Wie und warum wird gehackt?

Seminar aus Internetrecht

ao. Univ. Prof. Dr. Zankl

WS 2008

Dipl.-Ing. (FH) Thomas Wana

Matrikelnummer: 0508132

Inhalt

Wie und warum wird gehackt? 3
Der Begriff des Hackers 3
White-Hats 3
Black-Hats 4
Bereicherungsvorsatz 4
Schädigungsvorsatz 7
Verbotene Inhalte 8
Sonstiges 9
Wie wird gehackt? 10
Probe 10
Penetrate 13
Persist 20
Propagate 21
Paralyze 21
Abkürzungsverzeichnis 23

Wie und warum wird gehackt?

Der Begriff des Hackers

Bevor wir uns den Fragen „Wie und warum wird gehackt?“ widmen können, muss zunächst der Begriff des *Hackers* bzw. des *Hackens* behandelt werden. Der oder die DurchschnittsbürgerIn würde diese Begriffe, dem von Hollywood und den Medien geprägten Bild folgend, undifferenziert mit einer gefährlichen, potentiell kriminellen Person gleichsetzen, die in jede Art von Computersystemen eindringen und großen Schaden anrichten kann, mindestens sich jedoch sehr gut mit „Computern“ auskennt, die meiste Zeit mit ihnen verbringt und selten das Tageslicht zu Gesicht bekommt. So einfach das die meisten Menschen sehen, so problematisch ist dieser Begriff für die betroffenen Personen selbst. Das Abgrenzungsmerkmal ist die eigene Einstellung zu Gesetzestreue und *Hacker-Ethik* - also im Grunde die Unterscheidung zwischen *Gut* und *Böse*, über die der gemeine Hacker-Begriff keine Aussage trifft.

In der Hacker-Szene selbst hat sich für diese Unterscheidung eine Einteilung nach „Hutfarben“ eingebürgert, analog zum Erkennungsmerkmal von Gut und Böse in Western-Filmen, wobei die Übergänge natürlich fließend sein können. Im Folgenden werden also zwei Extreme aufgezeigt, anhand derer sich allerdings die Motivation und die – für Juristen letztendlich interessante – Rechtmäßigkeit bzw. Rechtswidrigkeit des Hackens sehr gut darstellen lässt.

White-Hats

Als *White-Hats* werden solche IT-Sicherheitsexperten bezeichnet, die ihre Dienste und Wissen ausschließlich zu guten Zwecken anbieten bzw. verwenden. Sie bezeichnen sich selbst als *Hacker*, im scharfen Kontrast zum negativ gefärbten populären Hackerbegriff. In Abgrenzung zu diesen bezeichnen sie *böse* Hacker als „Cracker“.

White-Hats sind einerseits Profis aus der IT-Sicherheitsbranche, die z.B. selbständig für Unternehmen die IT-Sicherheit überprüfen (**Penetration-Testing**) bzw. beraterisch tätig sind (**Consulting**) oder hauptberuflich für ein Unternehmen für diesen Bereich verantwortlich sind. Es handelt sich also um normale Werk- bzw. Dienstvertragsverhältnisse.

Andererseits handelt es sich um Personen aus dem akademischen Umfeld, zumeist, aber nicht nur, um überdurchschnittlich begabte Studenten facheinschlägiger Studienrichtungen. Sie machen den Großteil der bekannten Hacker-Szene aus. Sie verbindet die Faszination an der Technik, spielerische Leichtigkeit und Kreativität im Umgang mit der Technologie, das Finden von Wegen, auf neuartige Weise den Maschinen Verhalten zu entlocken, das so ursprünglich gar nicht geplant war. In dieser Szene ist das Verdienen von **Anerkennung** die maßgebliche Triebfeder. Das Eindringen in fremde Systeme ist **intellektuelle Herausforderung**, bei der auf keinen Fall Schaden angerichtet werden darf. Hier kann man durch Aufzeigen neuartiger Hacker-Techniken oder durch besonders kunstvolles Ausnützen von zuvor noch nicht entdeckten Sicherheitslücken (sog. **Zero-Days**) zu höchsten Ehren gelangen, während unethisches, also gegen die Hacker-Ethik verstoßendes, Verhalten zutiefst verpönt ist. So bedingt die Hacker-Ethik u.a., dass neu entdeckte Sicherheitslücken unbedingt dem Hersteller gemeldet werden müssen, und zwar auf so eine Weise, dass dieser rechtzeitig mit Updates reagieren kann, ohne durch vorzeitige Veröffentlichung der Sicherheitslücke in Bedrängnis zu geraten (**responsible disclosure**). Oder dass der Systemadministrator eines Systems, in das über eine Sicherheitslücke eingedrungen worden ist, von dieser Lücke verständigt wird, damit dieser sie schließen kann. Es besteht also eine freiwillige Selbstbindung an das „Gute“.

Black-Hats

Black-Hats sind gleichsam das Gegenteil der White-Hats. Von Gesetzestreue weit entfernt, ist ihre Motivation nicht etwa die intellektuelle Auseinandersetzung mit der Technologie, sondern geprägt von krimineller Energie. In der Szene werden sie als „Cracker“ bezeichnet. Ein großer Teil der Computerkriminalität spielt sich hier ab.

Bereicherungsvorsatz

Black-Hats hacken unter anderem, um sich oder Dritte unrechtmäßig zu bereichern.

E-Banking- und Kreditkartenmissbrauch

Sehr bekannt und effektiv ist das Ausspähen von E-Banking-Zugangsdaten und/oder Kreditkartennummern. Diese werden oft über **Phishing** in Erfahrung gebracht, bzw. Kundendatenbanken mit gespeicherten Kreditkartennummern gezielt angegriffen und ausgelesen. Mit den so in Erfahrung gebrachten Daten können Geldtransaktionen bzw. Warenbestellungen zugunsten der Angreifer durchgeführt werden.

Seit einiger Zeit werden auf Kreditkarten sogenannte „Card Verification Codes“ (CVC, auch „Card Security Code“, CSC oder „Card Verification Value“, CVV oder CVV2) abgedruckt. Diese Nummern sollen die Sicherheit von Kreditkartentransaktionen im Allgemeinen verbessern. Dieser Code ist nicht auf dem Magnetstreifen abgespeichert, wodurch die Sicherheit erhöht werden soll, dass eine Transaktion mit einer echten Kreditkarte durchgeführt worden ist und nicht etwa mit einer (für den Karteninhaber unbemerkt durch Auslesen des Magnetstreifens erlangten) Kopie. Für den Bereich der Online-Transaktionen gilt, dass der Händler den CVC zwar als Teil des Bestellvorgangs abfragen, diesen Code aber nicht speichern darf[1]. Somit kann ein Angreifer – bei durch CVC geschützten Webshops – mit gestohlenen Kreditkarten nichts anfangen, denn der CVC fehlt. Dies setzt natürlich voraus, dass die Händler tatsächlich den CVC nicht speichern.

Aufgrund der häufigen Vorkommen, hohen Schadenssummen und des damit einhergehenden Imageschadens der Händler als auch der Kreditkartenanbieter wird auf diesem Gebiet einiges unternommen. Es sei hier „Verified by VISA“ erwähnt: bei diesem Angebot übersendet der Kunde nicht mehr dem Händler die Kreditkartennummer so dass dieser die Transaktion mit VISA vornimmt, sondern der Händler leitet den Kunden auf eine Seite von VISA weiter, wo dieser seine Kreditkartendaten eingibt und VISA meldet dann dem Händler nur noch ob die Zahlung erfolgreich war oder nicht. Somit werden die Daten nur noch an einer einzigen, vertrauenswürdigen Stelle eingegeben. Um Phishing vorzubeugen, wird zudem eine, vom Karteninhaber selbst gewählte, Nachricht auf der Webseite ausgegeben. Nur VISA kann diese Nachricht kennen. Auch andere Anbieter kennen ähnliche Verfahren, z.B. MasterCard mit „SecureCode“.

Fälle von Hacks im Zusammenhang mit Kreditkartennummern kommen immer wieder vor. So wurden Anfang 2007 über 45 Millionen Kreditkartennummern bei einem Angriff auf den US-Einzelhändler TJX gestohlen, wobei den Kunden ein geschätzter Schaden von mindestens 750.000,- € entstand.[2]

Erpressung

Gehackt wird auch, um klassische Erpressungen iSd § 144 StGB durchzuführen. Das kann von der Drohung gegen einen Einzelnen, von diesem durch Hacken erlangte sensible Informationen zu

[1] z.B. für VISA: http://usa.visa.com/download/merchants/rules_for_visa_merchants.pdf
[2] http://www.spiegel.de/netzwelt/web/0,1518,474626,00.html

veröffentlichen, bis zur Bedrohung ganzer Konzerne oder Regierungen mit massiven **Distributed-Denial-of-Service-Attacken**, ausgeführt mit **Botnets**, reichen.

Vor der Fußball-EM 2004 beispielsweise wurden zahlreiche Online-Wettbüros mit eben solchen DDoS-Angriffen bedroht und erpresst.[3]

Betrug

Phishing ist das Paradebeispiel für den klassischen Betrug iSd § 146 StGB im Internet. Mit der Phishing-Mail oder -Webseite wird jemand über Tatsachen getäuscht, z.B. eine gefälschte Login-Seite seiner Bank, was bei diesem zu dem Irrtum führt, dass es sich um die echte E-Banking-Seite handelt, und dieser Irrtum zu dem Verhalten führt, dass der Getäuschte seine Zugangsdaten auf dieser Seite eingibt, welche der Angreifer dann seinerseits verwendet, um Überweisungen zu tätigen, was den Betrogenen im Vermögen schädigt.

Phishing[4] kommt in den verschiedensten Varianten vor, sei es als (gefälschte) E-Mail der Bank mit der Aufforderung, zur „Überprüfung der Zugangsdaten“ ebendiese einzugeben oder als getarnte Webseite, die wie die Login-Seite des E-Bankings aussieht.

Wirtschaftsspionage, politische Spionage

In Zeiten zunehmender Vernetzung nehmen auch Fälle gezielter Hackerangriffen zum Zwecke der Wirtschaftsspionage oder sogar politischer Spionage zu.

2004 wurde in einem berühmt gewordenen Hack der komplette Quellcode des Router-Betriebssystems IOS des kalifornischen Herstellers und Marktführers Cisco gestohlen.[5] Der Wert dieses Codes muss schon rein aufgrund der Entwicklungszeit und der dafür aufgewendeten Personalkosten in die zig-Millionen US$ gehen. Besonders brisant ist auch, dass es mit dem Besitz des Quellcodes einfacher wird, Sicherheitslücken in IOS zu finden – was dann weitere Angriffe zur Folge haben kann. Aufgrund der weiten Verbreitung von IOS (> 60% Marktanteil bei kritischer Netzwerkinfrastruktur) multipliziert sich die Gefährlichkeit noch.

Im Sommer 2007 wurden Fälle bekannt, wonach die Chinesische Volksbefreiungsarmee und somit der Chinesische Staat angeblich gezielt Regierungscomputer der deutschen Bundesregierung infiziert haben, um diese ausspähen zu können.[6] Auch die USA[7] und Großbritannien[8] meldeten solche Angriffe.

Identitätsdiebstahl

Ein sehr aktuelles Problem stellen Fälle von Identitätsdiebstahl dar. Hierbei werden gezielt Kundendatenbanken gehackt, um an möglichst viele Daten von Personen zu kommen. Diese Daten werden nun verwendet, um z.B. Duplikate offizieller Papiere zu erhalten oder Kredite auf Kosten der Geschädigten aufzunehmen.

Um einen Einblick in diese Aktivitäten zu erhalten, sei an dieser Stelle ein Auszug der FAQs der

---

[3] http://www.heise.de/newsticker/DDoS-Erpressung-gegen-Online-Wettbueros--/meldung/48613

[4] Das Wort Phishing selbst ist ein Kunstwort aus „fishing“, also dem „Fischen“ nach Zugangsdaten, und „phreaking“. Letzteres wiederum ist ein Begriff aus frühester Hackerzeit, den 1970er-Jahren, und bezeichnet eigentlich das „Hacken“ von Telefonanschlüssen, um gratis Ferngespräche führen zu können - soll also die etymologische Verbindung zur Hackerwelt unterstreichen.

[5] http://www.heise.de/security/Angeblich-Cisco-Quellcode-gestohlen--/news/meldung/47410

[6] http://www.heise.de/newsticker/China-spaeht-angeblich-PCs-des-Bundeskanzleramtes-aus--/meldung/94899

[7] http://www.spiegel.de/netzwelt/web/0,1518,503678,00.html

[8] http://www.spiegel.de/netzwelt/tech/0,1518,503921,00.html

Seite http://www.identitytheft.org/ zitiert:

„**How does the imposter take your identity?**

It is easy. All that is needed is your social security number, your birth date and other identifying information such as your address and phone number and whatever else they can find out about you. With this information, and a false driver's license with their own picture, they can begin the crime. They apply in person for instant credit, or through the mail by posing as you. They often provide an address of their own, claiming to have moved. Negligent credit grantors in their rush to issue credit do not verify information or addresses. So once the imposter opens the first account, they use this new account along with the other identifiers to add to their credibility. This facilitates the proliferation of the fraud. Now the thief is well on his/her way to getting rich and ruining your credit and good name. "

Das Problem scheint besonders in den USA verstärkt aufzutreten, womöglich weil nur wenige, relativ leicht zu beschaffende, Daten ausreichen (siehe Zitat oben). Der FTC (Federal Trade Commission, die US-Handelsbehörde) zufolge werden jährlich mehr als 8 Millionen Amerikaner Opfer von Identitätsdiebstahl, der Schaden betrage rund 52 Milliarden US$[9]. Es gibt Regierungs-Webseiten, die vor der Gefahr warnen[10], und sogar eigene Organisationen, die sich eigens diesem Problem widmen[11].

Betrug mit Online-Werbung

Botnets werden auch dazu verwendet, um Betrug mit Banner-Werbung auszuführen. Der Angreifer richtet beispielsweise einen Google AdSense-Account[12] ein und bindet diesen auf einer Webseite ein. Ein Botnet wird in der Folge eingesetzt, um diese Seite besonders oft aufzurufen und Klicks auf die Werbe-Links zu simulieren, wodurch dem Betreiber der Seite (also dem Angreifer) Geld gutgeschrieben wird (**Click fraud**). Google und andere Pay-per-click-Anbieter versuchen diesem Problem mit ausgefeilten statistischen Methoden Herr zu werden[13], allerdings können die Anbieter hier einen gewissen Interessenskonflikt nicht leugnen: denn höhere Klickraten auf ausgelieferte Werbung suggeriert den (zahlenden!) Werbekunden höhere Reichweite, wodurch der Anbieter wiederum höhere Werbepreise verlangen kann.

Spamming

Spamming, also das Versenden unerwünschter Werbemails, ist wegen der schieren Zahl an versendeten Spam-Mails pro Tag immer noch ein sehr lukratives Geschäft. Auch hier werden **Botnets** eingesetzt, um Mails in riesiger Zahl im Internet zu verbreiten.

Zwar können E-Mail-Adressen durch automatisches Generieren erraten werden; da jedoch die Werbe-Ausbeute einer Spam-Kampagne umso höher ist, je mehr gültige E-Mail-Adressen angespammt werden, herrscht auch großes Interesse daran, möglichst viele solcher Adressen zu bekommen. Es wird daher auch gezielt gehackt, um an User-Datenbanken inklusive derer E-Mail-Adressen zu gelangen.

Da es das Spam-Problem schon seit Jahren gibt, muss es sich wohl für die Spammer rechnen. Die Top-3-beworbenen Produktkategorien[14] sind Rolex-Uhren mit 0,0075% Klicks pro Spam-Mail,

[9] http://www.computerwoche.de/knowledge_center/security/1870796/
[10] http://www.ftc.gov/bcp/edu/microsites/idtheft/
[11] http://www.idtheftcenter.org/
[12] https://www.google.com/adsense
[13] „Tuzhilin-Report": http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf
[14] http://www.nytimes.com/2006/07/03/technology/03drill.html?_r=1

gefolgt von Werbung für Arzneimittel mit 0,02% Klickrate. Unübertroffen ist Werbung für Internet-Pornographie: ganze 5,6% „Response“ pro versendetem Spam-Mail. Dabei steigt die absolute Anzahl an Klicks linear mit der Anzahl der versendeten Spam-Mails, was die Aggressivität der Spammer erklärt.

Wissenschaftler der UC Berkeley und der UC San Diego haben im Rahmen einer Forschungsarbeit[15] eine eigene Spam-Kampagne aufgezogen, um den Ertrag einer solchen abschätzen zu können. Dabei schätzen sie den täglichen Umsatz einer Spam-Kampagne für Arzneiprodukte auf ca. 7.000 US$ oder 3,5 Millionen US$ pro Jahr.

Dem gegenüber schätzt eine Studie der EU-Kommission[16] den weltweit durch Spam verursachten Schaden (Datenübertragungskosten, Personalkosten, etc.) auf ca. 10 Milliarden €.

Eine technische Gegebenheit hat maßgeblich für den explosionsartigen Anstieg von Spam-Mails in den letzten Jahren Mitschuld. Das „Simple Mail Transfer Protocol“ (SMTP) ist im Internet für die Weiterleitung von E-Mails von Mailserver zu Mailserver (MTA – Mail Transfer Agent) bzw. Mailprogramm zu Mailserver (MUA – Mail User Agent) verantwortlich. Dieses Protokoll ist sehr alt und zur Zeit seiner Schaffung war Spam, oder allgemein gefälschte E-Mails, kein Thema. Das damalige Internet (bzw. Arpanet als Vorläufer des heutigen Internet) mit seinen fast ausschließlich dem akademischen Umfeld zugehörigen Teilnehmern war überschaubar und weiterführende Überprüfungen nach Absender, Empfänger und Inhalt einer Nachricht überflüssig. In dieser Zeit konnte von egal welchem Punkt des Internets eine Verbindung zu einem beliebigen Mailserver aufgebaut werden und über diesen Mail versendet werden, auch wenn der Mailserver gar nicht zum eigenen Netz gehörte. Dies öffnete der explosionsartigen Verbreitung von Spam-Mails Tür und Tor. Erste Maßnahmen wie das Kontrollieren des „Relaying“ konnten nur bedingt Abhilfe schaffen. Es wäre eigentlich ein weltweiter Umstieg auf ein neues, sichereres Mailprotokoll notwendig, doch das ist aufgrund der Verbreitung von SMTP nur schwer durchzusetzen. Deshalb haben sich verschiedene Zusätze zu diesem Protokoll entwickelt, die sich langsam durchsetzen, wie z.B. Greylisting oder SPF (Sender Policy Framework).

All diese Maßnahmen dienen dem Ziel, sicherzustellen, dass ein bestimmter Benutzer auch tatsächlich über einen bestimmten Mailserver senden darf, und der Mailserver nicht etwa für Spamming missbraucht wird. Spammer haben daher wachsendes Interesse daran, in Rechner in einem Netz einzudringen und über den „autorisierten“ Mailserver dieses Netzes Spam-Mails loszubekommen. Das wird auch im großen Stil mittels Botnets gemacht.

Dialer

Unter einem **Dialer** versteht man ein Programm, das sich in den Einwahlvorgang zum Internetprovider einklinkt und bewirkt, dass nicht die kostengünstige Telefonnummer zum Orts- oder Internettarif gewählt wird, sondern eine kostenpflichtige Mehrwertnummer des Angreifers, der somit an den Telefoniegebühren des Geschädigten verdient.

In Zeiten von Einwahlzugängen per analogem Modem oder ISDN noch sehr weit verbreitet, verliert diese Art von Angriff immer mehr an Bedeutung.

Schädigungsvorsatz

Black-Hats hacken nicht nur um sich unrechtmäßig zu bereichern, sondern auch, um Dritten Schaden zuzufügen, sei es als Auftragsarbeit oder aus eigenen Motiven.

---

[15] http://www.icsi.berkeley.edu/pubs/networking/2008-ccs-spamalytics.pdf
[16] http://europa.eu/rapid/pressReleasesAction.do?reference=IP/01/154&format=HTML&aged=1&language=DE&guiLanguage=en

Datenmanipulation, Datenvernichtung

Der wichtigste Fall der Schädigung durch Hacken ist sicherlich die Datenmanipulation bzw. Datenvernichtung. Hier geht es um das gezielte Verändern bzw. Zerstören von Daten, z.B. der Kundendatenbank oder der Buchhaltung. Je nach Größe des Unternehmens und des IT-Know-Hows können solche Angriffe durchaus existenzbedrohend sein. Gibt es keine, keine brauchbaren oder nur veraltete Sicherungen beispielsweise der Buchhaltung eines Kleinunternehmers kann diesem so ein Schlag die Existenz kosten. Noch gefährlicher, weil subtiler, ist die gezielte Manipulation von Daten, da hier Fehler, z.B. Buchungsfehler, erst allmählich auftauchen, wenn etwa frühere Sicherungen schon längst ausgealtert sind.

Auch das gezielte Verändern von Konfigurationsdateien, in denen oft beträchtliche Stunden an Detailarbeit stecken, fällt hier unter diesen Begriff.

Wichtig zu erwähnen ist in diesem Zusammenhang, dass solche Angriffe nicht nur von Außen kommen können, sondern auch in vielen Fällen von Innen, beispielsweise von entlassenen Mitarbeitern, die ihre Zugänge noch haben und auf diese Art und Weise Rache üben wollen.

Sabotage

Nicht nur die Manipulation oder Vernichtung von Daten sind denkbare Motive für Hacking, sondern auch außerhalb des Rechners die Beeinflussung oder Zerstörung von angeschlossenen Maschinen und/oder Infrastruktur. Das kann reichen von der Beeinflussung von automatisierten Fertigungsabläufen in der Industrie, bei der bei heutiger Just-In-Time-Fertigung sofort Umsatzausfälle in schwindelerregenden Höhen entstehen können, bis hin zu Sabotage von kritischer Infrastruktur (Wasser, Kraftwerke, Verkehrsregelung, etc.). Unter dem Stichwort *Cyberterrorismus* bzw. *Cyberwarfare* sammeln sich verschiedene Szenarien, bei denen ein politischer Angreifer von außen erheblichen Schaden in Gesellschaften anrichten kann.

In diesem Zusammenhang erwähnt sei eine relativ neue Art von Bedrohung, nämlich der gezielten Spionage bzw. Sabotage durch manipulierte Firmware bzw. Hardware. Das Verhalten eines Mikrochips ist von außen so gut wie gar nicht in Erfahrung zu bringen, dadurch ist es zumindest denkbar, dass Regierungen mit ihrem Einfluss auf Hardware-Hersteller Hintertüren in exportierte Chips einbauen lassen, die sich später noch als „nützlich“ erweisen könnten. Das FBI ermittelt bereits in diese Richtung[17] im Zusammenhang mit gefälschten chinesischen Mikrochips.

Rufschädigung, Existenzvernichtung

Denkbar ist auch Hacking, um z.B. unter der Identität des Opfers E-Mails an Kollegen, Arbeitgeber, Verwandte mit bloßstellendem Inhalt zu senden oder peinliche Einträge in Foren zu verfassen, um das Ansehen des Opfers zu schädigen.

In besonders schlimmen Fällen wird auch gehackt, um das Opfer strafrechtlicher Verfolgung auszusetzen, indem belastendes Material beispielsweise in Form von Kinderpornographie auf dem Rechner des Opfers abgelegt und dies anschließend den Strafverfolgungsbehörden anonym gemeldet wird. Hier geht es nur noch um die reine Existenzvernichtung des Opfers.

Verbotene Inhalte

Neben unrechtmäßiger Bereicherung und Schädigung kann auch das Zurverfügungstellen verbotener Inhalte ein Motiv für Hacking sein.

[17] http://www.infowar-monitor.net/modules.php?op=modload&name=News&file=article&sid=1700

Verbotene Inhalte

Terroristische bzw. verbotene politische Gruppierungen nutzen längst das Internet für die interne Kommunikation bzw. zur Verbreitung von Propagandamaterial. Eine Methode, um zu dem dafür benötigten Webspace zu gelangen, ist das Hacken von Webservern, auf denen das Material dann abgelegt wird. Auch Kinderpornographie oder urheberrechtlich geschütztes Material findet so eine Plattform für die Verbreitung, wird aber mehr und mehr durch P2P-Austausch[18] verdrängt.

Politische Messages

Eine effektive Methode, um zu Aufmerksamkeit zu gelangen, ist weiters das sogenannte **Website-Defacement**. Dabei werden Webserver, auf denen möglichst bekannte Webseiten liegen, gehackt, und diese Webseiten abgeändert oder ganz durch eigene Webseiten mit den gewünschten Botschaften ersetzt. Durch die erregte Aufmerksamkeit ist die Werbewirkung enorm.

Ein Beispiel aus jüngerer Zeit ist das Defacement mehrerer Webseiten der NPD durch Netzaktivisten des Chaos Computer Clubs Ende 2008[19].

Sonstiges

Verbreitung und Verschleierung

Auch wird gehackt, um die eigenen Aktivitäten zu verschleiern. Es ist nicht sehr ratsam, von seinem eigenen Internetanschluss daheim eine Hackerkarriere starten zu wollen. Nur allzu leicht wird die eigene Identität aufgrund der IP-Adresse auffliegen. Klüger ist es, erst über zwei oder drei schon vorher gehackte Rechner einen Angriff auf ein neues Ziel zu starten. Es wird also auch gehackt, um sich solche „Sprungbretter“ zu schaffen.

Neuerdings gewinnen auch Anonymisierungsnetzwerke wie TOR mehr und mehr an Bedeutung. Diese Netzwerke sind werden von hunderten Freiwilligen getragen und bestehen aus Nodes, die untereinander verschlüsselt kommunizieren und Anfragen untereinander weiterreichen, ohne darüber Protokoll zu führen. Es kann im Nachhinein (mit einer gewissen Sicherheit) nicht nachvollzogen werden, wer wann von wo wohin welche Verbindungen mit welchem Inhalt aufgebaut hat.

[18] Peer-To-Peer: über Tauschbörsen-Netzwerke wie gnutella oder -Protokolle wie bittorrent werden Dateien nicht mehr von einem Server zum Client übertragen, sondern unter den Teilnehmern untereinander ausgetauscht. Das macht die Lokalisierung und/oder das Sperren der Downloads schwierig bis unmöglich.

[19] http://www.heise.de/newsticker/25C3-NPD-Webseiten-fest-in-Hackerhand--/meldung/121009

Wie wird gehackt?

Nachdem wir uns nun die Motive, warum gehackt wird, näher angesehen haben, soll nun auf die Frage eingegangen werden, wie gehackt wird.

Die Einteilung der Methoden findet sinnvollerweise chronologisch statt, nach der Angriffs-Phase, wobei auch hier wieder die Grenzen fließend sind.

Probe

In dieser Phase geht es darum, im Vorfeld des Angriffs möglichst viele Informationen über das Ziel zu sammeln, um die Chancen auf Erfolg möglichst zu erhöhen.

Footprinting

Beim Footprinting wird versucht, einen „Fußabdruck“, also ein Profil, des Angriffsziels zu ermitteln. Der Angreifer versucht beispielsweise herauszufinden, welche Server mit welchen Betriebssystemversionen eingesetzt werden, welche Services darauf laufen, von wo aus welche Services erreichbar sind, welche Benutzerkennungen es im System gibt (oft abzuleiten aus auf der Webseite publizierten E-Mail-Adressen), etc.

Hierzu gibt es eine Vielzahl von Möglichkeiten. Wie schon erwähnt, gibt oft die eigene Unternehmenswebsite wertvolle Informationen preis, z.B. welche E-Mail-Accounts es gibt, aber auch Informationen über Ansprechpartner, gegen die ein Angriff mit **Social Engineering** Erfolg haben könnte. Jede beliebige Unternehmenswebsite dient hier als gutes Beispiel.

Die WHOIS-Datenbank ist eine verteilte Datenbank, die von den 5 großen RIRs[20] betrieben wird und Informationen zu jedem vergebenen IP-Netz und Domainnamen beinhaltet. Die Nutzung ist frei und von jedermann durchführbar. Durch eine WHOIS-Abfrage gegen den Domain-Namen bzw. IP-Adressen-Block des Angriffsziels lassen sich wertvolle Informationen wie z.B. die technischen und administrativen Kontakte (personenbezogene Daten!) des Ziels in Erfahrung bringen. Eine WHOIS-Abfrage von wana.at bringt sowohl meine (damalige, zur Zeit der Registrierung gültige) Postadresse und Telefonnummer als auch Informationen über den Server-Housing-Provider ans Licht:

```
...
personname:     THOMAS Wana
organization:
street address: Schwanengasse 248
postal code:    A-2821
city:           Lanzenkirchen
country:        Austria
fax-no:         +43262745727
nic-hdl:        TW2923682-NICAT
changed:        20070719 13:48:43
source:         AT-DOM

personname:
organization:   eTel Austria GmbH & Co KG
street address: Thomas A. Edison Strasse 1
```

[20] Regional Internet Registry: Diese Registrys haben die Oberhoheit über die weltweit verfügbaren IP-Adressen und geben diese an LIRs, Local Internet Registries, z.B. ISPs, weiter. Die 5 RIRs sind zur Zeit: RIPE NCC für Europa, Russland und den nahen Osten, ARIN für Nordamerika, LACNIC für Mittel- und Südamerika, APNIC für Asien und Australien, und AFRINIC für Afrika.

```
postal code:    7000
city:           Eisenstadt
country:        Austria
phone:          +4305010110
fax-no:         +4305010115274
e-mail:         domain.admin@etel.at
nic-hdl:        EAGC3059838-NICAT
changed:        20071023 08:50:06
source:         AT-DOM
...
```

Auch das DNS kann zwingend jederzeit von jedermann abgefragt werden. Aus diesem kann ein Angreifer sofort ermitteln, welche Mailserver im Einsatz sind, und erste Rückschlüsse über den logischen Aufbau des Netzes erhalten. Bei schlecht konfigurierten DNS-Servern ist sogar noch ein Zone Transfer möglich, d.h. mit einer einzigen Abfrage kann der Angreifer alle (!) im DNS registrierten Server im Netz des Ziels abfragen. Aus der folgenden Abfrage erfährt der Angreifer, welche Mailserver die Universität Wien betreibt, und wie sie heißen:

```
dr.gonzo@pctw:~$ host univie.ac.at
univie.ac.at has address 131.130.1.84
univie.ac.at mail is handled by 10 zidmx1.univie.ac.at.
univie.ac.at mail is handled by 10 zidmx2.univie.ac.at.
univie.ac.at mail is handled by 10 zidmx3.univie.ac.at.
```

Port-Scanning

Während man beim Footprinting noch keine verdächtigen Spuren beim Angriffsziel hinterlässt, da es sich um legale Abfragen handelt, geht man beim Port-Scanning bereits einen Schritt weiter. Hier wird bereits direkt Kontakt mit den anzugreifenden Servern aufgenommen und diese auf auf Netzwerkports lauschenden Diensten abgefragt. Diese Scans sind direkt im angegriffenen Netz sichtbar und können – je nach Aggressivität – von Abwehreinrichtungen erkannt und blockiert werden. nmap ist ein Werkzeug für Portscanning, ein Aufruf könnte so aussehen:

```
dr.gonzo@pctw:~$ nmap www.wana.at

Starting Nmap 4.62 ( http://nmap.org ) at 2009-01-19 19:44 CET
Interesting ports on mars.wana.at (212.88.179.74):
Not shown: 1708 filtered ports
PORT     STATE  SERVICE
25/tcp   open   smtp
53/tcp   open   domain
80/tcp   open   http
443/tcp  open   https
993/tcp  open   imaps

Nmap done: 1 IP address (1 host up) scanned in 11.592 seconds
```

Hier läuft also ein Mailserver, DNS-Server und Webserver.

Fortgeschrittenere Produkte wie *Nessus*[21] machen das Scannen noch komfortabler, indem sie die gefundenen Services auch gleich auf bekannte Sicherheitslücken abtesten.

Sniffing

Schon zur Hälfte bereits in der nächsten Phase, dem konkreten Angriff, angesiedelt, dient das Sniffen dem Beschaffen von Informationen durch Abhören von Netzwerken. Bei kabelgebundenen Netzen ist dazu physischer Zugriff auf das Netzwerk notwendig, auch sind heutige Netzwerke aus

[21] http://www.nessus.org/

Performancegründen segmentiert, d.h. es wird nur ein geringer Teil des gesamten Verkehrs auf dem Kabel zu sehen sein (**ARP-Spoofing** bzw. **-Flooding** kann hier allerdings Abhilfe schaffen).

Einen wahren Boom hat das Sniffen mit der explosionsartigen Verbreitung von WLAN-Netzen erfahren. Zum einen ist hier kein physischer Zugriff auf ein Kabel notwendig um das Netz sehen zu können, wie heutzutage jeder in seiner eigenen Wohnung feststellen kann, es kann also nicht verhindert werden, dass Dritte die Datenpakete in diesen Netzen sehen können. Deshalb wurden Verschlüsselungsprotokolle eingeführt, um Datenpakete zumindest unleserlich zu machen, wenn die Ausbreitung schon nicht kontrolliert werden kann, diese litten jedoch zu Beginn unter gravierenden Mängeln (WEP 56, 64, 128 Bit), bei denen es nur eine Frage von Minuten ist, um sie zu knacken. Einen viel besseren Sicherheitsstandard bietet WPA, das heute noch als sicher gelten kann. Gerüchteweise gibt es aber bereits erste erfolgreiche Angriffe auf dieses Protokoll[22].

Selbst wenn das Ursprungsnetz gute Sicherheit bietet, kann Verkehr immer noch auf dem Weg durch das Internet gelesen werden. Einen verlässlichen Schutz gegen Sniffing bietet hier nur der konsequente Einsatz von verschlüsselten Protokollen wie z.B. ssh bzw. der Einsatz von SSL bzw. TLS. Klartext-Protokolle wie POP3 sind nicht nur unsicher sondern mittlerweile schon als *gefährlich* zu erachten, da das Sniffen eines Passwortes, vor allem in Verbindung mit öffentlichen WLANs, nur eine Frage von Sekunden ist. Das Standardwerkzeug für Sniffing ist Wireshark[23], ist aber nicht nur für Hacker, sondern auch für Netzwerktechniker und Softwareentwickler ein extrem wertvolles Werkzeug.

Hier möge auch ein Problem bezüglich des Verbietens solcher Software angesprochen werden. Natürlich kann und wird Wireshark von Hackern missbraucht werden, aber auch von Technikern im Rahmen ihrer Arbeit zu legitimen Zwecken verwendet werden. Soll solche Software verboten werden? Der umstrittene sogenannte „Hackerparagraph“ § 202c dStGB lautet:

§ 202c Vorbereiten des Ausspähens und Abfangens von Daten

(1) Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er

1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder
2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist,

herstellt, sich oder einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft.

(2) § 149 Abs. 2 und 3 gilt entsprechend.

Es besteht die Gefahr, dass Software wie Wireshark unter den Abs. (1) Z 2 subsumiert und der Einsatz damit kriminalisiert wird, was letztendlich eigentlich der „guten“ Seite, den Systemadministratoren, die geeigneten Werkzeuge nimmt, um sich gegen Angriffe zur Wehr zu setzen. Diese Entwicklung ist mit größter Aufmerksamkeit zu verfolgen.

[22] http://www.heise.de/newsticker/WPA-angeblich-in-weniger-als-15-Minuten-knackbar--/meldung/118514
[23] http://www.wireshark.org/

Penetrate

Es gibt sehr viele Möglichkeiten und Techniken, um in fremde Systeme einzudringen und fremde Daten auszuspähen. Hier werden nur einige davon vorgestellt, die besonders im Web-Bereich von Bedeutung sind.

Brute Force

„Brute Force“, also rohe Gewalt, ist es, wenn ein Angreifer der Reihe nach verschiedene Passwörter durchprobiert, um Zugangssperren zu überwinden. Dies ist wohl die älteste und primitivste Angriffstechnik, aber immer noch überraschend effektiv, weil es mit der Qualität der von den Benutzern gewählten Passwörter oft nicht weit her ist (Name des Freundes/der Freundin, Haustiers, Kinder, Geburtstage, bzw. primitive Abwandlungen davon). Das kann natürlich auch automatisiert ablaufen, z.B. aufgrund eines „Wörterbuches“, das mitsamt Abwandlungen der darin enthaltenen Wörter durchprobiert wird (*Dictionary Attack*). Die Chancen auf Erfolg können wesentlich erhöht werden, wenn durch vorhergehendes **Social Engineering** das Umfeld des Opfers in Erfahrung gebracht wird, eben um das Wörterbuch mit Namen von Freunden und Verwandten etc. anzureichern.

Die Erfolgschancen von Brute-Force-Angriffen sind umso höher, je mehr Passwörter man in einer bestimmten Zeitspanne durchprobieren kann, man spricht von der *Geschwindigkeit* des Brute-Force-Angriffes. Die Idee ist natürlich keineswegs neu; die Idee, Passwörter automatisiert durchzuprobieren, gibt es schon genauso lange wie die Authentifizierung mittels Passwörtern selbst. Das erste Mal diesem Problem ausgesetzt waren netzwerkfähige Multitasking-fähige Betriebssysteme der 70er-Jahre wie UNIX, bei denen sich der User auch über ein Netzwerk am Rechner anmelden kann, und ebendort ein Passwort eingeben muss. Gegenmaßnahmen wurden daher schon frühzeitig erdacht. In so gut wie allen Betriebssystemen ist eingebaut, dass nur eine gewisse Anzahl von Fehlversuchen akzeptiert wird, bis ein User-Account gesperrt wird (und nur noch durch einen Administrator entsperrbar ist, der die Ursache für die Sperre untersucht). Auch können Verzögerungen von einigen Sekunden zwischen fehlgeschlagenen Anmeldeversuchen Brute-Force-Angriffe sehr effektiv verhindern – müssen doch bei starken Passwörtern Millionen und Abermillionen von Versuchen getätigt werden, bis ein Passwort erraten wurde. Das würde dann eben entsprechend Millionen und Abermillionen von Sekunden dauern.

Letztere Methode, nämlich Verzögerungen zwischen Fehlversuchen, ist übrigens dem Sperren von Accounts eindeutig vorzuziehen, denn dies kann auch missbraucht werden – man melde sich nur x Mal mit einem fremden Usernamen und falschen Passwort bei einem Dienst an, und der User ist aus dem Dienst ausgesperrt (bis zur nächsten Entsperrung).

Besonders Webapplikationen kamen hier in letzter Zeit wieder in die Schlagzeilen, weil in vielen diese einfachen Gegenmaßnahmen gegen Brute Force nicht implementiert sind. Erst vor einigen Wochen wurde ein Administratorkonto von Twitter mit Brute Force gehackt[24], worauf der Angreifer die Twitter-Konten einiger Berühmtheiten (u.a. Barack Obama, Britney Spears) übernehmen konnte. In der darauffolgenden Analyse zeigte sich, dass weder eine Verzögerung zwischen Passwort-Fehlerversuchen, noch eine automatische Sperre des Accounts nach x Fehlversuchen in Twitter implementiert ist, was diesen Angriff wohl erst möglich gemacht hat.

Man muss aber auch sagen, dass der Fall bei Webapplikationen etwas anders als bei Shell-Logins gelagert ist. Einzelne Requests eines Webservers werden in der Regel von jeweils eigenen Serverprozessen verarbeitet, die natürlich Ressourcen (in erster Linie RAM) benötigen. Es ist daher günstig, wenn einzelne Requests nur so kurz wie möglich dauern. Baut man

[24] http://www.heise.de/security/Lehren-aus-dem-Twitter-Hack--/news/meldung/121286

Fehlversuchsverzögerungen ein, leben diese Requests viel länger, d.h. es laufen viel mehr Requests parallel und es werden entsprechend mehr Ressourcen benötigt. Es wird daher spannend zu beobachten sein, welche Gegenmaßnahmen hier aufkommen. Die Idee, möglichst viele Ressourcen durch Requests aufzubrauchen, findet sich auch beim **SYN-Flooding**.

Wie immer empfiehlt es sich, überall starke Passwörter zu verwenden.

SQL-Injection

SQL steht für „Structured Query Language“ und ist die Standard-Abfragesprache gegen Datenbanken. So gut wie alle modernen Webapplikationen verwenden eine Datenbank im Hintergrund, um z.B. Berechtigungen von Benutzern zu speichern. SQL-Injection, also das Einschleusen von SQL durch den Angreifer, ist dort möglich, wo Benutzereingaben ungefiltert in SQL-Abfragen inkludiert werden, beispielsweise die Abfrage, ob ein Benutzer Administratorrechte erhalten soll oder nicht (die Benutzereingabe ist hier der Benutzername). Durch geschickte SQL-Injection ist es möglich, diese Abfragen zu manipulieren, so dass der Angreifer mehr Rechte erhalten kann als ihm zustehen.

Besonders PHP ist in dieser Hinsicht chronisch krank, denn von Haus aus hat PHP keine Unterstützung für sogenannte „Bind-Variablen“ wie Perl mit DBI oder Java mit JDBC. Daher muss der Programmierer die SQL-Abfrage samt Parameter „händisch“ zusammenbauen. Um das sicher zu machen, müssen Benutzereingaben konsequent auf gefährliche Zeichen wie das Anführungszeichen gefiltert werden – was aus Unwissenheit oder Bequemlichkeit nur allzu oft nicht gemacht wird.

Ein Beispiel: die Abfrage wird in der Webapplikation (PHP) folgendermaßen aufgebaut:

```
$sql = „SELECT * FROM accounts WHERE username='$username' AND
password='$password'“;
```

wobei $username und $password mit den Feldern der Login-Maske befüllt werden (register_globals spielt hier keine Rolle). Die Logik hinter dieser Abfrage ist, dass eine Zeile zurückgeliefert wird, wenn Username und Passwort korrekt sind, ansonsten nicht. An diesem grundsätzlichen Aufbau ist zunächst nichts auszusetzen.

Dieser Code ist aber anfällig für einen SQL-Injection-Angriff, denn wird als Passwort die Zeichenfolge **`xxx' OR '1=1`** übermittelt, wird aufgrund des Statements oben folgende Anfrage an den Datenbankserver geschickt:

```
SELECT * FROM accounts WHERE username='admin' AND password='xxx' OR '1'='1'
```

was immer eine Zeile zurückliefern wird und der Angreifer nun als Admin eingeloggt ist. Richtig wäre es gewesen, die Variablen $username und $password vor Einsetzen in den $sql-String oben zu filtern:

```
$username = mysql_escape_chars($username);
$password = mysql_escape_chars($password);
```

Cross-Site-Scripting

Cross-Site-Scripting (XSS) ist im Moment das größte Thema in Bezug auf die Sicherheit im Web. Es existieren sehr viele verschiedene Techniken, um Cross-Site-Scripting auszuführen, doch alle haben ein gemeinsames Ziel. Das Web-Protokoll HTTP ist „stateless“, hat also keinen Begriff für fortgesetzte sogenannte Benutzer-Sessions. Diese werden behelfsmäßig über „Cookies“ realisiert, also einer eindeutigen Identifikationsnummer („Session-ID“), die üblicherweise beim Einloggen erzeugt und im Browser eines Benutzers gespeichert wird und bei jeder Anfrage an den Webserver erneut mit übertragen wird, so dass die Webapplikation dahinter erkennen kann, dass es sich um den

selben User wie ein paar Requests davor handelt, und so dem Benutzer die Illusion geben kann, eine durchgehende Session mit der Webapplikation zu haben, also z.B. in einer Webmail-Oberfläche eingeloggt zu sein. Das bedeutet aber auch, dass jedermann, der diese Session-ID auslesen diese Session auch übernehmen kann („Session-Hijacking“) und so in diesem Beispiel ungestört die E-Mails des anderen Benutzers mitlesen kann. Dem Ziel, an die Cookies und somit an die Session-IDs eines Benutzers zu kommen, dient Cross-Site-Scripting.

Cookies werden als „Sicherheitsmaßnahme“ immer nur an den Server zurückgeschickt, von dem sie gesetzt wurden. Cross-Site-Scripting ist nun, wenn es der Angreifer schafft, in die Webapplikation Skriptcode einzuschleusen, der dann vom Browser des Opfers ausgeführt wird. Dieser Skriptcode liest die Cookies des Browsers aus und sendet diese zurück an den Angreifer. Im Beispiel mit dem Webmail könnte also etwa eine manipulierte E-Mail zu diesem Ergebnis führen, wenn die Webmail-Webapplikation keine besonderen Vorsichtsmaßnahmen dagegen trifft wie z.B. Skriptcode in E-Mails zu filtern.

Spoofing

Spoofing heißt ganz allgemein „Manipulation“. Die Manipulationen sind hier dergestalt, dass der Angreifer zu Datenströmen des Opfers gelangt, die nicht für jenen bestimmt sind. Hier gibt es wiederum zahlreiche Möglichkeiten, um Angriffe auszuführen. Da es sich um Manipulationen auf Netzwerkebene handelt, bietet es sich an, die verschiedenen Methoden nach dem betroffenen ISO-OSI-Layer[25] zu reihen. Aufgrund der extrem weiten Verbreitung wird hier nur auf Ethernet-Netzwerke in Verbindung mit dem Internet eingegangen.

Die ersten Techniken betreffen den OSI-Layer 2, den „Data-Link Layer“. Diese Schicht regelt die Kommunikation von Netzwerkgeräten, die sich ein physisches Medium teilen. Im Falle von Ethernet ist das also ein „Segment“ oder der Teil des Netzwerkes bis zum ersten Router (der verschiedene Netze miteinander verbindet). Auf diesem Layer werden Netzwerkgeräte mit ihrer MAC-Adresse angesprochen. Dies ist eine 48-Bit breite, weltweit eindeutige, Adresse. Besonders in WLANs ist sogenanntes MAC-Filtering gehäuft anzutreffen. Dabei wird nur Netzwerkgeräten mit bestimmten MAC-Adressen der Zugriff auf das Netzwerk gestattet. Das ist bei WLANs tendenziell wichtiger als bei kabelgebundenen Netzwerken, da bei jenen der Zugriff auf das Netzwerk viel einfacher ist. Diese sehr zweifelhafte „Sicherheitsmethode“ lässt sich durch **MAC-Spoofing** leicht aushebeln, denn moderne Netzwerkkarten erlauben es, ihre MAC-Adresse temporär zu ändern. Durch vorheriges Mitlauschen im Netz können „gültige“ MAC-Adressen schnell in Erfahrung gebracht werden. MAC-Adressen-Filterung wird im professionellen Umfeld schon allein wegen der schlechten Wartbarkeit dieser Filterlisten selten eingesetzt.

Im Internet regelt das ARP-Protokoll[26] die Auflösung zwischen MAC- und IP-Adressen. Jeder Rechner, der über ein Ethernet am Internet angeschlossen ist, verwaltet intern eine ARP-Tabelle, die die Zuordnung von MAC- zu IP-Adressen speichert. Diese Zuordnungen gelten eine gewisse Zeit lang, danach werden sie erneuert. Das ARP-Protokoll kennt dafür Protokoll-Nachrichten wie „Wer ist MAC-Adresse 00:bf:9f:11:32:bb“ oder „Ich habe MAC-Adresse 00:bf:9f:11:32:bb und meine IP-Adresse lautet 192.168.0.1“. Beim **ARP-Spoofing** wird diese Tabelle „vergiftet“ (*Cache poisoning*) und in ihr durch gefälschte ARP-Nachrichten Einträge ersetzt. So kann man problemlos den Verkehr eines Rechners zum Gateway umleiten und analysieren, ohne dass der Angegriffene dies bemerkt.

[25] Das ISO-OSI-Modell ist ein Standardmodell in der Netzwerktechnik, um die Abstraktionsebenen in einem Netzwerk zu beschreiben. Der außerordentlichen Komplexität von Netzwerken wird mit Abstraktion in Schichten („Layers“) begegnet. Das ISO-OSI-Modell hat 7 Schichten, beginnend beim Layer 1, dem „Physical Layer“, hinauf zum Layer 7, dem „Application Layer“ (manche sehen scherzhaft auch noch einen Layer 8, der ebenfalls für viele Fehler verantwortlich sein kann: den „Human Layer“).

[26] ARP – Address Resolution Protocol

(Damit verwandt ist das **ARP-Flooding**: dabei werden Netzwerkgeräte wie Switches, die Netzwerkverkehr aus Performancegründen nur zu den adressierten Teilnehmern durchschalten, dazu überredet, den gesamten Verkehr und nicht nur den für den Host bestimmten Verkehr weiterzuleiten).

Nachdem sich Layer 2 um die Kommunikation innerhalb eines Segments kümmert, kümmert sich Layer 3 darum, wie verschiedene Segmente miteinander kommunizieren können. Auf das Internet umgelegt heißt das „Routing", das Verbinden verschiedener Netze miteinander. Hier tauchen IP-Adressen erstmals auf. Auch hier kann man einiges tun. Mit **IP-Spoofing** werden Datenpakete verfälscht, so dass sie von einem anderen Absender zu kommen scheinen. **DHCP-Spoofing** manipuliert das *Dynamic Host Configuration Protocol*, mit dem Rechner automatisch IP-Adressen zugeordnet bekommen dergestalt, dass diese ihren gesamten Verkehr über einen Rechner des Angreifers senden, indem ihnen ein anderes Netzwerk-Gateway zugewiesen wird.

Layer 3 regelt den Transport einzelner Datenpakete quer durch das Internet, gibt aber weder Garantien für die Zustellung ab oder für die Reihenfolge zugestellter Pakete. Dies wird auf Layer 4 bewerkstelligt, besonders durch das TCP-Protokoll („Transmission Control Protocol"). Viele wichtige Protokolle basieren auf dem TCP-Protokoll, im Wesentlichen alle, bei denen die Zustellung von Datenpaketen garantiert sein muss (bzw. verlorengegangene Pakete automatisch erkannt und nochmals zugestellt werden) und diese in der selben Reihenfolge ankommen müssen in der sie abgesendet wurden. Das trifft zu auf den gesamten Verkehr im WWW und auf den Mail-Verkehr, um die beiden wichtigsten zu nennen. Auch kennt TCP das Konzept der „Verbindung" (*state*), eine Verbindung wird für eine Kommunikation auf- und wieder abgebaut. TCP bewerkstelligt dies durch sogenannte Sequence Numbers, das sind im Wesentlichen einfache Zähler, bei denen die eine Seite der anderen Seite mitteilt, wie viele Bytes bereits erfolgreich übertragen wurden. Ein Angriff auf diesem Layer ist das **TCP-Hijacking**, also das „Entführen" einer bereits bestehenden TCP-Verbindung. Dazu müssen die nächsten Sequence Numbers richtig berechnet werden. Durch Optimierungsmaßnahmen in TCP (Sliding window, Nagle's Algorithm, etc.) und bewusste Gegenmaßnahmen gegen diesen Typ von Angriff (Sequence Number Randomization) ist das alles andere als trivial.

Für den Bereich des Internets werden die OSI-Layer 5-7 zusammen betrachtet. Auch auf diesem „Application Layer" gibt es eine Vielzahl von Spoofing-Angriffen.

DNS-Spoofing manipuliert die Namensauflösung von Domainnamen zu IP-Adressen entweder durch eingeschleuste Datenpakete oder direkt auf dem Rechner des Opfers (*/etc/hosts*), wodurch z.B. www.netbanking.at nicht mehr auf das E-Banking der Erste Bank aufgelöst wird, sondern auf einen Rechner des Angreifers, der so in Besitz der Login-Daten des Opfers kommt. DNS-Pakete lassen sich besonders leicht fälschen, da DNS nicht über TCP, sondern UDP kommuniziert. UDP ist der (sehr) kleine Bruder von TCP und bietet fast nichts von dem was TCP bietet: keine garantierte Paketzustellung, kein Erhalt der Reihenfolge, kein Konzept von Verbindungen (*state-less*), sondern im Grunde ist UDP nur IP plus Portnummern[27]. Dafür ist aber auch der Kommunikationsaufwand viel geringer. DNS verwendet ebenfalls ein recht altes Protokoll, das noch nicht wirklich mit Sicherheitsproblematiken im Hinterkopf entworfen wurde. Abhilfe soll das mit Kryptographie verstärkte DNSSEC schaffen[28].

Nicht zuletzt im Web-Bereich sehr verbreitet ist das **URL-Spoofing**, bei dem Links so getarnt werden, dass sie auf den ersten Blick für das ungeübte Auge korrekt aussehen, in Wirklichkeit jedoch auf einen Server des Angreifers zeigen, z.B. http://www.netbanking.at@131.130.1.78/ (moderne Browser geben hier eine Warnung aus). Benutzer sind durch die zahlreichen verwirrenden

[27] http://www.networksorcery.com/enp/protocol/udp.htm
[28] http://www.dnssec.net/

Meldungen ihrer Programme, denen sie täglich ausgesetzt sind, in gewisser Weise trainiert, technische Informationen, die sie nicht verstehen, zu ignorieren. Noch viel besser sind sie durch die kryptische Darstellung von GET-Parametern in URLs im Browser-Fenster darauf trainiert, seltsamen Zeichenfolgen hinter dem Domainnamen keine Beachtung zu schenken. Der durchschnittliche User versteht die obige Adresse als „Netbanking und noch ein paar Zeichen dahinter“ und wird sich nichts dabei denken. In Wirklichkeit aber führt diese Adresse nicht zu www.netbanking.at, sondern zu 131.130.1.78, das ein System des Angreifers sein kann. Wegen der Gefährlichkeit dieses URL-Spoofings geben moderne Browser bereits Warnungen aus, wenn solche URLs erkannt werden. Darauf sollte man sich allerdings nicht zu sehr verlassen, ein neuer, kreativer Ansatz, den User zu täuschen, wird sicher bald gefunden werden.

Man-in-the-middle

Man-in-the-middle bedeutet, dass ein Dritter eine Datenverbindung automatisiert abhört, die empfangenen Datenpakete belauscht, gegebenenfalls manipuliert, und an den wahren Empfänger weiterleitet bzw. diese Manipulation auch in die Gegenrichtung ausführt. Im Gegensatz zum reinen Sniffing, bei dem nur gelauscht wird, ist hier der Aufwand höher, z.B. weil es sich um einen verschlüsselten Datenstrom handelt, der zuerst dekodiert werden muss, dann gefiltert, manipuliert, danach wieder verschlüsselt und weitergeleitet. Um dies zu verhindern, müssen Verschlüsselungsprotokolle zwingend Maßnahmen vorsehen, mit denen die Gegenseite kryptografisch sicher identifizieren werden kann. Das wird gerne unterschätzt: warum soll ich mich um die Identität meines Kommunikationspartners kümmern, die Daten sind ja verschlüsselt! Aber gerade *weil* verschlüsselte Daten übertragen werden, die normalerweise sensiblere und geheimere Daten sind als solche, die im Klartext übertragen werden (orf.at vs. E-Banking), ist das Interesse viel höher zu wissen, *wer* eigentlich mein Kommunikationspartner ist, dem ich den Datenstrom schicke!

Für das Internet hat hier die Firma Netscape eine Vorreiterrolle eingenommen und mit SSL einen Standard etabliert, der starke Verschlüsselung und Authentifizierung bieten kann. Hier wird Verschlüsselung durch asymmetrisch/symmetrisch gemischte Verfahren erreicht. Datenintegrität und nicht-Verfälschbarkeit durch (signierte) Prüfsummen. Nicht-Abstreitbarkeit durch digitale Signaturen, und nicht zuletzt Vertrauen in die Identität des Kommunikationspartners durch den Einsatz von „Zertifikaten“. Ein Zertifikat ist die Bestätigung einer übergeordneten, vertrauenswürdigen Stelle, dass der Kommunikationspartner tatsächlich der ist, für den er sich ausgibt. Dies wird durch digitales Signieren des fremden Public Keys erreicht sowie durch Vergleich des im Zertifikat gespeicherten Domainnamens mit dem angesurften Domainnamen. Dieses Zertifikat kann wiederum von jemand anderem signiert werden, wodurch eine „Chain of trust“ bis zu einigen wenigen, besonders vertrauensvollen, Stellen hergestellt wird. Auf die genaueren Details dieser sehr komplexen Materie kann hier leider nicht im Detail eingegangen werden.

Allerdings: auch Verschlüsselung und Zertifikate bieten keinen absoluten Schutz gegen einen Lauschangriff, denn ist der Private Key des Empfängers bekannt, kann sich ein Man-in-the-middle als dieser ausweisen und den Verkehr mitlesen, ohne dass der Sender dies bemerken kann (*Private key compromise*).

Trojaner, Viren und Würmer

Viren und Würmer zählen zur ältesten Kategorie sogenannter Malware, also Schadprogrammen, sie sind aber nach wie vor aktuell. Viren zeichnet aus, dass sie sich – genauso wie ihre biologischen Pendants – verbreiten können und auf neuen infizierten Systemen Schaden anrichten können. Manche Viren begnügen sich mit der reinen Verbreitung, während andere Viren gezielt auf

größtmöglichen Schaden ausgerichtet sind, z.B. Löschen aller Daten auf der Festplatte. Für den Virus ist aber es, wie in der Biologie, von Vorteil, wenn der Wirt so lange wie möglich nichts von der Infektion mitbekommt, so dass der Virus sein Programm ungestört ablaufen lassen kann. Sie können dabei z.B. **Rootkits, Backdoors** oder **Keylogger** installieren, um möglichst viel über den Wirten zu erfahren und die Verbreitungschancen zu erhöhen, oder sich in einem **Botnet** anmelden und auf Befehle warten.

Der Unterschied zu Würmern ist subtil und fließend. Ein Virus kann sich normalerweise nur dann verbreiten, wenn sein Schadprogramm durch eine Benutzeraktion oder ein Ereignis von außen angestoßen wird, er wartet also passiv auf seine Aktivierung. In Zeiten, in denen Computer noch so gut wie gar nicht vernetzt waren, war der einzige Übertragungsweg für einen Virus ein Datenträger, den ein Benutzer aktiv von einem Computer zum anderen tragen musste (Boot-Sector-Virus etc.). Würmer sind hier aggressiver, in dem sie sich aktiv versuchen, zu verbreiten, indem sie z.B. das Adressbuch des Benutzers auslesen und sich gezielt an die dort gefundenen Adressen per E-Mail weitverversenden. Der Benutzer muss zur Verbreitung gar nichts mehr dazutun!

Der erste Computerwurm, der weltweite Aufmerksamkeit erregte, war der im Mai 2000 ausgebrochene Loveletter-Wurm (auch bekannt als „I-love-you-Wurm“). Dies war der erste Wurm, der sich per E-Mail explosionsartig verbreitete und richtete Schäden in Milliardenhöhe an. Dies war auch der Beginn einer wahren Serie von Sicherheitsdebakeln in Microsoft-Produkten. Als Reaktion darauf hat der Konzern seine Sicherheits-Strategie grundlegend geändert und sich verstärkt auf dieses Thema konzentriert. Die Betriebssysteme Windows XP und Windows Vista können zu Recht als überdurchschnittlich sicher gelten. Dennoch werden immer wieder Sicherheitslücken in ebendiesen Betriebssystemen bekannt, dies liegt jedoch auch an dem extrem hohen Marktanteil dieser Produkte (Home-Bereich: ca. 90%): es zahlt sich nun mal aus, in genau diesen Produkten nach Sicherheitslücken zu suchen, denn wenn eine gefunden wird, hat sie einen umso größeren Impact.

Es kann also durchaus sinnvoll sein, auf besonders sicherheitskritischen Netzwerkgeräten (z.B. Firewalls) relativ exotische Betriebssysteme und/oder Hardwareplattformen einzusetzen, z.B. OpenBSD auf Alpha. Selbst wenn hier eine Sicherheitslücke bekannt wird, wird sie dadurch noch schwieriger auszunutzen, was die Lösung noch um einige Zehnerpotenzen sicherer macht.

Viren- und Würmerautoren werden immer kreativer. So ist der im Jänner 2009 ausgebrochene Wurm „Conficker“ einer der funktionsreichsten Würmer überhaupt. Zur Zeit der Bearbeitung dieser Seminararbeit breitet sich der Wurm aktiv weiter aus und hat bereits über 2,5 Millionen PCs infiziert[29]. In Österreich hat dieser Wurm besonders in Kärnten zugeschlagen, denn Anfang Jänner 2009 wurden über 3000 PCs der Kärntner Landesregierung infiziert, die dadurch tagelang unbrauchbar gemacht wurden. Auch die Kärntner Krankenanstaltengesellschaft KABEG wurde von Conficker heimgesucht. Eingeschleppt wurde er wahrscheinlich durch einen einzelnen, infizierten USB-Stick[30].

Trojanische Pferde werden häufig im Zusammenhang mit Viren und Würmern erwähnt. Dabei handelt es sich um vordergründig harmlos „aussehende“ Programme, die der Benutzer nichts ahnend installiert, die jedoch auch Schadprogramme beinhalten, beispielsweise eben Viren oder Würmer. Die Analogie zum „Trojanischen Pferd“ aus der Mythologie ist augenfällig.

[29] http://www.heise.de/newsticker/Studie-2-5-Millionen-PCs-mit-Conficker-Wurm-infiziert--/meldung/121684
[30] http://www.heise.de/security/Conficker-in-Kaernten-Nach-der-Landesregierung-nun-die-Spitaeler--/news/meldung/121570

Social Engineering / Social Hacking

Eine untechnische Methode, um an geheime Informationen zu gelangen, ist das sogenannte Social Engineering. Sie ist eng mit dem Betrug verwandt, wobei eine Person bewusst getäuscht wird, so dass sich diese irrt (z.B. über die Identität des Angreifers) und Informationen preisgibt. Das klassische Beispiel ist der Anruf Sonntag Abends (Bereitschaft) bei einem Systemadministrator mit der Bitte, das Passwort auf einen bestimmten Wert zurückzusetzen, weil man auf dringend benötigte Dateien für das Meeting Montag früh zugreifen müsse.

Zur wahren Meisterschaft in dieser Disziplin brachte es der Hacker *Kevin Mitnick* in den 80er-Jahren. Er war zu dieser Zeit eine der meistgesuchten Personen in den USA. Mitnick verbüßte eine 5-jährige Haftstrafe wegen seiner Aktivitäten und ist mittlerweile IT-Sicherheitsberater. Nach seinem Dafürhalten ist Social Engineering die bei weitestem effektivste Methode, um an Passwörter zu gelangen. Das „Standardwerk" zum Social Engineering ist sein Buch *„The Art of Deception: Controlling the Human Element of Security"*[31].

Social Engineering ist auch in der Probe-Phase nützlich, um an Informationen über das Opfer zu gelangen.

Sicherheitslücken auf Systemebene

Obwohl im Web-Bereich kaum noch relevant, möchte ich hier auch das große Themengebiet rund um Sicherheitslücken auf Systemebene ansprechen.

Programmiersprachen wurden im Laufe der Zeit immer komplexer; man spricht auch von „Generationen" von Programmiersprachen. Zu Beginn wurde mehr oder weniger in Binärcode programmiert, danach brachten Assembler bereits einige Unterstützung in Form von Mnemonics, Sprungmarken usw. Man ist hier aber immer noch sehr „nah an der Maschine" dran, weshalb nur relativ einfache und/oder kurze Programmabschnitte sinnvoll in Assembler programmiert werden, heutzutage überhaupt nur noch wenn es um absolute Performance geht (so sind die zentralsten und zeitkritischsten Komponenten moderner Betriebssysteme immer noch in Assembler geschrieben, z.B. der Interrupt-Handler in Linux). Die nächste Abstraktionsstufe waren prozedurale Sprachen wie C, in denen immer noch sehr hardwarenah, aber auch schon mit einem gewissen Komfort programmiert werden kann. Diese Sprachen werden auch Programmiersprachen der 3. Generation genannt. C ist ein überaus erfolgreicher Vertreter dieser Spezies. Sehr viele der modernen Programmiersprachen basieren auf C, haben viel von seiner Syntax übernommen, und im hardwarenahen Bereich ist C immer noch die *lingua franca* der Informatik. Das Betriebssystem Linux ist vollständig in C geschrieben. In der Windows-Welt der 90er-Jahre ist C++, die objektorientierte Weiterentwicklung von C, die vorherrschende Sprache. Was die Systemnähe betrifft ist C++ im Wesentlichen gleich wie C, weshalb sich beide Welten die nachfolgend beschriebenen Probleme teilen.

In den 70er-, 80er- und 90er-Jahren begegnet uns in der Informatik also eine Welt, die hauptsächlich aus auf C basierenden Programmen besteht.

Buffer Overflows

Die erste große und älteste Gruppe von Sicherheitslücken, die speziell systemnahe Sprachen wie C betreffen, sind die sogenannten Buffer Overflows. Sie alle nutzen die Eigenart aus, wie Programme von Betriebssystemen in den Speicher geladen und dort ausgeführt werden. Die von-Neumann-Architektur, nach der alle heutigen Computer aufgebaut sind, bringt es mit sich, dass Programmcode und Daten in einem *gemeinsamen* Speicher (dem Arbeitsspeicher) geladen und dort

[31] Kevin Mitnick: „The Art of Deception: Controlling the Human Element of Security", Wiley & Sons, ISBN 0471237124

zur Ausführung gebracht werden. Der sogenannte Stack wird im selben Segment abgelegt wie Daten, wächst nur in die entgegengesetzte Richtung. Grob gesagt[32] besteht die Sicherheitslücke darin, dass von außen kommende Daten den Stack überschreiben können, weil Längenlimits nicht korrekt abgefragt werden. Auf dem Stack befindet sich auch die Rücksprungadresse von Subroutinen, die somit überschritten wird, und dadurch der Angreifer nun die Kontrolle über die weitere Ausführung des Programms erlangt hat.

Auch weitere Gruppen von Sicherheitslücken wie **Format String Bugs** und **Heap Overflows** basieren auf dem Prinzip, dass ein Angreifer von außen den Stack oder den Datenspeicher so manipuliert, dass er in Folge die Kontrolle über den Programmfluss erhält. Bei all diesen Lücken muss der Angreifer den einzuschleusenden Schadcode (*Shell* Code) oft selbst in Assembler erstellen und dabei diverse Schwierigkeiten überwinden wie z.B. begrenzter Platz im Speicher oder Weglassen gefilterter Zeichen (Alphanumeric Shellcode[33], UTF-8-compatible Shellcode[34] für XML).

In Webapplikationen werden derart systemnahe Sprachen so gut wie gar nicht mehr eingesetzt. Stattdessen werden *interpretierte Sprachen* – Perl, PHP, Ruby, Python z.B. – oder in *virtuellen Maschinen* ablaufende Sprachen – Java, C# - eingesetzt. Diese Sprachen leiden schon aufgrund ihres Aufbaues, da sie weiterführende Abstraktionen der Sprachen der dritten Generation sind, nicht unter den geschilderten Problemen. Für klassische Buffer-Overflows sind Webanwendungen daher so gut wie nicht anfällig. Weiterhin in C bzw. C++ sind aber die allermeisten Server geschrieben, wie Webserver (Apache, IIS), Mailserver (Sendmail, qmail, Postfix, Exim, ...), DNS-Server (Bind), weshalb auch hier ab und zu noch ein Fehler dieser Art auftaucht. Moderne Betriebssysteme nutzen allerdings auch schon Features moderner Hardware aus, die es für einen Angreifer unmöglich machen, eine derartige Sicherheitslücke auszunutzen, selbst wenn sie existiert (Stichwort *non-executable Stack*).

Persist

Die Techniken dieser Phase dienen dazu, den durch den Angriff erlangten Zugriff auf das Computersystem auch für die Zukunft aufrechtzuerhalten.

Backdoors und Rootkits

Backdoors, also versteckte „Hintertürchen“, können entweder nach dem erfolgreichen Angriff installiert werden oder sind sogar schon getarnt in installierter Software vorhanden. Über sie kann der Angreifer jederzeit wieder Zugriff auf das System erhalten, er muss also einen Angriff nicht nochmal ausführen.

Starke Verbreitung finden momentan sogenannte Rootkits, die fast immer auch Backdoors mit beinhalten. Rootkits „lösen“ die sich für den Angreifer stellende Problematik, bei seinem Zugriff auf das System nicht aufzufallen, indem sie z.B. vom Angreifer gestartete Prozesse oder angelegte Dateien verstecken, Datei-Zeitstempel nicht schreiben etc. Rootkits installieren sich sehr tief im System, so dass sie es beispielsweise auch verhindern können, dass sie von Rootkit-Scannern, die es freilich auch gibt, erkannt und entfernt werden. Ein **Keylogger** ist ebenfalls in so gut wie jedem Rootkit vorhanden, der alle Tastatureingaben des Benutzers, insbesondere Login-Daten, mitspeichert. Hier gibt es unglaublich ausgereifte und funktionsreiche Rootkits, die Angreifer

[32] Auf die technisch sehr interessanten Einzelheiten kann hier leider nicht genauer eingegangen werden, zur Lektüre und als Einstieg empfiehlt sich aber das „Standardwerk“ in dieser Hinsicht: http://www.phrack.org/issues.html?issue=49&id=14#article

[33] http://www.phrack.com/issues.html?issue=57&id=15#article

[34] http://www.phrack.org/issues.html?issue=62&id=9#article

einfach nur noch downloaden und auf dem kompromittierten System installieren müssen.

Propagate

Ist der Zugriff auf ein Computersystem durch die vorhergehenden Phasen relativ gesichert, gilt es, sich auszubreiten. Es wird also nach Information gesucht, welche anderen Systeme gehackt werden können, und die Phasen beginnen von neuem. Dabei werden bereits vorgestellte Techniken eingesetzt: **Sniffing**, um im Netzwerk nach Passwörtern zu lauschen, und **Spoofing**, um die Ausbeute des Sniffings zu erhöhen. **Rootkits** helfen durch Keylogging ebenfalls dabei, an Passwörter für neue Maschinen zu gelangen. **Viren** und **Würmer** werden eingesetzt um noch weitere Verbreitung zu erreichen.

Gesondert erwähnen möchte ich an dieser Stelle die sogenannten **Botnets**. Hierbei handelt es sich um logische Netzwerke gehackter Systeme, die der Kontrolle eines Einzelnen oder einer kleiner Gruppe von Personen unterstehen. Über verschlüsselte Kommunikation können diese den *Bots* oder *Zombies* Befehle erteilen, z.B. um eine **Distributed Denial-of-Service-Attacke** auszuführen (siehe unten). Es existieren bereits Botnetze mit riesigen Ausmaßen (1 Million Bots und mehr) und entsprechendem Schädigungspotential.

Paralyze

Gerade bei Schädigungsabsicht ist auch die Paralyze-Phase wichtig. Sie fasst Techniken zusammen, die das Opfer „lähmen“ sollen (engl. t*o paralyze*).

(Distributed) Denial-of-Service

Denial-of-Service-Attacken (kurz DoS-Attacken) sind solche Attacken, mit denen ein Dienst eines Anbieters unbrauchbar gemacht werden soll, so dass er diesen Dienst nicht mehr erreicht werden kann und dadurch eine entsprechende Schädigung eintreten soll. Dies wird z.B. erreicht, in dem Programmfehler ausgenutzt werden, so war es bei einer frühen Version von Windows 98 möglich, durch ein einziges, 64 kB großes Datenpaket bestimmter Beschaffenheit das System zum Absturz zu bringen. Wurde dieses Paket alle paar Minuten ausgesandt hatte das zur Folge, dass der Rechner am Netzwerk nicht mehr einsatzfähig war (ohne entsprechende Filterung; zu dieser Zeit noch nicht weit verbreitet; sogenannter *Ping of Death*). DoS-Attacken können aber auch durch gezielte Überlastung eines Dienstes durch eine große Anzahl von Anfragen in kurzer Zeit ausgeführt werden.

Ein Beispiel für so einen Angriff ist das **SYN-Flooding**. TCP-Verbindungen werden über den sogenannten 3-way-handshake aufgebaut. Der verbindungsaufbauende Teil sendet zuerst ein SYN-Paket, der Empfänger sendet daraufhin (sofern er die Verbindung zulässt) ein SYN+ACK, was der erste Kommunikationspartner mit einem ACK bestätigt. Die Verbindung gilt nun als aufgebaut. Beim SYN-Flooding wird nun einfach eine sehr große Anzahl von SYN-Paketen an einen Server gesendet, ohne um sich um die Antwort zu kümmern. Der Server sendet nun eine ebenso große Anzahl von SYN+ACK-Paketen zurück, und muss sich aber intern merken, dass hier eine Verbindung im Entstehen ist, sonst kann er das später eintreffende ACK-Paket des Clients nicht zuordnen. Das bedeutet, dass für jedes SYN-Paket eines Clients irgendwo ein Vermerk geführt werden muss. Die Anzahl dieser Vermerke ist begrenzt, ein Server kann nicht unendlich viele Verbindungen gleichzeitig pflegen. Durch SYN-Flooding wird also erreicht, dass dieser Vermerkspeicher stets gefüllt ist, wodurch legitime Verbindungen nur mehr erschwert oder gar nicht mehr aufgebaut werden können. Eine Abhilfe für dieses Problem ist z.B. der Einsatz von SYN-Cookies.

Auch das Verstärken von Netzwerkverkehr (**Traffic Amplification, Smurf Attacks**) ist eine Möglichkeit um ein Angriffsziel in die Knie zu zwingen. Dabei macht sich der Angreifer die Eigenart eines Dienstes zu nutze, dass dieser mehr Daten zurückschickt als ihm Daten gesendet werden. Zum Beispiel eine 200 Bytes lange Fehlermeldung für eine Anfrage von 10 Bytes. Das an sich wäre noch kein Problem, denn damit würde sich der Angreifer nur selbst blockieren. Interessant wird es, wenn der „verstärkende“ Dienst so manipuliert werden kann, dass er die Fehlermeldung nicht an den Angreifer, sondern an ein anderes Opfer schickt – hier kann der Angreifer mit einem Verstärkungsfaktor von 20 die Leitung zum Opfer blockieren, eine entsprechend potente Anbindung des Dienstes vorausgesetzt.

Eine frühe Version des MS-SQL-Servers zeigte hier ein besonderes Verhalten. Wenn dieser eine fehlerhafte Anfrage bekam, sendete er diese an den Anfragenden zurück und fügte eine eigene Meldung hinzu. Diese Kommunikation fand über UDP statt, so dass die Absenderadresse sehr einfach zu fälschen war. Hatte man nun zwei MS-SQL-Server, genügte es ein einziges (!) fehlerhaftes Datenpaket mit der Absenderadresse des anderen Servers an den ersten Server senden. Dieser generierte eine Fehlermeldung und sendete diese an den vermeintlichen Absender zurück – an den zweiten SQL-Server. Dieser bekam die Anfrage, generierte eine Fehlermeldung, und sendete diese an den ersten Server zurück, weil dies ja der Absender war. Dieses Spiel ging endlos weiter, und in kürzester Zeit waren beide Server bzw. die Leitung zwischen ihnen lahmgelegt.

Auch im Bereich des E-Mailens ist so ein Angriff denkbar; jedoch wird das von allen Mailservern als *Double-Bounce* erkannt und rechtzeitig unterbunden.

Solange diese Angriffe von einem einzigen oder einigen wenigen Angreifern ausgehen, kann man ihnen dadurch Herr werden, dass diese gesperrt werden. Eine neue Dimension von DoS wurde allerdings mit Distributed-Denial-of-Service erreicht, die vor allem mit **Botnets** durchgeführt werden. Dabei wird den am Botnet teilnehmenden Zombies der Befehl erteilt, Anfragen in großer Zahl gegen das Opfer abzusetzen, was bei diesem innerhalb kürzester Zeit durch die schiere Anzahl an Anfragen in die Knie zwingt. Diese Anfragen auszufiltern und von gewünschten „echten“ Anfragen zu unterscheiden ist praktisch unmöglich, da sie von allen möglichen Ecken des Internets kommen, wenn das Botnet bereits eine entsprechende Verbreitung gefunden hat.

Abkürzungsverzeichnis

ACK	Acknowledge (TCP-Flag)
ARP	Address Resolution Protocol
CVC	Card Verification Code
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
DDoS	Distributed Denial of Service
dStGB	Deutsches Strafgesetzbuch
FAQ	Frequently Asked Questions
FTC	Federal Trade Commission
HTTP	Hypertext Transfer Protocol
IOS	Internetwork Operating System
ISO	International Organization for Standardization
IP	Internet Protocol
ISDN	Integrated Services Digital Network
IT	Informationstechnologie
LAN	Local Area Network
LIR	Local Internet Registry
MAC	Media Access
MTA	Mail Transfer Agent
MUA	Mail User Agent
OSI	Open Systems Interconnection
P2P	Peer-to-Peer
POP3	Post Office Protocol 3
RIR	Regional Internet Registry
SMTP	Simple Mail Transfer Protocol
SPF	Sender Policy Framework
SQL	Structured Query Language
ssh	Secure Shell
SSL	Secure Sockets Layer
SYN	Synchronize (TCP-Flag)
TCP	Transport Control Protocol

TLS	Transport Layer Security
TOR	The Onion Router
UDP	User Datagram Protocol
URL	Uniform Resource Locator
WEP	Wired Equivalent Privacy
WLAN	Wireless LAN
WPA	Wi-Fi Protected Access
WWW	World Wide Web
XSS	Cross-Site-Scripting