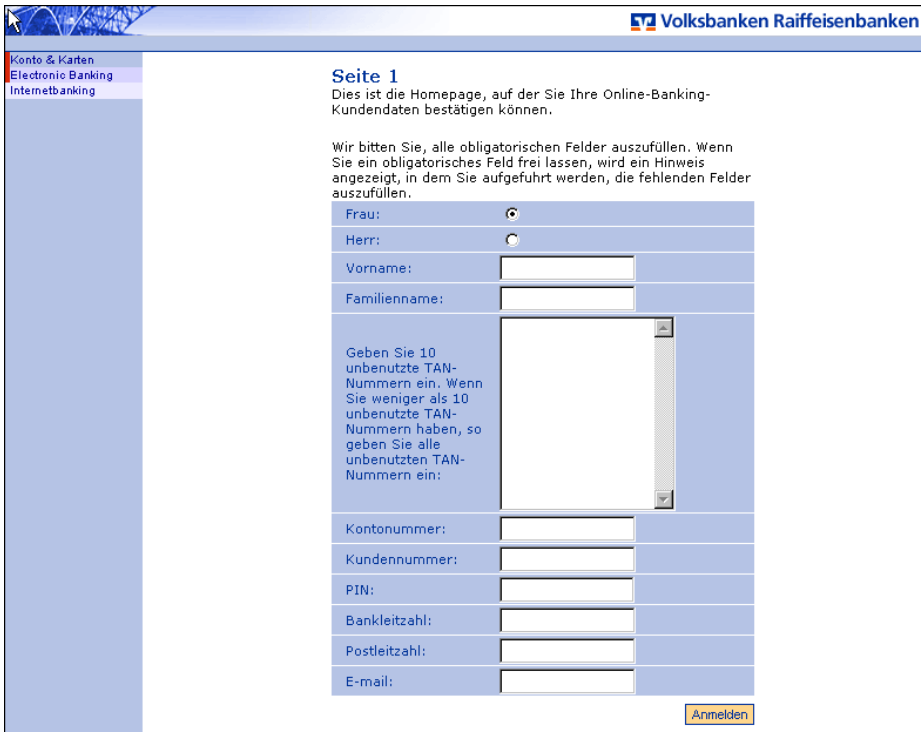
In diesem Beispiel handelt es sich um die gefälschte und nachgeahmte Webseite des Bankinstituts Volksbanken Raiffeisenbanken.

Diese nachgemachten Webseiten haben das Ziel, den Orginalseiten so ähnlich wie nur möglich zu sein, so dass möglichst wenige Benutzer Verdacht schöpfen. Dies wird erreicht, indem der Benutzer den selben Aufbau der Seite sowie die

[11] Corporate Design - das gesamte visuelle Erscheinungsbild eines Unternehmens - dient dem Zweck des Wiedererkennungseffektes

'Klickfolge' vorfindet. So wird erreicht, dass sich das Opfer in einer 'vertrauten' Umgebung befindet und nicht misstrauisch wird. Wie man an dem Quellcodebeispiel (weiter oben) leicht erkennen kann, verwenden die Angreifer für die Weiterleitung auf ihre gefälschten Seiten URLs, die der der Orginalseite sehr ähneln. Die kleinen, oft unmerklichen Abweichungen in der Adresse machen es den Benutzern fast unmöglich zu erkennen, ob sie sich auf einer Phishingseite befinden oder nicht.

Wie schon im Text der Phishing Mail zu lesen ist, soll der Empfänger nun auf dieser Seite seine Kundendaten eintragen. Hier werden dann meistens Kundendaten, Kontonummer, Zugangspin und TAN Nummern verlangt.



Bei diesem Beispiel sieht man sehr schön, dass die Phishingseite im Corporate Design der Orginalseite gehalten ist und welche Informationen vom Benutzer verlangt werden.

Webseiten, die das Ziel haben solche Informationen zu sammeln sind oft nur wenige Stunden lang nach dem Versand der Phishing Mails online. Leider reicht dieser Zeitraum oft schon aus, um an viele Daten von Kunden zu kommen. Mit diesen Kontozugangsdaten in Verbindung mit einer TAN Nummer nimmt der Phisher die Identität seines Opfers an und verwendet die Informationen für illegale Transaktionen. Den Inhabern dieser Konten ist es dann fast unmöglich nachzuweisen, dass eine andere Person diese Transaktionen vollzogen hat und erhalten deshalb auch nicht mehr ihr Geld zurück.

Die durch solche Attacken entstanden Schäden sind kaum abzuschätzen, werden jedoch auf mehrere hundert Millionen Euro bis hin zu Milliarden-Beträgen geschätzt.

2.5.2.5 Spear - Phishing Beim Spear - Phishing (engl. 'spear', zu deutsch 'Speer') besorgen sich die Phisher vorher gezielt E-Mail Adressen über eine zentrale Stelle wie z.B. die Newsgroup einer Hochschule. An diese Adressen schickt der Angreifer dann Phishing Mails, die vorgeben, von den lokalen Banken einer Stadt zu kommen. Wenn der Phisher zum Beispiel E-Mail-Adressen von Studenten in Stuttgart gesammelt hätte (mit einem Harvester2.5.1 Programm), würde er in seiner E-Mail vorgeben, sie käme von der LBBW, die in Stuttgart sehr bekannt ist. Hierbei wäre die Chance, das ein Student ein Konto bei diesem Bankinstitut hat, sehr viel höher als wenn er diese Phishing Mail an völlig willkürlich gesammelte Adressen schicken würde. Diese Art von Phishing Mails hat auf Grund der Trefferwahrscheinlichkeit einen wesentlichen höheren Wirkungsgrad.

2.5.2.6 Phishing mit Trojanischen Pferden Bei dieser Art versteckt sich in der Anlage einer E-Mail ein Trojanisches Pferd (siehe auch Pharming 2.5.3). Sobald der Benutzer die Anlage öffnet, verändern diese kleinen Programme die Hosts - Datei [12] des Betriebssystems. Damit erreichen Sie, dass sogar beim Aufruf der Orginalseite immer eine gefälschte Seite erscheint. Diese Umleitung auf eine andere Seite fällt dem Benutzer nur sehr schwer auf, da die richtige Adresse von ihm angegeben worden ist.

2.5.2.7 Instant Messenger In diesem Bereich fand das Phishing seinen Anfang. Ende der 90er wurden die ersten Benutzer einschlägiger Instant Messenger wie z.B. ICQ aufgefordert ihre Kontakt- und Zugangsdaten auf Webseiten anzugeben. Das Prinzip hat sich nicht groß verändert. Nur die Konsequenzen eines übernommenen Chataccounts sind verschwindend gering gegenüber dem Verlust eines größeren Geldbetrags.

2.5.3 Pharming

Beim Pharming handelt es sich um eine Weiterentwicklung des Phishings. Der Begriff entstand durch die Tatsache, dass die Betrüger große Server - Farmen oder Sammlungen von gefälschten Seiten unterhielten. Bei dieser Technik wird mit Hilfe von Trojanischen Pferden, Viren oder Malware[13] Programmen gezielt die lokale Hosts - Datei der Betriebssysteme verändert, so dass der Benutzer trotz richtiger Eingabe der URL auf die vom Phisher gefälschte Seite gelangt. Der Trick hinter der Sache ist, dass sich ein Betriebssystem eine interne Liste in der Hosts - Datei hält, in der die Zuordnungen der Internetadressen zu den

[12] Die HOSTS-Datei ist eine lokale Textdatei die der Zuordnung von Hostnamen und IP-Adressen dient.

[13] Als Malware bezeichnet man Computerprogramme, welche vom Benutzer unerwünschte (schädliche) Funktionen ausführen

jeweiligen IP - Adressen stehen, um nicht jedes mal bei einem DNS - Server anzufragen. Sobald diese Datei verändert wurde kann dieser Rechner nur noch die gefälschten Seiten besuchen. Zwar gibt der Benutzer immer die richtige Internetadresse an, diese ist nur ab diesem Zeitpunkt immer auf eine falsche IP - Adresse gemapt.

Durch korrumpierte DNS - Server oder DNS - Flooding, bei dem einem Rechner eine Adressauflösung auf Verdacht untergespielt wird, bevor dieser beim DNS - Server nachfragen kann, werden die Benutzer trotz richtiger URLs auf gefälschte Server umgeleitet.

3 Spam Bekämpfung

3.1 Der verwendete HoneyPot - ProxyPot

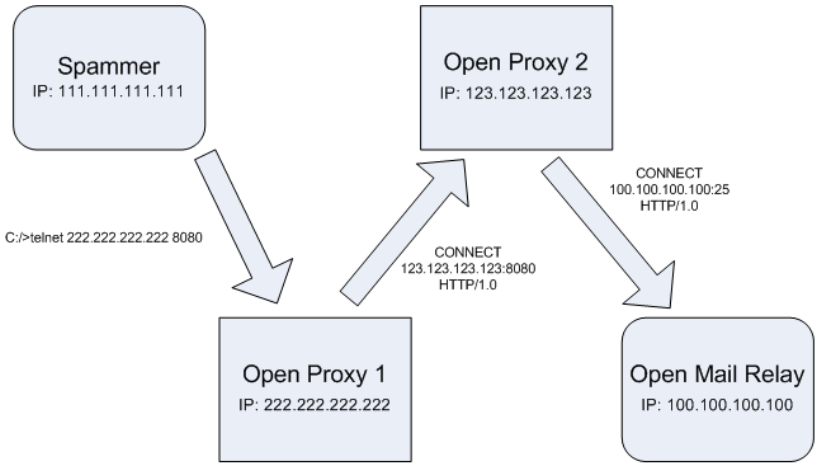
Es existiert zwar eine große Anzahl an frei verfügbaren Honeypot-Produkten auf dem Markt, für unsere Zwecke war allerdings der "Bubblegum Honeypot" von Alan Curry am besten geeignet. Dieser Honeypot, hier auch Proxypot[14] genannt, ist speziell darauf zugeschnitten, Spammer zu fangen. Er beinhaltet standardmäßig einige Funktionalitäten, die wir bei anderen Produkten erst umständlich Konfigurieren oder sogar selbst Programmieren müssten. Allerdings handelt es sich um ein kompaktes und nicht 100% ausgereiftes Programm, der gesamte Perl-Code ist in einer Datei mit knapp 6000 Zeilen untergebracht. Der Proxypot kann kostenlos unter `http://proxypot.org/download.html` heruntergeladen werden.

3.1.1 Warum ein HTTP-Proxy und kein SMTP-Relay?

Möchte man im Internet etwas Illegales machen, was nicht nachverfolgbar sein soll, so ist es eine gute Idee, dies über einen Proxy zu tun. Das "Opfer" sieht in diesem Fall nur die Adresse des Proxies und nicht die Adresse des tatsächlichen Täters. Diese Adresse steht nur in den Logfiles des Proxies, die oft nach einigen Tagen gelöscht werden oder nicht einmal existent sind. Geht man nun noch über mehrere Proxies in unterschiedlichen Ländern, so wird es fast unmöglich die tatsächliche Adresse des Angreifers herauszufinden.

Nach diesem Prinzip funktioniert auch oft der Versand von Spam[15]. Das für uns interessante Szenario funktioniert mit einer Kombination aus offenen Proxies und einem offenen Mail-Relay.

[14] Der Begriff Proxypot kommt von "Open Proxy Honeypot"
[15] Siehe hierzu auch Kapitel 5.3



Im hier skizzierten Szenario verbindet sich der Spammer mit dem Open Proxy 1, dann über einen HTTP CONNECT-Befehl mit dem Proxy 2 und letztendlich wieder über einen HTTP CONNECT mit dem offenen Mail Relay. Damit dies funktioniert müssen bei den beteiligten Maschinen einige Konfigurationsfehler vorhanden sein:

- Alle Maschinen müssen komplett offen sein, d.h. von jedem Rechner Verbindungen akzeptieren.
- Der HTTP CONNECT Befehl muss aktiv sein und darf nicht geprüft werden. Dieser Befehl ist eigentlich zum Tunneln von sicheren SSL-Verbindungen gedacht, kann aber auch auf diese Weise missbraucht werden um Proxy Chains aufzubauen.
- Es wird bei der Verbindung auf das Mail Relay als Protokoll HTTP angegeben, was gelogen ist, da an dieser Stelle SMTP verwendet wird. Proxy 2 muss also alle Strings auf das Mail Relay zulassen, auch wenn es keine gültigen HTTP-Befehle sind.

Unser Ziel ist es nun, die Position von Proxy 1 einzunehmen, da wir nur so an die tatsächliche Adresse des Spammers kommen können. Stehen wir an Stelle von Proxy 2, so können wir zwar die IP von Proxy 1 sehen, aber nicht die tatsächliche Adresse des Spammers.

Der Spammer hat nun in unserem Fall prinzipiell drei Möglichkeiten, sich mit dem Proxypot zu verbinden:

- Über Port 1080 mit einer SOCKS- (Version 4 oder 5) Anfrage
- Über Port 8080 mit dem oben beschriebenen HTTP CONNECT
- Über Port 3128, der Proxypot simuliert hierbei einen Squid-Proxy

Nachdem er sich erfolgreich mit dem Proxypot verbunden hat, hat er wiederum mehrere Möglichkeiten:

- Er verbindet sich direkt mit einem Mail Relay. In diesem Fall simuliert der Proxypot diese Verbindung und Kommunikation mit dem Relay komplett. Der Spammer hat das Gefühl, ein offenes Relay gefunden zu haben und sendet seinen Spam. Dieser wird natürlich nicht weitergesendet sondern auf der Festplatte gespeichert und in den Logfiles protokolliert. Da der Proxypot nicht weiß, welcher Art das zu kontaktierende Mail Relay ist, kann die Simulation natürlich auch nicht 100% perfekt sein. Man kann allerdings in den Konfigurationsdateien einstellen, welcher Mailserver simuliert werden soll.

- Er verbindet sich mit einem weiteren Proxy. Der Proxypot simuliert nun alle anderen, folgenden Proxies in der Kette. Irgendwann wird der Spammer sich auf einen Mailserver kontaktieren wollen, dann wird dieser wie oben beschrieben ebenfalls simuliert.

- Er sendet einen Socks oder HTTP-Request, der nichts mit den beiden oben beschriebenen Möglichkeiten zu tun hat. Hier ein Beispiel aus den Proxypot Logfiles:
  ```
  Accepted connection on 141.62.88.100:8080 from 222.137.175.138:2052,
  created process 4471
  Received HTTP command from client: "GET http://www.proxygrade.com/
  proxygrade.php?hash=BF0967506947519B8C97 HTTP/1.0\r\n"
  Faking empty reply
  Cleaned up child process 4471
  ```
  Der Client sieht bei dieser Anfrage folgendes:
  Anfrage:
  ```
  C:/>telnet 141.62.88.100 8080
  GET http://www.proxygrade.com/proxygrade.php?hash=BF09675069
  47519B8C97 HTTP/1.0
  ```
  Antwort:
  ```
  HTTP/1.0 200 OK Content-Length: 7 Content-Type: text/html;
  charset=iso-8859-1 Proxy-Connection: close
   
  Verbindung zu Host verloren.
  ```
  Er bekommt also eine HTTP 200 OK Meldung und eine leere Antwort zurück.

3.1.2 Installation des Linux-Systems

Wir entschieden uns als Basisplattform für den Proxypot für ein SuSE 10.0 Linux, da dies bekannt, kostenlos und schnell verfügbar ist. Prinzipiell ist es egal, welches Linux genutzt wird. Bei der Installation wurde darauf geachtet, alle nicht benötigten Dienste abzustellen bzw. erst gar nicht zu installieren. Um den Projektrechner sicherer zu machen mussten noch einige Änderungen

durchgeführt werden. Zunächst wurde der Rechner mit nmap gescannt. Hierbei erkannten wir, dass noch einige Ports offen waren, dessen Dienste wir nicht benötigten. Diese wurden über den runlevel-editor im yast abgeschalten. Im Detail handelte es sich um folgende Dienste:

- mdns responder (Kümmert sich um Apple rendevouz DNS requests)
- postfix MTA (Port 25) (Könnte gefährlich sein falls die Maschine übernommen wird. Es könnte problemlos(er) echter Spam von der Maschine aus versendet werden)
- portmap (wird nur für RPC-Dienste benötigt, also in unserem Fall nicht)
- Alle Samba-Dienste

Es wurden noch einige andere Dienste entfernt, diese waren allerdings nicht an einen spezifischen Port gebunden und somit in diesem Kontext nicht interessant. Man sieht in den Reports von Nessus und von nmap den vorher-nachher Unterschied.

Nmap-Scan vorher:

```
C:\>nmap 141.62.88.100 -oN -A -p1-65535
Starting Nmap 4.03 ( http://www.insecure.org/nmap )
at 2006-05-24 11:33
Interesting ports on 141.62.88.100: (The 65531 ports scanned
but not shown below are in state: closed)
PORT STATE SERVICE
22/tcp open ssh
111/tcp open rpcbind
631/tcp open ipp
1720/tcp filtered H.323/Q.931
Nmap finished: 1 IP address (1 host up) scanned in 943.286 seconds
```

Nmap-Scan danach:

```
Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2006-05-24
15:16
Interesting ports on 141.62.88.100: (The 65534 ports scanned but not
shown below are in state: closed)
PORT STATE SERVICE
22/tcp open ssh
Nmap run completed -- 1 IP address (1 host up) scanned
in 619.110 seconds
```

Port 22 musste offen gelassen werden, da die Administration der Maschine über SSH erfolgt. Nach dem Einrichten des SSH-Zugangs auf dem Projektrechner kam beim Versuch sich remote einzuloggen immer die Fehlermeldung "No more authentication methods available". Diese Meldung kommt, wenn alle verfügbaren Möglichkeiten zum Login ausgeschöpft sind, und keine funktioniert hatte. Da bei uns kein PKI Login eingerichtet war, bedeutete dies, dass mit dem

Passwort-Login etwas nicht stimmte. Nach kurzer Suche war das Problem gefunden. Im SSH-Server von Linux musste erst in der Datei /etc/ssh/ssh_config die Option PasswordAuthentication auf "yes" gestellt werden. Der direkte root-login über SSH wurde zur Sicherheit verboten. Die alten, unsicheren SSH-Versionen vor 1.99 wurden ebenfalls verboten.

Außerdem wurde mit Nessus noch ein Security Check durchgeführt. Die Nessus-Protokolle liegen dieser Dokumentation im Anhang bei.

3.1.3 Installation des Proxypot

Die Installation der Software gestaltet sich theoretisch relativ einfach, praktisch ist dies jedoch ein recht langatmiger Prozess.

Zunächst muss das Programm von`http://proxypot.org/download.html`heruntergeladen werden. Die Dateien liegen alle als Perl-Sources vor. In der Datei Proxypot-0.7.tar.gz finden sich folgende Programme:

- proxypot (Aktuelle Version: 20050321), der eigentliche Proxypot
- spamstat - Dieses Programm dient zum Analysieren der gesammelten Nachrichten und zum Erstellen von Reports
- spamstat-cgi - Dieses Programm kann dynamisch Reports erstellen, die dann über einen Webserver abgerufen werden können.
- deliverone - Hiermit kann man einzelne Testmails tatsächlich versenden, die von den Spammern genutzt werden, um die Funktionalität des Proxies zu überprüfen. Wird von uns nicht verwendet, da wir sämtliche Maildienste auf dem Rechner aus Sicherheitsgründen komplett deaktiviert haben. Wir benutzten zum Zurücksenden der Testmails eine andere Methode.
- spamstatcvt - Um alte Spamstat Datenbanken auf das aktuelle Format zu konvertieren.
- log2mbox - Mit diesem Tool können aus (alten) Logfiles die darin enthaltenen Mails extrahiert werden.

Um diese Programme installieren zu können müssen einige Vorbedingungen erfüllt werden. Zunächst muss sich auf dem Rechner eine lauffähige Perl-Umgebung befinden. Wie diese installiert wird, soll hier nicht weiter beschrieben werden, da dies sich zwischen den unterschiedlichen Linux-Distributionen stark unterscheidet.

Als erstes müssen einige benötigte Perl-Module nachinstalliert werden, da diese nicht standardmäßig installiert sind. Die Module können auf zwei verschiedene Arten installiert werden:

1. - www.cpan.org in den Browser laden
   - Einen Link zu dem Modul suchen
   - Das Modul herunterladen

- Das Modul in ein geeignetes Verzeichnis entpacken
Auf der Konsole:
- perl Makefile.PL
- make
- make test
- su (falls man nicht als Root angemeldet ist)
- make install
- exit
Dies ist jedoch sehr umständlich und dauert recht lange.

2. Es geht einfacher, indem man zunächst das installer-Modul von CPAN (nennt sich auch CPAN) installiert. Nun kann man Module mit einer einfachen Zeile auf der Konsole installieren:
`perl -MCPAN -e 'install Modulname'`
Man muss hierzu nun nur noch den Modulnamen kennen. Allerdings funktioniert diese Art der Installation mit einigen Modulen nicht problemlos, diese muss man dann auf die oben beschriebene Art installieren.

Welche Module im einzelnen benötigt werden, kann auf`http://proxypot.org/download.html` nachgelesen werden. Da es relativ viele Module sind und einige sich nicht problemlos installieren lassen, dauert dieser Vorgang relativ lange. Man kann circa einen Arbeitstag einplanen bis alle Module installiert sind.

Hat man alle Vorbereitungen abgeschlossen, kann der eigentliche Proxypot installiert werden:
$ gunzip -c Proxypot-*.tar.gz | tar xvf -
$ cd Proxypot-*/
$ perl Makefile.PL
$ make
$ make install

Ob die Installation erfolgreich war, kann man nun feststellen indem man einfach den Befehl "Proxypot" in die Konsole eingibt. Kommen keine Fehler und es erscheint in der Logdatei des Proxypot eine Zeile wie "Bubblegum Proxypot XXX starting", war alles erfolgreich.

Um nun Spamstat problemlos nutzen zu können, muss noch das Modul MIME::Parser gepatcht werden. Der Patch ist eine diff-Datei und kann wie üblich mittels "patch" eingespielt werden. Hierbei ist nur das Problem, dass nur ein Patch für die 5 Jahre alte Version 5.411 verfügbar ist, die aktuelle Version ist 5.420. Somit muss auch eine alte Version der MIME-Tools auf dem System installiert werden.

Es sollte drauf geachtet werden, dass die im System installierte Festplatte ausreichend groß ist, bzw dass das Logfile und das MailDir auf einer großen Platte gespeichert werden. Im Betrieb kann der Proxypot problemlos mehrere Gigabyte an Daten pro Tag sammeln.

3.1.4 Konfiguration des Proxypot

Der Proxypot wurde zu Beginn des Experimentes sehr defensiv und auf Sicherheit konfiguriert. Nachdem wir bemerkten, dass alles problemlos funktionierte, lockerten wir die Restriktionen ein wenig. Die Möglichkeiten des Proxypot sind allerdings immer noch bei weitem nicht ausgereizt.

Unser "Grundgesetz" bei der Konfiguration lautete: Alles darf rein, nichts darf raus. Es werden also alle Antworten des Proxies komplett simuliert. Auf einen HTTP-GET Request kommt immer ein HTTP-200-OK zurück, auf ein Reconnect auf einen SMTP-Server kommt eine simulierte SMTP-Verbindung. Die Verbindung zu dem tatsächlichen Zielrechner wird nie aufgebaut.

Es sollen nun die wichtigsten Konfigurationsdateien, ihre Funktion und unsere Einstellungen kurz beschrieben werden. Die Dateien sind auf unserem System im Verzeichnis /etc/proxypot zu finden.

- allow-deny: In dieser Datei kann festgelegt werden, welche Anfragen und Adressen erlaubt sind, und welche nicht erlaubt sind. Dies wird über Regelsätze festgelegt. Eine Regel besteht immer aus zwei Teilen, der erste Teil gibt die Ziel- oder Quelladresse oder -Port an, der zweite Teil beschreibt, was mit dieser Anfrage geschehen soll. Als Aktionen stehen zur Auswahl: Simulieren einer HTTP(s)-Verbindung, eines SOCKS-Proxies, einer SMTP-Verbindung oder die Funktionsweise eines echten Proxies. Die Aktionen können jeweils noch in drei Simulationsgrade unterteilt werden. Da es hier sehr viele Einstellmöglichkeiten gibt, die in der Konfigurationsdatei selbst auch sehr gut dokumentiert sind, werden an dieser Stelle nur zwei Beispiele gezeigt:

  1. `destaddr=141.62.0.0/16`
     `action=deny`
     In der ersten Zeile wird beschrieben, dass die darunterstehende Aktion nur ausgeführt wird, wenn der Client sich auf eine IP-Adresse im angegebenen Block verbinden möchte. In diesem Fall sind Verbindungen auf Rechner im Bereich141.62.xxx.xxx (das Netz der HdM) sicherheitshalber komplett verboten. Es würde zwar auch ohne diese Einstellung niemals eine echte Verbindung zu einem anderen Rechner aufgebaut werden, aber sicher ist sicher.
  2. `destport=8080`
     `action=httpc1`
     Wenn sich ein Client auf den Port 8080 verbindet (nicht am Proxy, an dem Zielrechner hinter dem Proxy), soll eine HTTP-Verbindung gefaked werden. Jede syntaktisch gültige eingehende Anfrage wird mit einer leeren HTTP 200 Antwort beantwortet. Es gibt hier auch noch zwei weitere Arten der Emulierung. Bei httpc2 wird die Existenz des Zielrechners überprüft, falls er existiert werden alle gültigen Anfragen mit HTTP 200 beantwortet. Wird httpc3 angegeben, so wird die Verbindung zum Zielrechner hergestellt und ein HTTP Connection Request durchgereicht wenn das Ziel explizit erlaubt wurde.

- dcc_*: Über die dcc_* Dateien kann konfiguriert werden, dass der Proxy an einem DCC Netzwerk teilnimmt. Da diese Funktion für uns in diesem Moment nicht relevant ist, wurde sie komplett deaktiviert und wird somit hier auch nicht behandelt.

- debugfile: Wenn der Proxypot das Signal SIGUSR1 erhält, werden alle Variablen in die hier angegebene Datei geschrieben. In unserem Fall handelt es sich um die Datei /etc/proxypot/proxypot_debugfile

- direct_http_commands: Um herauszufinden, ob es sich bei dem Proxy um einen funktionierenden Proxy handelt, gibt es prinzipiell zwei Möglichkeiten:

  1. Es wird ein HTTP CONNECT auf Port 80 eines anderen Servers gemacht und kontrolliert, ob die Verbindung durchkommt und wie lange sie benötigt. Diese Art der Abfrage kann über die normale allow_deny-Tabelle abgefangen werden.
  2. Es wird direkt ein HTTP GET-Befehl an den Proxy gesendet und überprüft, ob die Zieldatei vom Proxy angefasst wird. Das Verhalten bei diesem Befehl wird in dieser Datei eingestellt. Es gibt hierbei die Möglichkeiten allow, deny und fake. In unserem Fall wurde diese Option auf fake eingestellt, da ein Erlauben der Requests zu riskant wäre und ein totales Verbot uns vermutlich weniger Spam einbringen würde.

- incoming: Dies ist eine der wichtigsten Konfigurationsdateien. Hier wird eingestellt, welche Ports am Honeypotrechner offen stehen und welche Dienste dahinter liegen. Folgende Ports sind an unserem Honeypot offen:

  - Port 3128 (Default Port für Squid), es wird HTTP emuliert
  - Port 8080 (Default Port für Webcache), hier wird ebenfalls HTTP emuliert
  - Port 1080 (Default SOCKS Port), hier wird dynamisch SOCKS v4 oder v5 emuliert, je nachdem welche Anfrage eingeht.
  - Außerdem ist noch Port 22 für remote SSH offen, dies hat aber nichts direkt mit dem Honeypot zu tun.

- logfile: Hier kann angegeben werden, wo das logfile für den Proxypot liegt. In unserem Fall liegt es unter /var/log/proxypot

- logrotate_seconds: Das Logfile des Proxypot wird nach einer bestimmten Zeit wieder von vorne begonnen. In unserem Fall steht dieser Wert momentan auf 10 Tage, es ist aber zu überlegen diesen Wert kleiner zu machen und das Logfile regelmäßig über einen Cron-Job automatisiert zu sichern.

- maildir: Hier kann der Pfad angegeben werden, wo die gesammelten Nachrichten gespeichert werden sollen. Dieses Verzeichnis befindet sich bei uns unter /etc/proxypot/maildir-proxypot. Dieses Verzeichnis sollte auf einer großen Festplatte erstellt werden, da in kürzester Zeit immense Datenmengen zusammen kommen können.

- max_connects_per_destination_*_per_10_minutes: Hier kann eingestellt werden, wie viele Verbindungen zu einem Ziel innerhalb von 10 Minuten aufgebaut werden dürfen. Diese Einstellung kann in unterschiedlichen Granularitäten erfolgen. Es kann eingestellt werden, wie viele Verbindungen global, egal zu welcher Maschine in 10 Minuten aufgebaut werden (bei uns 800) bis hin, wie viele Verbindungen zu einem Host aufgebaut werden dürfen (bei uns 10).

- max_simultaneous_connections_per_client_*: Wie der Name der Datei schon suggeriert, besteht hier die Möglichkeit, die maximale Anzahl von Verbindungen pro Client zur begrenzen. Auch hier kann man wieder unterschiedliche Einstellungen für alle Verbindungen (bei uns 100), für Verbindungen die von der IP-Adresse des Clients mit der Subnetzmaske /24 (hier: 30), der Subnetzmaske /16 (hier: 60) oder eines Rechners (hier: 30) aufgebaut werden, begrenzen.

- max_fake_proxy_chain_length: Die Einstellung in dieser Datei hat in unserem Fall keine Auswirkung, da wir in der allow_deny-Table keine Verbindungen zu anderen Proxies erlauben. Ansonsten würde die Kette der miteinander verbundenen Proxies (egal ob realer Proxy oder simulierter Proxy) auf die hier eingetragene Zahl begrenzt werden, eine Kette über maximal 100 Rechner ist erlaubt. Die Begrenzung dient in erster Linie zur Vermeidung von DoS-Angriffen über geschleifte virtuelle Proxies.

- max_real_proxy_chain_length: siehe max_fake_proxy_chain_length

- max_kilobytes_per_second: Dient zur Begrenzung der vom Proxypot verwendeten Bandbreite. Unsere momentane Einstellung ist 500kb/s, was allerdings nur bei der schnellen Standleitung der HdM Sinn macht. Bei einem DSL 1000-Zugang wäre eine Einstellung von ca. 100kb/s maximal noch sinnvoll. Man sollte bei sehr schnellen Zugängen diese Einstellung allerdings eher mit Hinblick auf die anfallende Datenmenge wählen, bei maximal 5 Gigabyte Spam pro Tag (da der Spam als Datei gesichert wird und dazu noch in der log-Datei protokolliert wird, entspricht dies einer gespeicherten Datenmenge von circa 10 Gigabyte / Tag) sind in dieser Datei 57 kb/s einzustellen. Im "täglichen Betrieb" des Proxypot ist der momentane Wert also noch anzupassen.

- runas_user: Sollen von Proxypot well-known-Ports[16] gebunden werden, muss der Proxypot unter der Benutzerkennung root gestartet werden. Da

[16] Portnummer kleiner 1024

dies bei eventueller Kompromittierung des Proxypot einige Sicherheitsrisiken birgt, kann in dieser Datei konfiguriert werden, unter welcher Benutzerkennung der Proxypot weiterlaufen soll nachdem die Ports gebunden sind. Es muss hier also ein existenter User des Systems eingetragen werden, der Zugriff auf alle für den Proxypot notwendigen Dateien hat.

- smtp_servertype_default: Da sich der Proxypot bei uns nicht auf einen echten SMTP-Server verbindet, muss angegeben werden, welcher Servertyp simuliert werden soll. Gibt man in diese Datei als Servertype "help" ein und startet den Proxypot, so gibt er alle simulierbaren SMTP-Servertypen aus. Wir entschieden uns hier für den qmail-Server, da er recht weit verbreitet ist und seine Ausgaben sehr minimalistisch und somit leicht zu simulieren sind.

3.1.5 Bekanntmachen des Proxypot

Dieser Teil des Projektes war wohl einer der Langwierigsten und Kompliziertesten. Wir mussten erst einiges über die Vorgehensweise der Spammer herausfinden - wo bekommen diese die Mailadressen her - um endlich einen Erfolg verbuchen zu können. Dass der Proxypot von selbst gefunden wird mag möglich sein, wir hatten ihn allerdings circa einen Monat lang laufen ohne dass ein einziger Request von außen kam. Dies ist eigentlich auch kein Wunder, da es selbst bei einer großen Anzahl von Scannern sehr lange dauert bis alle im IP-Namespace möglichen Rechner gescannt sind. Es sind außerdem auch nur die wenigsten Spammer, die selbst IP-Ranges scannen, es gibt sehr viel schnellere und bequemere Wege an die Adressen von offenen Proxies zu kommen, wie unten beschrieben.

3.1.5.1 Über Webseiten und Gästebücher Der erste Ansatz war die Verbreitung der Adresse des Proxypot über diverse Webseiten. Die Adressen wurde für den Menschen unsichtbar, für Bots sichtbar auf diversen Webseiten publiziert. Markiert man den weißen Bereich unter dem Dilbert-Comic auf`http://www.hdm-stuttgart.de/~ms096/`, so erkennt man dies. Allerdings scheint es keine Bots zu geben, die wirklich nach diesen Angaben suchen, oder sie wurden einfach nicht gefunden. Es gibt keine Hinweise darauf, dass eine Verbindung auf den Proxypot durch diese online im HTML-Code publizierten Daten zustande kam.

Der gleiche Effekt kam durch die Verbreitung des IP-Adresse durch Gästebücher, die dafür bekannt sind, oft von Bots und Harvestern gescannt zu werden zustande, nämlich keiner. Offensichtlich scheint es nicht üblich zu sein, hier Hinweise auf offene Proxies zu finden. Also mussten wir uns nach einer anderen Möglichkeit zur Verbreitung der Adresse umschauen.

3.1.5.2 Über Open-Proxy Listen Es gibt im Internet einige Listen von offenen Proxies, die gegen unterschiedliche Gebühr (ca. 30 bis 100 Dollar / Monat) abonniert werden können. Dafür erhält man sehr umfangreiche Listen

von diversen offenen Proxies, die sehr aktuell sind. Unser erstes Problem war, überhaupt in solch eine Liste aufgenommen zu werden. Wir versuchten dies mit der Liste von `http://tools.rosinstrument.com/proxy/`, da diese sehr viele Besucher hat. Wir gaben die Adresse und die relevanten Ports in das Testformular ein, und warteten ab. Leider kann man als nicht angemeldeter User nicht sehen, welches Ergebnis der Test liefert. Eine Anmeldung kam auch nicht in Frage, da die Mitgliedschaft ca 30 USD/Monat kostet. Allerdings passierte nichts, es kam kein Spam.

Der zweite Versuch war, unseren Proxy tatsächlich für die Dauer des Tests auf dem entsprechenden Port zu öffnen. Dies war zu diesem Zeitpunkt noch möglich, da wir noch keinen Traffic auf der Maschine hatten. Also bauten wir ein kleines Perl-Skript, das den entsprechenden Port öffnete (Port 8080) und alle dort eintreffenden GET-Anfragen einfach an die angeforderte Adresse weiterleitete. Wir stoppten den Proxypot und bereiteten auf der Website alles für den Test vor. Dann starteten wir das Perl-Skript und starteten den Test. Das Perl-Skript wurde nun sofort wieder beendet, es lief insgesamt keine 10 Sekunden. Offensichtlich war diese Aktion allerdings erfolgreich, denn einige Stunden später kamen die ersten GET-Anfragen auf den Rechner. Nochmals einige Stunden später kam der erste Spam, über ein HTTP RECONNECT auf ein offenes SMTP-Relay auf Port 8080 unserer Maschine an. Dies war gegen 16 Uhr Mittags am 3. Juni 2006. Zunächst tröpfelten die Mails sehr langsam ein und wir hofften, bis zum Abend die Grenze von 100 Spam-Mails überschritten zu haben. Allerdings hatten wir bis zum Abend schon ca. 7000 Mails gesammelt![17]

3.1.5.3 Über Foren Durch Zufall stießen wir auf das Forum von www.proxz.com. Hier kann man nach Anmeldung offene Proxies in ein Forum posten. Wir posteten unseren "Proxy" unauffällig in einer Liste, zwischen einigen realen Proxies. Nach kurzer Zeit hatte das Posting mehrere tausend Visits und der Spam-Durchsatz auf unserer Maschine stieg stark an. Offensichtlich werden die hier geposteten offenen Proxies, die eigentlich zur Anonymisierung dienen sollen, recht schnell missbraucht. Über unsere Maschine wurden nach diesem Posting auch einige Brute-Force-Angriffe auf Yahoo, Microsoft-Server und einige andere Dienste gefahren, was sehr interessant zu beobachten war aber nicht Gegenstand unseres Projektes ist. Es wurden auch viele GET-Requests auf Webseiten mit offensichtlich illegalem Inhalt geloggt, es ist also auf keinen Fall sinnvoll einen offenen Proxy ins Internet zu stellen, da man über die von seiner Maschine durchgeführten Aktionen verantwortlich gemacht werden kann. An dieser Stelle soll noch einmal wiederholt werden, dass keine dieser Requests nach außen weitergegeben wird, jede Anfrage wird und wurde mit einem leeren HTTP 200 OK beantwortet.

[17] Momentan liegen wir in der Größenordnung von 100.000 bis 300.000 gefangenen Spam-Mails pro Tag, allerdings ist der Proxypot noch stark restriktiert. Es gibt Proxypot-User, die von einigen Millionen gefangener Mails pro Tag berichten.

3.1.6 Probleme

3.1.6.1 Bei der Installation Bei der Installation des Proxypot gab es keine großen Probleme. Der Proxypot selber lässt sich sehr einfach installieren, man sollte nur die Dokumentation des Proxypot genau lesen und alles so machen, wie es dort beschrieben ist.

Was schon eher ein wenig nervig war, war die Installation der zahlreichen Perl-Module. Viele Module lassen sich nicht einfach über das CPAN-Modul installieren und müssen komplett von Hand heruntergeladen und installiert werden. Bei einigen Modulen musste sogar noch die make-Datei angepasst werden, was sehr viel Zeit benötigte. Insgesamt kann man für die Installation der Module einen guten Tag Arbeit einplanen, die Installation und Konfiguration dauert noch wesentlich länger.

3.1.6.2 Beim Betrieb des Proxypot Beim ersten Betrieb des Proxypot bekamen wir beim Start eine Fehlermeldung "Use of uninitialized value in subroutine entry at /usr/bin/proxypot line 1633, <GEN32> line 34". Da wir den Fehler nicht selbst beheben konnten, schrieben wir uns in die Proxypot-Mailinglist ein und fragten um Rat. Nach ein paar Stunden kam eine Mail von Alan Curry, dem Entwickler des Proxypot, mit einer Erklärung des Fehlers und einem Patch. Nachdem der Patch eingespielt war, funktionierte alles problemlos. Der Fehler kam daher, dass der Proxypot die IP-Adressen von allen in der Maschine befindlichen Netzwerkkarten erfragt. Da in unserem Fall neben den normalen devices eth0 und lo noch das virtuelle Interface sit0[18] installiert war, kam von diesem keine IP-Adresse zurück. Der Proxypot beschwerte sich daraufhin wegen einem uninitialisierten Wert. Der Patch checkt nun, ob es sich um echte oder virtuelle Interfaces handelt und fängt ab, wenn keine gültige IP-Adresse zurück kommt.

Ein weiteres Problem, das sich im Lauf der Zeit herauskristallisierte, war die anfallende Datenmenge. Bekommt man in einer Nacht 100.000 Spam-Mails, die natürlich alle komplett (inklusive Header-Daten und sonstigen Informationen) auf der Platte in ein Verzeichnis gespeichert werden, wird es schwer, mit diesem Verzeichnis noch irgendetwas anzufangen. Man kann sich den Inhalt nicht mehr anzeigen lassen, man kann keinen grep über die Dateien machen, man kann sie nicht mehr problemlos kopieren. Das einzige was Sinn macht, ist spamstat darüber laufen zulassen und die dabei erzeugten Statistiken anzuschauen.

Auch dir Größe des Logfiles bereitet Probleme, da in diesem alle ankommenden Requests geloggt werden - auch der komplette Inhalt aller ankommenden Mails. Somit kann es leicht vorkommen, dass das Logfile pro Tag ein Gigabyte größer wird. Die einzige Abhilfe ist, den Wert in der Konfigurationsdatei logrotate_seconds kleiner einzustellen.

Auch die Konfiguration des Proxypot dauerte einige Zeit, man muss die Werte in den Konfigurationsdateien hier laufend anpassen, bis man das gewünschte Verhalten des Programmes bzw. der Spammer bekommt. Da dies von der eige-

[18] Dient zur Umsetzung von IPv4 - IPv6 Adressen

nen Rechnerkonfiguration und vom Netzwerk- bzw Internetanschluss abhängig ist, kann man hier keine generellen Ratschläge geben. Hier hilft nur ausprobieren und beobachten.

3.2 Stupid Bot Trapping

Dieses Verfahren zielt darauf ab die IP - Adressen der Spammer herausfinden. Hier wird ein Verfahren angewandt, das auch außerhalb des Internets Anwendung findet. Immer wenn man seine Postadresse z.B. bei einem Gewinnspiel angibt, wird ein kleiner Fehler in die Adresse eingebaut, so dass die Post trotz des Fehlers noch ankommt. Wenn nun Post zugestellt wird, die diesen Fehler in der Anschrift hat, kann man nachvollziehen wer die Adresse unrechtmäßig weitergegeben hat.

Die gleiche Technik wird nun in abgewandelter Version im Internet verwendet. Dabei macht man sich die Vorgehensweise der Bots der Harvester zu nutze. Da die gesammelten E-Mail-Adressen der Harvester nicht auf Plausibilität überprüft werden, besteht hier die Möglichkeit dem Programm dynamisch generierte E-Mail-Adresse unterzuschieben.

Mit einem serverseitigen Skript wird bei einem Zugriff des Harvesters auf eine Webseite dynamisch eine E-Mail-Adresse mit der IP-Adresse des Rechners, auf dem das Programm läuft, und der aktuellen Zeit generiert.

```
<a href="mailto:<?php echo date('YmdHis-') . "${_SERVER['REMOTE
_ADDR']}@subdomain.website.de"?>">meine_E-Mail@website.de</a>
```

Wenn jetzt der Harvester die E-Mail Adresse <meine_E-Mail@website.de> einsammelt, nimmt er in seiner Liste die E-Mail-Adresse <20060524152426-133.72.122.73@ subdomain.website.de> auf. Der erste Teile steht für die aktuelle Zeit des Zugriffs und der zweite Teil hält die IP - Adresse des Harvesters fest. Hierfür wird zum einen die Funktion 'date' von PHP verwendet und zum anderen wird aus dem serverseitig erzeugten Array '$_SERVER' mit dem Schlüssel '['REMOTE_ADDR']' die IP - Adresse des Servers geholt. Sobald an diese E-Mail Adresse Spam oder Phishing E-Mails verschickt werden, landen diese Nachrichten alle im Catch - All E-Mail Eingang[19] von subdomain.website.de und können dann ausgewertet werden. Sobald man solche E-Mails empfangen hat, ist schnelles Handeln gefragt, da diese IP - Adressen nicht lange gültig sind und nach einem Wechsel der IP schwer nachzuverfolgen sind.

Wie man im obigen Codebeispiel schön sehen kann, wird die dynamische E-Mail Adresse ganz normal im Browser als <meine_E-Mail@website.de> dargestellt. In dieser Darstellung kann man die Adresse beliebig auf gut besuchten Webseiten unterbringen, ohne dass sie den Benutzer stören würden. Empfehlenswert ist es, die E-Mail Adresse auf Webseiten, die schon etwas länger im Internet stehen und bei Suchmaschinen indiziert sind, einzustellen, da diese regelmäßig von den Bots der Harvester besucht werden.

[19] In der Catch-All Mailbox werden alle E-Mails gespeichert, die keinem gegebenen User zugeordnet werden können

Ein weiterer Vorteil dieser Taktik besteht darin, dass die meisten Spammer die Harvester Programme von ihrer Heimat - IP - Adresse laufen lassen, so dass man direkt an die Spammer - IP gelangt und nicht etwa die IP - Adresse irgendeines offenen Proxys aus Südostasien, über den der Spammer dann seine Nachrichten versendet.

3.3 Poisoning

Von 'Poisoning' oder 'Vergiften' spricht man nur im Zusammenhang mit Harvestern (2.5.1) oder anderen Programmen, die das Internet nach E-Mail Adressen durchsuchen. Hierbei besteht das einzige Ziel diesen Programmen möglichst viele nicht existente E-Mail Adressen anzubieten. Der Harvester sammelt fleißig alle Adressen ein, ohne diese auf Plausibilität oder Richtigkeit zu überprüfen. Durch eine große Masse an solchen bewusst gefälschten E-Mail-Adressen wird versucht die Datenqualität der Harvester zu verringern. Zum Beispiel hält dann ein solches Sammelprogramm nur 60% gültige Adressen in seiner E-Mail Liste, wodurch sich der angerichtete Schaden dieser Harvester verringert.

Für solche gefälschten oder nicht existenten E-Mail-Adressen gibt es inzwischen im Internet mehrere kleine bis größere E-Mail Sammlungen (E-Mail Farmen). Diese E-Mail Farmen sind auch untereinander verlinkt, so dass ein Harvester, der solche Links verfolgt, wieder auf eine Seite geleitet wird, die nur für ihn nutzlose Adressen anbietet.

4 Eine kleine Einführung in das Protokoll SMTP

Das Protokoll SMTP wurde schon im August 1982 von Jonathan Postel im RFC 821 vorgeschlagen. Ein Update gab es 2001 im RFC 2821. Es gibt noch 16 weitere RFCs die sich mit SMTP befassen, in unsrem Fall sehr interessant ist RFC 2505: "Anti-Spam Recommendations for SMTP MTAs". Diese Einführung in das Protokoll soll sich aber hauptsächlich auf die grundlegenden Funktionen beschränken, die meisten Sonderfunktionen wurden ohnehin in so gut wie keinen Mailservern implementiert. Außerdem sind sie für unsere Anwendungen nicht interessant.

```
HTTP/1.0 200 Connection established
220 mail ESMTP
HELO test.de
250 mail
MAIL FROM:<absender@mail.de>
250 ok
RCPT TO:<empfaenger@server.de>
250 ok
DATA
354 go ahead
From: <absender@mail.de>
```

```
To: <empfaenger@server.de>
Subject: Testmail
Hallo, dies ist eine kleine Testmail
Viele Grüße,
Matthias
.
250 ok 1148661689 qp 22072
QUIT
221 mail
Verbindung zu Host verloren.
```

Die oben gezeigte Kommunikation wurde übrigens mit unsrem Proxypot geführt - ist also komplett simuliert. Es folgt nun eine Auflistung aller verwendeten Befehle mit kurzer Beschreibung:

- In Zeile 2 sieht man die "Willkommens-Meldung" des Mailservers. Darauf hin begrüßt man ihn mit der `HELO`-Meldung und der Angabe des eigenen Rechnernamens. Als Alternative kann man hier auch den Befehl `EHLO` verwenden, der Server antwortet daraufhin mit einer Liste der verfügbaren Befehle. Man sieht hier in der dritten Zeile der folgenden Ausgabe schön, dass der Server sich beschwert, da nicht der echte Rechnername bei der Anmeldung genutzt wurde. Ausgabe des Befehls `EHLO` auf einem anderen Server :

  ```
  220-smtp.poczta.onet.pl ESMTP (3)
  250 Our local time is now Fri, 26 May 2006 18:43:39 +0200
  EHLO test.de
  250-smtp.poczta.onet.pl expected "EHLO p41A1B2BA.dip.t-dialin.net"
  250-SIZE 15000000
  250-8BITMIME
  250-PIPELINING
  250-CHUNKING
  250-ENHANCEDSTATUSCODES
  250-DSN
  250-X-RCPTLIMIT 5000
  250-AUTH=LOGIN
  250-AUTH LOGIN
  250 HELP
  QUIT
  221 2.0.0 smtp.poczta.onet.pl Out
  ```

- Nachdem der Server auf die `HELO`-Meldung mit der "`250 mail`"-Meldung geantwortet hat, kann man nun den Absender der Mail angeben. Dies geschieht mit dem Befehl "`MAIL FROM:<absender@mail.de>`". Dieses Kommando wird mit der "`250 ok`"-Meldung genehmigt.

- Nun kann als nächstes der Empfänger der Mail mittels"`RCPT TO:<...>`" angegeben werden. Soll die Mail an mehrere Empfänger versandt werden,

so kann man diesen Befehl einfach mehrmals hintereinander verwenden. Der Server antwortet jedes mal mit der“`250 ok`”-Meldung.

- Hat man alle gewünschten Empfänger eingegeben, so kann man den eigentlichen Mail-Text mit dem Befehl “`DATA`” beginnen. Nach Eingabe dieses Befehls kann man ganz normal seinen Mailtext eingeben. Es sollten allerdings die drei Angaben über Sender, Empfänger und Subject der Mail hier noch einmal gemacht werden, da diese vom empfangenden Mail-Programm meist interpretiert werden. Die Mail wird als beendet angesehen wenn man in einer Zeile nur einen einzelnen Punkt eingibt. Es folgt noch eine Meldung vom Server dass die Mail verschickt wurde und der dazugehörige Code.

- Man kann die Verbindung zum Server nun mit dem Befehl ‘`QUIT`” beenden.

Die erste Stelle des zurückgegebenen Statuscodes zeigt uns schon, um welchen Typ von Meldung es sich handelt:

- 2XX: Bestätigung, alles ok

- 3XX: Aufforderung zu einer Aktion

- 4XX: Temporärer Fehler, der Server verweigert die Aktion momentan , man kann versuchen die Aktion zu einem späteren Zeitpunkt zu wiederholen.

- 5XX: Fataler Fehler, weitere Versuche bringen nichts.

5 Versandarten von Spam

5.1 Fire and forget

Dies ist nicht unbedingt eine spezielle Versandart, eher ein Prinzip, nach dem gehandelt wird. Da extrem viele Nachrichten in sehr kurzer Zeit versendet werden müssen, wird nicht abgewartet ob der Versand der Nachricht erfolgreich war. Es wird meist nicht einmal die direkte Antwort des Servers abgewartet. Sollten ein paar Adressen nicht mehr erreichbar sein oder Nachrichten auf dem Weg verloren gehen, ist dies in diesem Fall auch völlig egal.

5.2 Offene Relays

Als das Protokoll SMTP entwickelt wurde war Spam noch kein Thema. Es konnte jeder User über jeden Mailserver ohne Autorisierung seine Nachrichten versenden. Heute bewährt sich diese Vorgehensweise nicht mehr. So genannte “Offene Relays”, also Mailserver, die von jedermann ohne Autorisierung genutzt werden können, werden meist sehr schnell entdeckt und zum Versand von Spam ausgenutzt. Dies verschwendet nicht nur Ressourcen des Netzwerkes und des

Rechners, sondern kann auch dazu führen, dass das Mail-Relay für illegale Zwecke genutzt wird. Viele Mail-Relays sind nicht komplett offen, können aber oft mit wenig Aufwand trotzdem genutzt werden.

So kann man beispielsweise bei einem Mail-Relay, bei dem nur existierende User Mails versenden können (z.B. kogut.o2.pl), einfach als Absenderadresse (die an dieser Stelle nicht auf Autorisierung geprüft werden kann!) verschiedene Kombinationen ausprobieren, hier also z.B. `MAIL FROM:<daniel@o2.pl>`. Hat man eine gültige Adresse gefunden, so ist der Mailversand kein Problem mehr. Bei manchen Servern muss der User nicht einmal existieren, die Nennung der lokalen Domain reicht schon.

Sollte der Server nun für illegale Zwecke genutzt werden, so hat man neben dem rechtlichen Problem (der Serverbetreiber ist meist haftbar für die Aktionen, die von seinem Server ausgehen) auch noch das Problem, eventuell auf eine Blacklist eingetragen werden zu können. Es ist nicht einfach, von so einer Liste wieder herunter zu kommen. Fast alle großen Mailserver blockieren die Kommunikation mit Rechnern, die auf einer Blacklist stehen. In einer SMTP-Session sieht dies folgendermaßen aus:

```
c:\>telnet mail.cjb.net 25
220 ESMTP
HELO test.de
250 mail.cjb.net Hello p54A0E.dip.t-dialin.net [84.160.233.135]
, pleased to meet you
MAIL FROM:<test@test.de>
250 2.1.0 <test@test.de>... Sender ok
RCPT TO:<test@stupidbot.cjb.net>
550 5.7.1 <test@stupidbot.cjb.net>... Rejected: 84.160.233.135
 listed at combined.njabl.org
```

In diesem Fall wird die IP-Adresse 84.160.233.135 geblockt, weil sie auf einer Liste von dynamischen IP-Adressen steht.

Im Internet existieren einige Seiten mit Adressen von offenen Relays, die wirklich guten Listen kosten 20 bis 100 Dollar im Monat. Dafür bekommt man einige zehn- bis hunderttausend Adressen komfortabel aufgelistet, man kann nach jeder Art von Proxy suchen und bekommt Angaben zur Verfügbarkeit, Geschwindigkeit, Ping-Times und zu Up- und Downtimes der Proxies und Relays.

5.3 Offene Proxies

Offene Proxies sind meist nicht direkt zum Versand von Spam geeignet, da sie meist selbst keinen lokalen Mailserver installiert haben. Trotz allem sind sie sehr interessant für Spammer, da sich über einen Proxy die Herkunft bzw. die tatsächliche IP-Adresse des Spammers verbergen lässt. Der Spammer verbindet sich hierzu auf den Proxy, und verbindet sich mittels einem HTTP-Reconnect

auf den tatsächlichen SMTP-Server. Dies geht nur, wenn der Proxy keine Autorisierung verlangt, was meist an falscher Konfiguration liegt. Im Detail sieht dieser Vorgang wie folgt aus:

```
C:/> telnet 141.62.88.100 8080
CONNECT 123.123.123.123:25 HTTP/1.0
HTTP/1.0 200 Connection established
220 mail ESMTP
HELO test.de
250 mail
...
```

Nach der Verbindung auf den offenen HTTP-Proxy 141.62.88.100 auf Port 8080 muss der CONNECT... Befehl eingegeben werden. Hierbei wird entweder ein weiterer HTTP-Proxy angegeben, oder, wie hier, ein SMTP-Server. Aus Sicht des SMTP-Servers kommt die Verbindung nun vom Rechner 141.62.88.100. Um herauszufinden, von wo die Verbindung tatsächlich kam, muss man den Admin des Proxys kontaktieren, dieser kann dann in seinen Logfiles nachschauen. Nutzt man mehrere Proxys hintereinander ("Proxy Chain"), am besten noch auf unterschiedlichen Kontinenten, so wird es sehr schwer nachzuvollziehen, von wo die Verbindung ursprünglich herkam.

HTTP ist nicht das einzige Protokoll das hier genutzt werden kann, es können auch HTTP-Post-Proxys, SOCKS4, SOCKS5 oder Squid-Proxys missbraucht werden. Auf unserem Projekt-Proxypot haben wir unterschiedliche Dienste am laufen:

- Auf Port 8080 wird ein offener HTTP-Proxy simuliert
- Auf Port 3128 wird ein Squid-Proxy über HTTP simuliert
- Auf Port 1080 wird ein offener SOCKS-Proxy simuliert, die Socks-Version wird je nach ankommendem Request dynamisch simuliert
- Auf Port 4471 wird ein "genereller" offener Proxy simuliert, je nachdem was für eine Anfrage ankommt (HTTP, SOCKS...) wird dynamisch entschieden welcher Dienst auf diesem Port simuliert wird

5.4 Zombies & Botnetze

Der Spam-Versand über Zombies, also von trojanischen Pferden infizierte Rechner (meist PC's) macht einen sehr großen Anteil am gesamten Spam-Aufkommen aus. Sehr oft kommen zur Zeit Meldungen über Festnahmen von Botnetz - Betreibern, so zum Beispiel ein sehr aktueller Fall, in dem ein Koreaner über 16000 infizierte Rechner ca 18 Millionen Mails pro Tag versendet hatte[20].

Die Methode des Versands über Proxies ist sehr aggressiv, da Kosten und Verantwortung des Spams direkt von Fremden getragen werden. Die Technik

[20] http://www.heise.de/newsticker/meldung/73352

ist, genauer gesagt, eine Kombination aus trojanischem Pferd und Virus oder Wurm. Ziel ist es, auf dem infizierten Rechner einen offenen Proxy / Relay zu aktivieren. Das wohl bekannteste Beispiel sind die Viren der Sobig-Familie, es wurden alleine im Jahr 2003 geschätzte 300 Millionen infizierter Mails versendet. Es soll hier beispielhaft die Funktion des ursprünglichen W32/Sobig.A beschrieben werden:

- Die erste Stufe des Virus kommt per Mail. Sie ist relativ leicht zu identifizieren, da die Absenderadresse immer big@boss.com lautet. Der Virus ist in Visual C++ geschrieben, und mit dem Laufzeitpacker Telock gepackt, um die Erkennung und die Analyse zu erschweren. Nun sucht der Virus auf der Festplatte nach Dateien mit den Endungen .txt und .html und versendet sich selbst an alle darin gefundenen E-Mail Adressen weiter. Danach lädt sich der Virus ein Datei aus dem Internet (`http://www.geocities.com/reteras/reteral.txt`) und lädt von den darin enthaltenen URLs die zweite Stufe. Die angegebene URL ist nur verfügbar, wenn der Virus "losgelassen" wird, und zwar nur für wenige Stunden oder Tage. Sobig.A lädt nun die zweite Stufe und führt sie aus

- Zweite Stufe: Dieses Programm ist das eigentliche trojanische Pferd und ist in Delphi geschrieben und ebenfalls mit Telock gepackt. Es ist relativ schwierig für Virenscanner diese Stufe zu erkennen, da sie ständig verändert wird. Dieses Programm kann von vielen unterschiedlichen Servern geladen werden, die Versionen auf den einzelnen Servern unterscheiden sich alle voneinander. Zudem werden sie, wenn eine Angriffswelle gefahren wird, alle paar Stunden ausgetauscht durch leicht veränderte Versionen. Dieses Programm hat viele verschiedene Funktionen, unter anderem eine Benachrichtigungsfunktion über HTTP an die Adresse `http://www.banking-concern.com/cgi-bin/index7.cgi`. Zu beachten ist, dass die Adresse des CGI-Skripts mit einer 7 endet, wir betrachten also gerade die siebte Version des Virus. Die beteiligten Seiten sehen auf den ersten Blick alle relativ unauffällig aus, es ist anzunehmen dass einige von ihnen gehackt wurden. Macht man allerdings eine whois-Abfrage auf `banking-concern.com`, so bekommt man allerdings einen relativ nichts sagenden und somit in dieser Hinsicht auffälligen Eintrag (nur die interessanten Zeilen sind dargestellt):

```
Registrant:
MN Corp
Box90226188
Sioux Falls, SD 57186
United States
Registered through: GoDaddy.com, Inc. (http://www.godaddy.com)
Domain Name: BANKING-CONCERN.COM
Created on: 22-Dec-04
Expires on: 22-Dec-06
```

```
Last Updated on: 21-Feb-06

Administrative Contact:
Taylor, Seth mor543212003@yahoo.com
MN Corp
Box90226188
Sioux Falls, SD 57186
United States

Domain servers in listed order:
PARK15.SECURESERVER.NET
PARK16.SECURESERVER.NET

Registry Status: REGISTRAR-LOCK
```

Es sind bei den Kontaktdaten keine genauen Angaben über den Registrar ("MN Corp.") und über den Administrativen Kontakt gemacht. Außerdem ist die Domain in dem Status "Registrar-Lock", dies bedeutet dass die Domain nicht einfach ohne Einwilligung des Registrars umgezogen werden kann. Dies alles weist nicht auf eine Bank hin.
Nun wird das trojanische Pferd installiert. Dieses installiert nun seinerseits einen Keylogger auf dem System und einen "Remote Access"-Teil. Über die gleiche "Hide and Seek"-Methode wie bei Stufe eins wird nun Stufe drei installiert. Diese wird in %windir%\g5aa.exe installiert und beinhaltet einen spezifischen Installer für den "Wingate Proxy Server". Dies ist ein eigentlich legales Programm, was hier allerdings für illegale Zwecke eingesetzt wird.

- Stufe 3: Nun wird Wingate 5.0.2 mit einer speziell angepassten Konfiguration aus dem zuvor installierten Installer installiert und gestartet. Dieses Programm öffnet folgende Ports:
  Port 555 - RTSP Streaming Media Proxy
  Port 608 - Remote Control Service
  Port 1180 - SOCKS Proxy server
  Port 1181 - Telnet Proxy server
  Port 1182 - WWW Proxy server
  Port 1183 - FTP Proxy server
  Port 1184 - POP3 Proxy server
  Port 1185 - SMTP Server
  Über Port 608 kann das Wingate-Programm über einen Gatekeeper-Client ferngesteuert werden. Über die anderen Ports können die entsprechenden Dienste genutzt werden. Interessant in unsrem Zusammenhang sind noch Port 1180 und 1182, da der Rechner über diese als anonymer Proxy verwendet werden kann. Über Port 1185 kann direkt über den Rechner Spam versendet werden. Da sich die infizierten Rechner in Stufe 2 an den "Besitzer" des trojanischen Pferdes zurückmelden, enthält dieser eine Liste

aller infizierten Maschinen. Wie viele dies sind ist nicht bekannt, es wird eine zweistellige Millionenanzahl geschätzt. Was passiert, wenn diese Maschinen alle zu einer DoS-Attacke oder zum Versand von Spam eingesetzt werden, kann sich wohl jeder Leser selbst vorstellen.

Man sieht an diesem Beispiel, dass Zombies und Botnetze eine sehr aggressive und gefährliche Methode zum Versand von Spam sind, da sie nicht einfach zu bekämpfen oder abzustellen sind. Aktuelle trojanische Pferde mit dieser Funktion sind ausgereifte Software-Frameworks mit komfortablen und einfach zu bedienenden Schnittstellen, update- und Erweiterungsmöglichkeiten und dürfen keinesfalls unterschätzt werden. Der nächste Schritt könnten Viren sein, die nicht mehr über zentrale Stellen kommunizieren, sondern komplette peer-to-peer Funktionalität integriert haben. Spätestens dann wird es beinahe unmöglich, ein Botnetz zu bekämpfen.

5.5 Versand über online-Mail-Dienste

Diese Möglichkeit ist relativ einfach und eher für kleine Mengen von Spam möglich (unter 10000 Mails). Allerdings ist hier eine sehr geringe Entdeckungswahrscheinlichkeit. Der Spammer meldet sich bei einem Online-Mail-Dienst wie z.B. web.de, gmx oder freemail an. Selbstverständlich nutzt er zur Anmeldung gefälschte Daten. Da die gesamte Kommunikation mit dem Mail-Dienst über HTTP läuft, ist hierbei auch der Einsatz eines anonymisierenden Proxy (z.B. YAP) oder Proxy-Chains möglich. Der Spammer beginnt nun, seinen Spam zu versenden, und zwar bevorzugt nachts, da zu dieser Zeit das Konto wegen wenig Personal meist nicht manuell, sondern nur automatisiert gesperrt wird. Bis dies nun bemerkt wird, sind meist schon einige 1000 Mails gesendet. Dies geschieht fast vollständig anonym, außerdem müssen bei diesem Verfahren keine komplizierten Protokolle verwendet werden, es ist quasi von jedermann durchführbar. Das Problem ist nur, dass man nicht besonders viele Mails auf diese Weise versenden kann - ein paar tausend Mails ist so gut wie nichts. Normalerweise wird nach Millionen abgerechnet, eine Million Mails kosten ca. 20 Dollar. Des Weiteren ist dieses Verfahren relativ langsam, da hier die Infrastruktur des Mail-Dienstes genutzt werden muss und somit Fire-and-Forget nicht möglich ist.

6 Methoden zur Erkennung von Spam

In diesem Kapitel soll eine kleine Auswahl interessanter und gebräuchlicher Methoden zur Erkennung von Spam vorgestellt werden. Es ist leider nicht möglich, hier alle verwendeten Technologien zu beschreiben, dies würde den Rahmen dieser Arbeit sprengen.

6.1 Erkennung ohne Analyse des Textes

Dieses Verfahren ist entweder zur groben Vorsortierung der Mails geeignet oder zur Kombination mit anderen Erkennungsverfahren. Die einzelnen Vorgehens-

weisen sollen hier alle nur kurz erwähnt werden, da sie allesamt sehr einfach und intuitiv sind.

6.1.1 Erkennung über RBL

In der RBL (Realtime Black List) sind die IP-Adressen aller bekannten Server, über die Spam versandt wird gespeichert. Kommt nun eine Mail von solch einer Adresse beim Mailserver / User an, so wird sie abgelehnt oder weggeworfen. Die Aufnahme in diese Listen geschieht meist automatisch, wenn erkannt wird dass von einer Adresse viel Spam ausgeht oder wenn Betreiber von Blacklists bei routinemäßigen Scans Adressen finden, die Mails von nicht autorisierten Adressen annehmen und weiterleiten. Aus solch einer Liste wieder herauszukommen ist schwer bis unmöglich. Deswegen sollten die Betreiber großer Mailserver in ihrem eigenen Interesse drauf achten, sich nicht als Spam-Versender missbrauchen zu lassen. Vor einiger Zeit kam GMX auf solch eine Liste, was einige Probleme brachte. Die Meldung kann man unter `http://www.heise.de/newsticker/meldung/37138` nachlesen.

6.1.2 Analyse des Mail-Headers

Es wird z.B. analysiert, um welche Zeit die Mail versandt wurde oder ob die Mail an eine Liste (viele Einträge im BCC-Feld) versandt wurde. Aus Kombination einiger Merkmale des Headers kann festgestellt werden, ob es sich um Spam handeln könnte.

6.1.3 Äußere Form der Mail

Es gibt einige äußere Merkmale, die auf Spam schließen lassen:

- Prozentualer Anteil von Satzzeichen am gesamten Text sehr hoch oder sehr niedrig
- Multipart- oder HTML-Nachricht
- Nachricht besitzt einen Anhang
- Nachricht enthält viele Großbuchstaben, Leerzeichen oder Sonderzeichen

Natürlich kann auch hier bei Auftreten eines Merkmales nicht sofort auf eine Spam-Mail geschlossen werden, es müssen Regelsätze erstellt werden, welche der Eigenschaften typischerweise auf Spam hinweisen. Diese Regeln müssen immer up-to-date gehalten werden, da sich die Charakteristik von Spam ständig ändert.

6.2 Erkennung über Inhaltsanalyse

6.2.1 Schlagwörter / Reguläre Ausdrücke

Die Suche nach bestimmten Ausdrücken in der Mail wird auch als "Blacklist-Methode" bezeichnet. Hierbei stehen alle verdächtigen Wörter in einer Liste,

kommt eine bestimmte Anzahl dieser Wörter in der Mail vor, so wird sie als Spam aussortiert. Man kann sich vorstellen dass diese Methode nicht besonders genau ist. Auch ist die Chance relativ hoch, dass "gute" Mails als Spam fehlinterpretiert werden. Deshalb wird diese Methode meist nur in Kombination mit anderen Erkennungsmethoden genutzt. Des weiteren ist das Problem, dass die Funktion des Schutzes direkt von der Güte der Listen abhängig ist. Da die Listen manuell erstellt werden müssen, ist hiermit ein großer Aufwand verbunden. Auch die Täuschung dieser Methode ist sehr einfach, man muss das Wort nur so abändern dass es nicht mehr in der Blacklist vorkommt, z.B. indem Viagra zu Via*gra geändert wird.

Eine Erweiterung dieses Spamschutzes ist die Verwendung von regulären Ausdrücken. Diese Methode setzt auf der oben beschriebenen Suche nach verbotenen Wörtern auf, jedoch werden hier mittels regulärer Ausdrücke alle denkbaren Schreibweisen berücksichtigt.

6.2.2 Lernverfahren

Bei Lernverfahren muss der User selbst eine Klassifikation der Mails nach Spam oder Nicht-Spam vornehmen. Das System lernt nun, "gute" und "schlechte" Mails voneinander zu unterscheiden.

Bayes-Filter

Bayes-Filter, auch Bayes'sche Filter genannt, haben sich zu einem der wichtigsten Technologien im Kampf gegen Spam entwickelt. Dieser Filter kann schon für sich alleine genommen eine sehr gute Erkennungsrate vorweisen. Da es sich hierbei um eine vom User eingelernte adaptive Technologie handelt ist sie sehr schwer vom Spammer zu überlisten. Ansätze zur Überlistung des Filters gibt es bei der Verwendung von Bildern, die den Spam-Text enthalten oder von absichtlich falschen Schreibweisen der Schlagwörter.

6.2.2.1 Funktionsweise des Filters Die prinzipielle Überlegung hinter dem Bayes-Theorem ist, dass die meisten Ereignisse voneinander anhängig sind und dass die Wahrscheinlichkeit eines zukünftigen Ereignisses aus vorherigen Ereigniseintritten abgeleitet werden kann. Da viele Begriffe in erster Linie in Spam-Mails verwendet werden und nicht in gewöhnlicher Mail-Kommunikation, kann man nun aus der Häufigkeit des Erscheinens solcher Begriffe in der Mail erkennen, ob es sich um Spam handelt.

Hierzu müssen die Begriffe erst dem Filter bekannt gemacht werden. Der User erstellt eine Datenbank mit Wörtern oder Token, indem er erhaltene Mails nach gültigen Mitteilungen ("Ham") und Spam klassifiziert. In die Analyse geht auch der ausgehende Mail-Verkehr des Users sowie der Inhalt schon bekannter Spam-Mails mit ein. Diese Lernphase dauert bei normalem Mailaufkommen ca. 2 Wochen. Es wird nun für jedes Wort eine Wahrscheinlichkeit ermittelt. In diese Berechnung geht ein, wie oft das Wort in guten und in schlechten Mails vor kam.

Eine Beispielrechnung:

Das Wort "Mortgage" kommt in 1000 Mails 200 mal vor und in 300 erwünschten Mails wird es 5 mal verwendet. Die Spam-Wahrscheinlichkeit wird nun wie folgt berechnet:

$$\frac{(200/1000)}{(5/300 + 200/1000)} = 0,9524$$

Für jede Mail wird nun ein Durchschnitt über alle Wörter der Mail berechnet, somit bekommt man am Ende eine Zahl heraus. Überschreitet diese einen einstellbaren Grenzwert, so wird die Nachricht als Spam klassifiziert. In der Literatur wird dieser Filter mit einer Erkennungsrate von 99,7% angegeben.

Ein Usability-Problem dieses Filter ist jedoch, dass wenn er in Unternehmen eingesetzt werden soll, eine unternehmensweite Begriffsdatenbank erstellt werden muss. Vordefinierte Datenbanken machen hier keinen Sinn, da bei einer Bank die Wörter "Kredit" oder "Hypothek" nicht als Spam klassifiziert werden dürfen, bei einem Arzt diese Wörter wohl recht selten in der Mailkommunikation auftauchen werden. Somit muss diese Datenbank sehr sorgfältig erstellt werden, damit keine wichtigen Nachrichten (z.B. Neukundenanfragen) verloren gehen. Vorteil dieser Methode ist, dass sowohl gute als auch schlechte Indikatoren gleichermaßen in die Bewertung einfließen.

Es gibt jedoch tatsächlich Anti-Spam Lösungen, bei denen die Ham-Datei schon fertig und unveränderbar in das Programm implementiert ist, z.B. bei Microsoft Outlook und Microsoft Exchange Server. Der Vorteil davon ist, dass der User das Programm nicht einlernen braucht. Dies bringt jedoch einige Nachteile mit sich. Da die Ham-Datei nicht änderbar ist, können Spammer herausfinden, welche Begriffe wie Klassifiziert werden und dadurch auf relativ einfache Weise den Filter umgehen, indem sie diese Begriffe einfach nicht verwenden. Auch ist die Erkennungsrate ist hierbei nicht besonders hoch, da die Begriffe nicht auf das Mail-Verhalten des Users angepasst sind.

Support Vector Machines (SVM's)

Ein noch recht neues Verfahren zum Erkennen von Spam sind die SVM's. Sie gelten als relativ robust, es scheinen allerdings noch keine aussagekräftigen Testergebnisse zur Verfügung zu stehen. Prinzipiell wird bei dieser Methode jedes Dokument als Vektor abgebildet. Für jedes neue enthaltene Wort entsteht dabei ein neuer Vektor. Haben wir die beiden Sätze "Das Haus ist gelb" und "Das Auto ist grün" so bekommen wir einen 6-dimensionalen Vektor. Somit wird die Dimension des Vektors mit steigender Anzahl von Nachrichten immer höher. Als Vergleichsreferenz zur Klassifizierung gibt es eine Sammlung aus ca. 10000 Nachrichtenmeldungen, diese ergeben 9962 unterschiedliche Wörter und somit einen 9962-dimensionalen Vektor. Dies bedeutet, dass sich auch jedes Einzeldokument als 9962-dimensionaler Vektor darstellen lässt. Da diese Datenmenge sehr groß ist, werden die Vektoren nun komprimiert und normiert. Da dieses Verfahren

im Endeffekt auch "nur" Spam und nicht-Spam unterscheiden soll, wird nun in dem Raum der Trainigsdaten eine n-dimensionale Hyperebene berechnet. Diese Ebene trennt Spam und nicht-Spam voneinander. Nun kann für die beiden "Seiten" der Hyperebene ein sogennanter Support-Vektor berechnet werden. Es entstehen somit zwei Vektoren: Einer für Spam und einer für nicht-Spam. Nun können neue Nachrichten klassifiziert werden, indem die Nähe zu den beiden Support-Vektoren bestimmt wird. Mittels Schwellwerten kann nun die Mail als Spam oder nicht-Spam eingestuft werden. Auch dieses Verfahren wird immer besser, aus je mehr Nachrichten das System gelernt hat. Die Hyperebene und somit auch die Support-Vektoren werden dabei ständig neu berechnet. SVM's haben den großen Vorteil, dass sie tatsächlich mit steigender Anzahl von Nachrichten immer besser werden, den von Bayes-Filtern oder neuronalen Netzen bekannten Effekt des Übertrainierens gibt es hier nicht. Durch die vorberechneten Support-Vektoren und den somit relativ einfach Vergleich sind sie auch sehr schnell.

Ein relativ ähnliches Verfahren ist die Erkennung mittels neuronaler Netze, auf die hier aber nicht näher eingegangen werden soll.

TFIDF Algorithmen

Diese Methode stammt ursprünglich aus dem Data Mining, genauer gesagt aus der Clusteranalyse. Diese Methode gewichtet Terme in Dokumenten in Bezug auf das Dokument und die gesamte Sammlung von Dokumenten. Das Gewicht eines bestimmten Terms (also in unserem Fall ein Schlüsselwort) wird wie folgt berechnet:

$$w_{ij} = tf_{ij} \cdot \log_2 \frac{N}{df_i}$$

Hierbei sind:

- wij ist das berechnete Gewicht des Terms i im Dokument j
- tfij ist die Häufigkeit des Terms i im Dokument j
- N ist die Gesamtanzahl an Dokumenten
- dfi ist die Anzahl der Dokumente, die den Term i enthalten

Diese Formel besteht aus zwei Teilen: Der Term Frequency (TF), die die relative Häufigkeit eines Wortes i in einer Nachricht j angibt, und der Inverse Document Frequency (IDF), diese gibt die Anzahl der Dokumente an, die Wort i enthalten. Durch den Logarithmus ist diese Funktion abgeschwächt.

Es werden also häufiger vorkommende Wörter niedriger gewichtet als selten vorkommende Wörter. Dies hat den Sinn, dass Wörter ohne Unterscheidungskraft (er, sie, es, und, also, nicht usw...) nicht stark in die Bewertung eingehen. In der deutschen Sprache sind meist die 50 - 100 am häufigsten vorkommenden

Wörter ohne Bedeutung. Nach dem Zipf'schen Gesetz machen die 100 häufigsten Wörter einen Anteil von ca. 40-60% aus.

6.2.3 Erkennung mittels zentraler Datenbank

Diese Form der Spam-Erkennung ist im Kern relativ einfach. Es wird von jeder beim User eingehenden Mail ein Hash berechnet. Dieser Hash-Wert wird in einer zentralen Datenbank gespeichert, dazu für jeden Hash ein Counter der die Anzahl gleicher Werte speichert. Überschreitet diese Anzahl eine bestimmte Schwelle, so wird die dazugehörige Mail als Spam klassifiziert und beim User abgeblockt. Das Verfahren für sich alleine genommen ist relativ leicht zu überlisten. Es muss nur in jede Mail eine zufällige Zeichenkombination eingefügt werden oder jede Mail personalisiert werden, damit jedes Mal ein neuer Hash entsteht. Das es sich um semantisch gleiche Mails handelt ist nun durch dieses Verfahren nicht mehr zu erkennen. Außerdem müssen genügend Leute daran teilnehmen, dass eine ausreichend große Datenmenge zur Verfügung steht.

6.2.4 Sonstige Methoden

Es gibt hier noch unzählige weitere Methoden zur Inhaltsanalyse von Mails. Nahezu alle Erkenntnisse der KI wurden hier umgesetzt. Einige weitere, hier nicht weiter beschriebene Methoden sind:

- Entscheidungsbäume
- k-Nearest-Neighbour (kNN)
- Rule Induction
- Genetische Algorithmen

6.3 Kombinationen aus obigen Methoden

Da jede der oben beschriebenen Methoden Vor- und Nachteile hat, ist eine Kombination aus mehreren Erkennungstechnologien sinnvoll. Prinzipiell ist das Mailverhalten von User zu User verschieden, somit macht es meist wenig Sinn, mit statischen Methoden zur Spambekämpfung zu arbeiten. Der User selber sollte entscheiden können was gute und was schlechte Mails sind, das System sollte dieses Verhalten lernen können.

In größeren Unternehmen ist es sinnvoll eine 3-Stufen-Erkennung anzuwenden. Hierbei sollte an den folgenden Punkten gefiltert werden:

- Gateway: Hier erfolgt die Grobfilterung. Alles, was gemäß der Mail-Policy des Unternehmens anhand eindeutiger Merkmale als Spam identifiziert oder generell unerwünscht ist, wird hier geblockt. Das können bestimmte Dateitypen sein (zum Beispiel Audio-, Video-Anhänge, ausführbare Dateien) oder E-Mails bekannter Spammer auf Basis von Blacklists.

- Server: Hier werden die Mail mittels statistischer Verfahren anhand der Spam/Ham-Datenbank des Unternehmens gefiltert. Da die Mails hier schon relativ aufwändig geprüft werden, kann diese Stelle auch gleich genutzt werden, um die Mail an den am besten geeigneten Mitarbeiter weiterzuleiten.
- Desktop: Die ca. 30% Mails, die hier als nicht klassifizierbar ankommen werden vom Mitarbeiter selber klassifiziert. Er lernt hierbei auch einen Filter an, der dann aufgrund seines persönlichen Mailverhaltens agiert.

Je weiter vorne die Filterung, desto wichtiger ist, dass keine unternehmensrelevanten Mails verloren gehen. Eine Neukundenanfrage sollte auf keinen Fall ungesehen ausgefiltert werden. Somit ist es besser, lieber eine Spam-Mail durchzulassen als eine nicht-Spam Mail auszufiltern.

7 Umgehung von Spam-Filtern

Es werden mehrere, sich teilweise deutlich voneinander unterscheidende Methoden zur Umgehung von Spamfiltern genutzt. In diesem Kapitel werden diese Methoden nacheinander kurz vorgestellt.

7.1 Veränderung von Schlagwörtern

Einfache Spam-Filter (so genannte Blacklists) suchen in erster Linie Schlagworte wie "Viagra" oder "Mortgages". Diese Schlagwortlisten müssen irgendwann von Hand erstellt werden und sind somit nie vollständig oder fehlerfrei. Der erste Ansatz zur Umgehung dieser Filter ist, diese Schlagworte so zu verändern, dass sie für den Filter nicht mehr erkennbar sind ohne dass die Semantik für den Leser verloren geht. Aus Viagra könnte so z.B. V-I*A_G-R+A werden. Mittels guter regulärer Ausdrücke können jedoch auch solche Konstrukte erkannt werden.

Es gibt auch den umgekehrten Weg, dass einfach alle Wörter zusammenhängend zu schreiben. So wird z.B. aus "Buy most complete Pharmacy online" die Zeichenfolge "BuyMostCompletePharmacyOnline". Oft hilft in diesem Fall, solche Zeichenfolgen nach verdächtigen Wörtern zu durchsuchen oder der Versuch, anhand der meist groß geschriebenen Anfangsbuchstaben den Originaltext wiederherzustellen. Werden die Anfangsbuchstaben nicht groß geschrieben, so macht auch die Nachricht normalerweise keinen Sinn mehr.

Auch die Verwendung von ausländischen Buchstaben beziehungsweise Buchstaben mit ausländischen Akzenten ist ein verbreiteter Weg, um einfache Blacklists zu überlisten. Der Unterschied zwischen Viagra, Vìagra und Víagra fällt dem Leser der E-Mail vermutlich nicht einmal auf. In diesem Zusammenhang ist auch Unicode zu einem Problem geworden.

7.2 Tarnung durch HTML-Mails

Durch die Möglichkeit in HTML Sonderzeichen mittels &# und ; zu maskieren, können nicht in der Mail offensichtlich erwünschte Wörter einfach "unsichtbar" gemacht werden. Viagra würde hier nun zu V i a g r a umgewandelt werden. Auch dies ist vom Spamfilter jedoch noch relativ einfach zu entdecken. Wesentlicher schwerer zu entdecken sind durch die Formatierungsmöglichkeiten von HTML versteckte Nachrichten. Auch hier gibt es mehrere Möglichkeiten. Eine sehr einfach Möglichkeit, die allerdings von den meisten Spamfiltern nicht erkannt wird ist die Tarnung des eigentlichen Textes durch dazwischenliegende Kommentare. So könnte man das Wort MONEY durch M<!–Hund–>O<!–Katze–>N<!–Maus–>E<!–Baum–>Y<!–Auto–> tarnen. Wird dieser Text von einem HTML-fähigen Mail-Programm gelesen, so wird er tatsächlich als MONEY ausgegeben. Da diese Methode sich zu einer der am weitesten verbreiteten Methode entwickelt hat, stufen gute Spamfilter schon das alleinige Vorhandensein von HTML-Kommentaren in Mails als verdächtig ein. Die Weiterentwicklung dieser Methode ist die Nutzung von so genannten Random-Tags. Random-Tags sind aus zufällig generierten Buchstabenfolgen erzeugte HTML-Tags. Da diese Tags nicht gültig sind, werden sie vom Mail-Programm bei der Anzeige einfach ignoriert. Diese Tags werden zufällig in den Text gestreut und verschleiern nun den tatsächlichen Wortlaut für den Spamfilter. BUY könnte folgendermaßen getarnt werden:

B</ladshf>U</rajlisdf>Y</dsfjkl>.

Bei HTML-Mails kann die oben beschriebene Methode zum Einfügen von Leerzeichen noch verfeinert werden, indem den Leerzeichen die Schriftgröße 0 zugewiesen wird. Beispiel:

```
V<font size=0> </font>i
<font size=0> </font>a
<font size=0> </font>g
<font size=0> </font>r
<font size=0> </font>a
```

Funktioniert diese Methode nicht mehr, so wird auf eine andere Schriftgröße ausgewichen, funktioniert dies auch nicht mehr so werden andere Füllzeichen verwendet. Die Methode, andere Buchstaben in kleinen, kaum sichtbaren Schriftgrößen (auch "Microdot" genannt) einzufügen, hat zudem noch den Vorteil dass dieses Wort "verschleiert" wird. Aus Viagra wird durch Vziagra in der Darstellung folgendes:

V iagra

Der Spamfilter erkennt hier (im besten Fall) allerdings nur Vziagra, ein für ihn harmloses Wort.

Eine weitere Möglichkeit zur Täuschung von auf statistischen Verfahren basierenden Spamfiltern ist es, unsichtbare "harmlose" Wörter einzufügen. Selbst wenn der Filter nun ein "verbotenes" Wort findet, kann es nun sein dass die Mail aufgrund der geringen prozentualen Anteile verbotener Wörter nicht als Spam eingestuft wird.

Eine hierbei sehr verbreitete Methode ist es, einfach einen zufälligen (harmlosen) Text in der gleichen Farbe wie der Hintergrund einzufügen. Geschieht dies noch in einer kleinen Schriftgröße, so fällt es dem Empfänger der Mail eventuell nicht einmal auf. Selbst wenn nun ein intelligenter Filter erkennt, dass der Text die gleiche Farbe wie der Hintergrund hat und somit für den Leser unsichtbar ist, kann dieser recht einfach getäuscht werden, indem nicht die gleiche Farbe, sondern nur eine ähnliche Farbe benutzt wird. So könnte z.B. der Hintergrund mittels <body bgcolor=#223340> eingefärbt werden, der versteckte Text könnte die Farbe <font color=#233239> erhalten. Für den Leser ist der Text nicht erkennbar, der Filter ist jedoch überlistet. Gute Filter können auch dies erkennen, da sie die Nähe von Farben berechnen können. Man nennt diese Methode auch "Camouflage".

Eine weitere Methode die auf dem Einfügen harmloser Textblöcke basiert ist das Einfügen von Ausschnitten aus online-Magazinen oder -Zeitungen. Damit der Empfänger der Nachricht nicht durch den Text abgelenkt wird, wird dieser einfach in Spitzklammern gesetzt. Die HTML-Engine des Mailclients erkennt diesen Text nun nicht mehr als Test, sondern als sinnlosen HTML-Befehl und rendert ihn nicht auf den Bildschirm. Es gibt noch unzählige weitere Möglichkeiten zum Verstecken von Text mittels HTML-Tags[21], auf diese soll hier aber nicht näher eingegangen werden.

Der "MIME-Trick" ist eine ältere Methode um Spam zu verstecken. Nutzt man einen HTML-fähigen Mailreader, so zeigt dieser bei eingehenden Mails die aus einem HTML-Teil und einem plain-Text-Teil bestehen per default die HTML-Nachricht an.[22]Da der Spamfilter jedoch die Mail als Ganzes analysiert kann der Spam im HTML-Teil versteckt werden, im plain-Text-Teil wird harmloser Text eingefügt. Dieser "Trick" ist der Grund, warum User von reinen (oder per default darauf eingestellten) plain-Text Mailclients oft vermeintlich sinnlose Mails ohne jeden Werbebezug bekommen.

7.2.1 Tarnung durch HTML-Tabellen

Durch geschickt angeordnete HTML-Tabellen können Spamfilter, die HTML auswerten können, komplett getäuscht werden. Die Vorgehensweise bei dieser Methode ist folgendermaßen:

Der zu verschleiernde Text wird in Zeilen untereinander geschrieben, beispielsweise:

BUY
MORE
VIAGRA

[21]Zum Beispiel mit dem <title> oder dem <marquee> Tag
[22]Zumindest bei Microsoft-Produkten

Im zweiten Schritt wird der Text in eine HTML-Tabelle eingepasst, die Tabelle besitzt eine Breite von mindestens einem Zeichen und maximal kürzer als das kürzeste im Text vorkommende Wort. Wir nehmen in diesem Beispiel eine Breite von einem Zeichen an.Die Höhe der Tabelle muss mindestens zwei Textzeilen betragen, wir nehmen eine Höhe von drei Textzeilen an. Heraus kommt nun folgende Tabelle:

B M V	U O I	Y R A	E G	R	A

Ein HTML-fähiger Spamfilter erkennt in diesem Fall beim Auswerten der Tabelle nur die (erlaubte) Zeichenfolge BMV UOI YRA EG R A. Durch die unterschiedlichen Formatierungsmöglichkeiten der Tabelle ist es sehr schwer, ein System zu entwickeln, das den in der Tabelle stehenden Text zuverlässig erkennt. Deswegen werden bei aktuellen Filtern alle E-Mails, die Tabellen mit vielen Zellen enthalten, als verdächtig eingestuft.

7.2.2 Tarnung durch Bilder

Wohl jeder kennt die vermeintlich von Banken ausgehenden Phishing-Mails, die zum Besuch einer bestimmten URL und zur Eingabe von UID, Passwort und TAN (!) auffordern. Wer diese Mails schon einmal genauer angeschaut hat wird bemerkt haben, dass sie meist aus einem einzigen Bild bestehen. Nicht nur der Text in der Mail, auch der meist darin enthaltene Link ist ein Teil dieses Bildes. Diesen im Bild enthaltenen Text zu erkennen und anschließend zu analysieren ist für einen Spamfilter nahezu unmöglich (verrauschtes Bild, leicht gedrehter Text, unterschiedliche Schriftarten...). Auch die in dem Bild enthaltene URL ist natürlich nicht die tatsächliche Ziel-URL. Oft erkennt man die tatsächliche URL wenn man mit der Maus über das Bild fährt in der Statuszeile des Mailclients. Es gibt allerdings auch Möglichkeiten, die in der Statuszeile angezeigt URL mit JavaScript auszutauschen, also kann man sich auch hier nicht wirklich drauf verlassen. Egal wo man auf das Bild klickt, man gelangt immer zur Ziel-URL. Dies zeigt, dass auf diese sehr einfache Weise nahezu alle Möglichkeiten zur Täuschung offen stehen. Realisiert werden diese Mails auch wieder durch die Verwendung von Multipart-Messages mit unterschiedlichen Mime-Typen. Im Text-Teil steht harmloser Text und die Einbindung des Phishing-Bildes als Hintergrund. Im image/gif-Teil sind die Daten des Bildes untergebracht.

7.2.3 Verschleiern verdächtiger URLs

Besonders im Bereich der Phishing-Mails ist es üblich, die URLs die das "Opfer" der Mail besuchen soll zu verschleiern. Niemand wird freiwillig auf www.hackme.ru seine Kontodaten eingeben. Es gibt allerdings einige Methoden, seine tatsächliche Adresse zu verschleiern. Fast alle beruhen auf der Möglichkeit, dass man vor der URL einen Benutzernamen angeben kann. Dies wird dann folgendermaßen

dargestellt: user@host.de. Das @-Zeichen trennt Benutzername und Hostadresse voneinander. Da hier allerdings auch noch die tatsächliche Zieladresse im Klartext sichtbar ist, muss diese noch versteckt werden. Hierzu gibt es verschiedene Arten, die Adresse dem Browser zu übergeben. Als Beispiel soll hier `http://www.heise.de` verwendet werden.

1. Zunächst kann man (z.B. über einen Ping) die IP-Adresse der Seite ausfindig machen und diese verwenden. Ein ungeübter User wird `http://www.bank.de@193.99.144.85` noch eher akzeptieren.

2. Man kann nun eine Umrechnung der IP-Adresse in eine Dezimalzahl vornehmen. Dabei werden zunächst die 4 Blöcke der IP-Adresse in Binärzahlen umgerechnet:
   193.99.144.85 wird nun zu
   11000001.01100011.10010000.01010101. Entfernen wir die Punkte so bekommen wir die Binarzahl
   11000001011000111001000001010101.

3. Wenn wir diese Zahl nun in eine Dezimalzahl umrechnen bekommen wir 3244527701. Geben wir nun diese Zahl in den Browser ein, so kommen wir auf www.heise.de. Nun können wir wieder einen "Benutzernamen" hinzufügen: `http://www.bank.de@3244527701` Diese Adresse ist schon wesentlich unauffälliger. Wir können die Adresse natürlich auch in andere Zahlensysteme umrechnen, z.B. oktal (es muss jedem Element eine 0 vorangestellt werden) oder hexadezimal (es muss jedem Element ein 0x vorangestellt werden)

4. Bei Internet Explorer kann man nun sogar noch einen Schritt weiter gehen. Setzt man als letztes Zeichen des Benutzernamens (vor dem @) das Steuerzeichen 0x01, so zeigt der Explorer unter bestimmten Umständen alles was nach diesem Zeichen kommt nicht mehr in der Statuszeile des Browsers an[23]. Mittels eines einfachen Scripts kann dieser Exploit ausgeführt werden:
   <button onclick="location.href=unescape('http://www.microsoft.de%01@www.heise.de');"> Test </button>
   Hierbei braucht man sich nun nicht einmal mehr die Mühe machen die URL zu verschleiern, da sie eh nicht angezeigt wird. Abhilfe kann hier geschaffen werden indem die Adresse von Hand in den Browser eingegeben wird.[24]

7.3 Täuschen von Spamerkennung mittels zentraler Datenbank

Wie zuvor schon beschrieben funktioniert die Erkennung von Spam mittels zentraler Datenbanken, indem bei dem User ein Hash über jede eingehende Mail

[23] Funktionierte im Test mit MS IE 5.x und 6.x
[24] `http://support.microsoft.com/default.aspx?scid=kb;de;834489`

gebildet wird. Dieser Hash wird online in eine zentrale Datenbank gespeichert. Kommen in kurzer Zeit eine grosse Menge an gleichen Hash-Werten in dieser Datenbank an, so kann man sich relativ sicher sein, dass es sich hierbei um Spam handelt. Das Gute an dieser Methode ist, dass sie eine relativ einfache Erkennung von Spam ermöglicht, ohne dass auf dem Client schwergewichtige Programme installiert sein müssen. Jedoch ist diese Methode relativ einfach zu täuschen. Es muss nur in jede Mail eine pseudozufällige Zeichenkette eingefügt werden. Durch den hiermit komplett veränderten Hash der Mail ist sie nicht mehr zentral als Spam zu erkennen.

Da es momentan (meiner Kenntnis nach) nur einen Anbieter gibt[25], der diesen Ansatz (im Kern) zur Erkennung von Spam verfolgt, ist das Einfügen zufälliger Zeichenkette in Mails noch nicht sehr verbreitet.

Fazit dieses Kapitels

Fazit ist, dass aktuelle Spamfilter sehr komplexe Programme sein müssen. Sie müssen eine komplette HTML-Engine beinhalten, und nicht nur die Syntax sondern auch teilweise die Semantik der HTML-Dokumente verstehen können. Durch die sich ständig verändernden Tricks der Spammer ist es wichtig, dass die Spamfilter über ein automatisches und regelmäßiges Update verfügen. Es zeigt auch, wie kreativ Spammer sind, und wie schwer ist es, einen vernünftig funktionierenden Spamfilter zu erstellen. Da der Grossteil der hier beschriebenen Tricks mit HTML zusammenhängt ist es besser, HTML ganz im Mailclient abzuschalten oder es zumindest nicht zur primären Darstellung zu nutzen.

8 Beispielhafte Auswertung einer durch den Proxypot identifizierten IP-Adresse

Es soll nun noch kurz auf beispielhafte Mechanismen eingegangen werden, die zur Identifizierung des Spammers und seiner Werkzeuge genutzt werden können. Selbstverständlich gibt es hier eine unüberschaubare Anzahl an Möglichkeiten zur Identifizierung der Spammer, es werden hier nur ein paar gebräuchliche Dienste vorgestellt.

8.1 ping

Der erste Schritt ist es, festzustellen, ob die Maschine überhaupt noch am Netz ist. Es können nur "frische" IP-Adressen verwendet werden, da sie nach einigen Stunden meist schon wieder offline sind.
Wir wollen nun im folgenden die Adresse 168.95.5.208 etwas genauer anschauen. Von dieser Adresse wurden einige Spam-Mails über unseren Proxypot versendet.

```
C:\>ping 168.95.5.208
Ping wird ausgeführt für 168.95.5.208 mit 32 Bytes Daten:
```

[25] SpamStopsHere `http://www.spamstopshere.de/`

```
Antwort von 168.95.5.208: Bytes=32 Zeit=354ms TTL=240
Antwort von 168.95.5.208: Bytes=32 Zeit=388ms TTL=240
[...]
```

Wir sehen, dass der Rechner hinter der Adresse noch online ist. Anhand der konstant hohen ping-Zeiten kann man vermuten, dass der Rechner weit entfernt liegt.

8.2 Blacklists

Wir können nun schauen, ob sich die IP-Adresse auf Blacklists befindet. Eine Nachfrage auf dnsstuff.com ergab, dass die IP tatsächlich schon in einigen Blacklists gelistet ist, z.B. unter`http://bl.csma.biz/cgi-bin/listing.cgi?ip=168.95.5.208`.

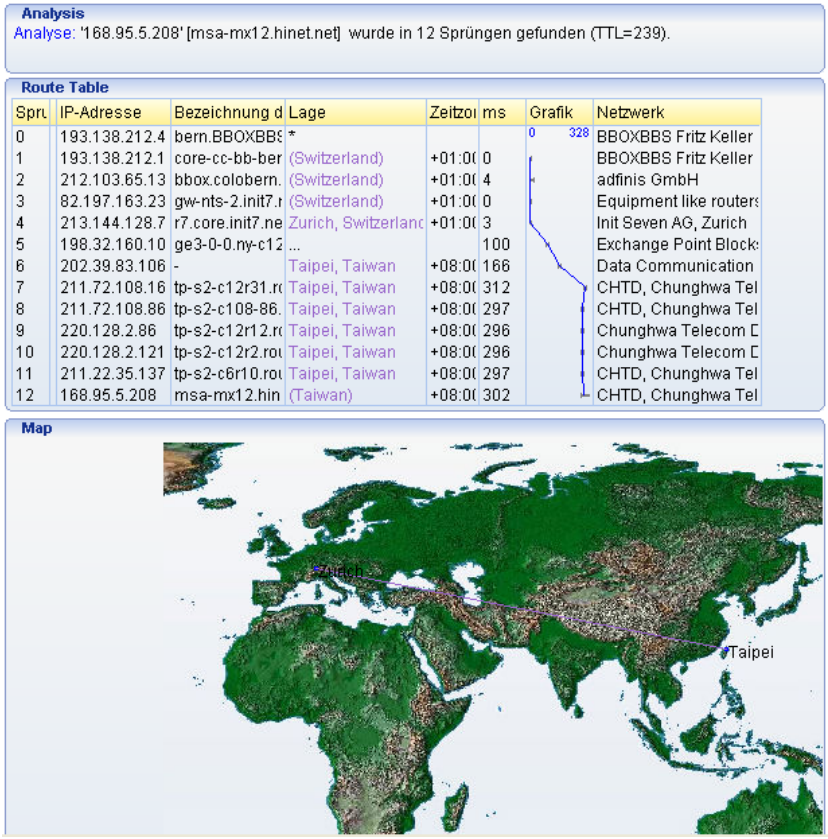
8.3 traceroute

Das "klassische" Traceroute liefert hier folgende Ausgabe:

```
1 2 ms 2 ms 2 ms XXX
2 * * * Zeitüberschreitung der Anforderung.
3 53 ms 54 ms 52 ms XXX
4 139 ms 141 ms 165 ms 62.156.131.150
5 142 ms 142 ms 142 ms sl-gw31-nyc-12-0.sprintlink.net [144.223.27.133]
6 138 ms 136 ms 137 ms sl-bb27-nyc-12-0.sprintlink.net [144.232.13.35]
7 155 ms 156 ms 156 ms sl-bb27-rly-0-0.sprintlink.net [144.232.20.165]
8 160 ms 162 ms 159 ms sl-bb22-rly-10-0.sprintlink.net [144.232.14.177]
9 224 ms 224 ms 222 ms sl-bb22-sj-10-0.sprintlink.net [144.232.20.186]
10 224 ms 224 ms 257 ms sl-st21-pa-12-0-0.sprintlink.net [144.232.8.241]
11 223 ms 225 ms 224 ms sl-chti-9-0.sprintlink.net [144.223.243.74]
12 222 ms 222 ms 222 ms pa-c12r11.USA-PAIX.router.hinet.net [202.39.83.189]
13 352 ms 2282 ms 357 ms tp-s2-c12r31.router.hinet.net [211.72.108.162]
14 353 ms 356 ms 351 ms tp-s2-c108-82.router.hinet.net [211.72.108.82]
15 350 ms 353 ms 351 ms tp-s2-c12r12.router.hinet.net [220.128.2.98]
16 353 ms 351 ms 352 ms tp-s2-c12r2.router.hinet.net [220.128.2.121]
17 678 ms 351 ms 351 ms tp-s2-c6r10.router.hinet.net [211.22.35.161]
18 354 ms 352 ms 351 ms msa-mx12.hinet.net [168.95.5.208]
```

Das Paket wird direkt nach Verlassen unseres Netzes von sprintlink zu hinet weitergegeben.

Hier noch die Ausgabe von VisualRoute. Der Scan wurde hier von einer Schweizer Website aus gemacht, von daher startet die Verfolgung in der Abbildung in Zürich.



Nun erkennt man schön, dass es sich hier um einen taiwanesischen Spammer handelt.

8.4 whois

Eine whois-Abfrage ist in diesem Fall nicht wirklich sinnvoll, da es sich im Beispiel um eine Subdomain handelt. Man bekommt somit nur die Daten der übergeordneten Domain heraus, die allerdings nicht unbedingt etwas mit den Daten der kompletten Domain zu tun haben müssen.

8.5 reverse dns

Da eine normale DNS-Anfrage zu einem Domainname die IP-Adresse findet, findet eine reverse DNS-Anfrage zu einer IP-Adresse den Domainname. In unserem Fall wurde der Dienst von `http://www.dnsstuff.com/` genutzt.

Der aufgelöste Domainname lautet im Beispiel folgendermassen:

```
168.95.5.208 PTR record: msa-mx12.hinet.net
```

Es handelt sich also um eine Subdomain von hinet.net. Dieser taiwanesische Provider ist bekannt für diverse zwiespältige Aktivitäten und wurde schon oft

in einschlägegen Foren diskutiert. Das "mx" im Namen deutet auf einen "Mail Exchange" hin. Um darüber Gewissheit zu bekommen, führen wir nun einen Portscan durch.

8.6 nmap Scan

Nmap lieferte folgende Ausgabe:

```
C:\>nmap -v -A -p1-1024 168.95.5.208
[...]
Host msa-mx12.hinet.net (168.95.5.208) appears to be up ... good.
Interesting ports on msa-mx12.hinet.net (168.95.5.208):
Not shown: 1011 closed ports
PORT STATE SERVICE VERSION
22/tcp filtered ssh
25/tcp open smtp Sendmail 8.8.8/8.8.8
62/tcp filtered acas
111/tcp filtered rpcbind
123/tcp filtered ntp
139/tcp filtered netbios-ssn
204/tcp filtered at-echo
330/tcp filtered unknown
391/tcp filtered synotics-relay
650/tcp filtered unknown
797/tcp filtered unknown
834/tcp filtered unknown
872/tcp filtered unknown
Device type: general purpose
Running: Sun Solaris 10|9
[...]
```

Es wurden hier die Ports 1 bis 1024 mit einem SYN Stealth Scan gescannt.

Sehr interessant ist, dass Port 25, wie vermutet, tatsächlich offen ist. Hier läuft ein Sendmail, Version 8.8.8. Es ist bekannt, dass diese Version einige Sicherheitslücken hat (z.B. die "Double Pipe Access Validation Vulnerability"). Es sind noch einige andere Ports gefiltert, diese scheinen in erster Linie nicht interessant zu sein. Möchte man einen Angriff auf die Maschine durchführen, so könnten sie allerdings sehr interessant werden.

8.7 nessus

Mit Hilfe des Security-Scanners nessus kann das vom Spammer genutzte System auf Sicherheitslücken untersucht werden. Über diese Lücken könnte ein eventueller Angriff auf dieses System vorbereitet werden und die vom Spammer genutzte Maschine kompromitiert werden. Dadurch könnte die Identität des Spammers vollständig offengelegt werden.

Da sich sowohl unsere ISPs als auch die Systemadministratoren der HdM von Nessus-Scans auf andere Maschinen verständlicherweise nicht begeistert zeigten, soll auf ein Beispiel verzichtet werden.

9 Ende

Als Abschluss möchten wir Prof. Kriha, Prof. Goik, Herrn van der Kamp, Herrn Czech und unseren Kommilitonen, im Speziellen Sebastian Roth und Andreas Springer für ihre tatkräftiger Unterstützung danken.

Es war ein sehr interessantes Projekt, wir haben viel gelernt, nicht nur über Spam. Ausserdem ist uns klar geworden, welche Dimensionen die Spam-Problematik inzwischen angenommen hat, und dass man schleunigst beginnen sollte, etwas dagegen zu unternehmen. Wir hoffen, diese Arbeit trägt ihren Teil dazu bei.
Im Anhang dieses Dokuments finden sie einen kleinen Auszug aus der erstellten Statistik, die aufgezeigt wieviele Spam Nachrichten über den Proxy 'verschickt' wurden, welche Größe die Nachrichten haben und an wieviele Empfänger der Spam gegangen wäre. Des Weiteren werden detaillierte Informationen zu den einzelnen Versender - IP -Adressen und den Empfängern geliefert.

Anhang

Auszug aus der SpamStat-Statistik

Evidence of open proxy spam

Intro . . . Summary . . . net . . . host . . . web . . . mail

Intro

This report contains evidence of email spam that has been sent using the illegal method known as proxy hijacking. If you need help understanding this report, see the proxypot information page.

Evidence summary

Message count

3170284 messages sent

Time frame

first at Sat Jun 3 14:46:00 2006 (1149338760.18729_1.spam)
  Hi !

last at Wed Jun 28 00:38:41 2006 (1151447921.13997_1.spam)
  ¡´¡´¥xÆQ¥Í§Þ²£«~¥þ®M¥un¢±¢°¢µ¢¯¢¯¤¸¡I¡I¤S¥i¥ H¸gÀç¤@¥÷¨Æ ~³á¡IOrtiz

Message sizes

smallest is 41 bytes (1149345931.19928_1.spam)
  no subject

biggest is 78442 bytes (1150534982.32714_1.spam)
  Travel Special! - TARKAN CONCERT - 07 AUG 2006

average message size 4353 bytes

Recipients per message

fewest recipients 1 (1150073378.8601_1.spam and 976410 others)
  =?windows-1251?B?0uXw7Ojt4Ov7IO/w6Ljs4CDP6+Dy5ebl6Q==?=

most recipients 60 (1150744601.26648_1.spam and 8438 others)
  ¥þ¥@¬É³Ì¨üª§Ä³ªº¤Q¤¡¸T¤ù ¤@¦¸¸¦¬¶°

average 6 recipients

Report totals

total 13800970445 bytes to 18900138 recipients

Breakdown by /24

/24	Messages sent	Bytes	Recipients	Details
66.185.126/24	445522	4518802331	2502715	Details
65.19.154/24	313210	285057116	1197487	Details
220.140.225/24	213850	1456569481	1196867	Details
89.208.6/24	146556	137477603	162838	Details
72.36.222/24	128288	280038155	1312202	Details
81.176.78/24	114510	1215438279	123931	Details
208.66.194/24	98117	91474419	373662	Details
220.131.171/24	85544	170781414	329021	Details
220.140.235/24	80522	147535702	434492	Details
220.130.54/24	79773	184822381	727160	Details
220.130.42/24	73059	236784468	569603	Details
213.219.205/24	64030	57567049	64030	Details
87.240.15/24	59837	122047064	63896	Details
217.129.81/24	58674	51225029	58674	Details
61.62.163/24	57084	311619563	596746	Details
89.208.7/24	52527	48169454	52527	Details
217.172.29/24	51865	246648880	62609	Details
201.252.74/24	48701	246686196	344932	Details
59.114.249/24	42640	70702353	208211	Details
200.69.237/24	39452	1423697589	207399	Details
210.245.197/24	37732	262271194	241086	Details
202.65.111/24	35903	153301987	143282	Details
59.61.109/24	34635	133461676	168517	Details
220.140.233/24	32863	59261046	189999	Details
220.131.139/24	31046	62668566	31046	Details
218.170.154/24	30659	132784041	308730	Details
…	…	…	…	
220.139.8/24	1	1282	1	Details
220.139.9/24	1	440	1	Details
222.78.93/24	1	746	3	Details

3170284 messages from 534 distinct /24s

Breakdown by host

Host	Messages sent	Bytes	Recipients	Details
220.140.225.230	200989	1433415324	1124694	Details
89.208.6.183	146556	137477603	162838	Details
72.36.222.50	128288	280038155	1312202	Details
81.176.78.131	112104	1146716408	121525	Details
66.185.126.4	103816	1014624553	584348	Details
65.19.154.70	101475	92286440	388232	Details
208.66.194.21	97643	88892177	373188	Details
65.19.154.91	92213	83986753	352784	Details
220.140.235.48	80522	147535702	434492	Details
220.130.54.49	67091	154695179	639927	Details
220.131.171.76	65516	136906974	232422	Details
65.19.154.68	62948	57253838	240387	Details
66.185.126.76	61435	703766980	350097	Details
87.240.15.9	59837	122047064	63896	Details
217.129.81.170	58674	51225029	58674	Details
213.219.205.87	57138	51296482	57138	Details
61.62.163.10	57084	311619563	596746	Details
65.19.154.92	56574	51530085	216084	Details
89.208.7.21	52527	48169454	52527	Details
217.172.29.17	51865	246648880	62609	Details
201.252.74.184	48701	246686196	344932	Details
66.185.126.83	47864	360317674	260284	Details
66.185.126.115	40005	383908015	223208	Details
200.69.237.104	39452	1423697589	207399	Details
220.130.42.143	39168	127179635	310177	Details
66.185.126.84	39103	410670787	213346	Details
210.245.197.11	37732	262271194	241086	Details
66.185.126.75	36812	267810858	200310	Details
59.61.109.243	34635	133461676	168517	Details
59.114.249.50	34588	53636922	175055	Details
220.130.42.144	33891	109604833	259426	Details
…	…	…	…	
222.64.111.43	1	664	1	Details
222.78.93.64	1	746	3	Details

222.122.60.243	1	14803	4	Details

3170284 messages from 1051 distinct hosts

Intro . . . Summary . . . net . . . host . . . web . . . mail

Breakdown by web site

Web site	Referencing messages	Bytes	Recipients	Details
a-vbs.com	220374	401845471	1710023	Details
hinet.net	189430	688549426	1799913	Details
yuyaotopao.net	131176	386265546	1110076	Details
cgiworld.net	128538	567279513	1325759	Details
algedistar.ru	81535	170149107	746658	Details
world-medical.ru	73945	69551663	82199	Details
lake-music.ru	73349	68613106	81377	Details
8k.com	54475	301366736	376239	Details
vcd-104.com	47500	105525709	452753	Details
uni.cc	43334	66756344	214527	Details
bpath.com	43266	186832765	453688	Details
uuu.to	43265	186828391	453678	Details
mobu.ru	38054	93826640	510746	Details
adult-movie-hosing.com	37732	262271194	241086	Details
…	…	…	…	
wertretwg.info	1	467	1	Details
y4k5i4iu6uyy.info	1	477	1	Details

1891974 references to 335 distinct web sites

Intro . . . Summary . . . net . . . host . . . web . . . mail

Breakdown by mail drop

Mail drop	Referencing messages	Bytes	Recipients	Details
yam.com	111182	525321625	1129290	Details
algedistar.ru	81535	170149107	746658	Details
vrsystem9@yahoo.com.ar	48701	246686196	344932	Details

processfiles.com	25334	78176849	25334	Details
claimprocessingfiles.com	21668	66379056	21668	Details
plasa.com	4988	7485378	9966	Details
yahoo.com.hk	2217	21045509	6977	Details
contemporaneocampobelo@yahoo.com.br	2115	11163366	11718	Details
o2.pl	1140	2244127	1140	Details
hinet.net	786	25507582	8402	Details
yahoo.com.tw	588	9087414	5735	Details
msk.ru	282	6543120	14079	Details
usa.com	279	373553	279	Details
hkdomestichelpers.com	215	2351673	849	Details
aspire.com.tr	98	7683431	103	Details
aspirenews.com	98	7683431	103	Details
mail.ru	73	1017798	73	Details
yahoocom.hk	11	191958	40	Details

301310 references to 18 distinct mail drops

Im Folgenden wird der versandte Spam von der IP-Adresse 66.185.126.4 analysiert und detaillierter beschrieben.

Evidence of open proxy spam from 66.185.126.4

Intro . . . Summary . . . main

Intro

This report contains evidence of email spam that has been sent using the illegal method known as proxy hijacking. If you need help understanding this report, see the proxypot information page.

This page of the report only contains messages that were sent from 66.185.126.4. To see *all* the available evidence in this report, go to the main page.

Evidence summary

Message count

103816 messages sent

Time frame

```
  first at Thu Jun 22 02:45:47 2006 (1150937147.27957_1.spam)
    Get lyour home payment lowelred
  last at Sat Jun 24 13:03:47 2006 (1151147027.17547_11.spam)
    Yolu canl lower home paymenlt by 30 percent
```

Message sizes

```
  smallest is 2485 bytes (1151145705.14742_8.spam)
    Your Rlle-.fi apprloval
  biggest is 19674 bytes (1150939465.30067_3.spam)
    Home lImprovement Approval
  average message size 9773 bytes
```

Recipients per message

```
  fewest recipients 1 (1150938880.29559_8.spam and 6775
others)
    Youlr approlval code
  most recipients 8 (1150939445.30049_5.spam and 21677
others)
    Tired of Hligh lHome Payments
  average 6 recipients
```

Sender totals

total 1014624553 bytes to 584348 recipients

Intro . . . Summary . . . main

Wenn man nun auf den Link klickt, bekommt man den Quelltext von der Spam – Mail angezeigt:

```
Return-Path: <ppohejyvgdgbl@direcway.com>
Delivered-To: <ahtong17@aol.com>
Delivered-To: <aiess300@aol.com>
Delivered-To: <ahsing@aol.com>
Delivered-To: <ahrc4640h@aol.com>
Delivered-To: <ahunt40607@aol.com>
Delivered-To: <aimeeree@aol.com>
Delivered-To: <aimmike98@aol.com>
Received: from direcway.com ([66.185.126.4]) by spam
([141.62.88.100])
          with ESMTP via SOCKS4 (1080) id
"483372,1151146987,11"
          (attempted proxy to 205.188.157.217:25);
```

```
          Sat, 24 Jun 2006 13:03:47 +0200
Received: from unknown (19.127.48.58)
       by mtu67.syds.piswix.net with SMTP; Sat, 24 Jun 2006
03:48:04 -0800
Received: from unknown (78.206.196.198)
       by mail.naihautsui.co.kr with SMTP; Sat, 24 Jun 2006
03:39:40 -0800
Received: from [23.53.154.89] by mx.reskind.net with ASMTP;
Sat, 24 Jun 2006 03:35:50 -0800
Message-ID: <419F817D.635FE68@direcway.com>
Date: Sat, 24 Jun 2006 03:19:11 -0800
From: "Elisa Dumas" <ppohejyvgdgbl@direcway.com>
X-Accept-Language: en-us
MIME-Version: 1.0
To: "Dion Herring" <ahtong17@aol.com>,
       "Dante Noble" <aiess300@aol.com>,
       "Clement Foley" <ahsing@aol.com>,
       "Coy Bartlett" <ahrc4640h@aol.com>,
       "Odell Mercado" <ahunt40607@aol.com>,
       "Maxwell Landry" <aimeeree@aol.com>,
       "Jarvis Durham" <aimmike98@aol.com>
Subject: Yolu canl lower home paymenlt by 30 percent
Content-Type: text/plain;
       charset="us-ascii"
Content-Transfer-Encoding: base64

UkU6IFByZS1BcHByb3ZlZCBIb21lIExvYW4NCg0KDQpSZS9GaSANCkhvbWUgR
XF1aXR5IExvYW5zDQpOZXcgSG9tZSBQdXJjaGFzZXMNCkRlYnQgQ29uc29saW
RhdGlvbg0KDQpTb21lIG9mIG91ciBSZWNlbnQgRGVhbHMNCg0KJDEwMCwwMDA
gZm9yIG9ubHkgJDM4Mi42NCBhIG1vbnRoDQokMTUwLDAwXJzdCBUaW1lIEJ1
eWVyPyBXZSBoYXZlIHBhY2thZ2VzIGZvciANCmFsbCBzcGVjaWFsIHNpdHVhd
GlvbnMNCg0KSG9tZSBFcXVpdHkgTGluZXMgb2YgQ3JlZGl0IC0tLSBQYXlpbm
cgb2ZmIGhpZ2ggaW50ZXJlc3QgZGVidD8gd2FudGluZyB0byByZW1vZGVsIHR
oZSBraXRjaGVuIG9yIGJ1aWxkIGEgZGVjayBtYXliZT8gDQpVc2UgaXQgYW55
d2F5IHlvdSBzZWUgZml0IHdoaWxlIGJlaW5nIGFibGUgdG8gd3JpdGUgb2Zm
IHRoZSBpbnRlcmVzdCBvbiB5b3VyIHRheCByZXR1cm4uIElmIHlvdSBwbGFuI
G9uIHVwZGF0aW5nIHlvdXIgDQpob21lIHRoaXMgcHJvZ3JhbSBjYW4ndCBiZS
BiZWF0Lg0KDQpCYWQgQ3JlZGl0IG9yIFNsb3cgY3JlZGl0IGlzIG5vdCBhIHB
yb2JsZW0uIE5ldyBCdXllcnMgd2VsY29tZSBhcyB3ZWxs
```

Wenn man nun diesen Quellcode in einen eml – Datei kopiert und mit einem Email – Programm öffnet bekommt man die Spam – Mail angezeigt:

RE: Pre-Approved Home Loan

Re/Fi
Home Equity Loans
New Home Purchases
Debt Consolidation

Some of our Recent Deals

$100,000 for only $382.64 a month
$150,000 for only 485.33 a month
$200,000 for only $642.31 a month
$250,000 for only $806.56 a month
$300,000 for only $960.11 a month
$350,000 for only $1,117.84 a month
$400,000 for only $1,254.55 a month
$450,000 for only $1,446.37 a month
and $500,000 for only $1,601.23 a month.

Taking advantage of this offer will benefit you in the following ways.

Saving Money -- Refinancing and paying off high interest debt makes sense

Lower Monthly Payments -- Monthly payment putting you in a pinch? No need to feel it we specialize in getting the lowest payment possible on all of our packages.

Special Need Packages --- Self Employeed or hard to prove income? First Time Buyer? We have packages for
all special situations

Home Equity Lines of Credit --- Paying off high interest debt? wanting to remodel the kitchen or build a deck maybe?
Use it anyway you see fit while being able to write off the interest on your tax return. If you plan on updating your
home this program can't be beat.

Bad Credit or Slow credit is not a problem. New Buyers welcome as well. We will get you the best package for you that fits your current situations needs.

We don't haggle or pressure you. We offer the deal you decide on whether to accept it or not.

60 Seconds Quick Response Form

http://acalizedneto.com/10f/

Regards,

Christina Morris

Anhang

Nessus-Scan vor Absicherung

Nessus Scan Report

SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 1
 - Number of security warnings found : 3
 - Number of security notes found : 9

TESTED HOSTS

 141.62.88.100 (Security holes found)

DETAILS

+ 141.62.88.100 :
 . List of open ports :
   o ssh (22/tcp) (Security warnings found)
   o sunrpc (111/tcp) (Security notes found)
   o ipp (631/tcp) (Security notes found)
   o sunrpc (111/udp) (Security notes found)
   o general/tcp (Security hole found)
   o general/icmp (Security warnings found)
   o general/udp (Security notes found)

 . Warning found on port ssh (22/tcp)

    The remote SSH daemon supports connections made
    using the version 1.33 and/or 1.5 of the SSH protocol.

    These protocols are not completely cryptographically
    safe so they should not be used.

    Solution :
     If you use OpenSSH, set the option 'Protocol' to '2'
     If you use SSH.com's set the option 'Ssh1Compatibility' to 'no'

    Risk factor : Low

 . Information found on port ssh (22/tcp)

    An ssh server is running on this port

 . Information found on port ssh (22/tcp)

    Remote SSH version : SSH-1.99-OpenSSH_4.1

    Remote SSH supported authentication : publickey,keyboard-interactive

 . Information found on port ssh (22/tcp)

    The remote SSH daemon supports the following versions of the
    SSH protocol :

. 1.33
. 1.5
. 1.99
. 2.0

SSHv1 host key fingerprint : b2:78:5d:e6:7f:91:ed:d3:62:fd:6f:10:ac:53:32:5a
SSHv2 host key fingerprint : 36:f0:87:71:89:dd:0b:08:f9:b9:e5:61:80:b8:31:63

. Information found on port sunrpc (111/tcp)

The RPC portmapper is running on this port.

An attacker may use it to enumerate your list
of RPC services. We recommend you filter traffic
going to this port.

Risk factor : Low
CVE : CAN-1999-0632, CVE-1999-0189
BID : 205

. Information found on port sunrpc (111/tcp)

RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is
 running on this port

. Information found on port ipp (631/tcp)

The service closed the connection after 0 seconds without sending any data
It might be protected by some TCP wrapper

. Information found on port sunrpc (111/udp)

RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is
 running on this port

. Vulnerability found on port general/tcp :

You are running a version of Nessus which is not configured to receive
a full plugin feed. As a result, the security audit of the remote host
 produced
incomplete results.

To obtain a complete plugin feed, you need to register your Nessus scanner
at http://www.nessus.org/register/ then run nessus-update-plugins to get
the full list of Nessus plugins.

. Warning found on port general/tcp

The remote host does not discard TCP SYN packets which
have the FIN flag set.

Depending on the kind of firewall you are using, an
attacker may use this flaw to bypass its rules.

See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html
http://www.kb.cert.org/vuls/id/464113

Solution : Contact your vendor for a patch
Risk factor : Medium
BID : 7487

. Information found on port general/tcp

Nessus was not able to reliably identify the remote operating system. It
might be:
IPCop (Linux Kernel 2.4 firewall)
IBM OS/400
Netilla Service Platform 4.0
The fingerprint differs from these known signatures on 6 points.
If you know what operating system this host is running, please send this
signature to
os-signatures@nessus.org :
:1:1:0:64:1:64:1:0:64:1:0:64:1:>64:64:0:1:1:2:1:1:1:1:0:64:5792:MSTNW:2:1:1

. Warning found on port general/icmp

The remote host answers to an ICMP timestamp request. This allows an
attacker
to know the date which is set on your machine.

This may help him to defeat all your time based authentication protocols.

Solution : filter out the ICMP timestamp requests (13), and the outgoing
ICMP
timestamp replies (14).

Risk factor : Low
CVE : CAN-1999-0524

. Information found on port general/udp

For your information, here is the traceroute to 141.62.88.100 :
192.168.2.100
192.168.2.1
?
217.0.68.138
62.154.22.138
129.143.101.141
129.143.101.41
129.143.1.29
129.143.101.202
141.62.88.100

This file was generated by the Nessus Security Scanner

Anhang

Nessus-Scan nach Absicherung

Nessus Scan Report

SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 1
 - Number of security warnings found : 2
 - Number of security notes found : 5

TESTED HOSTS

 141.62.88.100 (Security holes found)

DETAILS

+ 141.62.88.100 :
 . List of open ports :
   o ssh (22/tcp) (Security notes found)
   o general/tcp (Security hole found)
   o general/icmp (Security warnings found)
   o general/udp (Security notes found)

 . Information found on port ssh (22/tcp)

    An ssh server is running on this port

 . Information found on port ssh (22/tcp)

    Remote SSH version : SSH-2.0-OpenSSH_4.1

    Remote SSH supported authentication :
     publickey,password,keyboard-interactive

 . Information found on port ssh (22/tcp)

    The remote SSH daemon supports the following versions of the
    SSH protocol :

      . 1.99
      . 2.0

    SSHv2 host key fingerprint : 36:f0:87:71:89:dd:0b:08:f9:b9:e5:61:80:b8:31:63

 . Vulnerability found on port general/tcp :

    You are running a version of Nessus which is not configured to receive
    a full plugin feed. As a result, the security audit of the remote host
     produced
    incomplete results.

To obtain a complete plugin feed, you need to register your Nessus scanner
at http://www.nessus.org/register/ then run nessus-update-plugins to get
the full list of Nessus plugins.

. Warning found on port general/tcp

The remote host does not discard TCP SYN packets which
have the FIN flag set.

Depending on the kind of firewall you are using, an
attacker may use this flaw to bypass its rules.

See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html
http://www.kb.cert.org/vuls/id/464113

Solution : Contact your vendor for a patch
Risk factor : Medium
BID : 7487

. Information found on port general/tcp

Nessus was not able to reliably identify the remote operating system. It
might be:
IPCop (Linux Kernel 2.4 firewall)
IBM OS/400
Netilla Service Platform 4.0
The fingerprint differs from these known signatures on 6 points.
If you know what operating system this host is running, please send this
signature to
os-signatures@nessus.org :
:1:1:0:64:1:64:1:0:64:1:0:64:1:>64:64:0:1:1:2:1:1:1:1:0:64:5792:MSTNW:2:1:1

. Warning found on port general/icmp

The remote host answers to an ICMP timestamp request. This allows an
attacker
to know the date which is set on your machine.

This may help him to defeat all your time based authentication protocols.

Solution : filter out the ICMP timestamp requests (13), and the outgoing
ICMP
timestamp replies (14).

Risk factor : Low
CVE : CAN-1999-0524

. Information found on port general/udp

For your information, here is the traceroute to 141.62.88.100 :
192.168.2.100
192.168.2.1
?
217.0.68.138
62.154.22.138
129.143.101.141
129.143.101.41
129.143.1.29
129.143.101.202

141.62.88.100

This file was generated by the Nessus Security Scanner