W i l l k o m m e n !

Liebe Leserin, lieber Leser, dieses Dokument dient Ihnen als begleitende Information zum ebenfalls online verfügbaren Vortrag „Ermittlungsmöglichkeiten bei Linux“, der am 31.03.2004 im Rahmen der „Arbeitstagung Computerkriminalität, IAF NRW“ in Neuss gehalten wurde.
Beides zusammen, Folien und begleitende Information, ermöglicht jedem Interessierten, den Inhalt selbst nachzuvollziehen.

I Was macht TronicGuard?

Die TronicGuard GmbH tritt als Internet Service Provider für individuelle Hostingdienstleistungen und als Dienstleister im Bereich der IT-Security auf.
TronicGuard bietet Support für Unixoide Betriebssysteme der xBSD-Reihe und vertreibt Desktop- und Mini-Server-Systeme auf Basis der leistungsfähigen VIA Epia-Reihe.
TronicGuard bietet Content Management Systeme an. Im jüngsten Projekt wurde eine CMS-Umgebung für die Werkstätten eines europaweit vertretenen KFZ-Systemverbunds implementiert und an diese beworben.
Dabei bedient die TronicGuard hauptsächlich Kunden aus dem mittelständischen Unternehmensbereich.

II OpenBSD stellt sich vor

Was ist OpenBSD? [1], [2] (Quellenangaben „[x]“: Seite 13)

OpenBSD ist wie Linux ein Unixoides Betriebssystem und kann auf vielen Hardware-Plattformen eingesetzt werden (alpha, amd64, hp300, hppa, i386, macppc, sparc, sparc64 und ein paar weitere).
OpenBSD basiert auf ANSI- und POSIX-Standards und unterstützt damit fast alle gängigen Anwendungen, die von Linux bekannt sind. OpenBSD besteht wie Linux zum größten Teil aus standardisiertem C/C++ und daher lassen sich viele POSIX/ANSI-Anwendungen direkt unter OpenBSD kompilieren.
OpenBSD gilt als eines der sichersten, wenn nicht sogar als das sicherste open-source Betriebssystem. In den letzten 7 Jahren wurde lediglich eine remote ausnutzbare Sicherheitslücke in der Standardinstallation entdeckt.

Woher kommt OpenBSD? [1]

1995 begann der Amerikaner Theo de Raadt sein Projekt OpenBSD. Der Kern der Entwicklergemeinde ist in Amerika und Canada. Es gibt aber auch Entwickler in Russland, Australien, Deutschland und anderen Ländern.
OpenBSD basiert auf BSD 4.4-Lite2 der University of California in Berkeley und hat damit seinen Ursprung in den 70er Jahren im von AT&T entwickelten UNIX.

Was kostet OpenBSD? [1], [3]

OpenBSD ist ein open-source Betriebssystem und kann daher ohne Lizenzkosten verwendet werden.
Das Betriebssystem wird unter der BSD-Lizenz veröffentlicht. Die BSD-Lizenz ist noch liberaler als die von Linux bekannte GPL (Gnu Public License) und erlaubt, alles mit dem angebotenen System und dessen Quellcode zu tun, solange der Name des Urhebers und dessen Copyright-Notiz in die weiterentwickelte Software übernommen wird, und solange er nicht für die angebotene Software haftbar gemacht wird.

Beispiel:

Wer verwendet OpenBSD? [4]

OpenBSD wird von vielen kommerziellen und nichtkommerziellen Anwendern eingesetzt. Firmen wie Adobe Systems und Alteon Networks, Internet Service Provider wie Compartment und universitäre Einrichtungen wie die University of Michigan nutzen das Betriebssystem hauptsächlich im Sicherheitsbereich, aber auch z.B. als Datebank- oder Webserver.

TronicGuard setzt OpenBSD für alle Zwecke ein. Die angebotenen Hostingdienstleistungen wurden komplett mit OpenBSD als Basis realisiert.

Was kann OpenBSD besonders gut ?[1]

OpenBSD kann als Server-System (z.B. als Firewall, Webserver oder E-Mail-Server), aber auch als Desktop-System mit einer grafischen Oberfläche und Büroanwendungen verwendet werden.
Von besonderem Interesse sind natürlich die enthaltenen Sicherheitsfunktionen:

Firewalls mit OpenBSD

OpenBSD enthält den Paketfilter pf und ermöglicht damit den Einsatz leistungsfähiger und zugleich kostengünstiger Firewalls. Pf wird über eine klar strukturierte und leicht zu beherrschende Befehlssyntax bedient und bietet aktuelle Funktionen wie Stateful Packet Inspection (SPI), Traffic-Shaping, Redundanz und Verbindungsweiterleitung (NAT).

Bridges mit OpenBSD

OpenBSD kann, als Bridge konfiguriert, einzelne Hosts (Server) oder auch ein ganzes Netzwerk transparent überwachen und filtern.
Dafür werden zwei Ethernetkarten so konfiguriert, dass Datenpakete, die an einer Karte ankommen, an die jeweils andere Ethernetkarte weitergeleitet werden und umgekehrt. Dabei passieren alle Datenpakete den Paketfilter pf und können so gefiltert, gespeichert oder verändert werden.
Es wird also ein Computer für andere Computer transparent in die Datenleitung eingehängt und kann diese überwachen.

Vorteile durch den Einsatz von OpenBSD

OpenBSD ist ein sehr „schlankes“ Betriebssystem, dessen Basis-Installation sich auf ein Minimum gut ausgesuchter, sicher vorkonfigurierter Software beschränkt. Das Betriebssystem ist bekannt für seine sehr gute Dokumentation, und es lässt sich schnell, leicht und sicher konfigurieren.
Über 1000 verbreitete Programme sind zum sofortigen Einsatz vorbereitet.

Durch die gute Integration sicherheitsrelevanter Dienste (z.B. pf, kerberos, openssl, openssh) lassen sich effektiv und kostensparend sichere Netzwerkdienste einrichten und bestehende Netzwerke absichern.

Durch die extrem liberale Lizenz kann das System zu jedem beliebigen Zweck verwendet und verändert werden. Auch proprietäre closed-source Lösungen sind ohne Probleme möglich.

III Linux stellt sich vor

Linux in Worten und Zahlen [6], [7], [8]

IDC sagte im September 2002 ein Wachstum der Anzahl an installierten Linux-Servern im Zeitraum 2001 bis 2006 von 24,7% voraus. Damit werden 2006 voraussichtlich 6,6 Millionen Server Linux verwenden.
Im Jahr 2002 hatte Linux bei neuen Server-Lizenzen einen Anteil von 23,1% und bei neuen Client-Lizenzen 2,8%.
Die Zeitschrift c't zählte bereits 2001 15 Millionen Linux-Anwender.
Über 10.000 Entwickler arbeiten an Linux.
Der aktuelle Linux-Kernel 2.6.4 besteht aus ungefähr 6 Millionen Zeilen Quellcode von insgesamt 192MB Größe.

Was kann Linux im Vergleich zu OpenBSD?

Linux ist also erheblich weiter verbreitet als OpenBSD, und es gibt eine enorme Fülle an Distributionen (SuSE, RedHat, Debian, Slackware, Mandrake, ...), Entwicklern und Firmen, die Support für Linux leisten können.
Daher unterstützt Linux eine breite Palette an Hardware und läuft vom Kleinstrechner im embedded-Bereich bis zum Cluster aus tausenden von Servern, bestehend aus Mehrprozessorsystemen.
Die Firma FSMLabs bietet mit RTLinux Pro ein Echtzeit-Betriebssystem an (und mit RTCore/BSD auch eines auf BSD-Basis)[5].

Linux als Diagnosesystem [10]

Linux eignet sich aufgrund seiner Vielseitigkeit und hohen Flexibilität sehr gut als Diagnosesystem. Die meisten Programme können kostenlos eingesetzt werden und sind im Quellcode verfügbar.

Es gibt aber in der Regel niemanden, der für Probleme mit einem open-source Programm zur Verantwortung gezogen werden kann.
Da die Programme im Quellcode verfügbar sind, kann ein vor Gericht Beschuldigter nach eventuellen Fehlern in den verwendeten Programmen suchen und damit die Ermittlungsergebnisse bei einem Fehler in Frage stellen. Auf der anderen Seite kann durch die Einsehbarkeit des Codes die Qualität und Verwendbarkeit des Programms gezeigt werden.

Linux und entsprechende open-source Tools sind oft komplizierter zu bedienen als kommerzielle Pakete und oft muss auf der Kommandozeile gearbeitet werden.
Selbst kleine Fehler können unter Umständen schwerwiegende Folgen haben und z.B. die zu sichernde Festplatte überschreiben.

Besonders interessant für Ermittlungsarbeiten sind komplette Systeme, die direkt von einer CD booten können.

Die **Beschreibung zu Knoppix**, einer Linux-Distribution von Dipl. Ing. Klaus Knopper:

„KNOPPIX ist eine komplett von CD lauffähige Zusammenstellung von GNU/Linux-Software mit automatischer Hardwareerkennung und Unterstützung für viele Grafikkarten, Soundkarten, SCSI- und USB-Geräte und sonstige Peripherie. KNOPPIX kann als Linux-Demo, Schulungs-CD, Rescue-System oder als Plattform für kommerzielle Software-Produktdemos angepasst und eingesetzt werden. Es ist keinerlei Installation auf Festplatte notwendig. Auf der CD können durch transparente Dekompression bis zu 2 Gigabyte an lauffähiger Software installiert sein.“

Knoppix gehört zu den Linux Distributionen, die auf einer Live-CD angeboten werden. Diese Systeme eignen sich hervorragend für Arbeiten an PC deren Festplatteninhalt nicht verändert werden soll oder darf.

Knoppix-STD – Ein „Security-Linux-von-CD“ [14]

Knoppix-STD basiert auf Knoppix und wurde angepasst von dem Amerikaner Jason Liller.
Das System beinhaltet viele sicherheitsrelevante Tools, die es erlauben Netzwerke zu überwachen und Beweissicherung auf Computern durchzuführen.
Knoppix-STD wird, so wie Knoppix, zum kostenlosen Download unter der GPL-Lizenz angeboten.
Ein ähnliches Projekt, „FIRE“, wird unter [11] zur Verfügung gestellt.

Knoppix-STD startet automatisch von der CD und bietet die Möglichkeit, zwischen einer reinen Text-Konsole und einer grafischen Oberfläche zu wählen.
Dabei erkennt und konfiguriert es, soweit möglich, automatisch die vorhandene Hardware.

IV Sicherung einer Festplatte zur weiteren Untersuchung [12]

Wie die meisten unixoiden Betriebssysteme enthält Knoppix-STD das Programm „dd“, mit dem unter anderem bitgenaue Kopien von Datenträgern angefertigt werden können.
Linux benennt im System installierte IDE Festplatten nach dem Schema „hdX“ mit hda für die erste Festplatte, hdb für die zweite und so weiter. Installierte SCSI-Festplatten werden nach dem Schema „sdX“ benannt.

„dmesg“ zeigt die installierten Festplatten an:

hda: IC25N040ATCS04-0, ATA DISK drive
hda: attached ide-disk driver.
hda: 71969781 sectors (36849 MB) w/1768KiB Cache, CHS=4759/240/63, UDMA(100)
(Ausschnitt der relevanten Teile aus der dmesg-Ausgabe)

Die Partitionen einer Festplatte adressiert Linux mit einer angehängten Zahl. Die erste Partition der ersten Festplatte heisst hda1, die zweite hda2, usw.

root@0[knoppix]# fdisk /dev/hda

The number of cylinders for this disk is set to 2584.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 20.0 GB, 20003880960 bytes
240 heads, 63 sectors/track, 2584 cylinders
*Units = cylinders of 15120 * 512 = 7741440 bytes*

```
   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1       948   7164958+   7  HPFS/NTFS
/dev/hda2           948      1625   5119506    c  Win95 FAT32 (LBA)
/dev/hda3          1625      2584   7250544   a5  FreeBSD

Command (m for help):
```

Eine Partition kann über ihr Block-Device mit dd ausgelesen werden.
Folgender Befehl: *dd if=/dev/hda1 of=/backup/hda1.img*
beispielsweise würde die erste Partition der ersten IDE-Festplatte komplett auslesen und in die Datei hda1.img im Verzeichnis /backup schreiben.

Da wahrscheinlich kein lokaler Massenspeicher mit ausreichender Kapazität zur Verfügung steht, macht es Sinn, das Festplatten-Image auf einem entfernten System abzulegen:
dd if=/dev/hda1 |ssh Benutzer@Host 'cat >/ordner/hda1.img'

Mit dem oben gezeigten Aufruf kann das Image problemlos auch über das Internet übertragen werden. Da ssh verwendet wird, werden die Daten verschlüsselt übertragen.

Weitere Informationen über dd befinden sich in den Man-Pages:
man dd

V Verwendung einer solchen Sicherung (Sicherstellen von Bildern) [21], [22]

Das Abbild der Partition kann sowohl mit Linux als auch mit Windows-Programmen untersucht werden.
Da beim normalen Löschen einer Datei nur der auf diese Datei referenzierende Eintrag z.B. in der FAT (File Allocation Table), nicht jedoch die Datei selbst gelöscht wird, sind im Image auch alle gelöschten Dateien enthalten, die noch nicht von neuen Daten überschrieben wurden.

Unter Linux wird ein Image als „Loop-Back-Device“ eingebunden:
mount -o ro,loop,nodev,noexec /ordner/hda1.img /mount_point

Das Abbild ist nun einsatzbereit und kann untersucht werden.
Es ist auch möglich, eine komplette Festplatte inklusive aller ihrer Partitionen und ihrem MBR (Master Boot Record) auszulesen und einzubinden. Ausführliche, weitergehende Informationen finden sich dazu unter [13].

Jetzt werden vom Original, von der Image-Datei und allen darin enthaltenen Dateien MD5-Prüfsummen erstellt:

md5 /dev/hda1
md5 /ordner/hda1.img

Die Ausgabe für das Disk-Image wird so aussehen (Mit einem anderen Wert):

MD5 (/ordner/hda1.img) = fbf610de95f3c89f0c44z37a6d142b87

Es ist wichtig, dass die Prüfsummen von Original und Abbild identisch sind, denn nur dann liegt eine identische Kopie vor, mit der ordnungsgemäß gearbeitet werden kann.

find /mount_point -type f -exec md5 {} \; > /data/md5.all
Der letzte Befehl ermittelt die MD5-Summen aller Dateien im Festplatten-Abbild und schreibt diese in die Datei /data/md5.all

Nun können zum Beispiel alle auf dem PC gespeicherten Bilder unabhängig ihres Dateinamens und ihrer Erweiterung (also alle jpg-, gif-, bmp-, tif-, png-Bilder) folgendermaßen ermittelt werden:
find ./ -type f -exec file {} \; |egrep 'image|bitmap'

Die gefundenen Dateien werden dann angezeigt:

./images/apache_pb.gif: GIF image data, version 89a, 259 x 32
./images/mysql.png: PNG image data, 167 x 87, 8-bit colormap, non-interlaced
./images/mod_ssl_sb.gif: GIF image data, version 89a, 102 x 47

Diese Möglichkeit soll nur zur Anregung dienen!
Das Tool „foremost“ des United States Air Force Office of Special Investigations scannt ein komplettes Image oder auch direkt das Block-Device einer Festplatte anhand konfigurierbarer Header- und Footer-Bereiche beliebiger Dateien und findet damit Bilddateien auch in gelöschten oder versteckten Bereichen einer Festplatte und kopiert die Fundstücke direkt in ein beliebiges Verzeichnis.
Foremost ist (in einer etwas älteren Version) in Knoppix-STD enhalten.

Das National Institute of Standards and Technology veröffentlicht mit der National Software Reference Library regelmäßig aktuelle Prüfsummen bekannter Dateien zum Downloaden. Damit kann unter Umständen ein großer Teil der Bilder bzw. der zu untersuchenden Dateien von vornherein von der weiteren Untersuchung ausgeschlossen werden, denn mit den Prüfsummen lassen sich leicht Dateien finden, die zum Betriebssystem oder zu Standardsoftware gehören.
Insgesamt werden Prüfsummen für 2600 Softwarepakete angeboten.

VI Sicherheitsrisiken im Web

Durch die zunehmende Verbreitung des Internet und die zunehmende Komplexität der zugrunde liegenden Techniken werden Sicherheitsprobleme öfter, schwerwiegender und komplexer.
Ein Problem im Bereich des WWW soll hier kurz vorgestellt werden:

Cross-Site-Scripting [24]

Cross-Site-Scripting-Attacken sind für die betroffenen Benutzer einer Website unter Umständen schwer erkennbar, aber trotzdem sehr riskant. Sie können aber auch den Betreiber der anfälligen Site in Bedrängnis bringen.
Beispiel: http://www.xintrix.de/polnrw_css

Bei einer XSS-Attacke erlaubt eine Website das Einbinden und Ausführen/Übertragen fremden Codes in den eigenen. Wenn dies zum Beispiel in der Sicherheitszone einer Bank geschieht, dann hat der fremde Code damit unter Umständen Zugriff auf gespeicherte Cookies des Clients, die eventuell die aktuelle Session-ID enthalten, oder aber der fremde Code kann auf die betroffene Webseite Einfluss nehmen.

VII Netzwerküberwachung mit dsniff [15], [16]

Dsniff ist ein Programmpaket von Dug Song und ist im Quellcode und als fertiges Binärpaket für einige Betriebssysteme verfügbar.
Mit dsniff kann ein Netzwerk passiv „belauscht“ und aktiv beeinflusst werden.

Die Original-Beschreibung aus den FAQ:

dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI.

Dsniff ist in Knoppix-STD bereits enthalten.

Passwörter, URLs, E-Mails, IRC-Nachrichten und ICQ-Nachrichten im LAN abfangen

Ethernet-Netzwerken liegt eine logische Busstruktur zugrunde, bei der ein Datenpaket von allen angeschlossenen Netzwerksschnittstellen (z.B. Computern) empfangen wird, aber nur von dem im Paket enthaltenen Ziel (dem vorgesehenen Empfänger) akzeptiert, von allen anderen jedoch wieder verworfen wird.
BNC-verkabelte Netze sind auch physisch als Bus ausgelegt. In einem sternförmigen Netz mit einem Hub übernimmt dieser die logische Busstruktur, in dem er alle Pakete an alle angeschlossenen Anschlüsse kopiert.

Sobald eine Netzwerkschnittstelle in den Promiscious-Modus versetzt wird, hört sie auf, die nicht für sie bestimmten Pakete zu verwerfen und leitet sie an das Betriebssystem weiter.

Nun kann der Datenstrom ausgewertet werden.

Mit „dsniff“ kann nach im Klartext übertragenen Passwörtern gesnifft werden. Dsniff beherrscht dafür über 30 Protokolle (z.B. FTP, Telnet, Microsoft SMB, NFS, SNMP, SMTP, POP, IMAP) und zeigt automatisch die erkannten Passwörter an.

„mailsnarf“ speichert übertragene E-Mails im mbox-Format, so dass die gesnifften Mails komfortabel in einem normalen Mail-Client betrachtet werden können.

„msgsnarf“ findet übertragene Nachrichten von AOL Instant Messenger, ICQ 2000, IRC, MSN Messenger, und Yahoo Messenger und zeigt diese an.

„urlsnarf“ zeichnet alle von Web-Browsern angeforderten Web-Adressen (URLs) auf.

„filesnarf“ kann komplette per NFS (Network File System) übertragene Dateien herausfiltern und speichern.

Swichted Networks verwenden keinen Hub sondern einen Switch. Switches erkennen das Ziel eines Datenpakets anhand der Ziel-MAC (Media Access Control, Beispiel: *00:40:63:c1:1c:ee*) und leiten dieses Datenpaket nicht an alle angeschlossenen Geräte, sondern nur an das mit der passenden MAC. Dadurch baut ein Switch eine exklusive Verbindung zwischen Sender und Empfänger auf – dsniff wird also nicht mehr die Daten des gesamten Netzwerkes scannen können.

Um doch wieder an die fremden Daten heranzukommen, kann man versuchen mit „macof“ den MAC-Adressen-Speicher eines Switches mit extrem vielen zufällig generierten MAC-Addressen zu überfluten. Wenn dieser „fail-open“ arbeitet, dann wird er in diesem Zustand wie ein Hub alle Datenpakete an alle Anschlüsse weiterleiten.

Der zweite Weg beeinflusst gezielt nur den Host, der überwacht werden soll.
Computer speichern in ihrer lokalen ARP-Tabelle (Address Resolution Protocol) welche MAC-Adresse zu welcher IP-Adresse gehört, damit sie wissen, welchen Computer sie auf Netzebene für eine bestimmte IP ansprechen müssen:

C:\>arp -a
Schnittstelle: 192.168.0.50 on Interface 0x1000003

Internetadresse	*Physikal. Adresse*	*Typ*
192.168.0.100	*00-40-33-a4-1w-0f*	*dynamisch*
192.168.0.200	*00-40-33-8e-3e-11*	*dynamisch*

Das ARP-Protokoll ist verbindungslos und sieht vor, dass Computer neue Zuordnungen speichern, die ein anderer Computer im LAN ihnen zusendet, auch wenn sie dazu vorher keine Anfrage gestellt haben. Man kann also ungefragt bei einem Computer einen bisherigen Eintrag mit einer anderen MAC-Adresse überschreiben, solange dieser Eintrag nicht als „statisch“ in dem betreffenden System konfiguriert ist.

Um nun den Traffic eines Hosts im LAN abzufangen, überschreibt man bei diesem den ARP-Eintrag für das Gateway mit der eigenen MAC-Adresse und erreicht damit, dass der „vergiftete“ (ARP-Poisoning) Host nun die Daten statt an das Gateway, an den eigenen Rechner schickt. Dieses Vorgehen wird „arp-spoofing“ genannt.
Damit die Verbindung nicht abgebrochen wird, muss am eigenen PC IP-Forwarding aktiviert werden, damit diese Pakete an das eigentliche Gateway weitergeleitet werden.

Mit dem folgenden Befehl kann bei Linux-Systemen Forwarding aktiviert werden:
echo 1 > /proc/sys/net/ipv4/ip_forward

Jetzt fließen wieder fremde Daten durch den eigenen Computer, die mit den vorher genannten Tools untersucht werden können.
Für den angegriffenen Host ist dieser Vorgang bis auf den geänderten ARP-Eintrag transparent, da er ja der Meinung ist, seine Daten an den richtigen Kommunikationspartner zu senden.

Verbindungen stören und DNS-Anfragen umleiten

„tcpkill“ kann bestehende TCP-Verbindungen gewaltsam beenden und „tcpnice“ kann TCP-Verbindungen verlangsamen. UDP und ICMP sind davon jedoch nicht betroffen.
„dnsspoof“ kann DNS-basierte Authentifizierungen unterwandern, in dem Namensauflösungsanfragen (also welche IP sich hinter einem FQDN verbirgt) abgefangen und durch gefälschte Antworten ersetzt werden, die die IP des eigenen Computers enthalten. Dieses Vorgehen ist für MITM-Attacken gegen SSH und SSL-Verbindungen wichtig.

Schutz gegen dsniff

Sicherung auf Netzebene

Der erste Schritt zur Sicherung gegen Angriffe wie von dsniff ist der Einsatz von Switches, denn dann muss ein Angreifer auf Methoden wie arpspoofing oder MAC-flooding zurückgreifen.
Wenn die MAC-Adresse für das Gateway bei allen Clients statisch konfiguriert wird, ignorieren diese eventuelle ARP-Pakete mit anderslautenden MAC-Adressen für dessen IP und sind daher nicht mehr anfällig.
Ausserdem hinterlassen Methoden wie ARP-Spoofing und MAC-Flooding eindeutige Spuren im Netzwerk. Das Tool „arpwatch“ erkennt neue MAC-Adressen in einem Netzwerk und kann per E-Mail Alarm schlagen.
Netzwerke können ebenfalls durch Messung der Änderung von Latenzzeiten auf sniffing untersucht werden.

Verwendung von Verschlüsselung

Wo immer es möglich ist, sollten Netzwerkprotokolle durch verschlüsselte Varianten ersetzt werden. Denn dsniff und die *snarf-Tools erkennen nur im Klartext übertragene Daten.
Verschlüsselte Daten können zwar auch mitgeschnitten werden, deren Verschlüsselung setzt jedoch eine sehr hohe Hürde, wenn sie korrekt durchgeführt wird.
Einen 128Bit-Schlüssel per Brute-Force zu knacken dauert selbst mit einem Rechner-Cluster, der 10^18 (1.000.000.000.000.000.000) Schlüssel pro Sekunde testen könnte im Mittel ungefähr 5.000.000.000.000 Jahre, vorausgesetzt, der Verschlüsselungsalgorithmus hat keine Schwachstelle, die per Krypto-Analyse ausgenutzt werden könnte. [17]

VIII IT im Bereich der Internet Service Provider

Anfallende Verbindungsdaten (Logfiles)

Anfallende Verbindungsdaten werden meist direkt zum Zeitpunkt ihres Entstehens in Logfiles festgehalten.
Im Hosting-Umfeld fallen in der Regel hautpsächlich Logfiles für FTP-Verbindungen und -Übertragungen, Logfiles von Transaktionen mit den verwalteten Webservern und Logfiles der E-Mail-Server (SMTP und POP/IMAP) an.
Zusätzlich protokollieren Infrastruktur-Programme wie Firewalls, NIDS-Alarmsysteme, NTP-Zeitserver und SSH-Server anfallende Daten, und Datenbanksysteme speichern eventuell ganze Transaktionen.
Dabei sind die entstehenden Protokolldateien oft in einem Programm-individuellen Format gestaltet.

Webserver-Log [18], [19]

Im August 2003 hat Netcraft über 42.000.000 Webseiten untersucht und einen Marktanteil von fast 64% für den open-source Server „Apache“ festgestellt.
Besonders verbreitet ist Apache bei Webhostern, da durch seinen Einsatz keine Lizenzkosten entstehen.
TronicGuard verwendet ausschließlich Apache.
Damit liegt ein Großteil der HTTP-Verbindungsdaten im für Apache gebräuchlichen und von der W3C (World Wide Web Consortium) definierten CLF (Common Log Format) vor.

E-Mail-Server-Log [20]

E-Mails werden nach dem SMTP-Standard ausschließlich von MTAs (Mail Transfer Agent = E-Mail-Server) untereinander ausgetauscht. Damit ein Benutzer mit seinem MUA (Mail User Agent, z.B. Outlook) am E-Mail-Verkehr teilnehmen kann, übergibt er eine zu versendende E-Mail an den für ihn zuständigen MTA, damit dieser die Mail an den passenden Empfänger-MTA ausliefern kann. Der empfangende MTA speichert die angenommene Mail dann in der Mailbox des Empfängers.
Der Empfänger kann die an ihn gerichtete Mail mit einem Webmailer oder einem IMAP-Fähigen MUA direkt auf dem Server lesen oder sie mit dem POP-Protokoll vom MTA herunterladen und in seinem MUA speichern.

POP/IMAP dienen also nur dem „Zugänglich-machen“ einer bereits empfangenen und auf dem Server gespeicherten Mail.

Alle relevanten Daten über versendete und empfangene E-Mails finden sich daher in den Protokoll-Dateien des MTA.
Die POP/IMAP-Logs ermöglichen die Feststellung, ob und wann ein Benutzer seine Mails gelesen hat.

TronicGuard verwendet als Mail-Server „exim“ und als POP/IMAP-Server das „courier“-Paket – beide ebenfalls open-source.

Informationsgehalt der Verbindungsdaten

Log eines MTA's (exim)

Folgende Log-Einträge sind ein Auszug aus der Hauptprotokolldatei „exim_main“ des E-Mail-Servers. Dieser Auszug enthält nur Einträge eines bestimmten E-Mail-Nutzers. Es ist also sein gesamter E-Mail-Verkehr in einem gewissen Zeitraum zu sehen:

2004-03-27 11:18:04 1B7At9-0003Jy-00 <= owner-misc+M53613@openbsd.org H=openbsd.cs.colorado.edu [128.138.207.242] P=esmtp X=TLSv1:AES256-SHA:256 S=1883 id=406552D9.3040904@mindrot.org
2004-03-27 11:18:04 1B7At9-0003Jy-00 => abc1234 <wundram@tronicguard.com> D=procmail T=procmail_pipe
2004-03-27 11:18:04 1B7At9-0003Jy-00 Completed

Diese Zeilen beschreiben eine E-Mail die von „owner-misc...“ vom Server „openbsd.cs.colorado.edu“ am 27.03.2004 um 11:18 Uhr empfangen wurde. Die E-Mail hat die eindeutige ID „406552D9.3040904@mindrot.org“. Diese ID generiert der Versender und ist durch die enthaltene Absender-Domain einmalig. Das Übertragungsprotokoll war „esmtp“ (Extended Simple Mail Transfer Protocol) und der Datenstrom wurde zwischen dem sendenden und dem empfangenden Mail-Server per TLS (Transport Layer Security) verschlüsselt. Die E-Mail ist 1883 Byte groß und wurde dem Postfach abc1234 (dieses Postfach ist für wundram@tronicguard.com zuständig) zugeordnet. Die Punkte D=procmail und T=procmail_pipe zeigen an, dass die Mail auf dem empfangenden Server dem Programm „procmail“ (einem verbreiteten Mail-Filter-Programm) übergeben wurde. Procmail hat dann die abschließende Speicherung der Mail übernommen (in dem Fall: Ablegen in der Mailbox).
Um 11:18:04 wurde der gesamte Vorgang erfolgreich abgeschlossen.

2004-03-27 14:19:52 1B7DgE-0007nu-00 <= wundram@tronicguard.com H=p508d14d2.dip.t-dialin.net ([192.168.0.5]) [80.141.20.210] P=asmtp X=TLSv1:RC4-MD5:128 A=login:abc1234 S=3387961 id=200403271410.55140.wundram@tronicguard.com
2004-03-27 14:19:53 1B7DgE-0007nu-00 => abc4321 <peter.pan@tronicguard.com> D=localuser T=local_delivery
2004-03-27 14:19:53 1B7DgE-0007nu-00 Completed

Hier hat „wundram@tronicguard.com“ eine E-Mail versendet. „H=p508d14d2.dip.t-dialin.net ([192.168.0.5]) [80.141.20.210]“ gibt auch hier wieder den Absender-Host an, wobei zusätzlich die vom E-Mail-Client übergebene Adresse 192.168.0.5 enthalten ist. Diese Information ist für den eigentlichen Versand jedoch überflüssig und dient daher nur der Möglichkeit, die E-Mail zurück verfolgen zu können). Auch hier wurde der Datenstrom zwischen MUA und MTA TLS-verschlüsselt. Zusätzlich hat sich der MUA gegenüber dem MTA mit einem Benutzernamen und einem Passwort legitimiert (P=asmtp und A=login:abc1234).
Die E-Mail wurde an peter.pan@tronicguard.com geschickt, und da der gleiche Server für diese Adresse zuständig ist, wurde sie direkt lokal in das Postfach abc4321 übergeben (D=localuser T=local_delivery).
Auch diese Transaktion wurde erfolgreich beendet um 14:19:53 Uhr.

2004-03-27 15:28:25 1B7EnR-0001HE-00 <= abcdefgh@gmx.de H=pop.gmx.net (mail.gmx.net) [213.165.64.20] P=smtp S=1622 id=25286.1080397734@www39.gmx.net
2004-03-27 15:28:25 1B7EnR-0001HE-00 => abc1234 <wundram@tronicguard.com> D=procmail T=procmail_pipe
2004-03-27 15:28:25 1B7EnR-0001HE-00 Completed

Hier wurde eine E-Mail von „abcdefgh@gmx.de“ an „wundram@tronicguard.com“ geschickt. Der versendende MTA hatte die IP 213.165.64.20. Sein übergebener FQDN (Fully Qualified Domain Name) wich jedoch vom tatsächlichen ab: mail.gmx.net und nicht pop.gmx.net.
Bei der Übertragung kamen weder Verschlüsselung noch Authentifizierung zum Einsatz und daher einigten sich beide Server auf das Protokoll „SMTP“ (P=smtp)

2004-03-27 16:33:36 1B7FoV-0005R3-00 <= owner-misc+M53617@openbsd.org H=openbsd.cs.colorado.edu [128.138.207.242] P=esmtp X=TLSv1:AES256-SHA:256 S=2448 id=4065972E.20907@telia.com
2004-03-27 16:33:36 1B7FoV-0005R3-00 => abc1234 <wundram@tronicguard.com> D=procmail T=procmail_pipe
2004-03-27 16:33:36 1B7FoV-0005R3-00 Completed

Log eines Webservers (apache)

Das folgende Log-File des Web-Servers Apache protokolliert die Seitenabrufe für http://www.tronicguard.com. (Es wurden lediglich einige Einträge als Beispiel herausgegriffen.)

64.68.89.171 - - [27/Mar/2004:07:22:37 +0100] "GET /menu.js HTTP/1.1" 200 4850 "-" "Googlebot/Test"

Dieser Eintrag wurde von einem Computer mit der IP „64.68.89.171“ erzeugt, der am 27.03.2004 um 07:22:37 Uhr (der Zusatz „+0100“ signalisiert die Zeitzone „Europa/Berlin“ die Datei „menu.js“ aus dem Stammverzeichnis des Servers (Also: http://www.tronicguard.com/menu.js) angefordert hat. Dabei hat sich der Client als „Googlebot/Test“ zu erkennen gegeben. Der Statuscode ist „200“, die Anfrage war also erfolgreich. Dabei wurden 4850 Bytes übetragen.

si1006.inktomisearch.com - - [27/Mar/2004:12:11:22 +0100] "GET /robots.txt HTTP/1.0" 200 71 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)"
j3103.inktomisearch.com - - [27/Mar/2004:12:11:23 +0100] "GET /index.php?pid=01 HTTP/1.0" 200 8443 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)"

Hier hat der Webcrawler der Yahoo-Gruppe die Datei „robots.txt“ angefordert. Webcrawler bzw. Robots gehören in der Regel zu Suchmaschinenbetreibern und suchen für deren Datenbestand nach aktualisierten Informationen. Mit der Datei robots.txt können Webseitenbetreiber die Arbeitsweise der Robots steuern und bestimmte Bereiche vom „spidering“ ausschließen oder eine Seite für einen bestimmten Robot komplett sperren.
Natürlich muss sich ein Robot nicht an so eine Datei halten, daher handelt es sich hier eher um enthaltene Richtlinien.

211.XXX.XX.XX - - [27/Mar/2004:13:06:57 +0100] "GET / HTTP/1.1" 200 673 "http://www.website.org/site.html" "Mozilla/4.0 (compatible; MSIE 5.00; Windows 98"

Hier hat ein Client den Document-Root angefordert (http://www.tronicguard.com). Dabei wurden lediglich 673 Bytes Daten übertragen. Zusätzlich hat der Client einen Referer übermittelt: http://www.website.org/site.html
Dieser Referer zeigt an, auf welcher Seite sich der Surfer vorher befunden hat, woher er also gekommen ist.

XXX.XXX.lsu.edu - - [27/Mar/2004:15:21:11 +0100] "GET /scripts/nsiislog.dll" 404 - "-" "-"

Hier hat ein Client zwar eine gültige Anfrage gestellt. Der gewünschte Inhalt konnte jedoch nicht auf dem Server gefunden werden, und daher antwortete der Webserver mit dem Statuscode 404: „Page not found“.
Referer und User Agent wurden nicht übermittelt.
Hier handelt es sich augenscheinlich um eine ungewöhnliche Anfrage, bei der wahrscheinlich ein Fehler bei einem anderen Webserver ausgenutzt werden sollte.

xxxxx.dip.t-dialin.net - user1234 [27/Mar/2004:21:12:15 +0100] "GET /logs/ctry_usage_200403.png HTTP/1.0" 200 3541 "http://www.tronicguard.com/logs/usage_200403.html" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"

Dieser Eintrag zeigt, dass die angeforderte Ressource „logs/ctry_usage_200403.png“ von einem Client stammt, der sich gegenüber dem Server als Benutzer „user1234“ mit dem dazu passenden Kennwort identifiziert hat. Es handelte sich hier also um einen passwortgeschützten Bereich.

E-Mails „abhören“

Um E-Mails von und an eine oder mehrere E-Mail-Adressen bzw. E-Mail-Postfächern „abzuhören“, kann man aus Sicht des Internet Service Providers auf zwei verschiedene Arten vorgehen:

Anpassung des E-Mail-Servers (MTA-Spezifisch)

Diese Lösung greift in die Konfiguration des E-Mail-Servers ein. Der MTA wird über einen zusätzlichen Filter so angepasst, dass er bestimmte E-Mails an den eigentlichen Empfänger ausliefert und eine Kopie davon anfertigt.
Aufgrund der Vielzahl der verfügbaren MTAs (z.B. qmail, postfix, sendmail oder exim) muss für jeden MTA eine individuelle Lösung realisiert werden.
Bei dieser Lösung können natürlich auch E-Mails abgefangen werden, die per verschlüsseltem SMTP übertragen wurden, da der Datenstrom vom MTA wieder entschlüsselt wird.
Eine z.B. mit GPG/PGP (Gnu Privacy Guard / Pretty Good Privacy) verschlüsselte E-Mail ist davon jedoch nicht betroffen.

Verwendung von dsniff (mailsnarf)

Die zweite Lösung erfordert keine Anpassung am MTA.
Der gesamte Datenstrom des MTA's wird hierbei mit einem Sniffer (z.B. mailsnarf aus dem dsniff-Paket oder mit ethereal) überwacht und nach den gewünschten E-Mails gefiltert. Das Sniffing kann, wenn eine Bridge eingesetzt wird, für den Internet Service Provider vollständig transparent durchgeführt werden.
Natürlich scheitert dieses Vorgehen an korrekt konfigurierten, verschlüsselten SMTP-Sitzungen zwischen MTA's oder zwischen MTA und MUA.

IX Verschiedenes

Links mit weiteren Informationen / Quellen

[Print]
[1] „Absolute OpenBSD: Unix for the practical Paranoid“, Michael W. Lucas, No Starch Press
[6] WORLDWIDE CLIENT AND SERVER OPERATING ENVIRONMENTS FORECAST AND ANALYSIS, 2002-2006: MICROSOFT EXTENDS ITS GRIP ON THE MARKET," IDC, SEPTEMBER 2002
[17] „Web Security, Privacy & Commerce“ 2nd edition, Chapter 3, Simson Garfinkel and Gene Spafford, O'Reilly
[19] „Linux Apache Web Server Administration“, Kapitel 12, Charles Aulds, Sybex
[20] „Exim: the mail transfer agent“, Philip Hazel, O'Reilly
[23] „Maximum Protection“, Ryan Russell & Stace Cunningham, MITP-Verlag

[Web]
[2] http://www.openbsd.org
[3] http://www.openbsd.org/de/policy.html
[4] http://www.openbsd.org/users.html
[5] http://www.fsmlabs.com/products/products.html
[7] http://www.informationweek.de/index.php3?/channels/channel17/032314a.htm
[8] http://www.mug-d.de/mboerse/1999_2/linux.html
[9] http://www.heise.de/ct/01/19/162/
[10] http://www.knopper.net/knoppix/
[11] http://fire.dmzs.com/
[12] http://www.openbsd.org/cgi-bin/man.cgi?query=dd&apropos=0&sektion=0&manpath=OpenBSD+Current&arch=i386&format=html
[13] http://www.trekweb.com/~jasonb/articles/linux_loopback.shtml
[14] http://www.knoppix-std.org/
[15] http://naughty.monkey.org/~dugsong/dsniff/
[16] http://naughty.monkey.org/~dugsong/dsniff/faq.html
[18] http://news.netcraft.com/archives/2003/08/01/august_2003_web_server_survey.html
[21] http://foremost.sourceforge.net/
[22] http://www.nsrl.nist.gov/
[24] http://www.heise.de/security/artikel/38658/0