2 Gefährdungspotentiale, Hacking und Schutzmaßnahmen

In den letzten Jahren wurden immer mehr private Netze mit dem Internet verbunden. Der Trend, sich im Internet zu präsentieren, Informationen anzubieten, sowie diverse im Internet vorhandene Dienste zu nutzen, hält unvermindert an. Der Zusammenschluß vieler unterschiedlicher Netz über ein öffentliches Internet bringt aber auch verschiedene Sicherheitsrisiken mit sich.
Die folgende Einführung versucht, die wichtigsten Aspekte der IT-Sicherheit anzusprechen und eine Sensibilisierung für das Thema im Rahmen dieses Praktikums hervorzurufen. Dabei werden zuerst die Gefahren, welche von einer Rechnervernetzung ausgehen können, und dann die Maßnahmen zum Schutz der Rechnerressourcen aufgezeigt. Abschließend stellen wir noch einige Werkzeuge zur Durchführung von Angriffen vor.

2.1 Risiken

Daten und die ständige Verfügbarkeit von Informationen bekommen in der heutigen Zeit eine immer größere Bedeutung. Die Risiken, die eine Anbindung des Unternehmensnetzwerkes an ein öffentliches Netz wie das Internet mit sich bringen, können im Wesentlichen in drei Gruppen eingeteilt werden:

- **Diebstahl** von Daten, Know-How oder strategischen Plänen
- **Veränderung** und Fälschung von Daten
- **Blockieren** von kritischen Rechnersystemen

Je nach Bedeutung der schützenswerten Daten oder deren Verfügbarkeit kann der Schaden bei Eintritt einer dieser Fälle beträchtliche Ausmaße annehmen.

2.2 Angreifer

Die Angriffe können dabei von Personen und Organisationen mit unterschiedlichen Interessen und Ressourcen ausgehen:

- **Spione** mit sehr hoher Qualifikation und genauen Kenntnissen über das Angriffsziel. Diese Gruppe verfügt in der Regel auch über alle benötigten Ressourcen zur Durchführung eines effizienten Angriffes. Gründe für einen Angriff können finanzielle Interessen oder die Erreichung von Wettbewerbsvorteilen für die eigene Organisation sein.

- **Staaten und politische Organisationen** verfügen in der Regel über ähnlich gute Vorausetzungen für die Durchführung effizienter Angriffe, die Motivation liegt hier mehr im politischen Bereich.
- **Terroristische Organisationen** können aus religiösen oder politischen Gründen ebenso mit starken Ressourcen Angriffe im Netz starten.
- **Externe Einzelpersonen** können bei entsprechender finanzieller Motivation oder aus Machtinteressen auch beträchtliche Mittel und Qualifikationen aufbringen.
- **Personen innerhalb der Organisation** wie Angestellte oder externe Mitarbeiter verfügen über eine mehr oder weniger genaue Kenntnis der Infrastruktur sowie über gewisse Zugriffsrechte auf interne Rechnersysteme. Das vorhandene Fachwissen sowie die verfügbaren Ressourcen können sehr gut sein. Entsprechend hoch ist auch das Gefahrenpotential bei Angriffen aus den eigenen Reihen. Als Motivation kommen z.B. finanzielle Interessen oder Rache von unzufriedenen Mitarbeitern in Frage (interne Angriffe, siehe Seite 39).

Der breit gefaßte Begriff **Hacker** beinhaltet unterschiedliche Charaktere.
Sogenannte *Script Kiddies* oder *Wannabes* bringen wenig Know-How oder technische Ressourcen mit und sind oft nur neugierig auf die Möglichkeiten der Technologie oder wollen Aufmerksamkeit auf sich ziehen. Sie sind auf die Verwendung von vorhandenen Werkzeugen (Scanner, Exploits, Virus Constrution Sets usw.) angewiesen. Da ihnen aber meistens viel Zeit zur Verfügung steht und sie sich über die Ausmaße ihres Handelns nicht bewußt sind, gehen von ihnen beträchtliche Gefahren aus.
Black Hats bringen beträchtliches Wissen mit und brechen in Systeme ein, um sich Geltung und Anerkennung zu verschaffen.
Als *White Hats* werden die "guten" Hacker bezeichnet, welche keine bestimmten Systeme angreifen, sondern nur auf Sicherheitslücken und auf prinzipielle Einbruchsmöglichkeiten hinweisen wollen. Leider werden die von diesen hochqualifizierten Personen geschriebenen Werkzeuge auch von anderen Gruppen zu weniger noblen Zwecken verwendet.
Zuletzt findet man noch die sogenannten *Hacktivists*, die ihrer persönlichen politischen Überzeugung durch Angriffe im Netz Aufmerksamkeit verschaffen wollen.
Eine genaue Definition vieler Begriffe aus der "Szene" gibt es in den *Jargon Files* unter [Raym 01].

2.3 Informationsbeschaffung

Zur Informationsbeschaffung über die anzugreifenden Systeme können in einem ersten Schritt öffentlich zugängliche Informationsquellen wie Software-Dokumentationen, technische Literatur, Diskussionsforen, WHOIS-Datenbanken oder Internet-Seiten durchsucht werden. Dort finden sich z.B. Informationen zu Netzwerken und Systemen einer Organisation oder Hinweise auf Schwachstellen und Fehler in unterschiedlichen Programmen. Beispiele sind fehlerhafte Behandlungen von unerwarteten Zeichenfolgen oder Datenpaketen,

Puffer-Überlauf (*Buffer Overflow*)-Fehler, absichtlich eingebaute Hintertüren (*Trapdoors*) oder einfach nicht vorhandene Sicherheitsmaßnahmen in Programmen und Protokollen.
Eine weitere, nicht zu unterschätzende Gefahr geht vom sogenannten *Social Engineering* aus, also dem Sammeln von Informationen über persönliche Kontakte mit Menschen aus der anzugreifenden Organisation. Die Bandbreite reicht dabei vom Belauschen von Unterhaltungen in öffentlichen Räumen oder einem unscheinbaren Telefonanruf über die Vorgabe falscher Identitäten hin zu Bestechung und Bedrohung von Mitgliedern der Ziel-Organisation.
Natürlich spielen in diesem Zusammenhang auch die klassischen Methoden der Zugangsbeschaffung eine Rolle: Diebstahl von Authentifizierungsobjekten oder Einbruch.
Auf technischer Seite stehen dem Angreifen spezielle *Scan-Programme* zur Untersuchung der anzugreifenden Netzumgebung oder zum Auffindung von Schwachstellen in den Systemen zur Verfügung. Unter Ausnutzung der gefundenen Schwachstellen kann er dann weitere Informationen über die Systeme sammeln. Als Beispiele genannt seien hier das Verschicken von E-Mails mit gefälschtem Absender, das Abziehen der DNS-Informationen über DNS-Zonen-Transfers, das *Abhören* von unverschlüsseltem Datenverkehr von und zum Ziel-Netzwerk oder der Zugriff auf versehentlich *ungeschützte Dienste oder Bereiche* der Zielsysteme.
Bei *Man-in-the-Middle*-Angriffen wird in den Kommunikationspfad vom Client zum Server ein Fremdrechner eingebracht, über welchen sämtliche Kommunikation abgehört werden kann. Mit gefälschten Zertifikaten ist Letzteres bei unachtsamen Anwendern sogar für verschlüsselte Verbindungen möglich (Verschlüsselung siehe Seite 110).
Zu schwache Verschlüsselungen können mit entsprechenden Rechner-Ressourcen auf Angreiferseite auch entschlüsselt werden (*Brute Force/Exhaustive Key Search*, durchprobieren aller Schlüssel-Möglichkeiten oder verschiedene mathematische Algorithmen).

2.4 Angriffe

Je nach Art des Angriffes wird zwischen internen und externen sowie zwischen aktiven und autonomen (passiven) Angriffen unterschieden.
Die erste Unterscheidung zielt auf den Ausgangsort des Angriffes ab. Als intern werden diejenigen Angriffe bezeichnet, die von einem System im internen Netz aus ausgeführt werden. Externe Angriffe kommen aus nicht vertrauenswürdigen Netzen, also letzlich aus dem Internet. Von einem internen Angreifer gehen spezielle Gefahren aus, da er möglicherweisen auf viele Systeme schon Zugriff hat und detailliert über die Netzwerkstruktur Bescheid weiß. Zudem muß er zum Eindringen in die Systeme keine Firewall passieren und kann Dienste benutzen, die außerhalb des internen Netzes nicht verfügbar sind.

2.4.1 Aktive Angriffe

Aktive Angriffe werden vorsätzlich kontrolliert auf ein bestimmtes System ausgeführt.
Zum gezielten Eindringen in Systeme können die gesammelten Informationen ausreichen, um sich mit Fremd-Accounts auf normalem Wege Zugang zum System zu verschaffen. Ein interner Angreifer hat eventuell schon einen eigenen Zugang zum System. Des Weiteren können bestehende, bereits autorisierte Verbindungen vom Angreifer übernommen werden (*Session Hijacking*). Auch hier gibt es entsprechende Programme, welche z.B. die Übernahme von Telnet-Sessions ermöglichen. Ein Risiko stellen auch zu kurze und nicht statistisch unabhängig gewählte HTTP-Session-IDs bei Web-Angeboten dar.
Durch *DNS-Spoofing*, also durch Bereitstellung falscher Zuordnungen von IP-Adressen zu DNS-Namen im DNS-System (siehe Seite 145), und *IP-Spoofing* (Vorgabe einer falschen Absender-IP-Adresse, siehe Seite 89) kann der Angreifer dem System eine falsche Identität vorgeben.
Bei nicht genügend sorgfältig ausgewählten Passwörtern ist es dem Angreifer möglich, das Passwort aus gesammelten Informationen oder einfach durch Probieren zu erraten. Hilfestellung leisten hier Programme, die den Vorgang automatisieren und Passwörter nach ihrer statistischen Häufigkeit aus fertigen Wörterbüchern generieren (*Dictionary Attacks*).
Ist ein Angreifer erst mal ins System eingedrungen, kann er auf dem System weitere Informationen sammeln oder unter Ausnutzung lokaler Lücken versuchen, höhere Zugriffsrechte zu erlangen.
Viele Angreifer werden versuchen, ihr Handeln vor den Systemadministratoren zu verbergen. Dazu können sie alle Diagnosekommandos auf dem System oder gleich den Betriebssystem-Kernel durch entsprechend angepasste Versionen ersetzen (*Rootkits*), Hintertüren (*Backdoors*) dienen einem späteren Wiedereindringen ins System.
Nicht zuletzt kann auch die Netz-Infrastruktur (Stromversorgung, Kabel, Rechner usw.) angegriffen werden.

2.4.2 Denial of Service (DoS)

Ziel vieler aktiver Angriffe ist oft nur das Blockieren gewisser Dienste, sogenannte Denial of Service-Attacken. Dabei benötigt der Angreifer in der Regel keine besonderen Rechte auf dem Zielsystem. Gewisse Fehler in Programmen oder Betriebssystemen erlauben es, einen Prozeß oder das gesamte System allein durch das Zuschicken spezieller Datenpakete zum Absturz zu bringen oder die Systemlast stark ansteigen zu lassen. (z.B. mit *Ping of Death*, *WinNuke*, *Snork*, *Teardrop*)
Mit einer *SYN Flood* -Attacke kann durch Aufbau einer übermäßigen Anzahl von nicht bestätigten TCP-Verbindungen (Kommunikation wird nach Segment 2 in Abbildung 11 abgebrochen, der Server hält die Verbindung noch eine gewisse Zeit lang offen) ein Server dazu gebracht werden, keine weiteren TCP-Verbindungen mehr anzunehmen.
Viele Denial of Service-Attacks basieren auch einfach auf einer simplen Überlastung der Server- oder Netzwerkinfrastruktur. Durch das gleichzeitige Starten von Anfragen auf einer

großen Anzahl von Clients kann ein Internet-Server gezielt in die Knie gezwungen werden. Solche Angriffe sind besonders schwierig abzuwehren, da die Unterscheidung des Angriffs-Verkehrs vom Nutzverkehr nur sehr schwer möglich ist. Für solche Angriffe können auch z.B. durch einen Wurm (siehe Seite 41) kompromittierte Systeme verwendet werden, von denen aus wiederum der Angriff aufs eigentliche Zielsystem gefahren werden kann (*Distributed Denial of Service, DDoS*).
Weitere Formen von Denial of Service-Attacken sind das Vollschreiben von freigegebenen Speicherbereichen, die Überlastung von Mailservern mit einer Unmenge an E-Mails (*Spam*), die Überflutung eines Netzes mit ICMP-Anfragen (*Flood Ping* auf eine Broadcast-IP-Adresse in Verbindung mit IP-Spoofing) oder die Verbreitung falscher Routing-Informationen über dynamische Routing-Protokolle (z.B. RIP).

2.4.3 Autonome Angriffe

Autonome Angriffe werden von eingenständigen Programmen ausgeführt, welche, einmal im Netz losgelassen, selbstständig und unkontrollierbar weitere Systeme angreifen.
Die Verbreitung von autonomen Schadprogrammen in Form von Viren, Würmern und Trojanern konzentriert sich in der Regel nicht auf einzelne Organisationen.
Während *Würmer* nach dem Befall eines Rechners das Netz automatisch nach weiteren Opfern mit speziellen Sicherheitslöchern durchsuchen und diese befallen, werden *Viren* in der Regel durch Fehler in Programmen oder unachtsamen Umgang mit potentiell gefährlichen Dateien aktiviert und über lokal auf dem Rechner gespeicherte Informationen (z.B. Adressbücher) verbreitet. Während sich Viren früher vor allem über Disketten ausbreiteten, zum Beispiel im bei PCs ausführbaren Bootsektor, werden die heutigen Viren hauptsächlich in Form von ausführbaren Dateien oder als eigenständige oder in Dokumenten eingebettete Scripten über E-Mail verbreitet. Das Schadenspotential solcher Programme variiert von einer harmlosen Meldung zum Löschen von Dateien oder Sammeln (z.B. Tastaturscanner) und Verschicken von vertraulichen Informationen. Sehr beliebt ist in letzter Zeit die automatische Einwahl bei kostenpflichtigen und besonders teueren 0190er-Nummern über analoge oder ISDN-Modems (*Dialer*).
Trojaner werden zu einem ungefährlichen und oft durchaus nützlichen Programm hinzugepackt und mit diesem unerkannt auf dem System installiert und gestartet. Bei Installation und Start hat der Trojaner alle Rechte des aktuellen Benutzers und kann somit dessen Dateien beliebig verändern, löschen oder unerkannt weiterverschicken.
Nicht zu unterschätzen ist auch die beträchliche Server- und Netzwerklast, die solche Programme erzeugen können.

2.5 Schutzmaßnahmen

Der Schutz vor Gefahren aus dem Netz muß auf mehreren Ebenen erfolgen. Da ein System immer nur so gut wie seine schwächste Komponente ist, darf kein Bereich vernachlässigt werden. Die beste Firewall hilft nur wenig, wenn es den Benutzern möglich ist, sich über ein Modem einen direkten Internetzugang zu verschaffen oder wenn sie bereitwillig auf telefonische Anfrage ihre Passwörter verraten.

2.5.1 Sicherheitskonzept

Ein definiertes Sicherheitskonzept (Security Policy) ist die Voraussetzung für alle technischen Maßnahmen zum Schutz des internen Netzes.
Das Sicherheitskonzept beinhaltet neben den technischen Aspekten vor allem auch organisatorische Rahmenbedingungen. Es enthält die konkreten Ausführungsbestimmungen, die Ausarbeitung von Sicherheitszielen und generelle, von den konkreten Sicherheitszielen unabhängige Regelungen.

Bewertung und Schutzziel-Definition

Um den benötigten Umfang der Sicherheitsmaßnahmen abschätzen zu können, muß zuerst definiert werden, welchen Wert die zu schützenden Informationen und Systeme für die Organisation haben. Handelt es sich um unternehmens- und wettbewerbskritische Informationen, die keinesfalls in falsche Hände geraten dürfen oder reicht es, sicherzustellen, dass die Daten nicht verlorengehen? Der Wert kann dabei monetär bestimmbar sein oder eine schwerer zu bewertende Größe wie Image, Kundenvertrauen oder Wettbewerbsfähigkeit darstellen.
Für die in der Sicherheitsanalyse festgelegten Schutzziele (vgl. [Fuhr 98] und BSI-Sicherheitshandbuch, [7105 92])

- **Vertraulichkeit und Integrität**: Schutz vor Kenntnisnahme oder Veränderung der lokal übertragenen Daten und vor unbefugten Zugang zu lokalen Rechnern
- **Verfügbarkeit**: Schutz vor Angriffen auf die Verfügbarkeit der lokalen Netzkomponenten

ist der Schutzbedarf für verschiedene Teile des internen Netzes gegenüber Angriffen aus dem externen Netz anhand der folgenden Schutzklassen festzulegen:

- **niedrig**: Die Verletzung des Schutzziele durch zufällige Aktivitäten muß verhindert werden.
- **mittel**: ... mit einfachen Mitteln soll abgewehrt werden.
- **hoch**: ... mit qualifizierten Angriffen soll abgewehrt werden. Ein Restrisiko wird in Kauf genommen.
- **sehr hoch**: ... muß verhindert werden. Ein minimales Restrisiko wird in Kauf genommen. (Eine 100%ige Sicherheit ist unmöglich)

Kann keine sichere Trennung der Informationen innerhalb des internen Netzes vorgenommen werden, so bestimmen die Informationen mit dem höchsten Schutzbedarf das Schutzniveau.

Analyse der Ist-Situation

Zur Bewertung der aktuelle Situation sind folgende Fragen zu beantworten:

- **Wie sieht die Struktur des vorhandenen Netzes aus?**
  Diese Frage soll klären, welche Resourcen, z.B. Rechner oder Datenbanken, geschützt werden müssen.

- **Wie sieht das Kommunikationsprofil aus?**
  Hierunter versteht man die Zuordnung einzelner Internetdienste zu verschiedenen Nutzern, Zeiten und Authentisierungsverfahren. Speziell geklärt werden muß, welche Informationen durch die Filter nach außen hindurch- bzw. nach innen hereingelassen werden sollen.

- **Welche Zugänge nach außen werden benötigt?**
  Wird z.B. der Internet-Zugang über einen Internetprovider oder über einen Modempool für nur wenige Rechner realisiert?

- **Welche Netzwerk-Informationen sollen verdeckt werden?**
  Zur Beantwortung dieser Frage ist es notwendig, die im zu schützenden Netz verwendeten Internetadressen und Rechnernamen zu kennen. Zur Verdeckung der internen Netzstruktur kann z.B. eine Adressumsetzung am Firewall benötigt werden. Wichtig ist es z.B. auch, interne Benutzernamen nicht von außen ersichtlich zu machen.

- **Welcher Datendurchsatz ist zu erwarten?** Diese Frage ist aus zwei Gründen relevant: Zum einen darf ein Firewall im Normalzustand nicht an der Grenze seiner Belastbarkeit betrieben werden, da dieser dann keine weiteren kurzfristig auftretenden Spitzenbelastungen, wie sie z.B. bei einem Angriff entstehen können, verarbeiten kann. Zum anderen müssen in Hinblick auf eine starke Zunahme der Nutzung des Internets auch die nötige Datenübertragungsraten der nächsten Jahre berücksichtigt werden. Die Sicherheit eines Firewalls wird zu einem großen Teil von der Erfahrung der Administratoren bestimmt, so daß ein häufiger Wechsel der verwendeten Produkte zu einem Sicherheitsverlust führen kann. Andererseits kann auch die Administration einer veralteten Firewallsoftware zuviel Zeit beanspruchen. In diesem Fall ist der Wechsel zu einem neuen System sinnvoll.

Definition der Verantwortlichen und Bestimmung benötigter Ressourcen

Die Sicherheitsziele müssen letztlich von Personen umgesetzt werden. Daher muß von vornherein klar definiert sein, wer welchen Beitrag zur Sicherheit zu leisten hat, wer die Aufgaben koordiniert und wer für welchen Bereich zuständig ist. Hier sind Fragen zu beantworten wie

- Wer ist für die Administration des Firewalls und der sonstigen zentralen und dezentralen Sicherungssysteme verantwortlich?
- Wer ist macht die Protkolldatenauswertung? Hier muß z.B. auch festgelegt werden, welche Ereignisse protokolliert werden. Für Protokolldaten ist insbesondere die Zweckbindung nach §14 Bundesdatenschutzgesetz zu beachten.
- Wer ist für die Revision, also die periodische Sicherheitsüberprüfung der technischen Ausrüstung, verantwortlich?

Oft muß auch das Management noch von der Notwendigkeit für Investitionen und Bereitstellung von Personal überzeugt werden, da die Sicherheit oft nur als reiner Kostenfaktor wahrgenommen wird. In letzter Zeit ist allerdings auch dort eine erhöhte Sensibilität gegenüber Sicherheitsfragen zu beobachten.

Definition von Zugangsrechten und Anweisungen

Hier wird zunächst festgelegt, welche Personen oder Personengruppen auf welche Ressourcen und Informationen mit welchen Rechten zugreifen dürfen. Dazu zählen nicht nur rechnertechnische Aspekte, sondern unter anderem auch die Definition von Zugangsberechtigungen zu Räumen. Die besten Sicherheitsmaßnahmen werden sinnlos, wenn z.B. das Putzpersonal außerhalb der üblichen Arbeitszeiten unkontrollierten Zugriff auf offene Login-Sessions bekommen kann.
Abzuwägen ist auch zwischen zuwenig und zuviel Sicherheit ("Soviel wie nötig, aber so wenig wie möglich"). Zu oberflächliche Sicherheitsvorkehrungen können sich als ungenügend erweisen, zu strenge Maßnahmen verleiten die Benutzer zum Umgehen der Vorschriften (das Passwort auf der Unterseite des Mousepads...). Zudem ist zu berücksichtigen, daß die Zugangsberechtigungen fortlaufend zu aktualisieren sind (Mitarbeiter, die das Unternehmen verlassen haben).
Die definierten Regeln müssen dann schriftlich festgehalten werden und den betroffenen Personen mitgeteilt werden. Oft ist es auch nötig, die Einhaltung der Regelungen vertraglich zu vereinbaren.

Sicherungskonzepte

Kompromittierte Systeme müssen in der Regel vollständig aus einer Sicherung restauriert werden, im schlimmsten Fall sogar komplett neu aufgesetzt werden. In beiden Fällen ist ein aktuelles Backup unerläßlich. Für sicherheitsrelevante Informationen (Logfiles) ist zudem eine längere Archivierung ratsam. Bei personenbezogenen Daten kann andererseits gesetzlich vorgeschrieben sein, die Daten nach einer gewissen Zeit zu löschen.

Die Datensicherung darf sich allerdings nicht nur auf die Sicherheitssysteme oder die Nutzdatenbestände beschränken, sondern muß alle für den Betrieb der Infrastruktur notwendigen Geräte beinhalten (Router, Switches, Hubs usw.).
Die Backup-Mechanismen müssen regelmäßig überprüft und getestet werden, um sicherzustellen, daß die Wiederherstellung im Notfall auch wirklich funktioniert.

Änderungsprozesse (Change Management)

Die Sicherheitspolitik sollte hier festlegen, wie neue Verbindungswünsche behandelt werden, wer berechtigt ist, Änderungen an der Konfiguration der Systeme zu beabtragen und wer diese Änderungen durchführen darf.
Die Freischaltung bereits von anderen Mitarbeitern benutzter Dienste ist in der Regel kein Problem. Schwieriger wird es, wenn die gewünschten Ziele oder Dienste als Sicherheitsrisiko eingestuft sind oder ein Nutzer, der als notorischer Hacker bekannt ist, die Erlaubnis für die Nutzung des Telnetprotokolls beantragt. Für neue Dienste kann es auch vorkommen, daß die eingesetzten Systeme eine sichere Bereitstellung nicht ermöglichen (z.B. wenn noch keinen Proxy oder Filter existiert). Hier ist abzuwägen, ob der Dienst überhaupt und vorerst nur eingeschränkt (z.B. durch Paketfilter-Freischaltungen zu einigen wenigen Rechnern) bereitgestellt wird.
Schwierigkeiten können auch auftreten, wenn das Filter-Regelwerk oder die Anzahl der Benutzer unüberschaubar wird. Zum einen ist der Sicherheit der Systeme dann nicht mehr zu garantieren und zum zweiten kann bei einem Verstoß gegen geltende Sicherheitsbestimmungen der Überltäter gar nicht mehr lokalisiert werden.

Notfallpläne

Im Falle eines sicherheitskritischen Ereignisses muß feststehen, wie und von wem darauf reagiert wird (Incident Response). Zu beachten ist hier insbesondere, daß in so einem Fall unter Umständen gewisse Kommunikationsmittel nicht mehr zuverlässig benutzt werden können.

Schulungsmaßnahmen und Informationsmittel

Anwender und Administratoren müssen über aktuelle Sicherheitsprobleme und Maßnahmen auf dem Laufenden gehalten werden. Erst der richtige Umgang mit allen Komponenten ermöglicht die Einhaltung der Sicherheitsregeln. Um Akzeptanzprobleme von Seiten der Mitarbeiter gegenüber den Sicherheitssystemen zu verhindern, ist es sinnvoll, eine Benutzerordnung zu verfassen, die für die Benutzer wichtige Informationen in verständlicher Form zusammenfaßt. Diese Akzeptanz der Maßnahmen ist besonders wichtig, da die Eigenverantwortung der Anwender, die die Rechner im zu schützenden Netz einsetzen, eine wesentliche Grundlage für ein sinnvolles Sicherheitskonzept ist.

Kontrolle und Revision

Ein Sicherheitskonzept muß laufend überprüft und im Bedarfsfall angepaßt werden. Hier geht es nicht nur um die technische Überprüfung der Systeme, sondern auch darum, ob das Konzept immer noch den Anforderungen der Organisation gerecht wird.
Im Rahmen der Revision ist für einzelne Systeme wie Firewalls zu entscheiden, ob das in der Sicherheitspolitik festgelegte Kommunikationsprofil richtig umgesetzt wurde. Für die Erkennung von möglichen Angriffen sind zudem die Protokolldaten der Sicherheitssysteme auszuwerten.

2.5.2 Infrastruktur

- **Physikalische Gebäudesicherheit**
  Die physikalische Umgebung, die Arbeitsräume und vor allem das Rechenzentrum, muß gewissen Anforderungen gerecht werden. Je nach Bedarf sind Maßnahmen zum Schutz vor Feuer, Wasser oder Einbruch zu treffen. Benötigt wird auch eine zuverlässige Zutrittskontrolle, eventuell mit Überwachung, weiterhin eine gesicherte Stromversorgung, Klimatisierung von Rechnerräumen, in größeren Rechenzentren auch eine Wasserversorgung und Schutz vor Katastrophen. Auch EMV-Maßnahmen zur Minimierung der elektromagnetischen Abstrahlung (Abhörsicherheit) kann in speziellen Fällen notwendig sein.

- **Physikalische Netzwerksicherheit**
  Die Netzwerk-Infrastruktur ist zuverlässig bzw. redundant auszulegen. Möglichkeiten des Abhörens des Netzverkehrs oder eine Manipulation der Netzwerkverkabelung müssen beseitigt werden. Dazu können Kommunikationsleitungen z.B. in Druckrohren verlegt werden, die von Drucksensoren auf Änderung des Luftdruckes überwacht werden.
  Zu achten ist auch auf eine zuverlässige Anbindungen an externe Kommunikationsnetze.

2.5.3 Einzelne Systeme

Auch in einem durch Firewalls abgeschotteten Netz ist eine gewisse Sicherheit der Rechner unabdingbar. Dies betrifft die Sicherung der Zugriffsmöglichkeiten auf die Rechner über die Konsole genauso wie das Erschweren von internen Angriffen übers Netz. Außerdem ist es einer Firewall niemals möglich, einen 100%-igen Schutz zu gewährleisten.

- **Zugriffsberechtigung**
  Der Zugriff auf die Systeme darf nur autorisierten Personen möglich sein. Statische Passworte müssen eine gewisse Komplexität aufweisen und periodisch geändert werden. Eventuell müssen strengere Authentifizierungsmethoden wie Challange-Response-Algorithmen (der Anwender muß dabei das nur für eine Sitzung gültige Passwort aus einem statischen Passwort und einem vom Server bereitgestellten Challange-Parameter berechnen), Token- oder Smartcards eingeführt werden.

Diese Karten können in Verbindung mit einem statischen Passwort eine doppelte Authentifizierung bereitstellen. Auch biometrische Verfahren (Fingerabdruck, Laser-Abtastungen der Augen usw.) kommen zunehmend zum Einsatz.

- **Minimalisierung der Systemfunktionalität**
  Auf den Systemen sollten nur die wirklich benötigten Programme installiert und alle überflüssigen Dienste deaktiviert werden. Einige Dienste lassen sich auch auf einzelne Netzwerkkarten oder Quell-IP-Adressen einschränken. Für viele Betriebssysteme gibt es zudem spezielle Programme, welche alle Einstellungen des Systems so restriktiv wie möglich machen. Beispielhaft sei hier auf [ANKu 02] verwiesen.

- **Lokale Sicherheitsprogramme**
  Auf den Systemen selbst können bei Bedarf Paketfilter (siehe Seite 71), Scan-Programme (Virenscanner) oder Programme zur Überwachung der Systemdateien auf unerwartete Änderungen (Host basierte IDS-Systeme, siehe Seite 267) installiert werden. In diesem Zusammenhang besonders sicherheitskritisch sind Laptops. Diese werden von den Benutzern gerne dazu verwendet, Internetverbindungen aufzubauen. Wenn die Rechner dabei auch nicht gleichzeitig im internen Netz hängen, besteht trotzdem die große Gefahr, daß ein bösartiges Programm auf den Rechner gelangt. Wird der Rechner später wieder ans interne Netz angebunden, kann der Schädling dort ungehindert Informationen sammeln und sich weiterverbreiten. Bei Client-VPN-Verbindungen (siehe Abschnitt 5) vom Internet über ein VPN-Gateway ins interne Netz ist ebenso zu beachten, daß der Client trotz der gesicherten VPN-Verbindung in der Regel weiterhin vom Internet aus direkt angreifbar ist. Auf dem Client sollte daher zumindest ein lokaler Paketfilter installiert sein.

- **Software-Updates**
  Bekannt gewordene relevante Sicherheitslücken sollten umgehend mit verfügbaren Software-Updates oder Patches geschlossen werden.

- **Sicherheits-Audits**
  Sinnvoll sind m Rahmen der Revision die regelmäßige Überprüfung der Systemsicherheit mit entsprechenden Werkzeugen (z.B. nmap, nessus, siehe Seiten 53 und 55) und die Kontrolle wichtiger Systemdateien. Neue Erkenntnisse, bekannt gewordene Sicherheitslücken oder eine geänderte Sicherheitspolitik können Änderungen an den Systemen erfordern. Im schlimmsten Fall kann bei Audits auch ein erfolgter Einbruch oder Wurmbefall erkannt werden.

2.5.4 Das Netzwerk

- **Dokumentation**
  Um zu wissen, was im Netzwerk wirklich läuft, ist ein aktueller Netzwerkplan unerläßlich. Dort sollten möglichst alle angeschlossenen Systeme aufgeführt werden.

Von besonderem Interesse sind dabei Systeme, welche prinzipiell die Möglichkeit bieten, zusätzliche Verbindungen (z.B. mit Modem oder ISDN-Karte) zum Internet aufzubauen. Natürlich müssen auch alle Firewalls und Verbindungen zu externen Netzen aufgezeichnet sein.

- **Festlegung sicherheitskritischer Zonen**
  Vor allem in größeren Netzen gibt es Bereiche mit unterschiedlichen Anforderungen an die Zugriffsmöglichkeiten und an die Sicherheit. Einmal gibt es den nur für Mitarbeiter bestimmten internen Netzwerkbereich, eventuell noch einen Bereich, wo externe und interne Personen zugreifen müssen und weiter ein externes Netz mit Internet-Web- und -FTP-Servern. Auch innerhalb des internen Netzes kann es nötig sein, besonders kritische Netze extra zu schützen. Netzbereiche, welche besondere Aufmerksamkeit erfordern, werden als DMZs (demilitarisierte Zonen) bezeichnet. Insbesondere werden alle Netze mit Zugriff von externen Personen als DMZ behandelt. Auf DMZ-Topologien wird in den Folgekapiteln genauer eingegangen.

- **Maßnahmen im LAN (Local Area Network)**
  Um schon im lokalen Ethernet-Netz eine gewisse Sicherheit zu ereichen, sollten möglichst alle Rechner nur über Switches, nicht mehr über Hubs, ans Netz angebunden werden. Dadurch werden die Abhörmöglichkeiten innerhalb des LANs stark eingeschränkt. Außerdem können auf Switches auch die am Port erlaubten MAC-Adressen festgelegt werden. Das verhindert, daß fremde Rechner ans Netz gehängt werden können.
  Neuere Switches erlauben auch die Konfiguration sogenannter Private VLANs[11]. Diese ermöglichen es, die erlaubten Kommunikationsbeziehungen zwischen den einzelnen Switch-Ports auf Schicht 2 festzulegen. Dadurch kann z.B. festgelegt werden, daß Endgeräte, obwohl sie sich alle im selben IP-Netz befinden, nicht direkt miteinander kommunizieren können, sondern nur ihr Default-Gateway erreichen. Netze unterschiedlicher Abteilungen sollten ohne Verwendung von Private VLANs durch Router getrennt werden.
  Verschlüsselungsmaßnahmen (siehe Seite 110) können für spezielle Bereiche auch schon im LAN erforderlich sein, insbesondere bei drahtloser Kommunikation (Wireless LAN, WLAN).

- **Maßnahmen im WAN (Wide Area Network)**
  In Fernverkehrsnetzen können zwar dedizierte Leitungen für den Unternehmensverkehr zur Verfügung stehen, in der Regel entziehen sich diese Leitungen jedoch der direkten Kontrolle. Hier kann eine Verschlüsselung kritischer Daten unumgänglich

[11] VLAN bezeichnet ein Virtuelles LAN, welches sich nach außen hin wie ein gewöhnliches geswitchtes LAN-Segment verhält, jedoch physikalisch nicht aus einem eigenen Switch, sondern aus einer konfigurierten Gruppierung von Switch-Ports besteht.

werden, vor allem beim Einsatz von Richtfunkstrecken. Für die Betriebssicherheit sind bei Bedarf Backup-Verbindungen vorzusehen.

- **Netzwerkkomponenten**
  Heutige Netzwerkkomponenten werden immer intelligenter und sind fast immer übers Netz konfigurierbar. Spezielle Sicherheitsvorkehrungen betreffen hier die Einschränkung des Zugriffs auf die Komponenten nur von speziellen Management-Stationen aus, möglichst die Verwendung von verschlüsselten Zugriffsprotokollen (ssh statt telnet, https statt http) und die Berücksichtigung der Eigenheiten von Netzwerkmanagement- (SNMP, Simple Network Management Protocol, ist in aller Regel unverschlüsselt im Einsatz, die Default-Community-Strings [12] lauten durchwegs "public" und "private") und Routing-Protokollen (RIP erlaubt z.B. keine Authentifizierung).

2.5.5 Sicherheitssysteme im Netzwerk

- **Intrusion Detection-Systeme (IDS)**
  Diese Systeme haben die Aufgabe, den Netzwerkverkehr zu analysieren und auf auffällige Muster hin zu untersuchen. Dazu werden sie über speziell konfigurierte Mirroring-Ports an die Netzwerk-Switches angebunden. Über diese Ports gibt der Switch alle Datenpakete aus, welche das gespiegelte Netz passieren. Die Schwierigkeiten solcher Systeme liegen auf der Hand: Extrem große Datenmengen müssen in Echtzeit aufgenommen, untersucht und ausgewertet werden. Zudem müssen die Systeme intelligent genug sein, um normalen Netzwerkverkehr von bösartigen Aktivitäten unterscheiden zu können. Ein Einsatz dieser Systeme ist daher sehr genau zu überlegen und zu planen und ist mit entsprechendem personellen Aufwand bei Implementierung und Betrieb verbunden.
  Den IDS-Systemen ist ab Seite 262 ein eigenes Kapitel gewidmet.

- **Firewalls**
  Ohne Einsatz von Firewalls ist eine Anbindung des Netzes an nicht vertrauenswürdige Netzwerke nicht zu realisieren. Dabei geht es in den meisten Fällen um die Anbindung ans Internet, aber auch Zugänge von Partnerfirmen, Beratungsunternehmen oder anderen Intranets müssen unter Umständen darüber geführt werden. Je nach Anforderung kann ein einfacher Paketfilter, ein Proxy-Firewall oder eine mehrstufige DMZ -Struktur installiert werden. Das Thema Firewalls stellt den Hauptschwerpunkt des Praktikums dar und wird in den Kapiteln ab Seite 71 ausführlich behandelt.

- **Content Filtering-Systeme**
  Content Filtering-Systeme sind spezielle Programme, welche den Datenverkehr nach gefährlichen oder unerwünschten Inhalten durchsuchen. Sie werden in der Regel in den Kommunikationspfad eingeschleust und arbeiten z.B. als Mailrelay oder

[12] Passwörter für den Lese bzw. Schreib/Lese-Zugriff auf Geräte über SNMP.

HTTP/FTP-Proxy mit integrierter Filterfunktion.
Mit solchen Systemen kann der gesamte (oder auch nur der eingehende) E-Mail-, HTTP- oder FTP-Verkehr auf Viren oder sonstige gefährliche Inhalte (bösartige Programme, Active-X-Controls, Java Applets, Javascript, in Dokumenten enthaltene aktive Komponenten, aber auch Spam-Mails) untersucht werden. Je nach Konfiguration des Filters werden infizierte Dateien vollständig gelöscht, in einen speziellen Quarantäne-Ordner verschoben oder nach Entfernen des schädlichen Codes an den Client weitergegeben. Ensprechende Meldungen können an den Benutzer und den Systemadministrator zugestellt werden (z.B. Meldung im Web-Browser oder E-Mail).
Während die Filtersysteme mit allen gängigen Archivierungsformaten (ZIP, TAR, GZIP usw.) zurechtkommen können sie verschlüsselte oder mit Passwort geschützte Daten nicht untersuchen. Dies gilt insbesondere für HTTPS-Verbindungen (siehe Abschnitt 7.2.2).
Ein weiterer Nachteil dieser Systeme liegt in der relativ hohen Verzögerung der Kommunikation, die sich insbesondere beim Download großer Dateien über HTTP und FTP bemerkbar macht. Jede Datei wird vollständig auf dem System zwischengespeichert, ggf. entpackt und gescannt. Unschädliche Daten werden erst dann an den Client weitergegeben. Dadurch erhält die Client-Software zu Beginn der Download-Phase keine Daten, für den Anwender schaut es so aus, als ob der Download nicht funktionieren würde. Bei zu kurz konfigurierten Client- oder Server-Timeouts kann dies außerdem zum Abbruch der Verbindung führen. Um diese Effekte zu minimieren bieten einige Systeme die Möglichkeit, einen gewissen Prozentsatz der Daten sofort und ungescannt an den Client weiterzuleiten, was allerdings die Sicherheit der Filterung untergräbt.
Möglicherweise wünschenswert ist auch die Einschränkung des Zugriffs auf Internet-Inhalte (URL[13]-Filterung). Dazu werden von den Herstellern der Filter-Software Listen gepflegt, in denen die Internet-URLs nach Kategorien (z.B. Pornografie, Drogen, Presse, Religion, Spiele, Politische Parteien, Bildung usw.) klassifiziert werden. Diese Kategorien können dann im System erlaubt oder verboten werden. Das System lädt die aktualisierten Listen regelmäßig von einem Server des Herstellers herunter. Möglich ist auch eine manuelle Ergänzung oder Änderung der Listen.
Die Probleme solcher Systeme liegen auf der Hand: Es ist unmöglich, alle im Internet erreichbaren URLs zu erfassen. Webseiten und deren IP-Adressen ändern sich oder verschwinden, neue Sites entstehen. Außerdem gibt es einige Möglichkeiten für die Anwender, die Filtersysteme auszuhebeln, z.B. durch Angabe der URL in verschiedenen Formaten (z.B. `http://www.leo.org/` = `http://irgendwas@www.leo.org/` = `http://131.159.72.21/` = `http://131.159.072.021/` = `http://0203.0237.0110.0025/`, weitere Kombiantionen und Formate sind möglich, teilweise Browser-abhängig) oder die Verwendung von HTTP-Proxies im Internet

[13](Uniform Resource Locator, siehe 7.2)

(u.a. Übersetzungsdienste, aus `http://www.leo.org/` wird dann z.B. `http://unimut.fsk.uni-heidelberg.de/unimut/schwob?schwob_url=http%3A%2F%2Fwww.le`
Die Klassifizierung der URLs ist zudem oft "Geschmacksache" (Stichwort Zensur) und manchmal schlichtweg falsch.
Neuere Systeme enthalten auch Möglichkeiten zur Filterung von Grafiken, Werbebannern, Popup-Fenstern, Cookies oder HTML-Tags (siehe Anhang A.5.3) sowie Reporting-Mechanismen.
Erhältlich sind auch Client-seitige Filterprogramme, die direkt in den Web-Browser intergriert werden können. Neuere Web-Browser und E-Mail-Clients (z.B. Mozilla/Netscape) bringen zudem verschiedene Filtermöglichkeiten selbst mit.

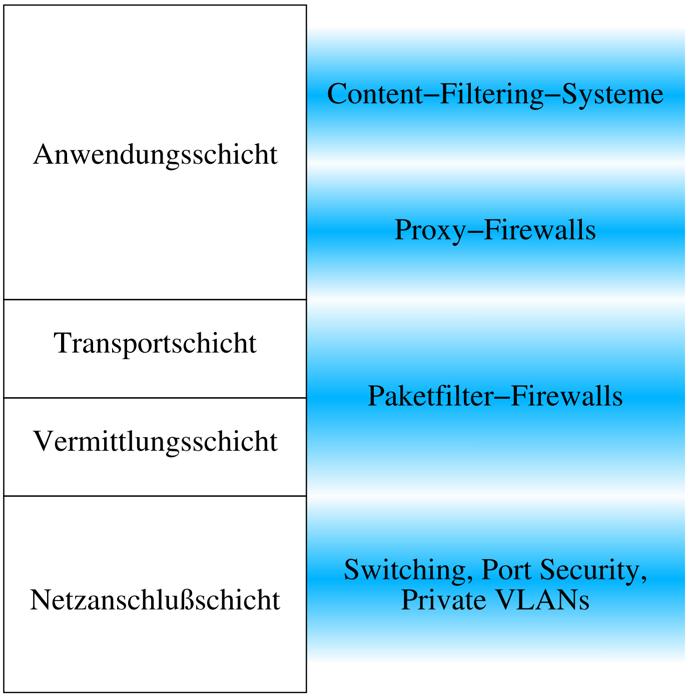

Abbildung 17: Sicherheitsmaßnahmen im Netzwerk

Abbildung 17 ordnet die netzwerkbezogenen Sicherheitsmaßnahmen in das Internet-Schichtenmodell ein. Die Grenzen zwischen den Systemen werden immer unschärfer, weil der Funktionsumfang neuerer Software ständig steigt und somit immer tiefer in die benachbarten Schichten vordringt.

2.6 Hacking-Tools unter Linux

Wir werden im Folgenden einige Programme für die Informationsbeschaffung über Systeme und die Durchführung von Angriffen auf Systeme vorstellen.

Es sei hier nochmal ausdrücklich auf die rechtlichen Anmerkungen zum Thema Hacking in Kapitel 0.5 und im Anhang A.3 hingewiesen.

2.6.1 Der Passwortcracker „Crack5.0“

Mit Passwortcrackern werden Passwörter entschlüsselt. Der wohl bekannteste Passwortcracker unter Unix ist der Crack5.0. Crack5.0 verschlüsselt Wörter aus Wörterbüchern und gängige Variationen davon in allen in Frage kommenden Varianten ($2^{12} = 4096$ Möglichkeiten[14]) und vergleicht das Ergebnis mit dem Eintrag aus der Datei `/etc/passwd` bzw. bei allen neueren Unix-Systemen in `/etc/shadow`.
Zur Installation von Crack5.0 muß sowohl das Quellcode-Archiv `crack50.tar.gz` (oder ähnlich) wie auch das Makefile `c50-linux-util-makefile` auf den Rechner geladen werden. Nach dem Entpacken des Archives müssen in der Datei `Crack` der Compiler und die Crypt-Library richtig referenziert werden:

```
# vanilla unix cc
#CC=cc
#CFLAGS="-g -O $C5FLAGS"
#LIBS=-lcrypt # uncomment only if necessary to use stdlib crypt(),
                                                     eg: NetBSD MD5

# gcc 2.7.2
CC=gcc
CFLAGS="-g -O2 -Wall $C5FLAGS"
LIBS=-lcrypt # uncomment only if necessary to use stdlib crypt(),
                                                     eg: NetBSD MD5
```

Außerdem muß unter `src/util` das Makefile durch `c50-linux-util-makefile` ersetzt werden. Anschließend kann der Passwortentschlüsseler mit `./Crack -makeonly` kompiliert werden.
Die Wörterbücher für die Angriffe werden mit `./Crack -makedict` erzeugt.
Da bei den meisten Unix Systemen die Passwörter nicht mehr in der `/etc/passwd`, sondern verschlüsselt in der `/etc/shadow` abgelegt sind, muß mit `./scripts/shadmrg.sv > passwd-shadow` ein File erzeugt werden, das die Informationen aus der `/etc/passwd` und der `/etc/shadow` vereint. In unserem Beispiel haben wir es `passwd-shadow` genannt. Dieses File ist das Inputfile für den Passwortcracker: `./Crack passwd-shadow`.
Nun wird versucht, die Passwörter zu knacken. Dieser Vorgang kann je nach Komplexität des Passwortes auch mehrere Tage dauern. Solange noch ein Crack-Prozeß läuft, ist das Programm noch nicht fertig:

```
ps ax | grep crack
```

[14]Die Art der Verschlüsselung wird durch die ersten beiden Zeichen des Passwort-Strings in `/etc/shadow` festgelegt ("Salzkorn").

```
19519 pts/2    RN   207:24 cracker -kill run/Ksectest.19434
 6951 pts/2    S      0:00 grep crack
```

Unter `run/` finden Sie die Files, in die das Programm seine Zwischenergebnisse speichert. Nach Beendigung der Berechnungen kann mit `./Reporter -quiet` das Ergebniss angezeigt werden:

```
./Reporter -quiet

---- passwords cracked as of Tue May 21 09:42:40 CEST 2002 ----

Guessed testuser1 [Hallo]  testuser [passwd-shadow /bin/bash]
Guessed testuser1 [Hallo]  testuser [passwd-shadow /bin/bash]
Guessed testuser1 [Hallo]  testuser,,, [passwd-shadow /bin/bash]
Guessed testuser1 [Hallo]  testuser,,, [passwd-shadow /bin/bash]

---- done ----
```

2.6.2 Der Portscanner „Nmap“

Nmap (Network Mapper) ist einer der bekanntesten Portscanner unter Unix. Bei den meisten Linux-Distributionen ist er als Paket enthalten. Er bietet eine Vielzahl von Optionen an, um die Art des Portscans zu variieren. In den Manpages `man nmap` finden Sie eine genaue Beschreibung der angebotenen Funktionen.
Ein einfacher Portscan sieht folgendermaßen aus:

```
nmap 10.50.181.8

Starting nmap V. 2.3BETA14 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (10.50.181.8):
Port    State       Protocol  Service
21      open        tcp       ftp
22      open        tcp       ssh
25      open        tcp       smtp
80      open        tcp       http
515     open        tcp       printer
1026    open        tcp       nterm

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Dabei können nicht nur einzelne Hosts, wie in dem Beispiel, sondern auch ganze Netze angegeben werden:

```
nmap 10.50.181.7-8

Starting nmap V. 2.3BETA14 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (10.50.181.7):
Port    State       Protocol  Service
135     open        tcp       loc-srv
139     open        tcp       netbios-ssn

Interesting ports on  (10.50.181.8):
Port    State       Protocol  Service
21      open        tcp       ftp
22      open        tcp       ssh
25      open        tcp       smtp
80      open        tcp       http
515     open        tcp       printer
1026    open        tcp       nterm

Nmap run completed -- 2 IP addresses (2 hosts up) scanned in 4 seconds
```

Der Status eines Ports kann entweder `open`, `filtered` oder `unfiltered` sein.

- `open`: auf dem Port werden Verbindungen akzeptiert.
- `filtered`: ein Firewall, Filter oder Router verhindert eine Überprüfung des Ports auf der Zielmaschine, die Scan-Pakete wurden ohne Rückmeldung verworfen.
- `unfilterd`: die Überprüfung des Ports hat ergeben, daß keine Verbindungen aufgebaut werden können (Rückmeldung `Connection refused`). Dieser Status wird nicht explizit angezeigt.

Beim Standardscan ohne Optionen wird ein vollständiger Drei-Wege-Handshake durchgeführt. Diese Scans sind in den Logfiles des gescanten Systems leicht auszumachen. Schwieriger mit der Lokalisierung wird es bei sogenannten TCP SYN Scans (`nmap`-Option `-sS`), Stealth FIN Scans (Option `-sF`), Xmas Tree (Option `-sX`) oder auch Null Scans (Option `-sN`). Bei all diesen Scans wird darauf gebaut, daß bei einer Verbindungsanfrage an einen nicht geöffneten Port ein RST zurückgesendet wird, wie in RFC 794 [Cerf 81] verlangt wird. Leider funktionieren die Scans bei einigen Windows-Versionen nicht, da dort der RFC nicht berücksichtigt wurde.

- TCP SYN Scan wird auch als "half open" scanning bezeichnet, da man keine vollständige TCP-Verbindung öffnet. An den zu testenden Port wird ein SYN Paket gesendet. Kommt ein SYN ACK Paket zurück kann man davon ausgehen, daß der Port offen ist. Bei einem RST ist der Port nicht geöffnet.

- Stealth FIN Scans senden auf die zu scannenden Ports FIN Packete.
- Bei Xmas Tree werden FIN, URG und PUSH Flag gesetzt.
- Beim Null Scan sind alle Flags ausgeschaltet.

Mit der Option `-p 1-10` werden die Ports 1 bis einschließlich 10 gescannt. Ohne Angabe eines Port-Bereiches werden alle Ports bis 1024 überprüft, die Portnummer wird in der Ausgabe ggf. durch den entsprechenden Dienst-Namen aus der Datei `/etc/services` des lokalen Rechners ersetzt.
Die Option `-O` versucht, über einen sogenannten "Fingerprint" (Charakteristische Reaktion des TCP/IP-Stapels des angesprochenen Systems auf gewisse Datenpakete) das Betriebssystem des gescanten Rechners herauszufinden, das Ergebnis muß aber nicht unbedingt stimmen.

2.6.3 Der Securityscanner „Nessus“

Eine spezifischere Möglichkeit zum Scannen von Systemen auf Schwachstellen bietet die Software `Nessus`. Das 'Nessus' Projekt wurde Anfang 1998 begonnen und zum erstenmal im April 1998 veröffentlicht. Zu dieser Zeit war SATAN der vollständigste freie Security Scanner, der aber mittlerweile veraltet ist. Es gibt natürlich auch verschiedene kommerzielle Scanner, die sehr hohe Lizenzkosten haben.
Die Funktionsweise des Nessus Security Scanners baut auf das Client-Server-Prinzip auf: Es kann beliebig viele Clients geben, die auf einen Server zugreifen, der dann den Scan vornimmt. Dabei können die Scanmöglichkeiten von Client zu Client unterschiedlich eingeschränkt werden. Natürlich darf man auch diesem Scanner nicht blind vertrauen. Jedes Scanergebnis muß interpretiert und von Hand überprüft werden.
Informationen und den aktuellen Sourcecode finden Sie unter `http://www.nessus.org/` und `http://www.scriptkiddie.de/nessus/`. Bei den meisten Linux-Distributionen ist bereits ein Nessuspaket dabei, das installiert werden kann. Wird aber der neueste Nessus benötigt, so muß er von Hand kompiliert werden:
Folgende Dateien sind dazu nötig:

- nessus-libraries-x.x.tar.gz
- libnasl-x.x.tar.gz
- nessus-core.x.x.tar.gz
- nessus-plugins.x.x.tar.gz

Zuerst müssen die Libaries installiert werden:

```
cd nessus-libraries
./configure
make
make install
```

Soll nur der Windows-Client benutzt werden, so ist der Cipher-Layer unnötig und kann gleich bei der Grundkonfiguration weggelassen werden: `./configure --disable-cipher`. Dananch folgt die Installation der `libnasl`:

```
cd libnasl
./configure
make
make install
```

Ebenso ist mit der Installation von `nessus-core` und `nessus.plugins` zu verfahren. Somit ist die Installation von Nessus abgeschlossen.
Um nun den Scanner in Betrieb nehmen zu können, ist zuerst die Konfiguration des Nessus-Servers `nessusd` nötig, wie man auch den Manpages `man nessusd` entnehmen kann. Zuerst legt Nessus sein Konfigurationsfile und sein public/private Schlüsselpaar an. Außerdem wird bereits der erste User mit ID und Passwort festgelegt.

```
nessusd --make-user=testuser,testpasswort
Generating primes: ...q..........pg
```

Mit `nessusd -L` können die so eingerichteten User angezeigt werden. Bei einem von Hand kompilierten Nessus ist das Softwareverzeichnis mit allen Konfigurationsfiles standardmäßig `/usr/local/etc/nessus/`. Bei SuSE liegen diese Files unter `/etc/nessus/`. Um noch weitere User mit Beschränkungen anlegen zu können, wird `nessus-adduser` (siehe auch `man nessus-adduser`) aufgerufen:

```
nessus-adduser
Using /var/tmp as a temporary file holder

Add a new nessusd user
----------------------



Login : test-user
Authentication method (cipher/plaintext) [cipher] :

Source restriction
------------------
```

```
You can, if you will, configure this account so that it can only
be used from a given host or subnet. For instance, you may want
test-user to be able to connect to this nessusd server only from
his work machine.

Please enter the host (or subnet) test-user is allowed to connect from.
A blank entry will allow him to connect from anywhere

The entry format must be an IP address followed by an optional netmask.
Hostnames are *not* accepted

Examples of valid entries :
   192.168.1.5
   192.168.1.0/24
   192.168.1.0/255.255.255.0

Invalid entry :
   prof.fr.nessus.org

Source host or network [anywhere] : 127.0.0.1

One time password : test-pass

User rules
----------
nessusd has a rules system which allows you to restrict the hosts
that test-user has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules syntax

Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)
accept 192.168.216.0/24
default deny


Login              : test-user
Auth. method       : cipher, can connect from 127.0.0.1
One time password : test-pass
Rules              :
accept 192.168.216.0/24
default deny
```

```
Is that ok ? (y/n) [y] y
user added.
```

Hier kann für jeden User festgelegt werden, welche User-ID er hat, ob Verschlüsselung für die Kommunikation benutzt wird, von welchen IP-Adressen aus sich dieser User anmelden darf, welches Passwort ihm zugeteilt ist und welche Netze er scannen darf.
Mit `nessusd -D &` wird der Nessusserver gestartet und in den Hintergrund geschickt.

```
ps ax | grep nessus
30998 pts/5    R      0:16 nessusd -D
```

Die Clientsoftware wird mit `nessus` gestartet. Bei der graphischen Oberfläche ist der Nessusserver, der Nessusport und die Verschlüsselung anzugeben. Bei `login` und `passwort` sind die User-ID und das Passwort anzugeben, die bei `nessus-adduser` auf dem Server festgelegt worden sind.

Unter `Target selection` werden die Zielsysteme angegeben, die überprüft werden sollen. Die Angabe kann mit DNS-Namen oder IP-Adressen erfolgen, wobei auch ganze Netze angegeben werden können: `192.168.216.0/25,192.168.216.190`. Unter `Plugins` (siehe auch Abbildung 18) wird festgelegt, auf welche Schwachstellen hin die angegebenen Server untersucht werden sollen. Um eine genaue Einschränkung treffen zu können, sollte immer zuerst die Beschreibung der Schwachstellen gelesen und interpretiert werden. Hier wird keine genaue Auflistung und Erklärung der verschiedenen Plugins gegeben, da die Angaben in der Software selbsterklärend sind.
Einige Security Checks brauchen Argumente, damit der Scan erfolgreich sein kann. Diese Argumente können unter `Prefs.` eingestellt werden. Unter `Scan options` werden verschiedene Optionen für den Scan festgelegt. Mit `User` können noch weitere Einschränkungen gemacht werden, da man bei Scans sehr vorsichtig sein muß, was man überprüft. Mit `Start the scan` wird die Überprüfung gestartet. Das kann unter Umständen sehr lange dauern. Nach Beendigung der Tests öffnet sich das Reportfenster, wie in Abbildung 19 dargestellt.

2.6.4 Der CGI-Scanner „Whisker“

Whisker ist ein CGI-Scanner, der Schwachstellen in Webserver-CGI-Programme[15], überprüft. Folgende Funktionalitäten sind in Whisker enthalten:

- Das CGI Verzeichnis kann vom vordefinierten Verzeichnis `/cgi-bin` auf eine Liste von bekannten CGI Pfaden oder von Hand definierten Verzeichnissen gewechselt werden.

[15] CGI steht für Common Gateway Interface und stellt einen Standard für die Integration ausführbarer Programme in Webserver dar. Diese Programme dienen in der Regel dazu, Webseiten in Echtzeit aus Informationen in einer extern Datenquelle, z.B. einer Datenbank, zu generieren.

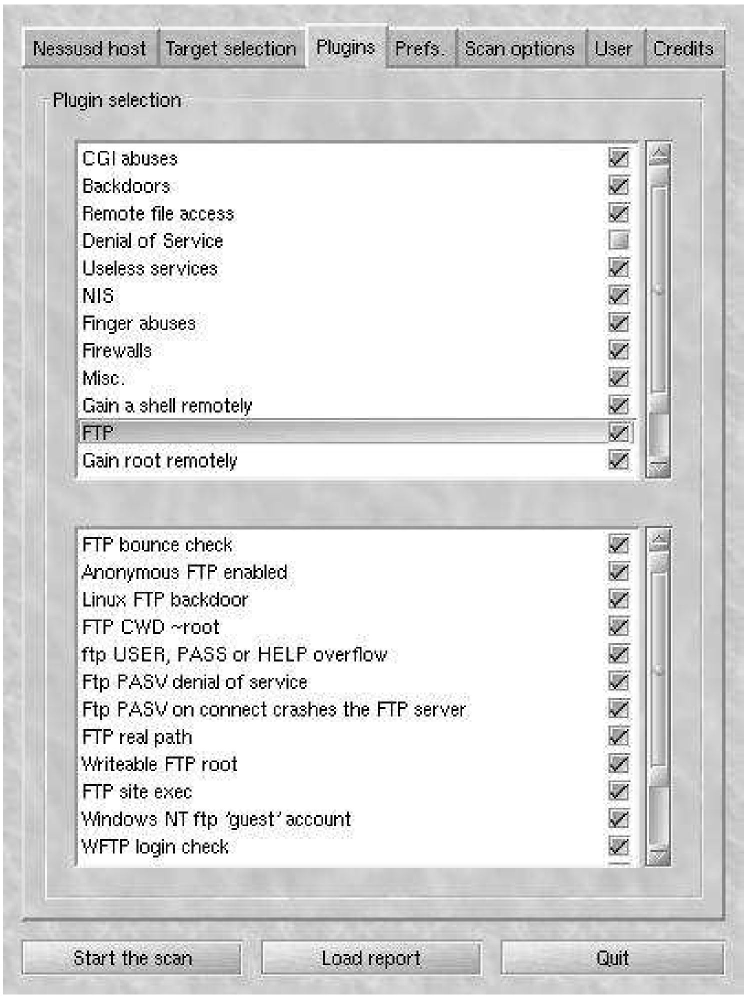

Abbildung 18: Screenshot der Nessus Plugins

- Zu Beginn des Scans überprüft Whisker, ob das angegebene Verzeichnis existiert. Ist es nicht vorhanden, so werden die angegebenen Verwundbarkeitsüberprüfungen aus Performancegründen unterlassen. Ebenso wird zuerst geprüft, ob das CGI Script vorhanden ist.
- Es wird der Webservertyp und die Version überprüft, um Checks von nicht unterstützten CGI Scripts zu minimieren.
- Das Feature 'Virtual Host' wird voll unterstützt.
- Whisker ist in Perl, einer unter Unix weit verbreiteten Interpreter-Sprache, geschrieben und somit leicht von Hand erweiterbar.
- Der Scan kann durch URL-Kodierung vor vielen IDS-Systemen getarnt werden.

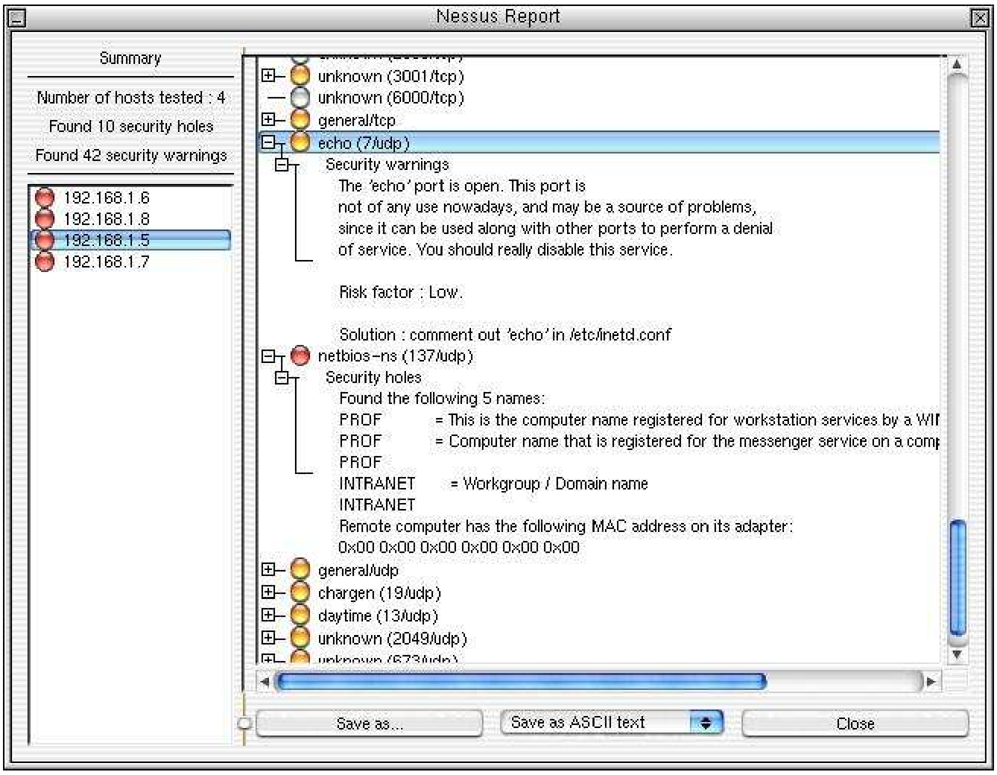

Abbildung 19: Screenshot des Reportfensters bei Nessus

Den Sourcecode finden Sie unter `http://www.wiretrip.net/rfp/` oder direkt bei `http://www.wiretrip.net/rfp/bins/whisker/whisker.tar.gz`[16]. Nach dem Herunterladen muß das Verzeichnis nur ausgepackt werden. Zum Ausführen ist Perl 5.0004 nötig. Eine Auflistung aller möglichen Optionen erhalten Sie mit `perl whisker.pl`:

```
-- whisker / v1.4.0 / rain forest puppy / www.wiretrip.net --

        -n+ *nmap output (machine format, v2.06+)
        -h+ *scan single host (IP or domain)
        -H+ *host list to scan (file)
        -F+ *(for unix multi-threaded front end use only)
        -s+  specifies the script database file (defaults to scan.db)
        -V   use virtual hosts when possible
        -p+  specify a different default port to use
        -S+  force server version (e.g. -S "Apache/1.3.6")
        -u+  user input; pass XXUser to script
        -i   more info (exploit information and such)
```

[16] `.tar.gz`-Archive können mit `tar -xvzf Datei.tar.gz` ausgepackt werden.

```
        -v   verbose.  Print more information
        -d   debug. Print extra crud++ (to STDERR)
        -W   HTML/web output
        -l+  log to file instead of stdout
        -a+  authorization username[:password]
        -P+  password file for -L and -U

        -I 1 IDS-evasive mode 1 (URL encoding)
        -I 2 IDS-evasive mode 2 (/./ directory insertion)
        -I 3 IDS-evasive mode 3 (premature URL ending)
        -I 4 IDS-evasive mode 4 (long URL)
        -I 5 IDS-evasive mode 5 (fake parameter)
        -I 6 IDS-evasive mode 6 (TAB separation) (not NT/IIS)
        -I 7 IDS-evasive mode 7 (case sensitivity)
        -I 8 IDS-evasive mode 8 (Windows  delimiter)
        -I 9 IDS-evasive mode 9 (session splicing) (slow)
        -I 0 IDS-evasive mode 0 (NULL method)

        -M 1 use HEAD method (default)
        -M 2 use GET method
        -M 3 use GET method w/ byte-range
        -M 4 use GET method w/ socket close

        -A 1 alternate db format: Voideye exp.dat
        -A 2 alternate db format: cgichk*.r (in rebol)
        -A 3 alternate db format: cgichk.c/messala.c (not cgiexp.c)

-- Utility options (changes whisker behavior):

        -U   brute force user names via directories
        -L+  brute force login name/password
             (parameter is URL; use with -a for username)

        + requires parameter;  * one must exist;

        (Note: proxy/bounce support has been removed until v2.0)
```

- `-h`: scannt einen einzelnen Host, wobei entweder der DNS Name oder die IP-Adresse angegeben wird.
- `-H`: kann eine ganze Liste von Maschinen scannen, die in einer Datei aufgelistet sind.
- `-s`: legt fest, anhand welches Datenfiles gescannt wird. Standard ist `scan.db`.

- `-V`: scannt auch virtuelle Hosts, falls vorhanden.
- `-p`: hier kann der Port festgelegt werden, hinter dem der zu scannende Webserver läuft.
- `-S`: legt fest, welche Webserverversion überprüft werden soll.
- `-i`: dadurch werden mehr Informationen zu den Schwachstellen und den Exploits ausgegeben.
- `-v`: verbose Modus, gibt generell mehr Information aus.
- `-d`: debug Modus
- `-W`: erzeugt HTML Ausgabe
- `-l`: hier kann eine Datei angegeben werden, in die das Scanergebnis geschrieben werden soll.
- `-a`: Angabe von User-ID und gegebenenfalls Passwort: userid:password
- `-P`: legt das für die Optionen `-L` und `-U` zu verwendende Passwortfile fest.
- `-I` Nummer: verschiedene Modi zum Umgehen von IDS Systemen.
- `-M` Nummer: verschiedene Modi zur Variation der Abfragemethoden.
- `-A` Nummer: alternative Datenbankformate.
- `-U`: Brute Force Angriff auf User-IDs per Directorystrukturen.
- `-L`: Brute Force Angriff auf User-IDs und Passwörter.
- Alle Optionen mit angehängtem `+` erwarten noch einen Parameter, der angegeben werden muß. Mindestens eine Option mit anghängtem `*` muß angegeben werden.

Ein möglicher Scan könnte folgendermaßen aussehen:

```
perl whisker.pl  -h www.muc.meinedomain.de -s barbara.db -V -p 80 -i -M 1
-- whisker / v1.4.0 / rain forest puppy / www.wiretrip.net --

= - = - = - = - = - =
= Host: www.muc.meinedomain.de  (virtual host)

= Server: Apache/1.3.9 (Unix) Debian/GNU

-  www.apache.org
```

```
+ 200 OK: HEAD /htdocs/

+ 200 OK: HEAD /html/

+ 200 OK: HEAD /perl/

+ 200 OK: HEAD /tools/
```

Wobei Whisker zwar einige Verzeichnisse gefunden hat, in denen eventuell zu überprüfende CGIs- Scripte liegen könnten. Die weiteren Tests haben aber gezeigt, daß keine der zu erwartenden Schwachstellen vorhanden war.

2.6.5 Rootkits

Ein Rootkit ist eine Sammlung von Dateien und Programmen, die es einem Angreifer erlaubt, zu einem späteren Zeitpunkt auf ein kompromittiertes System unerkannt zurückzukehren. Die Programme öffnen dazu eine Hintertür (backdoor) im System und versuchen gleichzeitig, alle eigenen Spuren (Dateien, Programme, laufende Prozesse, offene Ports usw.) zu verstecken oder zu beseitigen.
Viele Rootkits bringen auch einen Sniffer oder andere Mechanismen zum Sammeln von Daten über das System und seine Benutzer mit. Die Informationen, insbesondere mitgelesene Passwörter von `root` oder anderen Usern, werden mitprotokolliert und können vom Angreifer später abgeholt und weiterverwendet werden.

Grob kann zwischen zwei Arten von Rootkits unterschieden werden. **"Herkömmliche" Rootkits** ersetzt alle Systemprogramme zum Anzeigen von Systemzuständen (z.B. `ls`, `ps`, `netstat`, `lsof`) durch kompromittierte Binaries (ausführbare Dateien), welche die eigenen Dateien, Programme, laufende Prozesse offene Ports usw. nicht anzeigen und Server-Programme (`sshd`, `telnetd` usw.) durch Versionen mit speziellen Zusätzen.
Die neueren **LKM** (loadable Kernel Module) **-Rootkits** werden als Modul in den Kernel geladen und verändern dort die Eigenschaften des Kernels selbst, z.B. die Einträge im `/proc`-Dateisystem, aus welchen viele Befehle (z.B. `netstat`) die angezeigten Daten auslesen. Durch diese Art der Kompromittierung müssen keinerlei Änderungen an den Programmen des Systems gemacht werden. Diese Rootkits setzen allerdings voraus, daß der Kernel das Laden von Modulen unterstützt.
Rootkits gibt es für verschiedene Betriebssysteme, sowohl für gängige Unix-Derivate als auch für viele Windows-Versionen.

Auffinden von Rootkits unter Unix/Linux

Wurden keine speziellen Maßnahmen für das Erkennen von auffälligen Veränderungen im Dateisystem getroffen (`tripwire`, `md5sum`, darauf wird in späteren Kapiteln eingegangen) ist es nicht so einfach, ein Rootkit zu erkennen, da man davon ausgehen muß, daß die

Kommandos des Systems nicht mehr vertrauenswürdig sind und u.U. mehr Schaden anrichten als helfen (z.B. wenn ein `cat` plötzlich zum `rm` wird). Dennoch bestehen verschiedene Möglichkeiten, Hinweise auf ein Rootkit zu erhalten.

- **Den Rechner "von außen" untersuchen**
  Über Portscans von einem anderen Rechner aus können offene Netzwerk-Ports erkannt werden, welche z.B. von `netstat` lokal auf dem Rechner nicht anzeigt werden.
- **Vertrauenswürdige Programme auf den Rechner bringen**
  Da man den lokalen Kommandos nicht mehr vertrauen kann, sollten zur Untersuchung (notfalls über CD oder Diskette) neue Kommandos aufs System gebracht werden. Interessant ist in diesem Zusammenhang eine eigenständige Shell, die gleich alle wichtigen Kommandos integriert hat: `sash`. Außerdem kann hier das Werkzeug Carbonite (`http://www.incident-response.org/Carbonite.htm`) gute Dienste erweisen, welches trotz LKM-Rootkit offene Dateien und Prozesse anzeigt.
- **`/dev`-Dateisystem**
  Einige Rootkits verstecken sich unter `/dev` in der Hoffnung, dort nicht gefunden zu werden. Da `/dev` aber im Normalfall keine normalen Dateien enthält können solche Rootkits relativ einfach mit `find /dev -type f` gefunden werden.
- **`/proc`-Dateisystem**
  Im `/proc`-Dateisystem sind u.a. alle laufenden Prozesse und offene Netzwerkverbindungen verzeichnet. Sollten die Systemkommandos einige Dateien oder Prozesse nicht anzeigen findet man diese u.U. dort trotzdem. Leider gilt dies nicht für LKM-Rootkits.
- **Unerklärliche Prozessorlast?**
  Mit `top` läßt sich die Gesamt-CPU-Belastung des Systems sowie die von einzelnen Prozessen verursachte Last anzeigen. Weicht die Gesamtlast erheblich von der Summe der von den einzelnen Prozessen erzeugten Last ab ist zu vermuten, daß ein versteckter Prozeß, z.B. ein Passwort-Cracker, die Last verursacht.
- **Festplatten auf Blockebene durchsuchen**
  Mit `strings /dev/hda1 | grep verdaechtigeswort` können die einzelnen Partitionen (hier `/dev/hda1`) auf Blockebene nach verdächtigen Worten (Dateinamen und -inhalte) untersucht werden, z.T. auch in bereits gelöschten Dateien.
- **Systemaufrufe von Kommandos verfolgen**
  ein verdächtiges Kommando kann mit `strace` untersucht werden. `strace` zeigt alle Systemaufrufe an, die das Programm tätigt, so z.B. auch Zugriffe auf Dateien.
- **Kernelmodule anzeigen**
  Der Befehl `lsmod` zeigt (im Normalfall) alle geladenen Kernelmodule an. Allerdings können sich LKM-Rootkits auch vor `lsmod` verbergen.

- **Verdächtige Dateisysteme in einem anderen System untersuchen**
  Dazu kann das verdächtige System von einer anderen Partition, Festplatte oder von CD gebootet werden oder die Festplatte in einen anderen Rechner eingebaut werden, der von einer anderen Festplatte gebootet wird. Das zu untersuchende Dateisystem kann dann manuell gemountet werden.

- **Spezielle Rootkit-Erkennungsprogramme**
  Im Internet sind verschiede Programme zur Erkennung von Rootkits verfügbar. Beispielhaft sind hier drei Programme aufgelistet.

  – The Coroner's Toolkit (TCT): `http://www.porcupine.org/forensics/tct.html`
  – Chkrootkit: `http://www.chkrootkit.org/`
  – Rkdet: `http://vancouver-webpages.com/rkdet/`

Was tun mit dem infizierten Rechner?

Wenn möglich sollte der infizierte Rechner sofort vom Netz getrennt werden, um weiteren Schaden zu verhindern und um den Rechner genau untersuchen zu können. Das weitere Vorgehen hängt stark vom Einzellfall ab. Zu klären sind folgende Fragen:

- Wie ist der Cracker auf den Rechner gekommen?
  Die Schwachstelle im Netz muß erkannt und sofort geschlossen werden. Dabei kann es sich z.B. um einen undichten Firewall, eine veraltete, fehlerhafte oder nicht gepatchte Software, um eine Fehlkonfiguration oder um ein nach außen gedrungenes Passwort handeln.

- Wer war der Einbrecher und welches Ziel wollte er erreichen?
  Der Einbruch könnte von einem Programm (Wurm) oder von einer Person durchgeführt worden sein. Hatte der Einbrecher ein genaues Ziel oder wollte er nur in irgendeinen Rechner eindringen? Eventuell könnte es nützlich sein, weitere Informationen über den Einbrecher zu sammeln. Dazu könnte der Rechner speziell präpariert wieder ans Netz gehängt werden (Honeypot), da der Einbrecher, wenn er weitere Ziele angreifen möchte, irgendwann wiederkehren muß.

- Wie weit könnte der Angreifer bereits gekommen sein?
  Eventuell hat der Cracker sich schon Kennungen und Passwörter anderer Rechner besorgt und hat sich auch schon auf anderen Rechnern im Netz zu schaffen gemacht. Es ist zu klären, inwieweit Passwörter geändert, Authentifizierungsmethoden verschärft und weitere Rechner untersucht werden müssen.

- Was soll mit dem gecrackten Rechner selbst geschehen?
  Reicht es, nur das gefundene Rootkit zu entfernen oder muß der Rechner neu aufgesetzt bzw. aus einem Backup rekonstruiert werden? Im Zweifelsfall ist letzeres auf jeden Fall zu empfehlen.

Natürlich gelten die Maßnahmen nicht nur für Rootkits, sondern für jede Art von Einbruch in einen Rechner.

2.6.6 Denial of Service-Programme

In diesem Abschnitt werden einige Werkzeuge zur Durchführung von DoS-Attacken vorgestellt.

Teardrop

Teardrop wird kompiliert mit

`gcc -O2 teardrop.c -o teardrop`

und aufgerufen mit

`./teardrop src-ip dst-ip [-s src-prt] [-t dst-prt] [-n how-many]` .

Teardrop sendet von `src-ip` zu `dst-ip` (optional sind auch Quell- und Zielport festlegbar) `how-many` missgebildete IP-Fragmente. Angreifbar sind ältere Linux-Kernel sowie Windows 95 und NT 4.0 -Rechner.

Nuke

Kompilieren:

`gcc nuke.c -o nuke`

Ausführen:

`./nuke ip-addr`

Nuke schickt an `ip-addr` auf Port 139 (Netbios) nach dem Verbindungsaufbau Pakete mit gesetztem Urgent-Bit. Dies trennt ältere Windows 95 und Windows-NT-Rechner vollständig vom Netz.

Smurf

Kompiliert wird das Programm mit

`gcc smurf.c -o smurf` ,

aufgerufen mit

`./smurf target bcast-file num-packets packet-delay packet-size` .

Mit Smurf ist es möglich, das Netzwerk mit einer großen Menge von ICMP-Echo-Request-Paketen zu überfluten. Dazu spooft Smurf die Absender-Adresse (`target`) der Pakete und schickt von dieser Adresse aus eine beliebige Anzahl (`num-packets`, `0` für Fluten des Netzen) von ping-Paketen zu einer beliebigen Broadcast-Adresse, die in der Datei `bcast-file` enthalten ist. `packet-delay` gibt an, wie groß die Pause zwischen den einzelnen Paketen

in Millisekunden sein soll, `packet-size` bezeichnet die Paketgröße (< 1024).
Alle Rechner im zur Broadcast-Adresse gehörigen Netz antworten nun auf diese ping-Anfragen an `target`, nicht an den Rechner, der den Angriff gestartet hat. Der Rechner wird also mit einer Unmenge von ICMP-Echo-Reply-Paketen bombardiert.

Leider hat das Programm auf unserer Testinstallation nicht richtig funktioniert (Die angesprochenen Rechner haben auf den Broadcast-ping nicht geantwortet, auf `ping -b 10.50.187.255` schon, offenbar sind die von Smurf erzeugten ICMP-Echo-Request-Pakete fehlerhaft.).

Jolt

Kompilieren:

```
gcc jolt.c -o jolt
```

Aufrufen:

```
./jolt [-s src-addr] [-p port] dest-addr
```

Jolt bringt bei Windows-Rechnern (Win98, NT4/SP5 und 6, Win2K) die Prozessorlast sofort auf 100%, der Rechner reagiert nicht mehr.

Online-Demos

Die in letzter Zeit aufgekommenen Online-Demos haben zum Ziel, Web-Auftritte von Organisationen für eine bestimmte Zeit unerreichbar zu machen. Beispielhaft wird hier die Funktionsweise eines frei im Internet erhältlichen Baukastens zur Durchführung solcher DDoS-Attacken beschrieben.

Das Programm beschränkt seine Funktionsweise auf einen bestimmten Zeitraum. Dazu holt es sich von mehreren großen Web-Auftritten die aktuelle Uhrzeit:

```
linux:~$ telnet www.google.de 80
Trying 216.239.39.101...
Connected to www.google.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 302 Found
Content-Length: 151
Connection: Close
Server: GWS/2.0
Content-Type: text/html
Date: Fri, 24 May 2002 13:21:48 GMT
Location: http://www.google.de/
...
```

```
Connection closed by foreign host.
linux:~$
```

Liegt die Zeit im gewünschten Zeitraum versucht es, von einer oder mehreren vorgegebenen URLs eine Konfigurationsdatei zu laden, die das Angriffsziel beinhalten. Ist dies erfolgreich, so wird der Angriff gestartet. Kann die Konfigurationsdatei nicht geholt werden, so wird die fest in den Code eingetragene URL des Angriffsziels angegriffen. Die Konfigurationsdatei dient offensichtlich dazu, mit dem Programm kurzfristig auf eventuelle Abwehrmaßnahmen zu reagieren.

Der eigentliche Angriff besteht aus einer großen Menge normaler HTTP oder HTTPS -Anfragen auf das Ziel. Die "böswilligen" Anfragen können also nicht von "normalen" Anfragen unterschieden werden. Abwehrmaßnahmen sind aus diesem Grund sehr schwierig zu realisieren. Ziel der Attacke ist es, die Infrastruktur des Web-Auftrittes (Web-Server, Firewalls, Netzwerkanbindungen) durch simple Überlastung unerreichbar zu machen.
Die DDoS-Attacke kann nur "erfolgreich" sein, wenn das Programm auf genügend vielen Rechnern im Internet im gewünschten Zeitraum gestartet wird. Dazu wird das entsprechend vorkompilierte Programm im Internet angeboten, in der Hoffnung, daß möglichst viele Menschen das Programm herunterladen und anwenden.

2.7 Praktische Aufgaben

2.7.1 Scanner und Passwortcracker

1. Installieren Sie `nmap`, `nessus`, `whisker` und `crack` entweder als SuSE Paket (wenn vorhanden) oder durch Kompilieren des Quellcodes. Eventuell muß dazu noch das Kommando `make` im `YaST` installiert werden.

2. Scannen Sie mittels `nmap` die `192.168.216.253` , `192.168.216.128/26` und sich selbst:

   (a) ohne Optionen

   (b) mit einem Fingerprint

   (c) Port 20 bis 1000

   (d) mit einem FIN Scan

   und vergleichen Sie die System-Logdatei-Einträge des Ziel-Systems mit den Shell-Ausgaben von `nmap`. Was ist festzustellen und wie ist das zu interpretieren?

3. Aktivieren Sie beim Nessus alle Plugins außer "Denial of Service" und scannen Sie sich selbst. Speichern Sie den erzeugten Report ab. Wie ist das Ergbnis zu interpretieren?

4. Machen Sie mit Whisker einen HEAD und einen GET Scan Ihrer Maschine und vergleichen Sie die Logfileeinträge.

5. Legen Sie auf Ihrer Maschine drei Dummy User mit unterschiedlich schwierigen Passwörtern an. Erzeugen Sie aus der `/etc/passwd` und `/etc/shadow` das Inputfile und lassen Sie den Passwortcracker Crack5.0 laufen. Wie lange hat der Cracker gebraucht? Was läßt sich somit über die Beschaffenheit von Passwörtern sagen?

2.7.2 Rootkit

Auf dem Rechner `hacktest` (`192.168.216.252`) ist ein Rootkit installiert. Auf dem Rechner können Sie sich mit dem Benutzer `secpgast`, Passwort `pcsec` einloggen, das `root`-Passwort lautet `y;x:c_v,b.n-` .

1. Versuchen Sie, das Rootkit zu entdecken und soviele Informationen wie möglich über das Rootkit zu sammeln (dazugehörige Dateien, Prozesse, Backdoors, gesammelte Informationen).

2. Wie haben Sie den Rechner untersucht?

3. Was würden Sie als Reaktion auf das entdeckte Rootkit vorschlagen?

Sie können auf dem Rechner `hacktest` zur Lösung der Aufgaben beliebige Programme installieren und den Rechner nach Ihren Vorstellungen untersuchen. Sollten Sie das Rootkit entdecken verändern Sie es nicht, damit Ihre Kollegen noch was zu suchen haben. Sprechen Sie sich ggf. mit anderen auf dem Rechner eingeloggten Anwendern ab, bevor Sie den Rechner z.B. neu starten oder Software installieren.

2.7.3 DoS-Werkzeuge

1. Beschäftigen Sie sich mit drei der vorgestellten Programme. Kompilieren sie diese und versuchen Sie, die beschriebene Attacke durchzuführen.

2. Zeichnen Sie die von den Werkzeugen generierten Datenpakete auf und interpretieren diese.