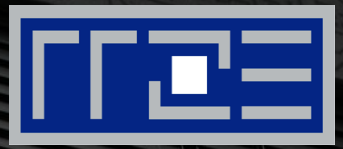
REGIONALES RECHENZENTRUM ERLANGEN [RRZE]



Elementare Sicherheitsmaßnahmen

Praxis der Datenkommunikation

18.11.2015, Holger Marquardt, RRZE

Gliederung

1. Angriffe
2. Schwachstellen
3. Maßnahmen
    1. Systemsicherung
    2. Paketfilter
    3. Netzwerkanalysen
    4. Erkennen von Angriffen
    5. Verhalten im Angriffsfall
4. Zusammenfassung

1. ANGRIFFE

Generelles

- Wer?
  - „Script-Kiddies“, Hacker, Cracker
  - Kriminelle mit „Grundlagenwissen“, Firmen, Staaten(?)
- Warum?
  - Ausspionieren von lokalen Daten
  - Ablage von urheberrechtlich geschützten oder strafbaren Daten
  - Demonstration von Fähigkeiten / persönlichem Ehrgeiz
  - Nutzung von sonst nicht zugänglichen Ressourcen
  - Wirtschaftskriminalität / (Industrie-)Spionage
- Wie?
  - (Distributed) Denial of Service Angriffe
  - Viren, Trojaner, Würmer
  - Spezifisches Angreifen von Schwachstellen
  - Remote Ausführung von eingeschleustem Schadcode („Exploit“)
  - Einbrüche durch gehackte / bekannte Passwörter

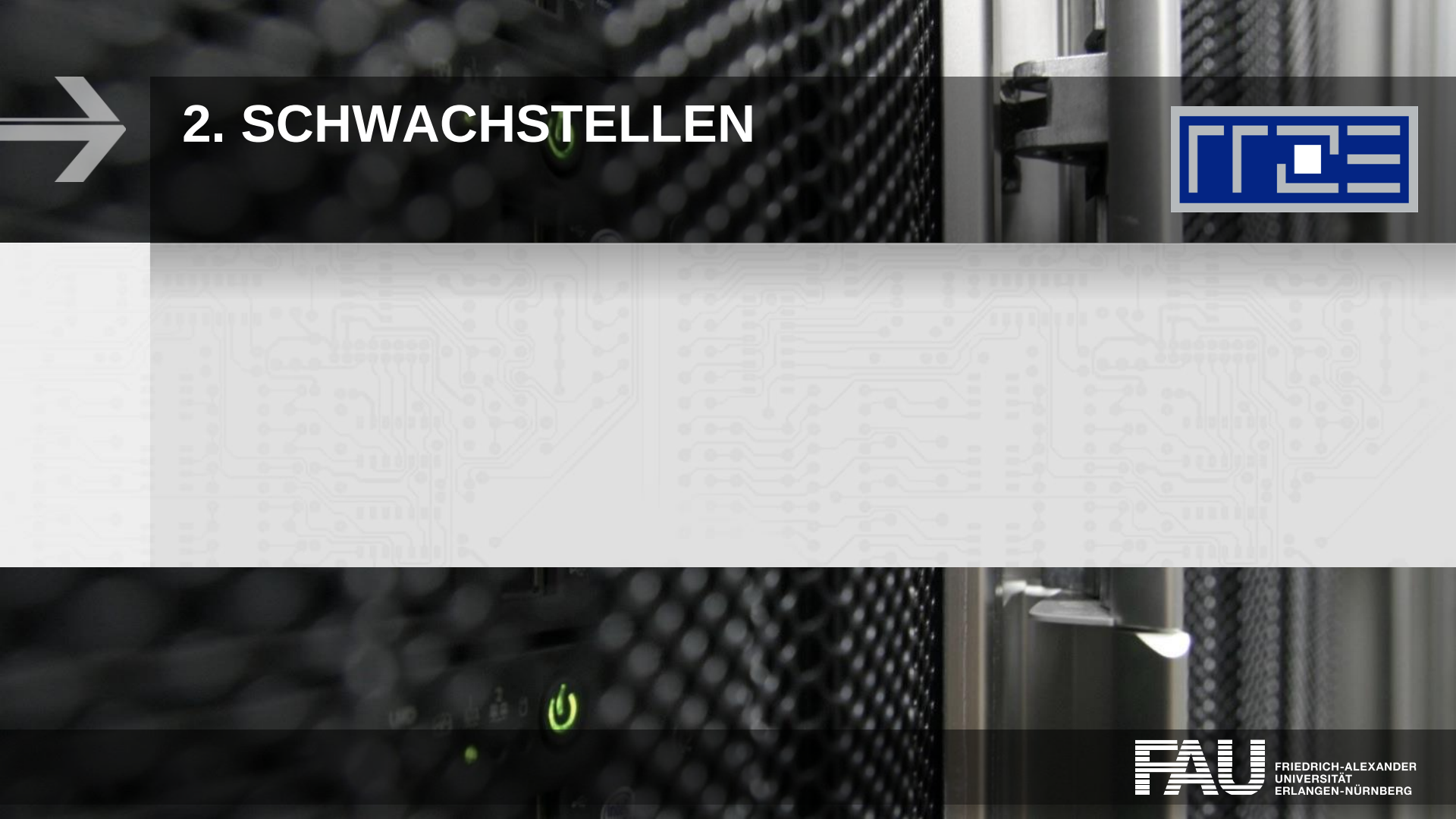
Methodik: Wie geht ein Angreifer vor?

- Feststellen des IP-Adressraums (Footprinting)
- Absuche nach Schwachstellen (Scanning)
- Suche nach Nutzernamen, freigegebenen Verzeichnissen (Enumeration)
- Erlangen von Benutzerrechten auf Systemen (Gaining Access) und Installation von Malware (Viren, Trojaner, Rootkits)
- Erhalten von Root-Rechten (Escalating privilege), z.B. Versuch über „Standard“-Logins (Beispiel: admin, admin)
- Social Engineering

Nach erfolgreichem Angriff

- Zugriff auf Nachbarsysteme (Pilfering)
  - Infos über System / Zugänge zu anderen Systemen sammeln, z.B. durch Sniffing
- Spuren beseitigen (Covering tracks)
  - Spuren in Log-Files löschen
  - System-Binaries (ps, netstat, …) durch Rootkits ersetzen (automatisierte Tools, die Angreifer verdecken, Bsp.: lrk4)
- Hintertüren schaffen (Creating Backdoors)
- Spam / Phishing
- (Distributed) Denial of Service Attacken (DDoS)
  - Zombies

2. SCHWACHSTELLEN



Bekannte Schwachstellen

- Benutzer installiert seine Software selbst
- Angreifbare Systeme
  - Systeme werden nicht gepatched
  - Benutzerkennungen ohne Passwörter
  - Kein Virenschutz
  - Keine Einschränkung der erreichbaren Dienste (Packetfilter, Firewalling)
- Verwendung unsicherer (unverschlüsselter) Programme (telnet, login)

Problemfall Authentifizierung / Berechtigungen

- Nutzer ist Sicherheitsrisiko Nr. 1
  - Typische Passwörter: Username, Vorname, Name des Hundes/der Katze, Geburtsdatum, „1234“
- Passwörter können erraten werden
- *Brute Force*: Systematisches Durchprobieren des gesamten Schlüsselraumes

> *Brute force is a trial and error method used by application programs to decode encrypted data through exhaustive effort (using brute force) rather than employing intellectual strategies. A brute force cracking application proceeds through all possible combinations of legal characters in sequence. Brute force is considered to be an infallible, although time-consuming, approach.*

Loginversuche “bruteforce” via SSH

Fallbeispiel: Heartbleed

- Entdeckung April 2014: OpenSSL-Versionen 1.01 bis einschl. 1.01f betroffen (Fehlerhafter Code erstmals 14. März 2012 veröffentlicht)
- Potentiell Zugriff auf privaten Schlüssel des Serverzertifikats und andere Daten im Arbeitsspeicher: Benutzernamen, Passwörter, persönliche Daten, etc.
- OpenSSL-basierte Dienste (d.h. https, radius, EAP, …) auf betroffenen Systemen akut unsicher geworden
- Sämtliche Zertifikate und Daten auf betr. Systemen kompromittiert

→ Akuter Handlungsbedarf bei Systemen und ausgestellten Zertifikaten

3. MAßNAHMEN

3.1 Systemmaßnahmen

Absicherung der Systeme

- Benutzerverhalten
  - Optimal: Benutzer installiert keine Software
  - Sensibilisieren: Software aus dem Internet birgt Risiken
- Verwendung *sicherer* Programme / Protokolle
  - z.B. ssh statt telnet, https statt http
  - Perfect Forward Secrecy (PFS)
- Systeme schwerer angreifbar machen
  - Systeme (automatisch) patchen
  - Keine Benutzeraccounts ohne Passwörter
  - Virenschutz
  - Einschränkung der erreichbaren Dienste
  - Paketfilter / Firewall

Überwachung der Systeme – Auditing

- Die (normale) Nutzung dokumentieren
- Verletzungen der Regeln melden / loggen
- Wirkung der Regeln überprüfen
- Unregelmäßigkeiten und Änderungen sollen erkennbar sein, Revisionskontrolle
- Informationen sollen verlässlich sein

Überwachung der Systeme – Logging

- Nutzung von Logfiles:
  - Optimale Lösung: dedizierter Log-Server
  - Pattern Matching: Meist Suche nach bestimmten Schwachstellen
    (Aktuelle Exploits: https://portal.cert.dfn.de/adv/archive/
    http://www.us-cert.gov/ncas/alerts)
- Betriebssystem-Funktionen:
  - UNIX: „*last*“ – letzter Login-Versuch von ..
  - Windows: Login-Monitoring über Registry aktivieren
  - Datenschutz beachten!
  - Vorsicht bei Logfiles mit personenbezogenen Daten

3. MAßNAHMEN

3.2 Paketfilter

Allgemein

- Filtern von bestimmten Netzwerkpaketen
- Filterung auf Layer 3 (adressbasiert) und/oder Layer 4 (portbasiert)
  - z.B. Filtern von Paketen an Rechner 131.188.79.246 Port 23 TCP (telnet)
- „Verbergen“ von Diensten
- Keine Überprüfung der IP/TCP/UDP Payload
- Keine Informationsveränderungen

Allgemein

- Reduktion der Angriffsfläche
- Bieten Schutz für verwundbare Dienste und Protokolle
- Erkennen von Angriffsversuchen (Logging)
- Sind auf Kontrolle durch den Administrator angewiesen
- Benötigen evtl. viele Ressourcen

Realisierung

- Zugelassene Protokolle müssen bekannt sein
  - Portbeziehungen
  - Unterscheidung incoming / outgoing traffic
- Positivlisten sind zu bevorzugen
  - Gezielt einige wenige Dienste erlauben
  - Rest per Default sperren
- Stateful Packet Inspection
  - „Intelligenter“ Paketfilter
  - Permanente Buchführung über Verbindungsbeziehungen
  - Dynamische Öffnung von Ports, z.B. für
    - Antwortpakete aus der Gegenrichtung
    - Für protokollmäßig erforderliche neue Datenverbindung (z.B. FTP Datenkanal)
  - Schwächer: TCP „SYN“-Filterung inbound

Grenzen

Paketfilter…

- lösen nicht das Insiderproblem
- können die Vertraulichkeit nicht gewährleisten
- beheben keine Fehler in der Implementierung zugelassener Dienste
- analysieren nicht den Inhalt von Paketen (benötigen demgegenüber aber weitaus weniger Ressourcen)
- sind nur so sicher wie die Regeln, die sie beschreiben

Probleme

- Zuordnung Dienst <---> Portnummer kann abgeändert werden (z.B. Webserver auf SSH-Port)
- Viele Applikationen sind mit Standard-Paketfiltern nur sehr schwer erfassbar (z.B. Skype, P2P-Programme,…)
- Generelle Vorgehensweise:
  - Sicherheitsregeln aufstellen (Security Policy)
  - Filterregeln erstellen, implementieren und testen
  - Dokumentation!

Einsatzmöglichkeiten

- Einsatz auf zuständigen Subnetz-Router:
  - Möglichkeit der Filterung von allen Paketen, die von Außen ins Subnetz gelangen

- Einsatz lokal auf Rechner:
  - Möglichkeit der Filterung von allen Paketen, die den jeweiligen Rechner erreichen (d.h. auch aus gleichem Subnetz)

Einsatz auf Routern

- Pro:
  - Paketfilter auf Router können das komplette Subnetz nach Außen hin schützen
  - (Teilweiser) Schutz gegen (D)DoS-Angriffe
  - Extrem performant
- Contra:
  - Kein Schutz gegen Angreifer im gleichen Subnetz
- Im Bereich der FAU:
  - Absicherung von ganzen Subnetzen durch Paketfilter auf den Routern
  - Als Dienstleistung vom RRZE angeboten
  - acl@fau.de

Einsatz auf Routern

- Beispielkonfiguration Cisco-Router im Bereich der FAU:
- Definition von Access-Control-Listen (ACL):

  ip access-list extended <name>
  permit <protocol> <rule>
  deny <protocol> <rule>

- Bsp.:

  ip access-list extended ACL42
  permit icmp any any echo-request
  permit icmp any any echo-reply
  deny icmp any any
  permit tcp any host 10.10.8.10 eq telnet
  deny ip any any

Einsatz lokal auf Rechner

- Pro:
  - Schutz auch gegenüber Rechnern im gleichen Subnetz
  - Flexibilität
- Contra:
  - Vom Prinzip her unsicherer: Pakete werden erst gefiltert, wenn sie den Rechner schon erreicht haben
  - Kaum Schutzmöglichkeit im Falle von Flooding- / (D)DoS-Attacken
  - Schwer zu verwalten bei mehreren inhomogenen Systemen
- Für nahezu alle Betriebssysteme verfügbar:
  - Windows (ab 2000): Integriert
  - Linux: IPTables
  - Solaris / BSD: IPFilter
  - „personal Firewalls“
- Erfordert Grundlagenwissen!

UTM / Next Generation Firewalls

- Deep Packet Inspection / Application Awareness
- Reputation based filtering / URL filtering
- User Awareness / Anti Malware / Anti Virus
- Globale Vernetzung
- Pro:
  - Oberes Ende des technisch Machbaren
  - Sehr hohe Schutzwirkung
  - Sehr individuell konfigurierbar (Nutzer <---> IP-Adressen)
  - Globale Vernetzung
- Contra:
  - Hohe Anschaffungs- und laufende Kosten
  - Datenschutz
  - Beliebig hoher Konfigurationsaufwand
  - Globale Vernetzung

3. MAßNAHMEN

3.3 Netzwerkanalyse

Netzwerkanalyse

- Offenbarung von unsicheren Diensten (Portscan)
- Informationen über die Systeme
- Monitoring von aufgebauten Verbindungen (Aufspüren von Eindringlingen)
- Erkennung fehlender Patches
- Erkennung fehlerhafter Konfiguration

→ Automatische Werkzeuge zum Entdecken von Sicherheitslücken. **Aber**:

- Genaue Analyse der Ergebnisse ist erforderlich
- Nur im eigenen Bereich!

Netzwerkanalyse

- OS-Fingerprinting:
  - Herausfinden des verwendeten Betriebssystems auf Zielrechner
  - Bsp.: nmap -O <hostname/ip>

- Suspekte Rechner vor Ort überprüfen
  - Welche Ports sind offen?
    - Unix / Windows: „netstat -an“
  - Welcher Prozess hält welchen Port offen?
    - Windows XP oder höher: Process Explorer (https://technet.microsoft.com/de-de/sysinternals/processexplorer.aspx)
    - Unix: lsof (http://freecode.com/projects/lsof)

Sniffing

- Passives Abhören des Netzwerks nach Schlüsselinformationen
- Anschluss ans Netzwerkmedium nötig oder Installation auf Zielmaschine oder Router
- Abgriff von Authentifizierungsdaten im Klartext, z.B. Port 23 (telnet), Port 21 (ftp), Port 80 (www)
- Standard-Programme:
  - Wireshark (OpenSource): https://www.wireshark.org
  - Tcpdump (Unix): http://www.tcpdump.org
  - snoop (Solaris)
- **Vorsicht**: StGB §202a, §202b

Beispiel: tcpdump

- Einsatz unter Linux / Unix
- Capture und Analyse von Netzwerkverkehr

X> tcpdump host Y.rrze.uni-erlangen.de

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:49:03.575215 IP X.rrze.uni-erlangen.de > Y.rrze.uni-erlangen.de: ICMP echo request, id 1, seq 4, length 40
12:49:03.575244 IP Y.rrze.uni-erlangen.de > X.rrze.uni-erlangen.de: ICMP echo reply, id 1, seq 4, length 40
12:49:04.579558 IP X.rrze.uni-erlangen.de > Y.rrze.uni-erlangen.de: ICMP echo request, id 1, seq 5, length 40
12:49:04.579566 IP Y.rrze.uni-erlangen.de > X.rrze.uni-erlangen.de: ICMP echo reply, id 1, seq 5, length 40
```

Portscanning

- Automatisiertes Überprüfen, ob bestimmte (TCP-, UDP-) Ports auf bestimmten Rechnern erreichbar („offen“) sind bzw. ob dort ein Dienst angeboten wird
- Extremfall: „full-scan“, durchprobieren aller Ports auf (ggf. mehreren) Rechnern
- Information: Erkennen von Ports, die nicht per Paketfilter geschützt sind
- Was erreichbar sein soll, ist damit auch sichtbar

- **Vorsicht**:
  - Portscanning auf fremde Rechner kann je nach Rechtslage / Providerrichtlinien als harmlos bis Straftatbestand bewertet werden (StGB §202c)

Beispiel: nmap

- www.nmap.org
- Portscanner, der auch Betriebssystem erkennen kann
- Multiplattform – für (fast) alle Betriebssysteme
- Kommandozeilen-Werkzeug aber auch graphische Oberfläche
- Root- / Administrator-Rechte notwendig
- Funktionen:
  - Host discovery
  - Portscanning
  - OS fingerprinting

nmap: Optionen

- -v: verbose
- -s: Scanning
  - -sP: Ping scanning: Senden von ICMP echo requests
  - -sS: Senden von „TCP-Pings“ mit gesetztem SYN-Bit (nur mit Root-Rechten!):
    - SYN-ACK: Port offen
    - RST: Port geschlossen
  - -sU: UDP scans: Welche UDP Ports sind offen?
- -O: Versuch, Betriebssystem, Uptime, etc. festzustellen
- -p <nr>-<nr>: Abgefragte Portnummern
- -g <nr>: Portnummer der Source setzen
- -Pn: Hosts werden vor Portscanning nicht angepingt

nmap in Aktion

X> nmap –sS –p 25 Y

Starting Nmap 6.40 (http://nmap.org) at 2015-11-17 13:52 CET
Nmap scan report for Y.rrze.uni-erlangen.de (131.188.x.y)
Host is up (0.00040s latency).
PORT STATE SERVICE
25/tcp filtered smtp

Nmap done: 1 IP address (1 host up) scanned in 1.32 seconds

Ausgabe von tcpdump:

15:50:02.63728 X.rrze.uni-erlangen.de > Y.rrze.uni-erlangen.de: icmp: echo request
15:50:02.63729 X.rrze.uni-erlangen.de.39254 > Y.rrze.uni-erlangen.de.http: . ack
15:50:02.63735 Y.rrze.uni-erlangen.de > X.rrze.uni-erlangen.de: icmp: echo reply
15:50:02.63738 Y.rrze.uni-erlangen.de.http > X.rrze.uni-erlangen.de.39254: R
15:50:09.03598 X.rrze.uni-erlangen.de.39232 >Y.rrze.uni-erlangen.de.smtp: S
15:50:09.03605 Y.rrze.uni-erlangen.de.smtp > X.rrze.uni-erlangen.de.39232: R ack

nmap: Analyse

- Ports können offen, geschlossen und gefiltert sein

	TCP - Port	UDP - Port
erfolgreiche Verbindung (SYN-ACK)	open	-
ICMP „Port unreachable“	closed	closed
Verbindung abgelehnt (RST)	closed	-
keine Antwort	filtered	open/filtered
Firewall, z.B. ICMP „Administratively Prohibited“	filtered	filtered

3. MAßNAHMEN

3.4 Angriffserkennung

Angriffserkennung

- Verdacht auf laufenden Angriff:
  - Rechner verhält sich „ungewöhnlich“
  - Netznutzung „ungewöhnlich“
- Systemanalyse:
  - Systemintegritätscheck (z.B. aide.sourceforge.net oder www.tripwire.org)
  - Auswertung von Log-Dateien
- Netzwerkanalyse:
  - Werkzeuge kennen und anwenden, die auch Angreifer verwenden
  - Monitoring von Netzwerkverkehr
  - Scannen von Ports und Rechnern
  - automatisierte Sicherheitstools, um Schwachstellen zu erkennen (z.B. IDS, Schwachstellenmanagement,…)

Intrusion Detection System

- Spezialisierte Software/Hardware, die Netzwerkverkehr sammelt und analysiert (bestehend aus Sensoren)
- Ziel: Feststellung, ob
  - ein potentieller Angreifer nach Sicherheitslücken sucht
  - gerade ein Angriff (von innen oder außen) stattfindet
  - die Firewall Schlupflöcher hat
  - ein Angreifer erfolgreich gewesen ist
  - Informationen geändert bzw. entwendet wurden
- Unterschied zur Firewall:
  - Firewalls filtern aktiv Datenverkehr anhand von Layer 3-4 Paketinfos, IDS warnen (passiv) vor „verdächtigen“ Paketen (inkl. Payload)

Intrusion Detection System

- Host-basiertes IDS (HIDS):
  - Systemintegrität: DB mit Soll-Zustand, Datei-Attributen, Checksummen
- Anomalie-basiertes (Netzwerk-) IDS:
  - IDS lernt *normales* Netzwerkverhalten und triggert auf abnormalem Verhalten (Möglichkeit, unbekannte Angriffe zu erkennen)
- Regel-basiertes (Netzwerk-) IDS:
  - Definition von *Signaturen*, welcher Netzwerkverkehr als Angriff bewertet wird – in diesem Fall Alarmierung (ohne dass Angreifer es merkt)
  - Zu Beginn Anpassung an die lokalen Bedingungen
  - Permanente Aktualisierung ist erforderlich

Beispiel Snort

- Snort (www.snort.org)
  - Regelbasierte, opensource IDS-Software
  - Untersucht Header und Payload von Paketen („Signatur“)
  - Flexibles Erstellen von Signaturen
  - Bereits große Menge an Regeln/Signaturen vorhanden, neue erscheinen unmittelbar, nachdem Exploits bekannt werden
  - Stateless Rules, d.h. Anwendung auf jedes neue Paket

Beispiel Snort

Beispielszenario: Portscan von Rechner 131.188.x.y nach Zielsystem 131.188.x.z:
(nmap –O –sS 131.188.x.z)

SNORT-Regel auf Zielsystem:

alert tcp 131.188.0.0/16 any -> 131.188.0.0/16 any (msg:"SCAN nmap fingerprint attempt";flags:SFPU;
reference:arachnids,05; classtype:attempted-recon; sid:629;rev:1;)

→ Erzeugt SNORT Warnung auf Zielsystem:

[**] SCAN nmap fingerprint attempt [**]
11/17-15:40:42.002683 131.188.x.y:35604 -> 131.188.x.z:7
TCP TTL:57 TOS:0x0 ID:24301 IpLen:20 DgmLen:60
**U*P*SF Seq: 0x732E0205 Ack: 0x0 Win: 0x800 TcpLen: 40
UrgPtr: 0x0 TCP Options (4) => WS: 10 NOP MSS: 265
TS: 1061109567 0

IPS: Intrusion Prevention System

- Verknüpfung von IDS und Firewall
- Erkennt nicht nur intelligent Angriffe, sondern blockiert sie auch (soweit möglich)
- Pro:
  - Höherer Schutz
  - Evtl. Datenschutzverträglicher
- Contra:
  - Fortlaufende Pflege des Systems nötig
  - Datendurchsatz beschränkt

IPS-System im Bereich der FAU

- Bis ca. 2011 an zentralen NAT-Gateways
- Abfangen von Angriffen/Viren/Trojaner/Malware,… soweit möglich von infizierten Klienten (z.B. WLAN-Laptops)
- → Schutz des Internets vor Angriffen „von Innen“
- Datenschutz-Aspekt: Abuse-Problematik eindämmen ohne Logging

3. MAßNAHMEN

3.5 Verhalten im Angriffsfall

Verhalten im Angriffsfall

- Keine Panik!
- Vorbereitet sein
- Abschalten des Systems
- Ggf. Informieren des Sicherheitsteams im RRZE (abuse@fau.de)
  - Sicherung des Datenbestandes
  - Anweisungen des RRZE befolgen
  - Neuinstallation bzw. Patchen & Virenschutz

Forensische Analyse

- Überprüfen der Integrität des Filesystems (HIDS)
- Wenn keine zusätzlichen Logins: evtl. Rootkits installiert
- Ohne Vorarbeit relativ aufwendig

4. ZUSAMMENFASSUNG

Zusammenfassung

- Vermeiden von typischen Fehlern:
  - Keine Pflege der Systeme
    - Dummy- sowie alte Benutzerzugänge
    - Veraltete Software mit Fehlern
    - Keine regelmäßige Kontrolle der Logs
  - Unverschlüsseltes Lagern bzw. Übertragen wichtiger Daten
  - Blindes Vertrauen in kompromittierte Systeme
- Prävention:
  - Regelmäßiges (automatisches) Patchen
  - Reduzierung der angebotenen Services
  - Passwörter und Logging
  - Einsatz von Scannern auf Rechnern in eigenen Netz-Bereichen
  - Regelmäßige Kontrolle der Logfiles
  - Werkzeuge zur Erhöhung der Sicherheit wie ssh

Sicherheit an der FAU

- Unterstützung bei der Konfiguration der Systeme durch das RRZE
- Bereitstellung von Antiviren-Software
- Automatisches Einspielen von Patches
- Erkennung von Scans
- Sicherung der Systeme in Eigenverantwortung
  - Sperrung aller Klartextdienste (Zugang beispielsweise nur per SSH)
  - Häufige Updates

Sicherheit an der FAU

- Sperrung kritischer Ports am Außenrouter zum X-WiN
- Paketfilter können auf Routern der FAU implementiert werden
  - ACLs werden ausschließlich vom RRZE gepflegt
- IT-Betreuer muss Schutzbedarf und Kommunikationsbeziehungen kennen
  - Eigene, dezentrale Lösungen an der FAU nicht erlaubt
  - Für hohen Schutzbedarf Firewall-Lösungen in Absprache mit RRZE
- Richtlinien: *www.rrze.fau.de/dienste/konditionen/datennetzrichtlinien.shtml*
- Kontakt: acl@fau.de

FRAGEN & ANTWORTEN

Vielen Dank für Ihre Aufmerksamkeit

holger.marquardt@fau.de

noc@fau.de