Security – Logging & Computer Forensics

Note Title 4/14/2011

Announcements

- Two articles you can use for "good faith" paper (which is due next Tuesday)
  - no extensions!
- Lab 5 is posted
  - reading due by Tuesday
  - checkpoint on Wed. the 20th
  - final version due Wed. the 27th
- Check finals schedule & notify me of any conflicts

Computer Forensics

The branch of forensics that specializes in recovering digital evidence from a sequence of unknown events, usually for eventual use in a court of law.

We've all seen CSI, but as more crimes "go digital", different expertise is needed.

FingerPrints are hard to fake, but what about digital evidence?

Digital DNA

- Username (typically in log files)
- Network Address
- CPU Seriel Numbers - on Pentum III's until ~2000
- Hardware / Software artifacts
- Software watermarks
  Ex: GUID in MS office (used to track Melissa Worm)
- Encryption keys

Tools

1. Disk imaging + hashing
2. Text or binary editors
   Ex: Unix strings
3. System Logs ← more today
4. Network Scanners
5. Software Scanners
6. Data recovery

Issues in Forensics

- Chain of custody

  First thing is to image & then never touch original

- Cryptographic hashes

  Periodically checked for tampering

A word of caution

Computer forensics is delicate, + without legal authorization, illegal!

(So stay on DETER or your own private machine!)

Even in legal investigations, care must be taken not to exceed the warrant.

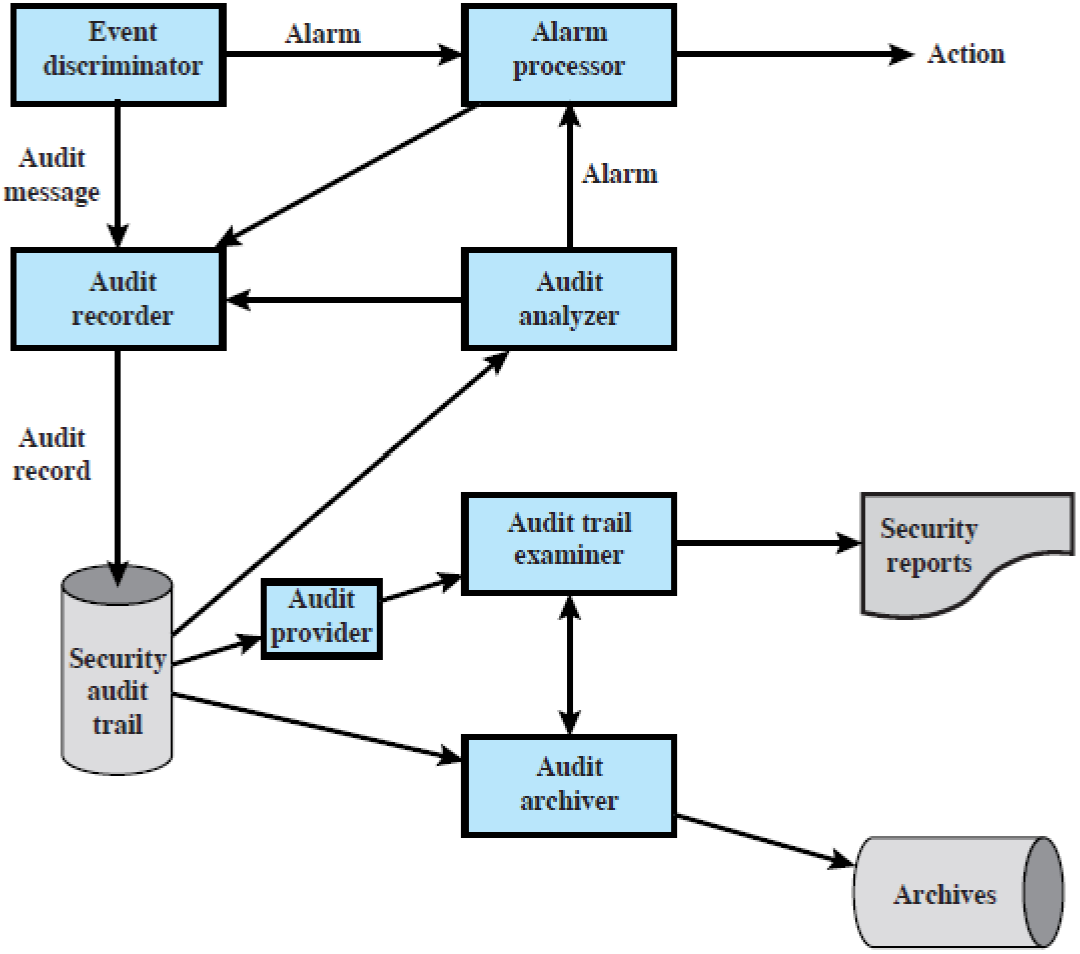
Main Element: Logging

2 issues:

① How to configure ahead of time

② How to use effectively after an issue has occured.

Auditing :


Event discriminator
Alarm
Alarm processor
Action
Audit message
Alarm
Audit recorder
Audit analyzer
Audit record
Audit trail examiner
Security reports
Audit provider
Security audit trail
Audit archiver
Archives

Auditing: What to collect

Issues:

- Amount of data:

  quantity versus efficiency/space

Possibilities:

- events relating to audit software
- events relating to security mechanisms
- intrusion detection & firewall events
- system management events
- system calls to OS
- remote access or logins
- access to some applications

Separation of audits:

1. System level

2. Application level

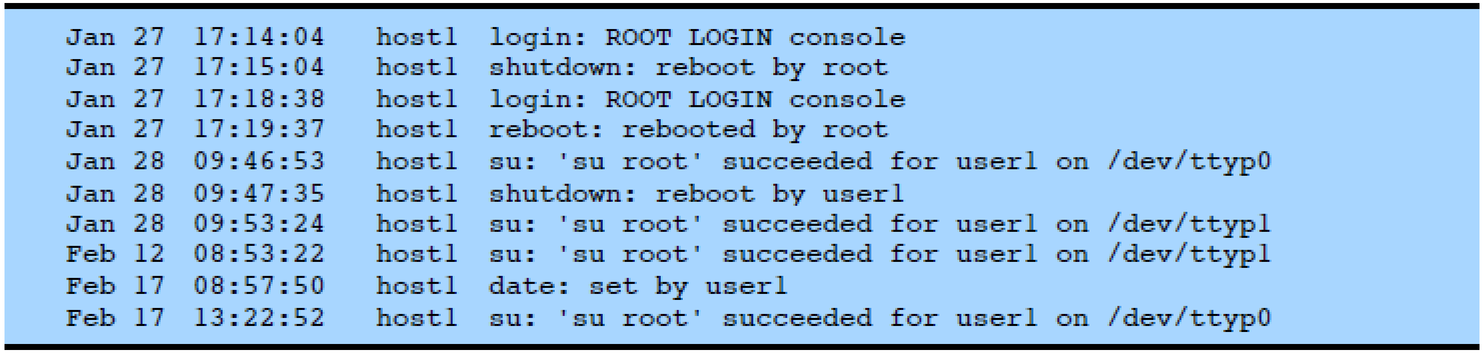
3. User level

① System Level

+ includes login attempts, devices used,
  + OS functions performed

- also useful for monitoring system performance

Ex:

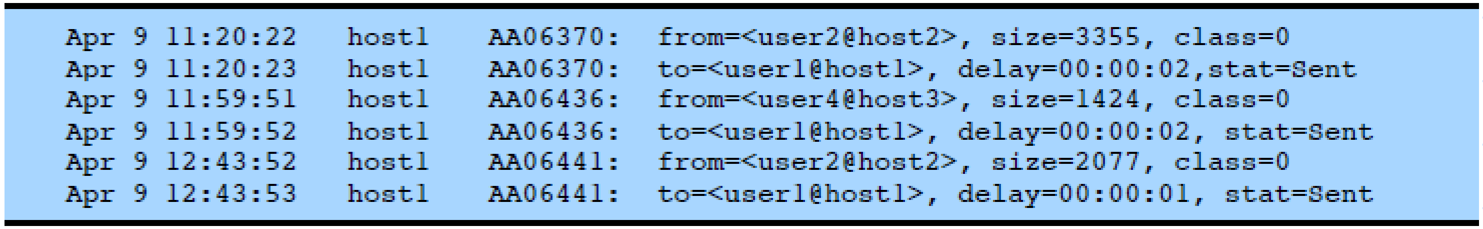


```
Jan 27  17:14:04   hostl  login: ROOT LOGIN console
Jan 27  17:15:04   hostl  shutdown: reboot by root
Jan 27  17:18:38   hostl  login: ROOT LOGIN console
Jan 27  17:19:37   hostl  reboot: rebooted by root
Jan 28  09:46:53   hostl  su: 'su root' succeeded for userl on /dev/ttyp0
Jan 28  09:47:35   hostl  shutdown: reboot by userl
Jan 28  09:53:24   hostl  su: 'su root' succeeded for userl on /dev/ttypl
Feb 12  08:53:22   hostl  su: 'su root' succeeded for userl on /dev/ttypl
Feb 17  08:57:50   hostl  date: set by userl
Feb 17  13:22:52   hostl  su: 'su root' succeeded for userl on /dev/ttyp0
```

② Application-Level

- detect security violations within an application, or flaws in application

Ex: mail delivery system:



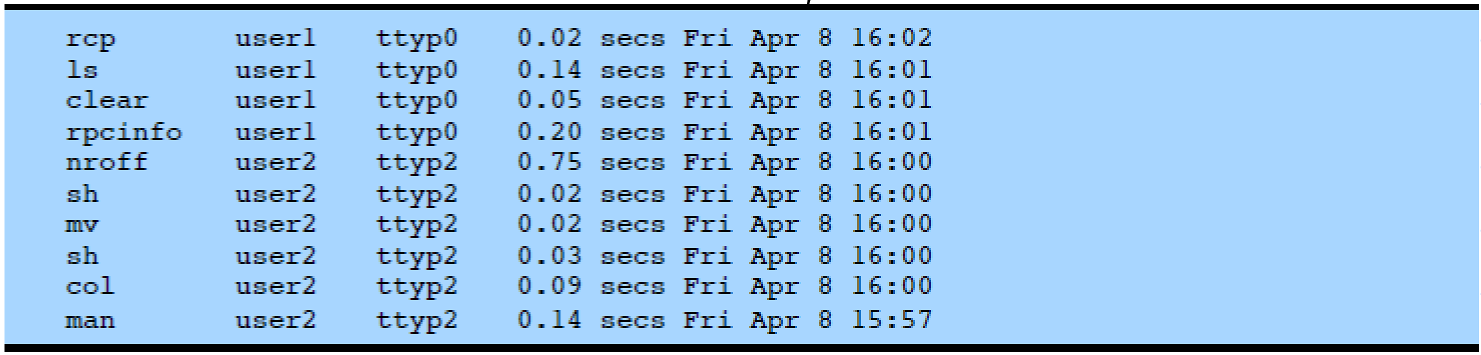
```
Apr 9 11:20:22   host1   AA06370:   from=<user2@host2>, size=3355, class=0
Apr 9 11:20:23   host1   AA06370:   to=<user1@host1>, delay=00:00:02,stat=Sent
Apr 9 11:59:51   host1   AA06436:   from=<user4@host3>, size=1424, class=0
Apr 9 11:59:52   host1   AA06436:   to=<user1@host1>, delay=00:00:02, stat=Sent
Apr 9 12:43:52   host1   AA06441:   from=<user2@host2>, size=2077, class=0
Apr 9 12:43:53   host1   AA06441:   to=<user1@host1>, delay=00:00:01, stat=Sent
```

vary greatly depending on app.

③ User-level

- holds users accountable
- can define "normal" behavior over time

Ex: Commands executed by users
(on UNIX system)



```
rcp       user1   ttyp0   0.02 secs Fri Apr 8 16:02
ls        user1   ttyp0   0.14 secs Fri Apr 8 16:01
clear     user1   ttyp0   0.05 secs Fri Apr 8 16:01
rpcinfo   user1   ttyp0   0.20 secs Fri Apr 8 16:01
nroff     user2   ttyp2   0.75 secs Fri Apr 8 16:00
sh        user2   ttyp2   0.02 secs Fri Apr 8 16:00
mv        user2   ttyp2   0.02 secs Fri Apr 8 16:00
sh        user2   ttyp2   0.03 secs Fri Apr 8 16:00
col       user2   ttyp2   0.09 secs Fri Apr 8 16:00
man       user2   ttyp2   0.14 secs Fri Apr 8 15:57
```

Physical Access

Any critical system will be kept in a secured location.

Why?

If you have physical access, you can break in!

So door access, modification of access priveledges, etc., is also relevant log information.

Protecting log data

Generally, 3 options

① Read / write on a host
- log files
- seperate server, encryption

② Write-once, read-many device
- CD-ROM
- Magnetic tapes

③ Write only device
- printer

Logging in Windows

Window Event Log:

Each event gets a numeric ID code, set of attributes (such as task, opcode, version, keywords), plus optional user supplied data

3 types of logs:

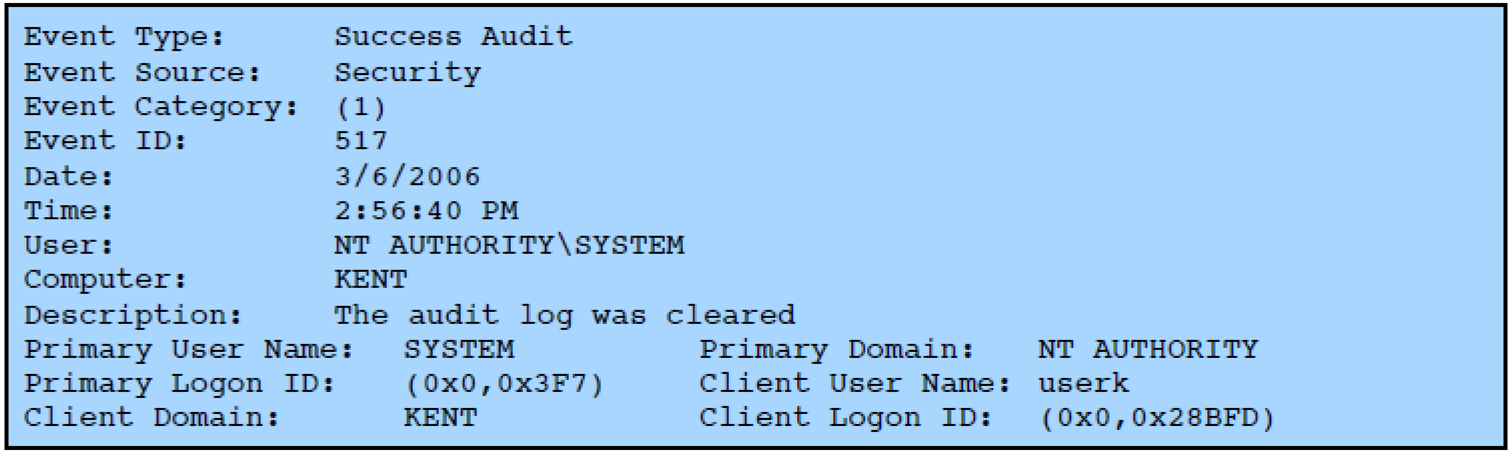
- system event log
- application event log
- security event log

Windows (cont)

Auditing can be enabled in 9 categories:

- Account logon events
- Account management
- Directory service access
- Logon events (local)
- Object access
- Policy changes
- Priveledge use
- Process tracking
- System events

Windows example



```
Event Type:       Success Audit
Event Source:     Security
Event Category:   (1)
Event ID:         517
Date:             3/6/2006
Time:             2:56:40 PM
User:             NT AUTHORITY\SYSTEM
Computer:         KENT
Description:      The audit log was cleared
Primary User Name:   SYSTEM          Primary Domain:    NT AUTHORITY
Primary Logon ID:    (0x0,0x3F7)     Client User Name:  userk
Client Domain:       KENT            Client Logon ID:   (0x0,0x28BFD)
```

UNIX logging

Syslog is the default found on all UNIX systems.

Elements:

- syslog(): API referenced by several standard utilities, & available to applications
- logger -command to add entries to system log
- /etc/syslog.conf
- syslogd

Not uniform across UNIX systems!

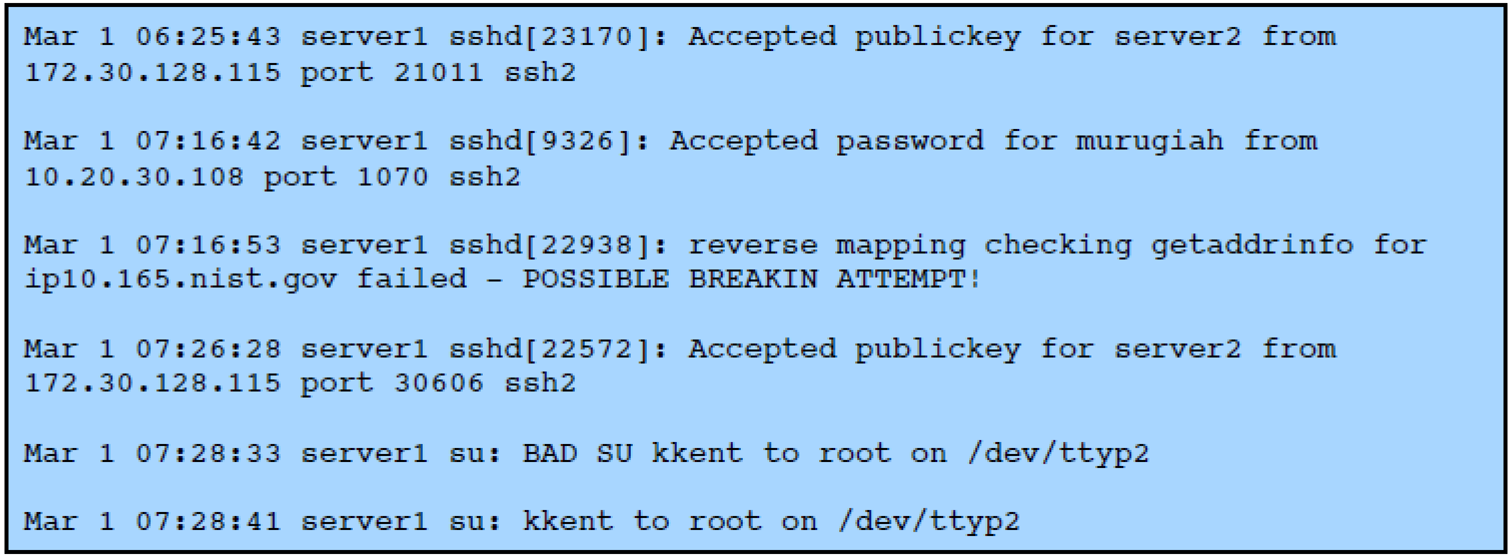
Basic services:

- capture relevent events
- store them
- transmit to central machine, a syslog server

Other functions:

- robust filtering: basic is only facility + priority, but adds host or program source or other filters
- log analysis: originally, no analysis
- event response
- log file encryption
- database storage
- rate limiting (to resist DDOS)

Unix example



```
Mar 1 06:25:43 server1 sshd[23170]: Accepted publickey for server2 from
172.30.128.115 port 21011 ssh2

Mar 1 07:16:42 server1 sshd[9326]: Accepted password for murugiah from
10.20.30.108 port 1070 ssh2

Mar 1 07:16:53 server1 sshd[22938]: reverse mapping checking getaddrinfo for
ip10.165.nist.gov failed - POSSIBLE BREAKIN ATTEMPT!

Mar 1 07:26:28 server1 sshd[22572]: Accepted publickey for server2 from
172.30.128.115 port 30606 ssh2

Mar 1 07:28:33 server1 su: BAD SU kkent to root on /dev/ttyp2

Mar 1 07:28:41 server1 su: kkent to root on /dev/ttyp2
```