

Honeypots

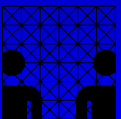
Einsatzmöglichkeiten beim Schutz von IT-Systemen und in der Erforschung von Angriffen

Andreas von Knobloch, Michael Krooß

{0knobloc|5krooss}@informatik.uni-hamburg.de

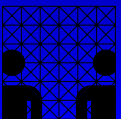
Fachbereich Informatik

Universität Hamburg

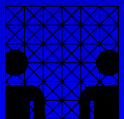


Inhalt

1. Überblick über Honeypots
Definition, Einsatzbereiche, rechtliche Aspekte, Risiken
2. Beispiele
BackOfficer Friendly, Honeyd, Honeynets
3. Analyse
4. Der Feind



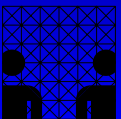
1. Überblick über Honeyspots



Definition Honeypots

„A security resource whose value lies in being probed, attacked or compromised.“

Lance Spitzner



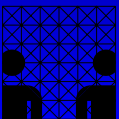
Definition Honeydumps

„A security resource whose value lies in being probed, attacked or compromised.“

Lance Spitzner

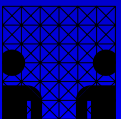
Forderungen:

- Es werden keine Produktionsdienste angeboten
- Es gibt keine autorisierte Nutzung
- Verbindungsaufbau zum Honeydum ist im Allgemeinen ein Scan oder ein Angriff
- Verbindungsaufbau vom Honeydum bedeutet das System wurde kompromittiert



Implementationsmöglichkeiten

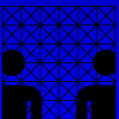
- Abhören von Ports
``netcat -l -p 80``
- Emulation von Diensten
u.U. mit bekannten Verwundbarkeiten
- Bereitstellung eingeschränkter Betriebssysteme
z.B. `chroot`-Umgebung unter Unix
- Bereitstellung von vollständigen Systemen
Einzelne Rechner oder Netze



Typen von Honey pots

Einteilung nach Marty Roesch, Entwickler von Snort (IDS)

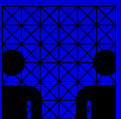
- Produktivitäts-Honey pots (production honeypots)
 - Zur Erhöhung der Sicherheit eines Netzwerks
 - Zum Schutz einer Organisation
- Forschungs-Honey pots (research honeypots)
 - Zum Sammeln von Informationen über Verwundbarkeiten, Bedrohungen und Angreifer (Werkzeuge, Motive, Taktiken)



Grad der Interaktion

Interaktionsmöglichkeiten für Angreifer

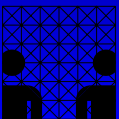
- Je höher die Interaktion desto mehr kann über Angriffe bzw. Angreifer gelernt werden
- Je höher die Interaktion desto höher die Risiken
- Je höher die Interaktion desto höher der Aufwand
- Produktivitäts-Honeypots bieten eher niedrige Interaktionsmöglichkeiten
- Forschungs-Honeypots bieten eher hohe Interaktionsmöglichkeiten



Produktivitäts-Honeypots

Einteilung nach Bruce Schneier

- **Vorbeugung vor Angriffen (prevention)**
Zur Täuschung bzw. Abschreckung von Angreifern, nur eingeschränkt wirksam
- **Erkennung von Angriffen (detection)**
Jeder Verbindungsaufbau stellt im Allgemeinen einen Angriff dar.
- **Reaktionen auf Angriffe (response)**
Untersuchung der Vorfälle wird nicht durch Produktionsabläufe behindert.

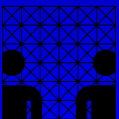


Produktivitäts-Honeypots

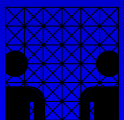
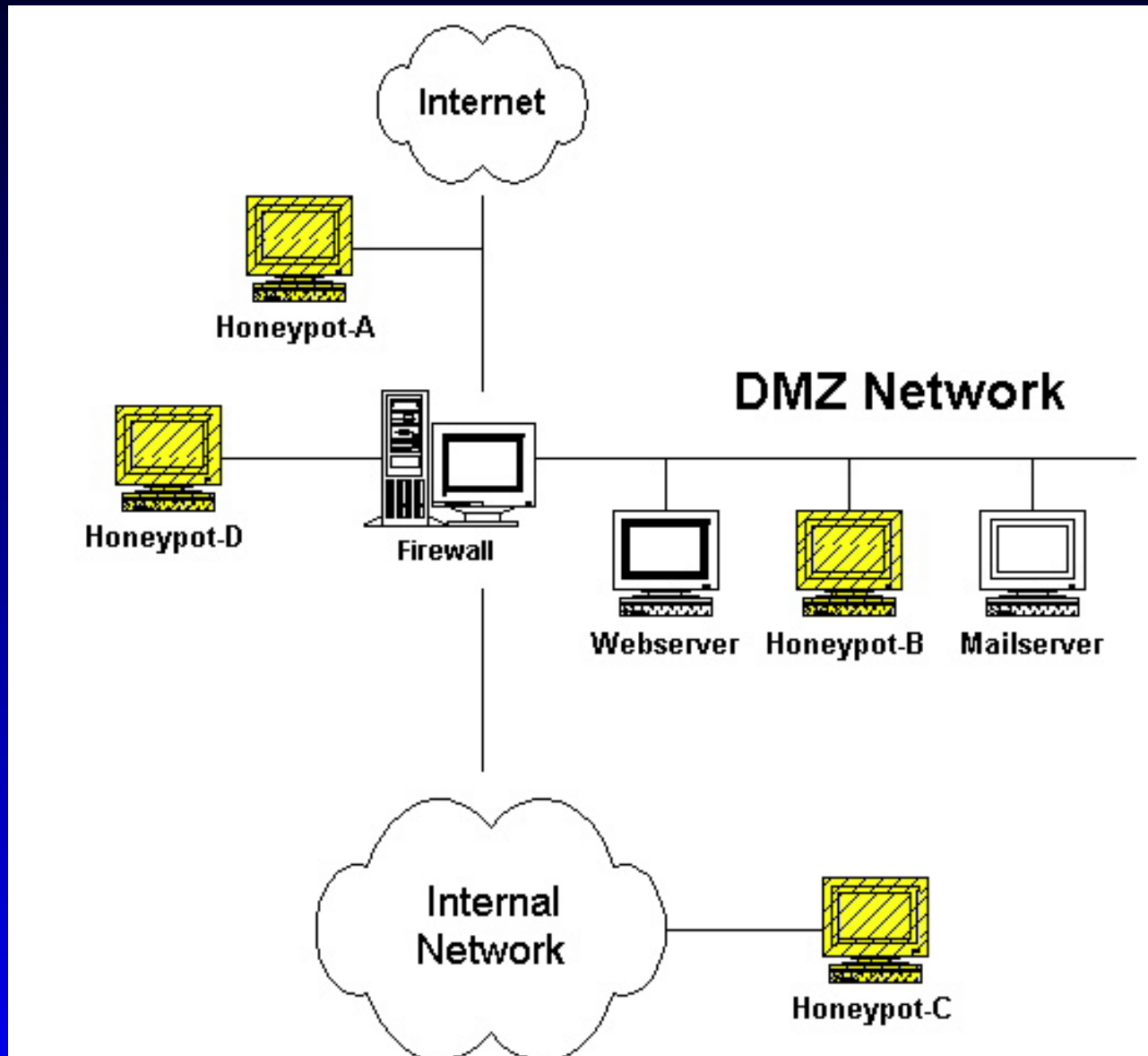
Einteilung nach Bruce Schneier

- **Vorbeugung vor Angriffen (prevention)**
Zur Täuschung bzw. Abschreckung von Angreifern, nur eingeschränkt wirksam
- **Erkennung von Angriffen (detection)**
Jeder Verbindungsaufbau stellt im Allgemeinen einen Angriff dar.
- **Reaktionen auf Angriffe (response)**
Untersuchung der Vorfälle wird nicht durch Produktionsabläufe behindert.

Honeypots sind ein Werkzeug, keine Lösung

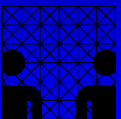


Platzierung (von Produktivitäts-Honeypots)



Forschungs-Honeypots

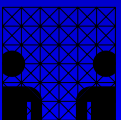
- Sammeln von Informationen über Angreifer
- Frühwarnsystem für neue Exploits oder Würmer



Forschungs-Honeypots

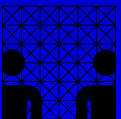
- Sammeln von Informationen über Angreifer
- Frühwarnsystem für neue Exploits oder Würmer

Most security measures are about keeping blackhats out. This one is different: It is about keeping the bad guys in.



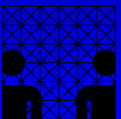
Rechtliche Aspekte

- **Datenschutz**
Dürfen von einem Angreifer übertragene oder gespeicherte Daten eingesehen und ausgewertet werden?
- **Verleitung, Anstiftung**
Stellt die Bereitstellung eines angreifbaren Systems eine Verleitung bzw. Anstiftung dar?
- **Haftung**
Bestehen Haftungsansprüche Dritter, wenn diese von einem kompromittierten Honeypot aus angegriffen wurden?



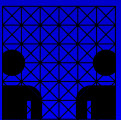
Vorteile von Honeypots

- gesammelte Daten haben einen hohen Informationswert
 - Wenige False Positives
 - Wenige False Negatives
- Einfachheit des Konzepts
- Betrieb ist relativ wenig ressourcenintensiv
- Return of Investment



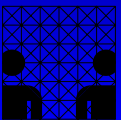
Nachteile von Honey pots

- Eingeschränkte Sicht
- Es werden keine verwundbaren Systeme geschützt
- Risiken
 - Erkennung des Honey pots z.B. durch Fingerprinting
 - Wenn kompromittiert sind weitere Angriffe insbesondere auf Dritte möglich

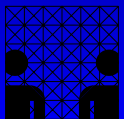


Zusammenfassung

- „Honeypots are cool“ (Lance Spitzner)
- Durch den Einsatz von Honeypots werden keine unsicheren Systeme geschützt!!!

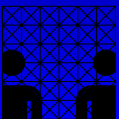


2. Beispiele



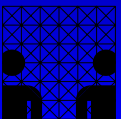
Beispiel 1: BackOfficer Friendly

- Entwickelt von Marcus Ranum (Network Flight Recorder)
- MS Windows oder Unix
- ursprünglich zur Erkennung und zum Antworten auf Verbindungsversuche des BackOrifice-Client
- Beobachtung von bis zu sieben Ports (FTP, Telnet, SMTP, HTTP, POP3, IMAP)



Beispiel 2: Honeyd

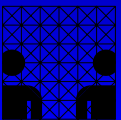
- Entwickelt von Niels Provos (Universität Michigan)
- Unix
- Beobachtung aller Ports möglich
- Beobachtung von Verbindungsversuchen zu nicht vergebenen IP-Adressen durch Blackholing bzw. ARP-Spoofing möglich
- Emulation von IP-Stacks verschiedener Betriebssysteme möglich
- Emulation der Dienste durch Plugins (nur TCP)



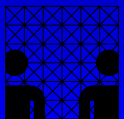
Exkurs: ARP-Spoofing

Vorgehen:

1. Funktionsweise des Address Resolution Protocol am Beispiel von IP und Ethernet
2. ARP-Spoofing von nicht vergebenen IP-Adressen in Broadcast-Netzen
3. ARP-Spoofing von nicht vergebenen IP-Adressen in geschichteten Netzen

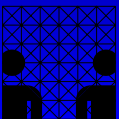


Address Resolution Protocol



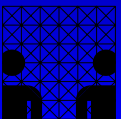
ARP Request

```
ETHER: ----- Ether Header -----  
ETHER:  
ETHER: Packet 15 arrived at 10:22:4.23  
ETHER: Packet size = 60 bytes  
ETHER: Destination = ff:ff:ff:ff:ff:ff, (broadcast)  
ETHER: Source       = 8:0:20:a1:3c:c, Sun  
ETHER: Ethertype = 0806 (ARP)  
ETHER:  
ARP: ----- ARP/RARP Frame -----  
ARP:  
ARP: Hardware type = 1  
ARP: Protocol type = 0800 (IP)  
ARP: Length of hardware address = 6 bytes  
ARP: Length of protocol address = 4 bytes  
ARP: Opcode 1 (ARP Request)  
ARP: Sender's hardware address = 8:0:20:a1:3c:c  
ARP: Sender's protocol address = 192.168.16.79, elmo  
ARP: Target hardware address = ?  
ARP: Target protocol address = 192.168.16.80, groover  
ARP:
```

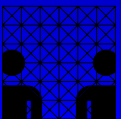
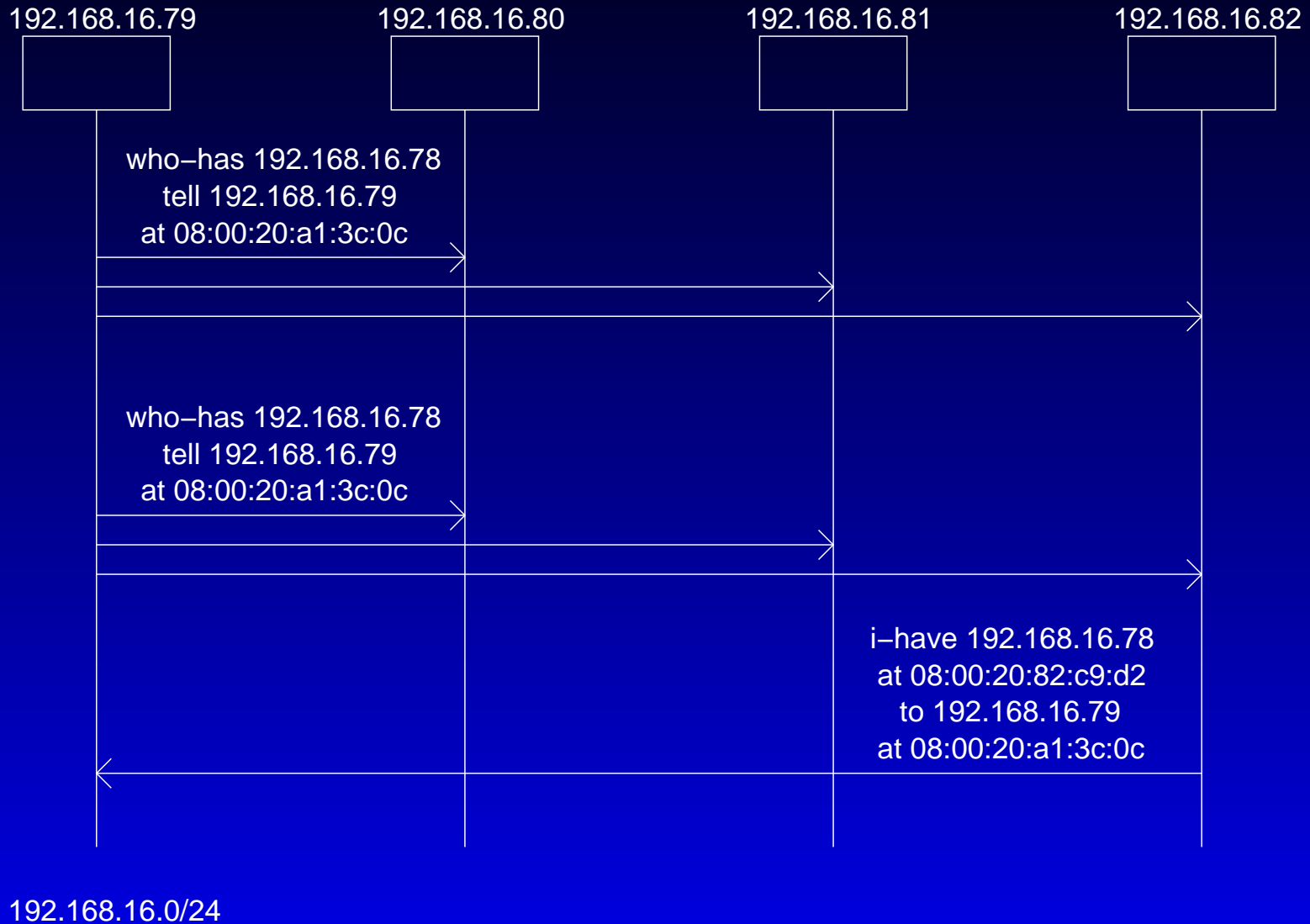


ARP Reply

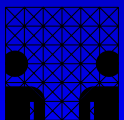
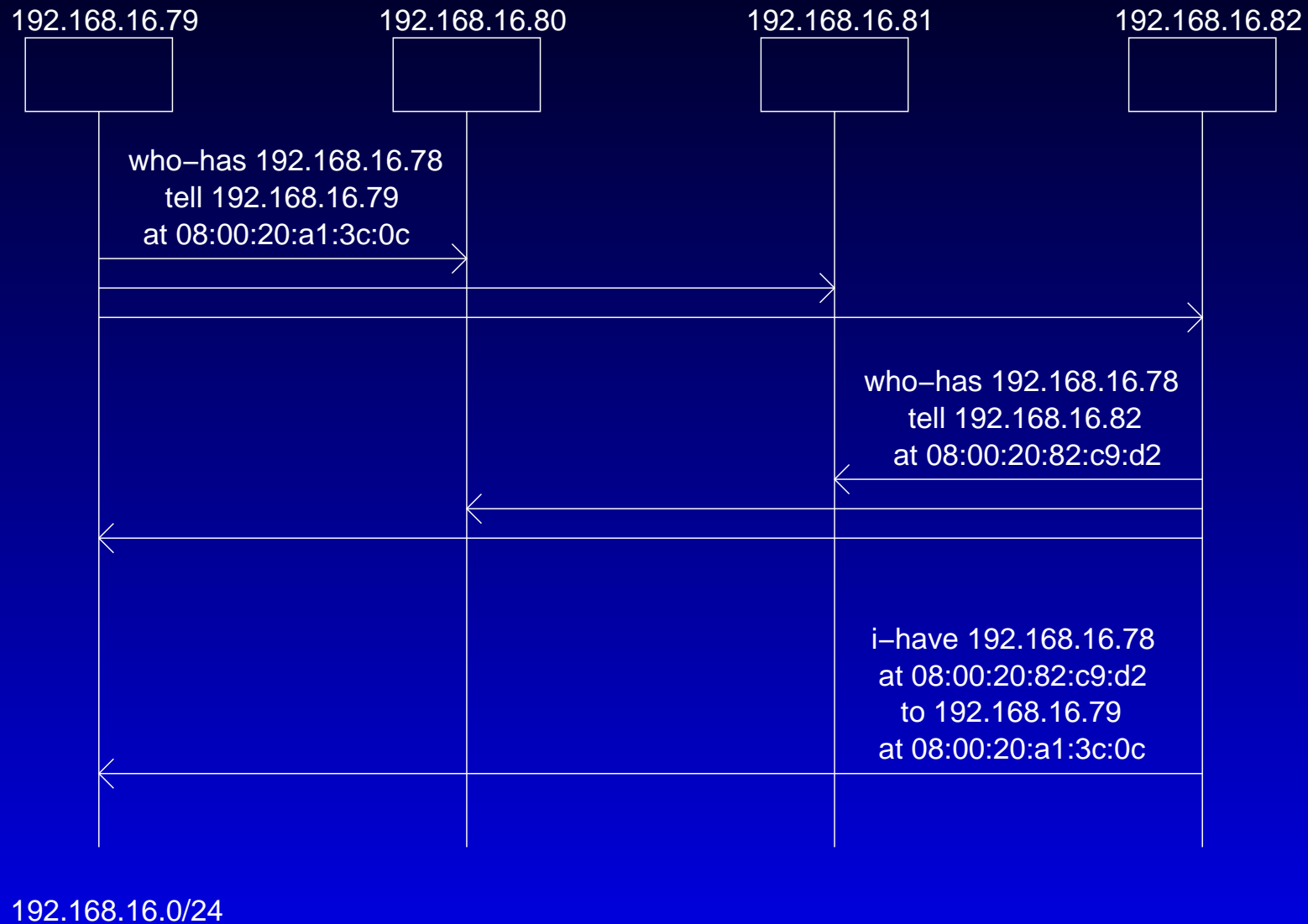
```
ETHER:  ----- Ether Header -----  
ETHER:  
ETHER:  Packet 16 arrived at 10:22:4.23  
ETHER:  Packet size = 42 bytes  
ETHER:  Destination = 8:0:20:a1:3c:c, Sun  
ETHER:  Source      = 8:0:20:a1:3c:2e, Sun  
ETHER:  Ethertype = 0806 (ARP)  
ETHER:  
ARP:  ----- ARP/RARP Frame -----  
ARP:  
ARP:  Hardware type = 1  
ARP:  Protocol type = 0800 (IP)  
ARP:  Length of hardware address = 6 bytes  
ARP:  Length of protocol address = 4 bytes  
ARP:  Opcode 2 (ARP Reply)  
ARP:  Sender's hardware address = 8:0:20:a1:3c:2e  
ARP:  Sender's protocol address = 192.168.16.80, groover  
ARP:  Target hardware address = 8:0:20:a1:3c:c  
ARP:  Target protocol address = 192.168.16.79, elmo  
ARP:
```



ARP-Spoofing (Broadcast)



ARP-Spoofing (Switching)

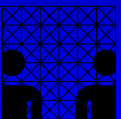


ARP-Spoofing bei Honeyd

Durch das Programm `arpd` werden unabhängig vom `honeyd` alle nicht vergebenen IP-Adressen dynamisch gespooft.

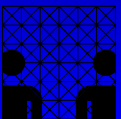
```
arpd -i hme0 192.168.16.0/24
```

```
honeyd -f /etc/honeyd.conf 192.168.16.0/24
```



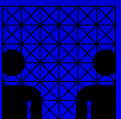
Beispiel 3: Honeynets

- Netzwerk aus Standard-Systemen
- Ein Honeynet ist eine Architektur, kein Produkt oder eine Software
- Einsatz vorwiegend zur Forschung, da hohe Interaktionsmöglichkeiten gegeben

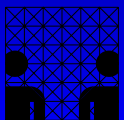
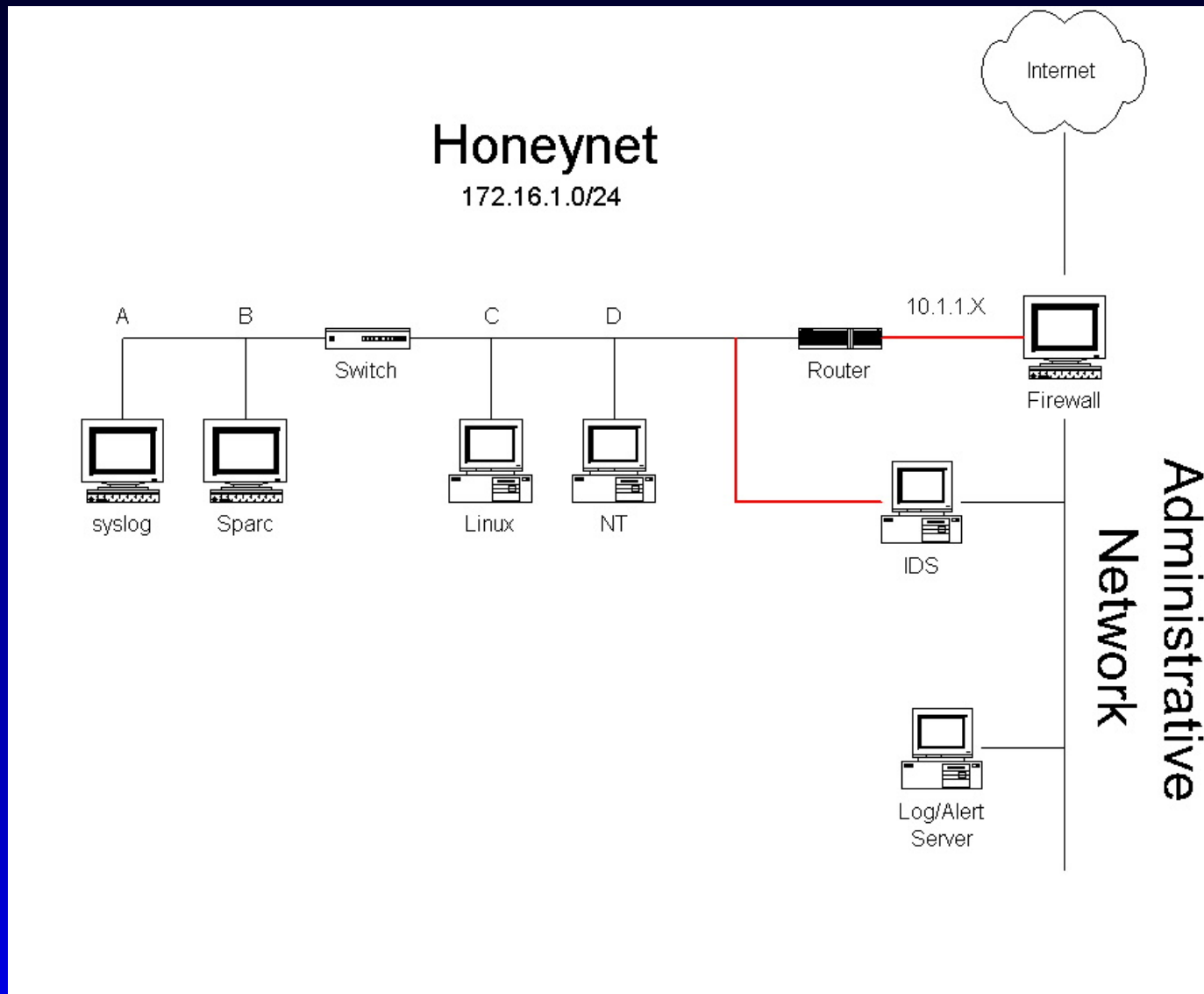


Implementierungsanforderungen

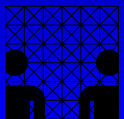
- Kontrolle:
 - Verhinderung von Angriffen auf Dritte
 - Automatisiert auf mehreren Ebenen
 - Angreifer darf nicht erkennen, dass er kontrolliert wird
 - Alarmgebung
 - manuelles Eingreifen möglich
- Datenerfassung:
 - Erfassung möglichst vieler Daten
 - Schutz gegen Modifizierung und Zerstörung



Honeynet-Architektur

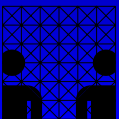


3. Analyse



Analysemittel

- Firewallalerts
 - bei Zugriff auf das System wird eine Warnmail versandt
- IDS(Intrusion Detection System)
 - Warnung und Erkennung bekannter Signaturen
 - Datenpakete werden abgefangen und nach Möglichkeit aufgeschlüsselt
- Systemlogs
 - Speichert alle Vorgänge, die auf dem Honeypot passieren
 - Systemlogs werden zur Sicherheit zusätzlich auf einem Logserver gesichert



Vorgehen des Angreifers

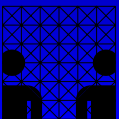
1. Auskundschaften

- Angreifer versucht Schwachstellen zu finden
- z.B. durch NetBios Scans oder DNS Probes

2. Angreifen

- Angreifer nutzt Exploit, um einzudringen
- Account mit root-Rechten wird erstellt
- Tools/rootkits werden per FTP runtergeladen
- Backdoor wird später ausgenutzt, um z.B. ein Denial of Service Tool zu installieren

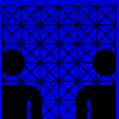
Durch Überwachung der Befehle kann der Zweck des Angriffs und der Erfahrungsgrad des Angreifers bestimmt werden.



Fortgeschrittene Analysetechniken

Passive Fingerprinting

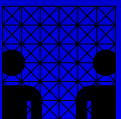
- versucht durch Analyse von Datenpaketen Informationen über den Angreifer zu erhalten
- liefert schlechtere Genauigkeit als active Fingerprinting
- kann kaum aufgespürt werden



p. Fingerprinting bei TCP

Wichtige Merkmale:

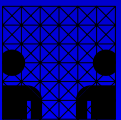
- Time to Live
- Window Size
- Don't Fragment Flag
- Type of Service



p. Fingerprinting bei ICMP

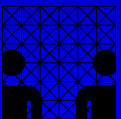
Beispiel Echo request:

- Datagramgröße
- Inhalt
- Timestamp
- ID



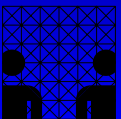
Forensics

- Schritt für Schritt Analyse des angegriffenen Systems
- Hauptaugenmerk auf dem Nachvollziehen der Schritte
- „Forensische Analyse ist eine Kunst, keine Wissenschaft.“

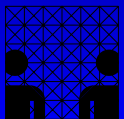


Vorgangsbeispiel

- Kopieren der kompromittierten Festplatten
- Mounten der Festplatten auf einem sicheren System
- „Graverobber“ filtert wichtige Informationen(logfiles, conf files etc.)
- gelöschte Inodes und Dateien wieder herstellen
- MAC(Modify/Access/Change) Times
- Inodes

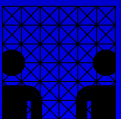


4. Der Feind



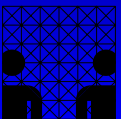
Die Bedrohung

- größtenteils „Skript-kiddies“
- Tools sind sehr verbreitet
- Diese verursachen häufige Zugriffe
- Ein unbekanntes System ist kein sicheres System



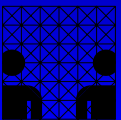
Taktiken

- Scannen der Rechner auf mögliche Schwachstellen
- Autorooter automatisieren diese Vorgänge
- Nach dem Scannen entscheidet Angreifer oder Tool, ob das System verletzlich ist
- Backdoors und Trojaner werden nach dem Eindringen installiert
- rootkits automatisieren Vorgänge auf kompromittierten Systemen



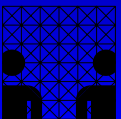
Tools

- sind schwierig zu programmieren
- IP Databasetool
- Exploittool
- Rootkits



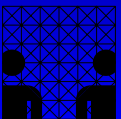
Motive

- DOS(Denial of Service)/DDOS(distributed Denial of Service)
- Identität verwischen
- IRC Bots und Server
- Anerkennung in der Community
- Daten im Internet verteilen



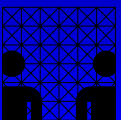
Trends

- neue Scantechniken
- Benutzung von Verschlüsselung
- bessere Rootkits
- Würmer
- Rechner werden nicht mehr gescannt, sondern direkt angegriffen

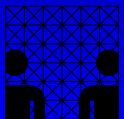


Würmer

- neuer „Typ“ von Angreifer
- voll automatisiert
- kombiniert die Fähigkeiten mehrerer Tools
- Repliziert sich selbst

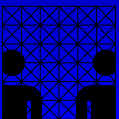


Resume



Literatur

- [1] David C. Plummer. An ethernet address resolution protocol, November 1982.
- [2] The HoneyNet Project. *Know your Enemy*. Addison-Wesley, 2001.
- [3] Lance Spitzner. *Honeypots — Tracking Hackers*. Addison-Wesley, 2002.



**Vielen Dank
für Ihre Aufmerksamkeit**

