# Syslog: Praxis und kommende Standards

Dipl.-Inform. Klaus Möller
DFN-CERT GmbH
Februar 2003

## Agenda

- Motivation Logging
- BSD Syslog
- Probleme
- Reliable Syslog
- Syslog Sign
- Vergleich
- Ausblick

# Logs

- Kleine Meldungen, ähnlich Logbuch Einträgen

```
Feb 04 14:10:59 shadow su:(to root) moeller on
  /dev/pts/2
```

- Von
  - Kernel, Systemprozesse, Anwendungen
  - Netzwerkgeräte: Drucker, Switches, Router, ...
- Für
  - Fehlersuche
  - Frühwarnung bei Angriffen
  - Spurensuche nach Angriffen
- In
  - **`/var/log/, /var/adm/, ...`**



# Motivation 1

- Angreifer wissen um Systemlogs
- Ziel: Angriffsspuren verwischen, d. h. alle verräterischen Meldungen ändern/löschen
  - IP-Adressen, Hostnamen
  - UserIDs
  - Terminalnamen
  - . . .
- Scripte und Programme dafür seit langem Teil von → Rootkits
  - **`http://www.packetstormsecurity.com/UNIX/penetration/log-wipers/`**

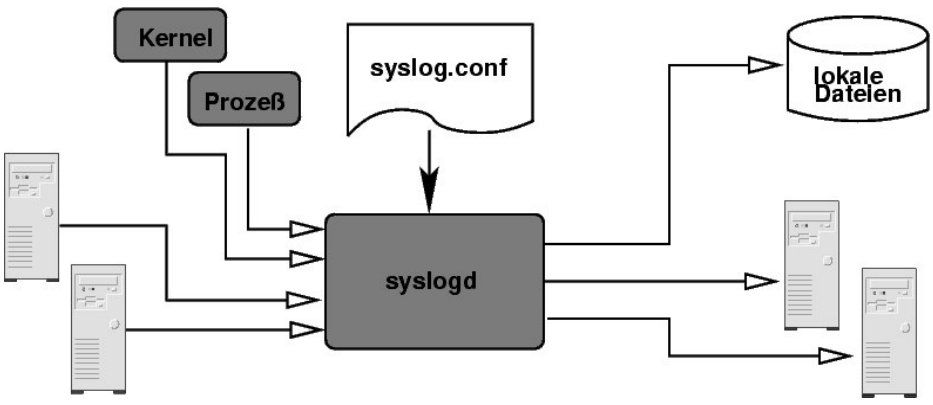
# Motivation 2

- Wenn Logs nur auf erzeugendem Host gespeichert
  - Auswertung mühselig (→ Windows Eventlog)
  - Keine Warnung zum Zeitpunkt des Angriffs
- Einziges Beweismittel nach erfolgreichem Angriff verloren
- Notlösungen
  - Logs direkt auf Drucker - Papierstau, Tinte alle
  - Logs auf CD brennen – Kompliziert, CD Wechsel
- Besser: Netzwerkfähiges Log-System



# Syslog

- BSD Syslog
  - Erste Version von Eric Allman für BSD Unix
  - 2002: Formeller Standard, RFC 3164
  - Netzwerkprotokoll, API und Implementierung



# BSD Syslog

- Steuerung über **`/etc/syslog.conf`**
  - Auswahl über Facility und Priority
  - Facility: Auslösendes Subsystem
    - **`KERN (0) – LOCAL7 (184)`**
  - Priority: Dringlichkeit der Nachricht
    - **`EMERGENCY (0) – DEBUG (7)`**
  - Speichern
    - **`mail.err;kern.info  /var/log/messages`**
  - Weiterleitung
    - **`*.alert  @193.174.13.2`**



# BSD Syslog Nachrichten

- Nachrichten werden als UDP Pakete versendet
- Eine Nachricht pro Paket, max. 1023 Bytes lang
- Klartext, nur druckbare ASCII Zeichen


- **`PRI = Facility * 8 + Priority`**
- Timestamp, Hostname und Tag sind optional
- Aber wichtig zur Korrelation von Nachrichten

# Wirkung eines Loghost

- Auszug aus t0rn v8 Rootkit

```
SYSLOGCONF="/etc/syslog.conf"

echo -n "          checking for remote logging... "

REMOTE=`grep -v "^#" "$SYSLOGCONF" | grep -v "^$" | grep
"@" | cut -d '@' -f 2`

if [ ! -z "$REMOTE" ]; then
    echo '          REMOTE LOGGING DETECTED '
    echo 'I hope you can get to these other computer(s): '
    echo
    for host in $REMOTE; do
        echo -n "              "
        echo $host
    done
    echo
    echo '          cuz this box is LOGGING to it...'
```



# Syslog Probleme 1

- Zuverlässigkeit
  - UDP → Nachrichtenverlust in Tests bis 70% !
  - Nachrichtenankunft in anderer Reihenfolge
- Informationsgehalt
  - Timestamp optional, kein Jahr, keine Zeitzone
  - Hostname, Tag ebenfalls optional

  ⇒ Korrelation von Meldungen schwierig
  - 7-Bit ASCII unzureichend
  - 1023 Zeichen zu wenig für einige Meldungen

# Syslog Probleme 2

- Vertraulichkeit
  - Nachrichten im Klartext – Jeder kann mitlesen
- Integrität
  - Kein Schutz gegen (Ver)Fälschung
  - Kein Schutz gegen Replay
  - Überfluten trivial möglich (IP-Spoofing)
    - **`http://www.packetstormsecurity.com/UNIX/penetration/log-wipers/sysfog.c`**
- **Authentizität**
  - Hostname kein FQDN
  - IP-Adresse des UDP Paketes nicht aussagekräftig



# Teillösungen

- Gegen Überfluten
  - Port **`514/udp`** am Netzübergang sperren
  - Empfang von Syslog Nachrichten abschalten, wo nicht benötigt
    - **`syslogd -t`** (Solaris)
    - **`syslogd`** (Linux, kein **`-r`**)
  - Wo Empfang notwendig: Auf erwünschte Quellen mit Paketfiltern einschränken
- Zeit mit **`ntp`** synchronisieren
  - Auf Authentifikation achten
- Logs möglichst häufig sichern (mind. 1x täglich)

# Reliable Syslog

- Vollständig neues Protokoll (RFC 3195)
  - Weiter nachrichtenbasiert
- Keine neuen Informationen
  - Priority, Facility, Timestamp, usw. wie BSD Syslog
  - Stützt sich auf BEEP für Transport und Sicherheit
  - Neuer Port: `601/tcp`
- Nicht kompatibel zu BSD Syslog
- Zwei Modi
  - RAW: Einfach, unstrukturiert
  - COOKED: Strukturiert, erweiterbar



# BEEP

- Block Extensible Exchange Protocol
- Grundgerüst, Anwendungsprotokoll definiert durch
  - Kern (BEEP Core, RFC 3080)
  - Profil (XML DTD)
  - Transport Mapping (derzeit nur TCP)
- Drei Grundinteraktionen in BEEP
  - Message – (ein) Reply
  - Message – (ein) Error
  - Message – (ein oder mehrere) Answer
- Asynchroner Nachrichtenaustausch möglich
- Nachrichten beliebiger Größe und Struktur (MIME)

# RAW Mode

- Nachrichtenformat analog BSD Syslog
  - Optionale Attribute jetzt verbindlich

```
MSG 1 0 . 0 50
DFN-CERT Loghost. Leg los.
END

ANS 1 0 . 0 61 0
<38>Oct  1 15:39:41 procert sshd[29616]: Connection from
    193.174.13.16 port 35688

ANS 1 0 . 61 58 1
<38>Oct  1 15:39:41 procert sshd[29616]: Enabling
    compatibility mode for protocol 2.0

NUL 1 0 . 119 0
END
```



# Cooked Mode

- Nachrichten sind XML Strukturen
  - Erweiterbar durch Änderung des Profils
  - Zusätzliche Informationen (**pathID, IAM**)

```
MSG 1 0 . 2235 242
Content-type: application/beep+xml
<entry facility='4' severity='6'
  hostname='procert'
  deviceFQDN='procert.cert.dfn.de'
deviceIP='193.174.13.1'
  timestamp='Oct  1 15:39:41'
  tag='sshd[29616]'
  Connection from 193.174.13.16 port 35688
</entry>
END
```

# Zusatzinformationen

- *PathID* Attribut
  - Pfad, den Nachricht bereits zurückgelegt hat

```
<path fromFQDN='shadow.cert.dfn.de'
 fromIP='193.174.13.17' toFQDN='certain.cert.dfn.de'
 toIP='193.174.13.5' linkprops='ULRI' pathID='173'>
</path>
```

- *IAM* Element
  - Identifiziert Absender und Rolle des Abs.
    - Erzeuger (*Device*), Weiterleiter (*Relay*), Empfänger (*Collector*)

```
<profile
  uri='http://xml.resource.org/profiles/syslog/COOKED'>
  <![CDATA[ <iam fqdn='shadow.cert.dfn.de'
  ip='193.174.13.17' type='device'/> ]]>
</profile>
```



# Reliable Syslog

- Sicherheit durch BEEP Profile
- TLS (RFC 2246)
  - Integrität: Sicherung gegen Verfälschung, Replay
  - Vertraulichkeit
- SASL (RFC 2222)
  - Authentifikation des Absenders
- Zusammen mit pathID gute Abschätzung der Vertrauenswürdigkeit von Nachrichten
- Auf Transportebene !
- Kein Schutz bei Speicherung

# Implementierung

- SDSC Secure Syslog
  - Version 1.0.0 Release Candiate 6
    - **`http://www.sdsc.edu/security/`**
  - Benötigt BEEP Bibliotheken von Roadrunner
    - **`http://www.roadrunner.se/`**
  - Zusätzlich: OpenSSL, Glib 2, Pkg-config
- RAW, COOKED und Security Profile
- Implementierung **ist** abwärtskompatibel zu BSD Syslog



# Syslog Sign

- Erweiterung von RFC 3164
  - **`http://www.ietf.org/internet-drafts/draft-ietf-syslog-sign-08.txt`**
- Abwärtskompatibel zu BSD Syslog

  ⇒ Gleiches Transportprotokoll (UDP) und Port (514)

  ⇒ Gleiches Nachrichtenformat

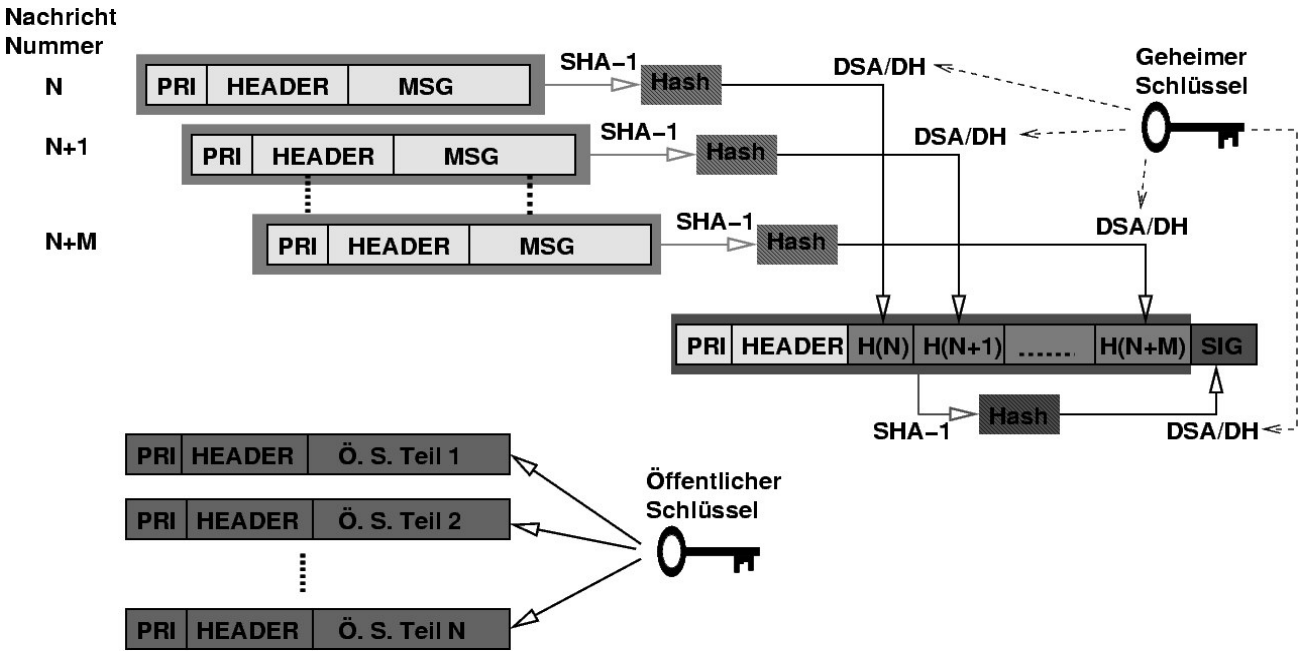
  Sicherheit durch digitale Signaturen
- Implementierung in Arbeit
  - Erweiterung des FreeBSD **`syslogd`**
  - **`http://eurobsdcon.org/papers/mietus_presentation.pdf`**

# Signierung 1

- Jede Nachricht wird signiert
  - Schutz gegen Verfälschung
- Signaturen implizit nummeriert
  - Schutz gegen Einfügen von Nachrichten und Replay
- Übertragung als „detached Signatures“ in normalen Syslog Nachrichten
  - Eine Nachricht überträgt Signaturen mehrerer vorangegangener Nachrichten
- Signiert wird mit beim Boot erzeugten Schlüsseln
  - Public-Key wird ebenfalls in Syslog Nachrichten übertragen



# Signierung 2



# Signaturprüfung

- Online
  - Bei Sortierung der Nachrichten mit Hashes ist effiziente Überprüfung möglich
- Offline
  - Gleicher Algorithmus
  - Manipulationen gespeicherter Nachrichten können bemerkt werden
    - Geheimer Schlüssel muß nicht aufbewahrt werden
  - Problem: PRI Feld muß zusammen mit Nachricht gespeichert werden
    - Neue Option (-v) beim Start von syslogd
- Prüfung der Schlüssel nicht Teil des Protokolls



# Vergleich

## Reliable Syslog

- Vertraulichkeit
- Zuverlässigkeit (TCP)
- Erweiterbar
- Aufwand für BEEP
- Nicht kompatibel zu BSD Syslog

## Syslog Sign

- Kompatibel zu BSD Syslog
- Schutz gespeicherter Nachrichten
- Schlüsselprüfung ungeklärt
- Prüfalgorithmus anfällig für Denial-of-Service
- Nachrichtenverlust (UDP)

# Ausblick

- Deutliche Verbesserungen der Sicherheit
- Dennoch Unzufriedenheit in Arbeitsgruppe
  - Evtl. noch ein drittes Protokoll ?
- Erste Implementierungen im Laufe des Jahres
- Weiterhin: Keine aussagekräftigen Logs von kompromittierten Systemen
  - Angreifer kann Nachrichten vor Erreichen des Log-Systems manipulieren oder abfangen



# Fragen

? ? ?