Informationssicherheit IT4 / IN6

Sommersemester 2010

Mirko Dziadzka

Wer bin ich?

- Studium Mathe/Informatik in Berlin
- 6 Jahre Sysadmin an der FH-Furtwangen
- 6 Jahre Softwareentwicklung und Betrieb im schweizerischen Bankenumfeld (Unix Cluster und Security)
- 10 Jahre Vorlesungen im Bereich Unix +Internet+Security

2

Wer bin ich?

- Seit 2005 in Regensburg bei der art of defence GmbH (Web-Application-Firewall)
- CCC, OWASP, GUUG

3

Wie wird das hier ablaufen?

- Vorlesung (eventuell mit Gastdozenten)
- Klausur - voraussichtlich 90 Minuten
- Voraussetzung zur Klausur ist die Abgabe der Übungsblätter

4

Übungen

- Vorlesungsbegleitend
- Übungsblätter in der Vorlesung
- Abgabetermin i.d.Regel 2-3 Wochen später
- Arbeit in Gruppen von 2 Personen möglich
- Abgabe der Übungsblätter und Vorführung der Resultate im Rahmen der Übungen bei mir

5

Übungen

- TODO: Teilnehmerliste

6

Kontakt

- Email: mirko.dziadzka@gmail.com
- Twitter: @MirkoDziadzka
- Web: https://mirko.dziadzka.de/vorlesung

7

Literatur

- Siehe Webseite
- Claudia Eckert: IT-Sicherheit, Oldenbourg Verlag, 6. Auflage
- Matt Bishop: Computer Security, Addison Wesley

8

Worum soll es in der Vorlesung gehen

- Umfrage und Diskussion

9

Worum soll es in dieser Vorlesung gehen

- Information Security:
  - Products
  - People
  - Procedures

Worum soll es in dieser Vorlesung gehen

- Die Vorlesung will ein Bewusstsein dafür schaffen, das Sicherheit im IT Bereich ein sehr komplexes Unterfangen ist
- Wir werden uns auf Verfahren und Protokolle, ihre Probleme und Möglichkeiten konzentrieren
- Best practices, HowTo

11

Einführung in den Bereich IT Security

- Ziele der Informationssicherheit
  - Confidentiality, Integrity, Authenticity
  - Availability
  - Non-repudiation
  - Anonymity

12

Einführung in den Bereich IT-Security

- Bedrohungsanalyse / Status Quo

Inhalt

- Einführung
- Probleme mit TCP/IP
- Netzwerksicherheit - Firewall und IDS
- Hostsicherheit - Hardening, Zugrifsrechte
- Kryptographie - Grundlagen und Anwendung
- Applikationssicherheit, Exploits & mehr
- Authentisierung / Autorisierung
- Web Application Security
- Prozesse

Probleme mit der TCP/IP Protokollfamilie

- Wie ist TCP/IP entstanden und warum gibt es Probleme

Ethernet

- Adressierung über (theoretisch) eindeutige MAC Addresse
- Kann trivial gefälscht werden
- War früher ein reines Broadcastmedium - jeder Teilnehmer konnte jeden Datenverkehr sehen.

Ethernet

- Heute in der Regel via Switch eine direkte Kommunikation zwischen zwei Rechnern
  - Kann via MAC-Flooding und MAC-Spoofing umgangen werden
  - Port Security

VLAN - 802.1Q

- Ein Mittel um den Datenverkehr zu trennen, allerdings sollte man sich vom Security Standpunkt aus nicht allzusehr darauf verlassen
  - switch ist in der Regel fail-opeen
  - http://www.corecom.com/external/livesecurity/vlansec.htm
- Authentisierung: 802.1X

WLAN

- Ethernet over the air - jeder kann senden und mithören
- Eingebaute Sicherheit:
  - WEP - öffentlich tot seit 2001
  - heute WPA und WPA2

19

WPA

- PSK (pre-shared-key) kann der verbindungsaufbau aufgezeichnet und dann mit brute-force der key ermittelt werden. Normale Passwörter sind hier ungeeignet.
- Ist akademisch gebrochen (seit 2008)

20

WPA2

- auch hier das Problem mit PSK - gute Passwörter wählen - d.h. einen kryptographischen Zufallsgenerator nehmen
- EAP stat PSK

21

WLAN Empfehlung

- Das WLAN ist als unsicheres Netz anzusehen und zu behandeln
- Es ist unsicherer als Kabelgebundenes Netz, da der physikalische Zugriff einfacher ist.

22

WLAN Empfehlung

- VPN over WLAN ist eine gute Möglichkeit den mobilen Zugriff zu schützen
  - Zugriff aus dem WLAN Netz nur auf den VPN Server, kein Zugang zum internen Netzt oder ins Internet

23

IPv4

- Interessante Header:
  - Fragmentierung - Evasion
  - TTL - Privacy
  - Source IP - Spoofing
  - IP-Options - Routing

24

ARP - Address Resolution Protocol

- Umsetzung von IP (Layer 3) auf Ethernet (Layer 2) Adressen
- Broadcast Anfrage, Unicast Antwort

```
0:80:c8:f8:4a:51 ff:ff:ff:ff:ff:ff arp
                 who-has 192.168.99.254 tell 192.168.99.35

0:80:c8:f8:5c:73 0:80:c8:f8:4a:51 arp
                 reply 192.168.99.254 is-at 0:80:c8:f8:5c:73
```


25

ARP

- Jeder Rechner im LAN sieht alle ARP Requests und kann jederzeit ARP Replies senden.
- Ein Rechner cached ARP Replies

ARP

- Es kann legitim sein, ARP Replies für fremde IPs zu senden
  - Virtuelle Maschinen + Routing
  - Cluster + Failover und umziehende IPs

DHCP

- Versorgt dem Client mit IP, DNS Server und Router
- Client vertraut DHCP Server
- Umleiten des Datenverkehrs möglich

IPv6

- Kommt ... irgendwann
- IPSec eingebaut
  - encryption + authentication
- Mobile IP
- Es fehlt die Erfahrung, also wird es am Anfang viele Probleme auch im Securitybereich geben

29

Routing

- Wie funktioniert IP Routing
  - lokales Interface
  - next Hop
  - speziellere Route hat Vorrang
  - Metrik / Kosten Funktionen
  - MIt spezieller Software Policy-basiertes Routing möglich

30

Routing

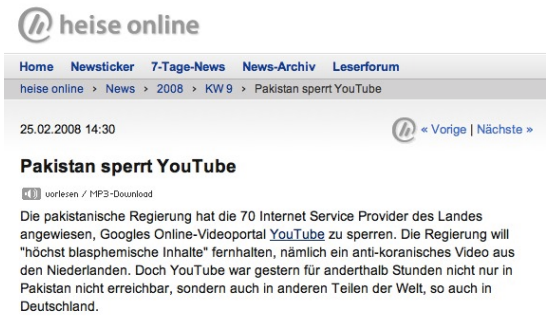
- Bestimmung des Next-Hop in kleines Netzen statisch, in größeren Netzen in der Regel über Routingprotokolle
  - Routingprotokolle ermöglichen automatisches Failover
  - RIP und OSPF innerhalb eines AS
  - BGP zwischen Autonomen Systemen

BGP

- Internet Core Routing Protocol
- Viel Vertrauen auf dieser Ebene
  - Peeringpartner nehmen in der Regel Routinginformationen Ihrer Peers ungeprüft entgegen (können in der Regel auch nicht prüfen)
  - Die Endpunkte (Transit) werden besser abgesichert

32

BGP



33

BGP

UDP

- Kleiner Wrapper um IP
- mehrere Ports per Rechner
- Checksum
- Anwendungen im Internet
  - DNS
  - Echtzeit Audio und Video Streaming

35

TCP

- Virtueller Datenstrom (geordnet mit Fehlerkorrektur und Retransmit)
- SRC + DST Port
- Sequence number, Ack Number, 3-way handshake
  - ISN sollte zufällig sein
- SYN, ACK, FIN, RST (PSH, URG)

TCP Angriffe

- Syn Fload
  - Belegt durch halboffene Verbindungen Ressourcen am Server
  - Lösung in SynCookies
- IP Spoofing aufgrund 3-way handshake heute schwierig
- RST Attacke auf lange Verbindungen

NAT / PAT

- IPv4 Knappheit führt zur Benutzung von privaten Netzen RFC 1918
  - 10/8, 172.16/12, 192.168/8
- Router setzt zwischen internen und offiziellen Adressen um
- NAT (PAT)- keine Sicherheitsmaßnahme

DNS

- spielt indirekt eine sehr große Rolle im IT Security Bereich
- Cache Poisioning
  - Altes Problem aber Dan Kaminsky / 2008 - neue Methode es auszunutzen
- Alles wird gut, wenn wir endlich DNSsec haben ;-)

DNS

```
mirkos-macbook:~ mirko$ dig any heise.de

; <<>> DiG 9.6.0-APPLE-P2 <<>> any heise.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6765
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 3

;; QUESTION SECTION:
;heise.de.                      IN      ANY

;; ANSWER SECTION:
heise.de.               1466    IN      A       193.99.144.80
heise.de.               245     IN      NS      ns2.pop-hannover.net.
heise.de.               245     IN      NS      ns.plusline.de.
heise.de.               245     IN      NS      ns.s.plusline.de.
heise.de.               245     IN      NS      ns.heise.de.
heise.de.               245     IN      NS      ns.pop-hannover.de.
heise.de.               4274    IN      MX      10 relay.heise.de.

;; ADDITIONAL SECTION:
ns.plusline.de.         400     IN      A       212.19.48.14
ns.heise.de.            14358   IN      A       193.99.145.37
ns2.pop-hannover.net.   1080    IN      A       62.48.67.66
```


SMTP - Simple Mail Transfer Protocol

- Keine Authentisierung
- ESMTP
  - STARTTLS - opportunistic encryption
  - SMTP-Auth
- Port 25, Port 587 (Message Submission), Port 465 SMTP over SSL

Spam

- gefühlte 95 Prozent aller emails sind SPAM
- SPF (Sender Policy Framework)
- DomainKeys

Spam

```
mirkos-macbook:~ mirko$ dig any nurfuerspam.de

; <<>> DiG 9.6.0-APPLE-P2 <<>> any nurfuerspam.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49804
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 4

;; QUESTION SECTION:
;nurfuerspam.de.                        IN      ANY

;; ANSWER SECTION:
nurfuerspam.de.         14400   IN      NS      ns.schlund.de.
nurfuerspam.de.         14400   IN      NS      dns.gmx.net.
nurfuerspam.de.         14400   IN      TXT     "v=spf1 redirect=gmx.net"
nurfuerspam.de.         14400   IN      SOA     dns.gmx.net. hostmaster.gmx.net. 2009052000 28800 7200 604800 86400
nurfuerspam.de.         14400   IN      A       213.165.65.50
nurfuerspam.de.         14400   IN      MX      10 mx0.gmx.net.
nurfuerspam.de.         14400   IN      MX      10 mx1.gmx.net.

;; ADDITIONAL SECTION:
dns.gmx.net.            13712   IN      A       213.165.64.1
ns.schlund.de.          10733   IN      A       195.20.224.97
mx1.gmx.net.            110     IN      A       213.165.64.102
mx0.gmx.net.            110     IN      A       213.165.64.100
```


Spam

```
mirkos-macbook:~ mirko$ dig any gmx.net @ns.schlund.de

; <<>> DiG 9.6.0-APPLE-P2 <<>> any gmx.net @ns.schlund.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50533
;; flags: qr aa rd; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;gmx.net.                       IN      ANY

;; ANSWER SECTION:
gmx.net.                86400   IN      NS      ns.schlund.de.
gmx.net.                300     IN      TXT     "v=spf1 ip4:213.165.64.0/23 ip4:74.208.5.64/26 -all"
gmx.net.                86400   IN      MX      10 mx1.gmx.net.
gmx.net.                86400   IN      A       213.165.65.50
gmx.net.                86400   IN      MX      10 mx0.gmx.net.
gmx.net.                86400   IN      SOA     dns.gmx.net. hostmaster.gmx.net. 2010032401 600 7200 604800 3600
gmx.net.                86400   IN      NS      dns.gmx.net.
```


Spam

- Was funktioniert?
  - Spam Filter am Endpunkt
  - Rechtliche Probleme in der Firma
  - Das Fernmeldegeheimnis ... ist ein Verbot des unbefugten Abhörens, Unterdrückens, Verwertens oder Entstellens, von .... Botschaften.

FTP

- Keine Verschlüsselung
- Getrennter Daten und Kontrollkanal, daher historisch interessant zum Scannen von Netzen (seit > 10 Jahren gefixtes Problem)
- Schwierig für Firewalls

HTTP

- Kein State
- Keine Authentisierung
- Web Server Security
  - dazu später mehr
- Browser Helper Programme
  - Wo kommen denn all die Trojaner her?

Netzwerkbasierte Sicherheit

- Firewall
  - Definition
  - Komponenten
  - Konzepte
- Intrusion Detection

Firewall

Was ist das ?

Firewall

"Als Firewall bezeichnet man ein organisatorisches und technisches Konzept zur Trennung von Netzbereichen, dessen korrekte Umsetzung und dauerhafte Pflege. Ein oft benutztes Instrument der Umsetzung ist ein Stück Hardware, das zwei physisch getrennte Netzbereiche genau so verbindet, wie es im Konzept zugelassen wird. Dieses Stück Hardware bezeichnet man als Firewall-Rechner/System oder verkürzt als Firewall.

http://www.iks-jena.de/mitarb/lutz/usenet/Firewall.html

Firewall

A firewall is a part of a computer system or network that is designed to block unauthorized access while permitting authorized communications. It is a device or set of devices which is configured to permit or deny computer based application upon a set of rules and other criteria.

http://en.wikipedia.org/wiki/Firewall_(computing)

Firewall

"An internetwork gateway that restricts data communication traffic to and from one of the connected networks and thus protects that network's system resources against threats from the other network"
"A firewall is not always a single computer. For example, a firewall may consist of a pair of filtering routers and one or more proxy servers running on one or more bastion hosts, all connected to a small, dedicated LAN between the two routers."

RFC 2828

Firewall

- Definition
  - Koppelung von Netzbereichen verschiedener Sicherheitsanforderungen
  - Realisierung einer Sicherheitskonzepts

Komponenten

- Paket Filter
- Proxy / Circuit-Level Gateway
- Application Level Gateway
- DMZ

Paket Filter

- Analyse des Datenstroms auf Paketebene
  - Durchsetzung der Protokolldefinitionen (check von Paketlängen, IP und TCP Optionen, ...)
  - Zugriffsmatrix
- Pakete können erlaubt oder geblockt werden, eventuell auch modifiziert (NAT)

Paket Filter

- Heute in der Regel mit Stateful Inspection auf Layer 3 + 4, d.h. mit Zustand
  - Erkennt Pakete, die zu einer erlaubten Verbindung gehören
- Router und eventuell Switche haben auch Paketfilter, aber in der Regel nur nur auf Layer 3 - IP.

Beispiel

```
 pkts bytes target     prot opt in     out     source               destination
 127M   11G ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
38181 3209K ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmp type 255
    0     0 ACCEPT     esp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     ah   --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            224.0.0.251         udp dpt:5353
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:631
   12   676 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:631
 151M   36G ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
 1559  101K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:8082
   53  3284 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:1723
7888K  473M ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:8083
2622K  157M ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:8086
3288S 1957K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
41995 2443K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:25
77345 4647K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:80
18643 1112K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:443
 777K   41M REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
```


Circuit-level gateway

- Proxy auf TCP oder UDP Ebene
  - Terminiert die connection und baut sie neu auf.
  - Trennung von externer und interner IP
  - Garantiertes korrektes TCP
- rinetd
- SOCKS (ssh -D 1080)

Application Level Gateway

- Terminiert UDP und/oder TCP Verbindung
- Eventuell Store-and-Forward
- Validiert Datenverkehr auf Application-Layer Level
- Kann komplexere Protokolle implementieren

Application Level Gateway

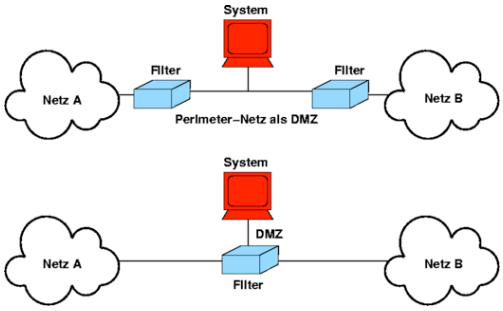
- Typische Beispiele:
  - DNS Server,
  - Mailserver,
  - Webproxy
- Content Filter (Viren)

DMZ

- Dual-homed
- Screened Host
- Screened Subnet

DMZ

- Spezieller geschützter Netzwerkbereich



Beispiele

- Einfacher Paktfilter mit NAT und DNS Proxy
- Mail und Webserver in der DMZ
- Webserver mit Zugriff auf eine Datenbank im internen Netzwerk

Hinweise

- Beim Design eines verteilten Systems bereits an die Firewall denken
  - Kommunikation vom sicheren system zum unsicheren
  - Kommunikation nür über TCP
  - Kommunikation dokumentieren

Hinweise

- Beim nachträglichen Einbau einer Firwall
  - erst mal mitlaufen lassen und Kommunikation protokollieren
  - Kommunikation verstehen + Dokumentieren
  - Monatsabschluss, Jahresabschluss, Failover

Hinweise

- Tunneln kann man nicht verhindern
- UDP am besten über Applikation Level Gateway (braucht man eh nur für DNS :-)
- Ausgehende Pakete nach Absender-IP filtern
- TCP Conections mit reject stat mit drop ablehnen

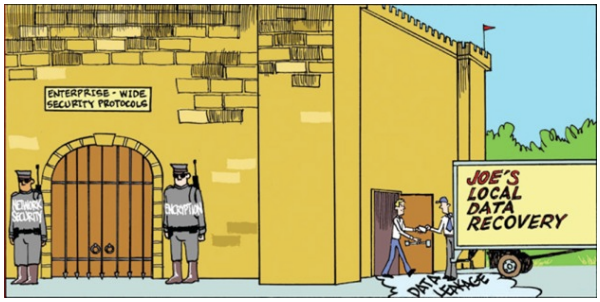
Administration

- Idealerweise durch ein eigenes gesichertes physikalisch getrenntes Netzwerk
- Alternativ über kryptographisch gesicherte Protokolle
- Audit Log
- Vier Augen Prinzip

Vorteile / Nachteile / Grenzen

- Bündelung von Sicherheitsdiensten
- Komplexe Anforderungen / Regelwerk
- Fehler in der Konfiguration sind schwerwiegend
- Webservices
- Risikokompensation

Grenzen



IDS - Intrusion Detection System

- Netzwerkbasiert
- Erkennen von Anomalien durch beobachten des Netzwerkverkehrs
- In der Regel Pattern basiert
- Kann als IPS auch Gegenmaßnahmen einleiten

IDS Evasion

- Payload Obfuscation
- Fragmentierung, Small packages, TTL Tricks
- Verschlüsselung
- Bandbreite
- Attacken gegen das IDS

Sinnvoller Einsatz

- “False Positives” vs. “False Negatives”
- “Informationsbeschaffung” im Intranet
- Gut in gesicherten Netzen (DMZ)
- Inline IDS

Hostbasierte Sicherheit

- Rechtekonzepte
- System Hardening
- Rootkits
- Intrusion Detection Systeme

Rechtekonzepte

- Discretionary Access Control
  - Der Eigentümer eines Files legt die Zugriffsrechte fest
- Klassisches Unix:
  - User / Group / Other
  - Read / Write / Execute (andere OS haben noch Create, Append, Delete, ...)

Rechtekonzept Unix

- Directory:
  - 'read': Inhalt ansehen (ls)
  - 'write': files anlegen und löschen
  - 'execute': zugriff auf files und Unterverzeichnisse
- Check on Open

Rechtekonzept Unix

- (Fast) Alle User sind gleich
- suid / sgid um einem Prozess temporär die Rechte eines anderen Beutzers zu geben
- Beispiele sind:
  - sudo, su, passwd, ping

Rechtekonzept Unix

Rechtekonzept Unix

- Root ist allmächtig - größtes Sicherheitsproblem in Unix
- Capabilities
  - Ein Recht hängt nicht mehr an der UID 0
  - Prozess kann Rechte dauerhaft abgeben

Rechtekonzept

- Verschiedene Unixe implementieren auch POSIX 1.e ACLs (Draft, der Verworfen worden ist)
- Diese sind mehr oder minder äquivlent zun Windows ACLs
- Komplexität durch Vererbung

Rechtekonzepte

- Mandatory Access Control
  - nicht der Eigentümer des Objects sondern eine Policy legt die Zugriffsrechte fest.
- Bell LaPadula
- Biba (Integrity Protection)
- andere und Mischmodelle

Bell LaPadula

- Ersten formal spezifiertes System
- 197x - US Air Force
- Schutz der Vertraulichkeit von Informationen
- Unterscheidet Zuständigkeitsbereiche und Zugriffslevel

Bell LaPadula

- Lesen:
  - $L(S) \geq L(O)$ && $Z(S) \geq Z(O)$
- Schreiben:
  - $L(S) \leq L(O)$ && $Z(S) \leq Z(O)$

Biba (Integrity Control)

- “Sichere Komponenten” dürfen keine “unsicheren Daten” lesen
- “Unsichere Komponenten” dürfen keine “sicheren Daten” schreiben

Implementationen

- SELinux - Security Enhanced Linux
  - Entwickelt von der NSA
  - Seit 2000 OpenSource
  - Heute im Standard Linux Kernel
  - Auch auf andere Unix artige OS portiert

SELinux - Konzepte

- Objecte (Files, Sockets, Prozesse, ...) und
- Subjecte (Prozesse, Benutzer)
- Labels: user:role:type:level
- unconfined_u:object_r:user_home_t:s0

SELinux - Konzepte

- Standard Installation nutzt nur Typen
- Ein Subject mit dem Typ X kann auf Objekte mit dem Typ Y zugreifen, wenn es dafür eine Regel gibt
- Ein Subject mit dem Typ X das ein Process mit dem Typ Z ausführt kann den Typ Y erwerben, wenn eine Regel das erlaubt (verallgemeinertes SUID)

SELinux Konzepte

- Standard Fedora oder RHEL kommen mit zwei Policies:
  - targeted
    - schränkt nur Netzwerkprozesse ein
    - Alles andere ist 'unconfined_t'
  - strict
    - Volle Policy

Hostsecurity

- Compartments / Jails / Chroot
  - Schränkt Zugriff auf das Filesystem und eventuell auf andere Ressourcen ein

System Hardening

- Modifikation eines Systems um möglichst wenig Angriffspunkte zu bieten
- Sinvolle Voraussetzung für die Installation eines Systems in der DMZ

System Hardening

Secure File Permissions Matter

Posted April 13, 2010 by Matt. Filed under Development.

Summary: A web host had a crappy server configuration that allowed people on the same box to read each others' configuration files, and some members of the "security" press have tried to turn this into a "WordPress vulnerability" story.

WordPress, like *all other* web applications, must store database connection info in clear text. Encrypting credentials doesn't matter because the keys have to be stored where the web server can read them in order to decrypt the data. If a malicious user has access to the file system — like they appeared to have in this case — it is trivial to obtain the keys and decrypt the information. When you leave the keys to the door in the lock, does it help to lock the door?

System Hardening

- Vorgehen
  - Entfernen aller unnötigen Dienste
  - Entfernen aller unnötigen SUID/SGID Files
  - Zugriffsrechte mehr einschränken, als das OS Default vorgibt
  - Einschalten von Auditing

Hostbasiertes IDS

- Dateisystem Integritätcheck
- Monitoring der Rechte der Datein und laufenden Prozesse
- Monitoring der offenen Netzwerkverbindungen

Rootkits

- Definition: “A rootkit is a software program or coordinated set of programs designed to gain control over a computer system or network of computing systems without being detected.”

Rootkits

- Application Level (sshd, login, ...)
- Library Level (patch von Systembibliotheken)
- Kernel Level (neue und veränderte System calls)
- Hypervisor Level (Blue Pill)
- Firmware / Hardware Level

Kryptographie

- Historisches
- Grundlagen der modernen Kryptographie
- Wichtige Anwendungen und Protokolle

Kryptographie

- Ziele:
  - Vertraulichkeit (confidentiality)
  - Integrität (integrity)
  - Authentizität (authentication)
  - Nichtabstreitbarkeit (non-repudiation)

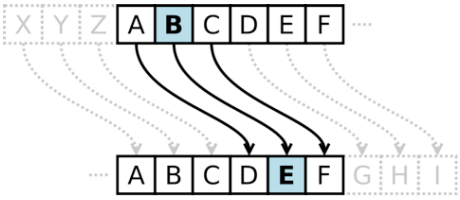
Kryptographie

- Historie
  - Transpositions-Chiffren
  - Substitutions-Chiffren

Transpositionschiffren

Substitutionschiffren

- Caesar Chiffre



Monoalphabetische Chiffren

- Der Schlüssel ist eine Permutation der Buchstaben des Alphabets
- Ver- und Entschlüsselung erfolgt durch anwenden der Permution auf die einzelnen Buchstaben des Textes
- Durch Analyse der Häufigkeiten der Buchstaben in natürlichen Sprachen zu brechen

Polyalphabetisch Chiffrecn

- Es gibt mehrere (N) verschiedene Monoalphabetische Keys $K = K_1, K_2, \ldots K_N$
- $C_i = \text{Perm}(K_{i \bmod N}, P_i)$
- Je größer N, umso sicherer die Chiffre

Polyalphabetische Chiffren

- Wenn N wesentlich kleiner als der verschlüsslte Text ist, kann man
  - durch Häufigkeitsanalysen N ermitteln und dann
  - mit Statistik die einzelnen Monoalphabetishen Chiffren brechen

One-Time-Pad

- Einziges beweisbar sicheres Verfahren.
- Polyalphabetisch
  - jede Permutation zufällig
  - soviele Permutation wie der originaltext lang ist
  - wird nur einmal verwendet

One-Time-Pad

- Probleme bei der Benutzung
  - Zufallszahlengenerator
  - Key Verteilung

Shared Key Chiffren

- Blockchiffre
  - Verschlüsselt einen Block (früher 64, heute meistens 128) mit einem geheimen Schlüssel
  - Schlüssellänge heute 128-256 bit (DES: 56 Bit)

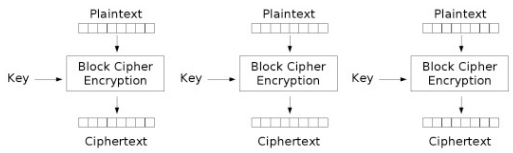
Blockchiffren

- Sowohl blocksize als auch schlüssellänge sind wichtig
  - Schlüssellänge: Brute-Force gegen den Schlüssel
  - Blocksize: Geburtstagsparadox, verschlüsselung des selben blocks
  - Bekannte Chiffren: DES, 3DES, AES, TwoFish

Modi of operations

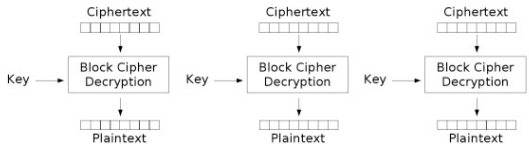
- Blockchiffre verschlüsselt immer nur ein paar Bytes, wie verschlüssel ich einen ganzen Text?

ECB - Electronic Codebook



Electronic Codebook (ECB) mode encryption

ECB - Electronic Codebook

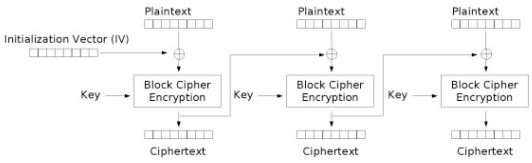


Electronic Codebook (ECB) mode decryption

Eigenschaften ECB

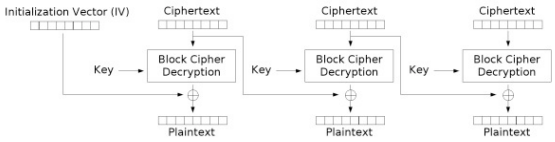
- Bitkippe im Cyphertext bleiben lokal im block
- zwei gleich Blöcke werden gleich verschlüsselt

CBC - Cipher Block Chaining



Cipher Block Chaining (CBC) mode encryption

CBC - Cipher Block Chaining

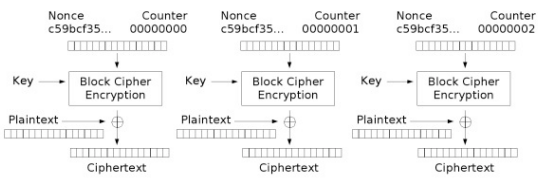


Cipher Block Chaining (CBC) mode decryption

Eigenschaften CBC

- Bitkipper im Cyphertext bleiben lokal of den aktuellen und den nächsten Block begrenzt.
- Identische Blöcke werden in der Regel verschieden verschlüsselt
- Gleiche Nachrichten werden bei gleichem Schlüssel und verschiedene IV verschieden verschlüsselt

CTR - Counter Mode



Counter (CTR) mode encryption

CTR - Counter Mode

- CTR - Counter Mode
  - erlaubt wahlfreien Zugriff beim Lesen
- OFB - Output Fedback Mode

Sonstiges

- Padding

Empfehlung

- AES 256 im CBC Mode benutzen
- Achtung: Encryption bietet keine 'integrity protection'

Stromchiffre

- Benutzung einer Blockchiffre als Stromchiffre
  - OFB
- Für jede nachricht muss zwingend ein anderer Initialisierungsvektor genommen werden, sonst ist das System angreifbar
- rc4
- A5/1

Vergleich von Strom und Blockchiffren

- Latency ist bei Stromchiffren geringer -> Audio und Video Streaming
  - ein bit kann encoded/decoded werden wenn es ankommnt
- Flipping a single given bit is possible
  - Bit 20 bei ihrem Kontostand

Public Key Kryptography

- Probleme mit Symetrischer Krypto:
  - Keymanagement
  - Signaturen
- Idee

Public Key Verfahren

Diffie-Hellman - 1976

- Key Exchange - Alice und Bob erzeugen durch öffentliche Kommunikation ein gemeinsames Geheimnis
- discrete logarithm Problem
- Alice: a, Bob: b; public: p, g, g^a, g^b mod p
- Geheimnis ist g^{ab}
- Ohne Authentisierung ist MITM möglich

El Gamal -1985

- Basiert auf Diffie-Hellman
- Alice wählt p,g,a und veröffentlicht (p,g,g^a) als public key
- Bob wählt zufälliges r -> g^{ar} ist shared key zwischen Alice und Bob
- Bob sendet (g^r , m * g^{ar})
- Alle Werte als Restklassen mod p

RSA -1978

- Problem der Primzahlzerlegung
- n = p * q
- e teilerfremd zu phi(n) = (p-1)(q-1)
- d * e == 1 mod phi(n)
- n und e sind öffentlich

RSA

- Sei m eine Nachricht (Zahl kleiner n)
- c = $Encrypt_{n,e}(m) = m^e$ mod n
- m = $Decrypt_{n,d}(c) = c^d$ mod n
- s = $Sign_{n,d}(m) = m^d$ mod n

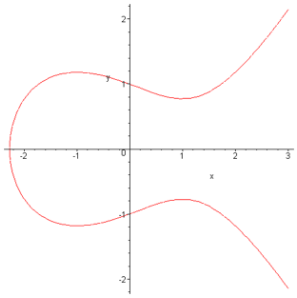
Andere

- DSA (Digital-Signature-Algorithm)

Andere

- ECC (Eliptic-Curve-Cryptography)
  - kürzere Keys als bei RSA (380 Bit)
  - weniger erforscht, Patente, wirtschaftlicher Druck

Elliptic Curve



$y^2 = x^3 + ax + b$

Andere

- Empfehlung: RSA 4096
- Padding beachten(!)

Hybride Verfahren

- Nachteile: von Public Key: langsam
- Hybride Verfahren
  - Wähle zufälligen symmetrischen Key K.
  - Verschlüssel Nachricht mit K
  - Verschlüsseln K mit Public Key Verfahren
  - Sende: $Enc_k(M) + RSA_E(M)$

Hash Funktionen

- Ziele
  - eindeutig, fixe länge
  - praktisch nicht umkehrbar
  - no second pre-image
  - praktisch keine Kollisionen erzeugbar

Hash Funktionen

- md4 - ist gebrochen
- md5 - 160 Bit
  - gilt heute als gebrochen bzg. Kollisionen
- SHA-1 - 256 Bit
  - gilt unter Kryptographen inzwischen als unsicher bezg. Kollisionen

Hash Funktionen

- SHA-2 (SHA-256 und SHA-512) gelten noch als sicher

Hash Funktionen

- Bekannte Probleme von SHA und anderen:
  - Extension Problem:
  - H(A) = H(B) -> H(A|X) = H(B|X)
  - Nachrichten so strukturieren, dass Extension erkannt wird

Integrität einer Nachricht

- Wie können wir feststellen, ob eine verschlüsselte Nachricht auf dem Transport verändert wurde?
- Wir übertragen $Enc_k(M + H(M))$
- Vorsicht:
  - Wenn Enc eine Stromchiffre und H keine kryptogtaphische Hashfunktion ist -> Änderung der Nachricht möglich

MAC

- Message Authentication Code
  - Hash mit Key
  - $HMAC(M,K) = H(K_1\ H(K_2\ M))$
- Integrität von M:
  - M , HMAC(M, K)

Digitale Signatur

- Publick Key Verfahren sind nur für kleine Nachrichten geeignet: 1024-4096 Bit (384 bei EC)
- Signiere nicht die Nachricht, sondern den Hash der Nachricht.

Andere Probleme

- Zufallszahlen
  - dürfen durch einen externen Beobachter nicht vorausgesagt werden können
  - Messen von physikalischen Ereignissen im Rechner, teilweise spezielle Hardware
  - diese Entropie wird in eine sicheren PRNG eingebracht

Andere Probleme

- Zufallszahlen
  - /dev/random vs. /dev/urandom
  - Viele Programmierumgebungen stellen eine Funktion für sichere Zufallszahlen zur Verfügung (in Python: os.urandom())
  - Aber auch diese Bibliotheken können Fehler enthalten

Zufallszahlen

- Epic Fail: Debian 2006-2008
  - When creating a new OpenSSH key, there are only 32,767 possible outcomes for a given architecture, key size, and key type
  - All SSL and SSH keys generated on Debian-based systems (Ubuntu, Kubuntu, etc) between September 2006 and May 13th, 2008 may be affected

Andere Probleme

- Primzahlen
  - Wie finde ich grosse Primzahlen?
  - probabilistische Tests: Miller-Rabin

Sicherer Kanal

- Ziele
  - Integrität
  - Authentizität
  - Vertraulichkeit

Sicherer Kanal

- session key, regelmässig neue keys
- Verschlüsselung
- hash über alle bisherigen Daten als MAC
- resistent gegenüber replay Attacken
- perfect forward secrecy

Sicherer Kanal

- Benutzen Sie SSL (mit keys auf client und server Seite)
- Wenn Sie sich wirklich für Implementierungsdetails interessieren: Practical Cryptography von Schneier

Digitale Signatur

- kombiniert Hash Funktion und Signatur
- M, $SIG_K(HASH(M))$
- wichtige Frage: Wie wird der Bithaufen M interpretiert
  - Struktur von M muss klar definiert sein

Digitale Signatur

- Wie weiss ich, welchen public key mein Kommunikationspartner hat?
  - Gelbe Seiten?!
  - Wer garantiert die Korrektheit der Einträge?
  - Zertifikate

Secret Sharing / Splitting

- Wie verteile ich ein Geheimnis (Passwort) unter N Personen, so dass M Personen zusammen KEINE Information haben und M +1 Personen das Geheimnis rekonstruieren können.

Anwendungen der Kryptographie

- SSL / TLS
- SSH
- Email: PGP, S/MIME
- VPN (IPSec, OPenVPN, PPTP)
- DNSSEC

SSL / TLS

- "Secure Socket Layer"
  - Erfunden von Netscape, ca. 1995
  - Heute Version 3
- "Transport Layer Security"
  - Standardisierte Nachfolger von SSL
  - Version 1.0, 1.1, 1.2

Ziele

- Sicherer Kanal
- Authentisierung des Servers (eCommerce)
- Optional: Authentisierung des Clients

Protokollbeschreibung

- ClientHello
- ServerHello + Certificate
  - Optional: CertificateRequest
- ClientKeyExchange
  - Optional: Certificate + CertificateVerify
- ChangeCipherSpec, Finished

SSL Speedup

- In der Anfangszeit von SSL (auch heute noch) war Public Key Crypto sehr Rechenaufwendig (10 SSL Connections pro Sekunde)
- HTTP wurde oft als HTTP/1.0 benutzt
- Resume einer Connection anhand der SessionID möglich (ohne neue PublicKey Operationen)

X509 Certificates

- ITU Standard für Public Key Infrastrukturen
- *Eine CA bestätigt die Identität des Eigentümers eines Public Keys*
- *Signatur über PublicKey, Namen und weitere Eigenschaften*
- ASN.1 basierend, dann PEM oder DER kodiert

Aufbau

- Version:
- Serial Number
- Signature Algorithm
- Issuer
- Validity
- Subject
- Public Key
- Extensions
- Signature

Beispiel

- openssl s_client -connect banking.postbank.de:443 -showcerts
- openssl x509 -in certificate -text -noout

Beispiel

- Version: 3 (0x2)
- Serial Number 26:65:19:20:52:6d:3e:d6:5f:e1:da:8a:ba:11:7d:00
- Signature Algorithm: sha1WithRSAEncryption
- Issuer: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/rpa (c)06, CN=VeriSign Class 3 Extended Validation SSL SGC CA
- Validity
  - Not Before: Jul 16 00:00:00 2009 GMT
  - Not After : Aug 15 23:59:59 2011 GMT

Beispiel

- Subject: 1.3.6.1.4.1.311.60.2.1.3=DE/2.5.4.15=V1.0, Clause 5. (b)/serialNumber=HRB6793, C=DE/postalCode=53113, ST=NRW, L=Bonn/streetAddress=Friedrich Ebert Allee 114 126, O=Deutsche Postbank AG, OU=Systems AG, **CN=banking.postbank.de**
- Subject Public Key Info:
  - Public Key Algorithm: rsaEncryption
  - RSA Public Key: (2048 bit)
    - Modulus (2048 bit): 00:cb:ae:35:94:6f:e3.....
    - Exponent: 65537 (0x10001)

Beispiel

- X509v3 extensions:
  - X509v3 Basic Constraints:
    - CA:FALSE
  - X509v3 Subject Key Identifier: 9A:E7:84:B2:3B:B7:B3:2B:0B ....
  - X509v3 Key Usage:
    - Digital Signature, Key Encipherment
  - X509v3 CRL Distribution Points:
    - URI:http://EVIntl-crl.verisign.com/EVIntl2006.crl
  - X509v3 Certificate Policies:
    - Policy: 2.16.840.1.113733.1.7.23.6
    - CPS: https://www.verisign.com/rpa

Beispiel

- X509v3 extensions
  - X509v3 Extended Key Usage:
    - TLS Web Server Authentication, TLS Web Client Authentication, Netscape Server Gated Crypto
  - X509v3 Authority Key Identifier: keyid:4E:43:C8:1D:.....
  - Authority Information Access:
    - OCSP - URI:http://EVIntl-ocsp.verisign.com
    - CA Issuers - URI:http://EVIntl-aia.verisign.com/EVIntl2006.cer
  - 1.3.6.1.5.5.7.1.12: .... 0`.^.\0Z0X0V..image/gif0!0.0...+.....Kk. (.....R8.).K..!..0&.$http://logo.verisign.com/vslogo1.gif

Beispiel

- Signature Algorithm:
  - sha1WithRSAEncryption
- Signature:
  - 1e:0d:f2:68:4b:bc:ad:61:d9:d0:6d:50:38:fb: 8d:24:23:a6 ....

Prüfen des Zertifikates

- Client hat Liste aller "vertauenswürdigen" Root Zertifikate.
- Server schickt Zertifikat oder Zertifikatskette
- Ist die Kette gültig?
- Sind alle Constraints erfüllt? (CA=True)
- Ist das Root Zertifikat bekannt?
- Stimmt der CN mit dem Namen des Webserver überein?

Probleme / Angriffe

- Inband TLS
  - SMTP + STARTTLS
- Wrapper TLS
  - HTTPS
  - Wie weiss der Server, welches Zertifikat er präsentieren soll? Der Servername kommt erst im HTTP Header(!) also nach dem SSL Verbindungsaufbau

Probleme / Angriffe

- DerClient generiert das PreMasterSecret (ist für Sicherheit verantwortlich)
- unzureichende Überprüfung der Ketten

Probleme / Angriffe

News-Meldung vom 05.11.2009 13:20 « Vorige | Nächste »

Schwachstelle im SSL/TLS-Protokoll

vorlesen / MP3-Download

Schwachstellen im SSL/TLS-Protokoll lassen sich Berichten zufolge von Angreifern ausnutzen, um in geschützte Verbindungen eigene Inhalte einzuschleusen. Betroffen wären neben HTTPS auch alle anderen Protokolle wie IMAP, die zur Transportsicherung TLS einsetzen. Über die genauen Auswirkungen des Problems wird noch diskutiert. Möglich wäre aber offenbar, etwa HTML-Inhalte von Webseiten während der Übertragung zu manipulieren und beispielsweise Schadcode zu injizieren.

Probleme / Angriffe

News-Meldung vom 29.07.2009 16:21 « Vorige | Nächste »

Studie: Warnmeldungen bei SSL-Zertfikaten so gut wie nutzlos

vorlesen / MP3-Download

Warnungen bei Unstimmigkeiten von SSL-Zertifikaten auf Web-Servern halten Anwender kaum davon ab, eine Webseite zu besuchen, haben Forscher der Carnegie Mellon University herausgefunden. In ihren Beobachtungen hatten mehr als 55 Prozent der Probanden die Warnmeldungen einfach ignoriert und weggeklickt. Neu ist diese Erkenntnis sicher nicht, allerdings haben die Forscher nun offenbar erstmals die Quantität des Problems vermessen.

Probleme / Angriffe

News-Meldung vom 30.09.2009 13:46 « Vorige | Nächste »

Trickzertifikat für SSL veröffentlicht [Update]

vorlesen / MP3-Download

Der Sicherheitsspezialist Jacob Appelbaum hat auf der Hacker-Mailingliste Noisebridge ein SSL-Zertifikat und den dazugehörigen privaten Schlüssel veröffentlicht, mit denen ein Webserver in verwundbaren Browsern keine Fehlermeldung provoziert – egal für welche Domain er das Zertifikat ausliefert. Phisher könnten dies etwa für Phishing-Angriffe ausnutzen und ihren Server als legitimen Bankserver ausgeben – was erst bei genauerer Prüfung des Zertifikats auffliegen würde.

Probleme / Angriffe

News-Meldung vom 30.12.2008 18:02 « Vorige | Nächste »

25C3: Erfolgreicher Angriff auf das SSL-Zertifikatsystem

vorlesen / MP3-Download

Sicherheitsforschern ist es gelungen, das Zertifikatsystem SSL für vertrauenswürdige Internet-Verbindungen zu kompromittieren. Durch eine sogenannte MD5-Kollision konnten sie ein Herausgeberzwischenzertifikat erstellen, das alle wichtigen Internet-Browser als vertrauenswürdig einstufen. Wer über ein solches Herausgeberzertifikat verfügt, kann sich bespielsweise SSL-Zertifikate für jede beliebige Internet-Domain erstellen. Damit können sich Angreifer als "Man in the Middle" in gesicherte Internetverbindungen einklinken und Daten ausspähen (etwa für Phishing) oder manipulieren, ohne dass Anwender eine Warnmeldung zu sehen bekämen.

Probleme / Angriffe

- ASCII-0 Bytes in ASN.1 Strings / ASN.1 Overflow
- 2009 Plain-text injection into HTTPS during re-negotiation
- CCC 2008: create a rogue Certificate Authority, accepted by all common browser
  - Trustmodell ist problematisch

PKI / CA

- Mit Public-Key-Infrastruktur (PKI, engl. public key infrastructure) bezeichnet man in der Kryptologie ein System, das digitale Zertifikate ausstellen, verteilen und prüfen kann.
- Certificate Authority: Organisation, die das CA-Zertifikat bereitstellt und die Signatur von Zertifikatsanträgen übernimmt.

Andere Trustmodelle

- SSH
  - keine Zertifikate
  - präsentiert bei unbekannten keys den hash über den key
  - meldet änderungen am Key

Andere Trustmodelle

- PGP
  - Web of Trust
  - Benutzer "beglaubigen" ihre Identitäten gegenseitig
  - Kann auch von CAs erfolgen (CACert HeiseCA)

Andere Trustmodelle

- Kombination von WebOfTrust und X509 Hierachie ist möglich. Zum Beispiel Identifiziert CACert die Benutzer anhand des WebOfTrust und stellt dann X509 Zertifikate aus

SSH

- genereller aufbau / trustmodell
- public key vs. passwort
- interessante optionen
  - forced commands
  - ProxyCommand
- Tunnel

VPN

- Szenarien
  - Firma zu Firma
  - Aussenstelle zu Zentrale
  - Home Office zu Firma
  - Road Warrior

VPN

- Technologien
  - IPSec
  - OpenVPN
  - PPTP
  - SSH

VPN

- Authentisierungsmöglichkeiten
  - Pre Shared Secret
  - X509 Zertifikate
  - Token basiert
    - X509, SecureID, ...
  - Username/Passwort gegen normale

VPN

- Fragen:
  - Wie weit vertraue ich den Daten, die via VPN hereinkommen
  - Remote Access aus dem Internetcafe???
  - Wo positioniere ich das VPN Gateway in der DMZ

Email

- Vertraulichkeit / Verbindlichkeit
- S/Mime (X509 basiert)
  - vor allem in der Business Welt im Einsatz
  - Setzt CA voraus (Kosten?)
  - Key-Recovery? Key-Escrow?
  - Storage vs. Transport vs. Signatur

Email

- PGP
  - in der nicht-kommerziellen Welt zu Hause
  - Web-of-Trust
  - in der Regel keine zentrale CA, aber heise.de, cacert.org, ...

DNS

- DNSSEC

Storage

- Warum verschlüsseln
  - gestohlen, verloren, defekte
    - Laptop
    - Platte
    - Mobile-Phone

Storage

- Wie verschlüsseln
  - Hardware encryption
  - Whole Disk Encryption
  - Partition Encryption
  - File System Level Encryption
  - Single File Encryption

Storage

- Disk Encryption
  - TrueCrypt - Windows, Mac, Linux
  - PGPDisk - Windows, Mac, Linux
  - LUKS - Linux
  - FileVault - MacOS
  - Bitlocker - Windows (Vista+ professional)

Storage

- Single File Encryption
  - zip ?
  - gpg

Storage

- praktische Fragen:
  - funktioniert das Backup Programm noch?
  - Key escrow?
  - Ist die Platte bootfähig?
  - Integration von Token als Authentisierung

Application Security

- Saltzer, Schroeder: The Protection of Information in Computer Systems, 1973
- formuliert Desing Prinzipien für sichere Systeme

Economy of mechanism

- Keep the design as simple as possible
- Review!
- Komplexität ist der größte Feind von Security

Fail-safe defaults

- Base access decisions on permission rather than exclusion
- expect a subsystem to fail anytime
- whitelist vs. blacklist

Complete Mediation

- Every access to every object must be checked for autority

Open design

- The design should not be secret

Seperation of privilege

- 4 Augen Prinzip

Least privilege

- Every program and every user of the system should operate using the least set of privileges necessary to complete the job.
- Limitiert den Schaden, der bei einem Fehler passieren kann

Least common mechanism

- Minimize the amount of mechanism common to more than one user ...
- verschiede Sicherheitstufen sollten getrennt werden
  - UID, Filesystem, Virtuelle Machine, eigener Rechner, eigenes Netzwerksegment

Psychological acceptability

- Wenn der Benutzer es nicht akzeptiert dann hilft aus das beste System nicht
- Passwort Policy

Side Channel Attacks

- Was ist das
  - Beobachte das System an den nicht definierten Schnittstellen
  - Laufzeit, Stromverbrauch, Speicherberbrauch, ...

Side Channel Attacks

- Beispiele
  - Smart Card key 'recovery'
  - historisch: passwort 'recovery'
  - Laufzeit von Logins via Internet
  - smart card: physikalischer Zugriff auf Key

Aktuelle Security Fehler

- Non-Web Bereich
- C artige Sprachen
  - Buffer Overflow (Stack und Heap)
  - Integer Overflow
  - Null Pointer
  - free memory

Gegenmaßnahmen

- Keine Fehler machen oder Fehler erkennen:
  - statische Source-Code Analyse,
  - Code Review
- Sicherheitsnetz:
  - Stack canary, non-executable stack, randomized address space

Authentisierung / Autorisierung

- Authentisierung: Wer bin ich
  - Pass
- Autorisierung: Was darf ich
  - Einreisen?
  - Auto fahren? (Führerschein)

Local

- Passwort Datenbank
- Passwörter in der Regel als Hash und nicht im Klartext
- Aber auch Hash muss geschützt werden - ein hash ist nur 'schwer' umkehrbar.
- Cracken von Passwörtern
- Gute Passwörter?

LAN

- LDAP
- Radius
- Kerberos
- Active Directory
  - LDAP + Kerberos
- Username+Passwort oder X509

Web

- Username + Passwort
- X509 Zertifikat
- HTTP Basic Auth
- HTTP Digest Auth
- Form Based
- Per Request oder per Session?

Cross Domain Validation

- OpenID
  - decentralized authentication
- OAUTH
  - authorize a second application

Two Factor Authentication

- Banken
  - PIN + TAN (iTAN, mTAN)
  - PIN + Challenge/Response
  - HBCI
- Password + RSA SecureID o.ä.
- Pro Transaktion oder pro Session

Autorisierung

- Liste von Rechten für den authentisierten Benutzer
- In der Regel in Datenbank, LDAP oder im X509 Zertifikat abgelegt.
- niemals(!) mit negativ Listen arbeiten (fail-safe defaults)

Web Security (Client)

- Der Browser ist das Einfallstor ins Firmennetz
- Plugins -> Buffer Overflow
  - Flash, Java, image-libraries

Privacy

- Cookies
- Flash_Cookies
- Java-Script
- Browser-Kennung
  - OS, Installierte Plugins, installierte Schriften, ...

Java-Script

- Code, der von einer fremden Seite kommnt und auf meinem Rechner ausgeführt wird.
- Theoretisch in einer Sandbox
- Same-Origin Policy
  - realisiert über den DNS Namen(!) nicht über die IP

Java Script

- Kann auf meinen DOM Baum zugreifen.
  - History Sniffing

Gegenmaßnahmen

- kein Flash, kein Java, keine sonstigen Plugins
- NoScript (JavaScript seletiv für benötigte Domains einschalten)
- Trennung von Browser und Produktiv-Netz (Visual Firewall via Terminal-Server)

Web Application Security

- OWASP - Open Web Application Security Project - http://owasp.org/
- OWASP TOP 10

OWASP TOP 10

- A1: Injection
- A2: Cross-Site-Scripting
- A3: Broken Authentication and Session ..
- A4: Insecure Direct Object Reference
- A5: Cross-SiteRequest-Forgery

OWASP TOP 10

- A6: Security Misconfiguration
- A7: Insecure Cryptographic Storage
- A8: Failure to restrict URL Access
- A9: Insufficient Transport Layer Protection
- A10: Unvalidated Redirects and Forwards

HTTP Grundlagen

```
GET / HTTP/1.1
Host: bithalde.de
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.9) Gecko/
20100330 Fedora/3.5.9-1.fc11 Firefox/3.5.9
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.8,de-de;q=0.5,de;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```


HTTP Grundlagen

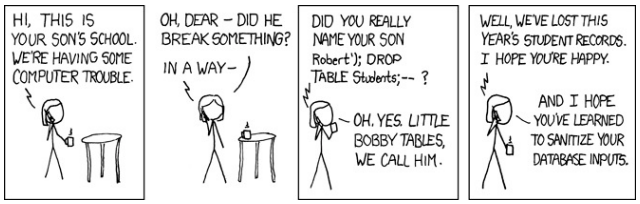
```
HTTP/1.1 200 OK
Date: Thu, 17 Jun 2010 13:55:20 GMT
Server: Apache/2.2.3 (CentOS)
Set-Cookie: jsession=3e0e9901cd4637cb1dd64274eebe47497c442e39c473dcd0214c3b525a5265cd;Path=/
Last-Modified: Thu, 29 Apr 2010 11:08:27 GMT
ETag: "690004-21c-4855e255b40c0"
Accept-Ranges: bytes
Content-Length: 540
Connection: close
Content-Type: text/html; charset=UTF-8

<html> .... 540 bytes
```


Injection Angriffe

- SQL Injection, Shell Injection, ...
- Interpretation von Daten als Code
- Der “Buffer Overflow” des Web Zeitalters

SQL Injection



SQL Injection

Cross-Site-Scripting (XSS)

- Injection Angriff auf HTML Ebene im Response
- Server wird dazu gebracht, Seite mit JavaScript Code auszuliefern, die so nie auf dem Server vorhanden waren

Cross Site Scripting

- Diebstahl von Daten
  - Authentisierung,
  - Trojaner?! - abgreifen von Kontodaten
- Würmer

Broken Authentication and Session Mangmt.

- sessionid in url (referer)
- cockie oder session id ist nicht zufällig
- Sessions haben keinen Timeout / Logout
- session fixation
  - change session id after login

Broken Session Management

- ‘forgot password’ procedure
- ‘secure cookies’

Session Highjacking

- Authentisierungs is Session Basiert
- Übernahme einer fremden Session

Insecure direct Object reference

- Datenbank-ids vom client auf berechtigung prüfen.
- Besser: Liste der ids in der Session halten und dem client nur eine reference auf die session geben
- dasselbe mit filenamen

Forced Browsing

- angriff, bei dem einfach die bekannten Parameter modifiziert und ausprobiert werden.

Cross Site Request Forgery

- GET Requests besonders anfällig
- POST requests mit JavaScript oder eigenen Forms möglich
- einbinden eines hidden parameters in die form, der pro session oder pro request eindeutig ist

Facebook Wurm

Lösungen

- Secure Software Development Process
- Source code analysis
- Pen Test / Vulnerability Scan

Lösungen

- Web Application Firewall
  - zweites Netz, Hot Patching
  - Blacklist, Whitelist
  - Session Protection, Form Protection

Prozesse

- Sicherheit im Unternehmen
- Compliance
- Relevante Standards

Sicherheit im Unternehmen

- Warum? Firmen machen nur Sachen, die wichtig fürs Geschäft sind

Sicherheit im Unternehmen

- Gefahren für's Geschäft
  - Brand im Rechenzentrum
  - Verlust aller Kundendaten and die Konkurenz
  - Schlechte Presse
- Compliance
  - Gesetze oder andere Vorschriften

Sicherheitsprozess

- Erfassen von Assets
  - Was sind die zu schützenden Werte
  - z.B. Kundendatenbank
  - z.B. die Fähigkeit, die Produktion zu managen

Sicherheitsprozess

- Definition von Schutzielen pro Asset
  - Welchen Schutz will ich erreichen
  - z.B. Kundendaten dürfen nicht zur Konkurrenz
  - z.B. Die Logistiksystem dürfen nie länger als 6 Stunden ausfallen

Sicherheitsprozess

- Bedrohungsanalyse
  - Was kann meine Schutziele gefährden
  - z.B. Brand im Rechenzentrum / Flugzeugabsturz / Erdbeben / Hochwasser
  - z.B. Hacker / Industrispionage
  - z.B Mitarbeiter der sauer auf den Chef ist

Sicherheitsprozess

- Bedrohungsanalyse 2
- Wie gefährlich sind die gefundenen Bedrohungen?
  - Wie warscheinlich ist das auftreten
  - Welchen Schaden können diese anrichten

Sicherheitsprozess

- Priorisierung der Bedrohungen
  - Schadenshöhe x Schadenswarscheinlichkeit

Sicherheitsprozess

- Erstellung eines Massnahmenkatalogs zur Verhinderung der Bedrohungen
- Kosten / Nutzen
- Priorisierung der Massnahmen

Sicherheitsprozess

- Umsetzen der Massnahme
- Kontrolle der Umsetzung
- gehe zu: Bedrohungsanalyse

Sicherheitsprozess

- Notfallplan
  - Was muss wer wann und wie tun, wenn eine der Bedrohungen eingetreten ist

Compliance

- Warum muss ein Unternehmen was tun, selbst wenn es keine Lust hat?
  - Basel II, SOX, KonTraG
  - PCI DSS
  - Datenschutzgesetz
  - HIPPA (Health Insurance Portability and Accountabillity Act)

Basel II

- 2004, 2009
- Regulierung im Bankenbereich
- Auswirkungen auf Kreditvergabe - deswegen werden auch Kreditempfänger einem Audit auf Risiken unterzogen

SOX

- Sarbanes-Oxley Act 2002
- Reaktion auf ENRON
- Gilt für alle börsengelisteten Firmen

KonTraG

- Gesetz zur Kontrolle und Transparenz im Unternehmensbereich
- Regelt die Haftung von Vorstand und Aufsichtsrat
- Verpflichtet zur Implementation eines Risikifrüherkennungssystems

PCI DSS

- Payment Card Industry Data Security Standard
- Relevant für alle Zahlungsdienstleister

Standards

- BSI Grundschutzhandbuch
- BS 7799, ISO 17799, ISO 27002
- ISO 27001

ISO 27001

- spezifiziert die Anforderungen für Herstellung, Einführung, Betrieb, Überwachung, Wartung und Verbesserung eines dokumentierten Informationssicherheits-Managementsystems unter Berücksichtigung der Risiken innerhalb der gesamten Organisation.

BSI Grundschutzhandbuch

- hilft bei Grundschutz
- Grundschutz: versicht auf Komplette Sicherheitsanalyse
- pauschale Gefährdungen
- Kochrezept für ein normales Schutzniveau

Grunschutzhandbuch

- Strukturanalyse
- Grundwerte (Vertraulichkeit, Integrität, Verfügbarkeit)
- Schutzbedarfserstellung pro System (niedrig-mittel, hoch, sehr-hoch)
- Basis Sicherheitscheck

Grundschutzhandbuch

- Bausteine
- Gefährdungskatalog
- Massnahmenkatalog

Bausteine

- B 1: Übergreifende Aspekte der Informationssicherheit
- B 2: Sicherheit der Infrastruktur
- B 3: Sicherheit der IT-Systeme
- B 4: Sicherheit im Netz
- B 5: Sicherheit in Anwendungen

Gefährdungskatalog

- G 1: Höhere Gewalt
- G 2: Organisatorische Mängel
- G 3: Menschliche Fehlhandlungen
- G 4: Technisches Versagen
- G 5: Vorsätzliche Handlungen

Massnahmenkatalog

- M 1: Infrastruktur
- M 2: Organisation
- M 3: Personal
- M 4: Hard- und Software
- M 5: Kommunikation
- M 6: Notfallvorsorge

ENDE

- Feedback - gerne per mail
  - mirko.dziadzka@gmail.com

ENDE

- Angebote von art of defence
  - Praktikum, Diplomarbeit, Jobs
  - Softwareentwicklung: Python, Java, C
  - Security Consulting / Support