

Logs for Incident Response

Anton Chuvakin, Ph.D., GCIA, GCIH, GCFA
Chief Logging Evangelist



Mitigating Risk. Automating Compliance.

Logs for Incident Investigations

A few thoughts to start us off ...

- **All attackers leave traces. Period!** 😊
- It is just that you *don't always know what and where*
- And *almost never know why...*
- **Logs** are the place to look, **first**

Goals

- Learn/refresh about **logs and logging**
- Refresh our knowledge of **incident response practices**
- Learn how **various logs are used at various stages** of incident response
- Learn about **log forensics**
- Learn how to **insider-proof logging**

Outline - I

- Incident Response (IR) Process
- Logs Overview
- Logs Usage at Various Stages of the Response Process
- How Log from Different Sources Help IR

Outline - II

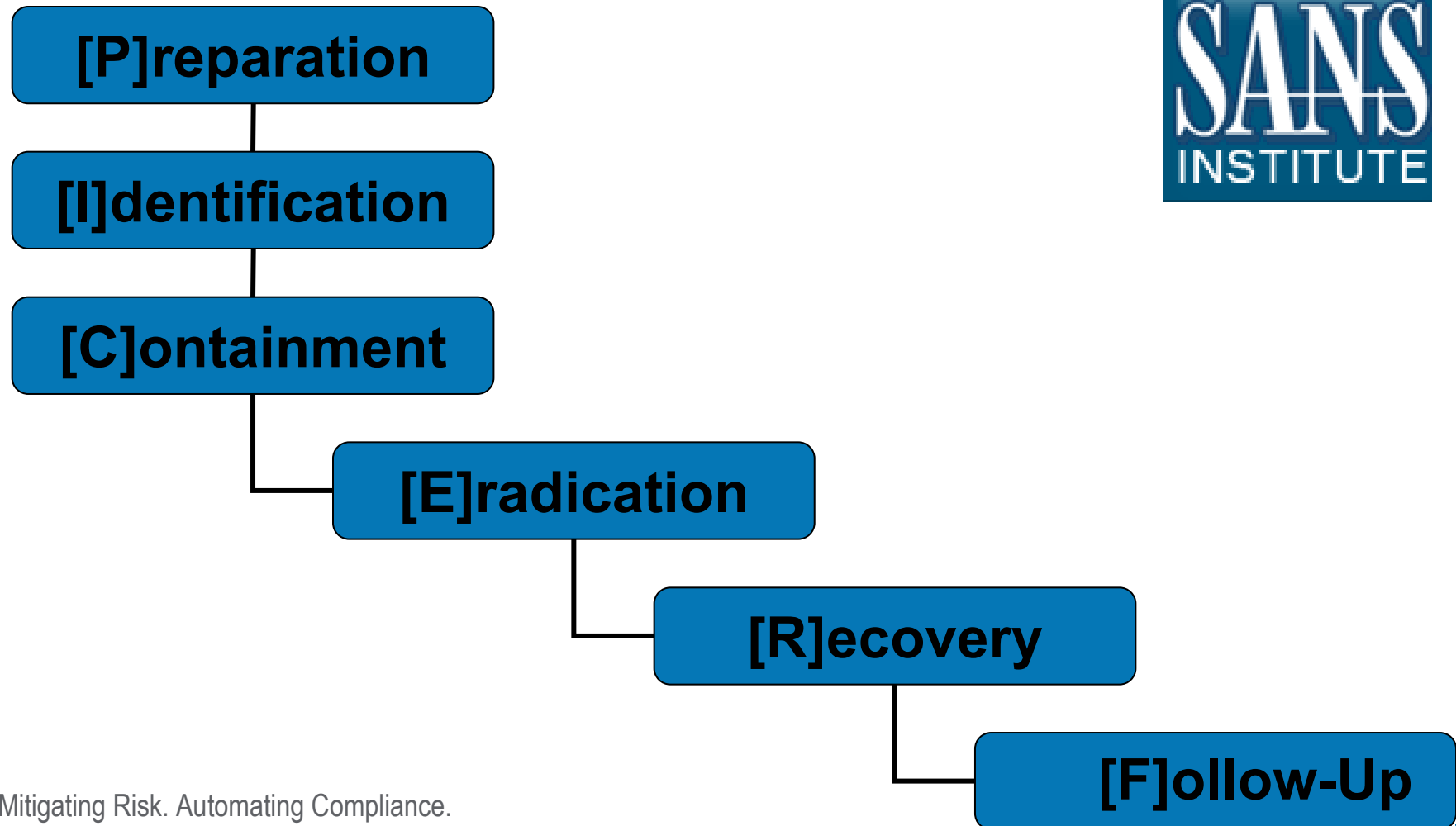
- Standards and Regulation Affecting Logs and Incident Response
- Incident Response vs Forensics
- Log Analysis and Incident Response Mistakes
- Bonus: Logs vs Insiders
- Bonus: Logs + Honeytokens Case Study

Incident Response Process

Incident Response Process

Incident Response Methodologies: SANS

- SANS Six-Step Process



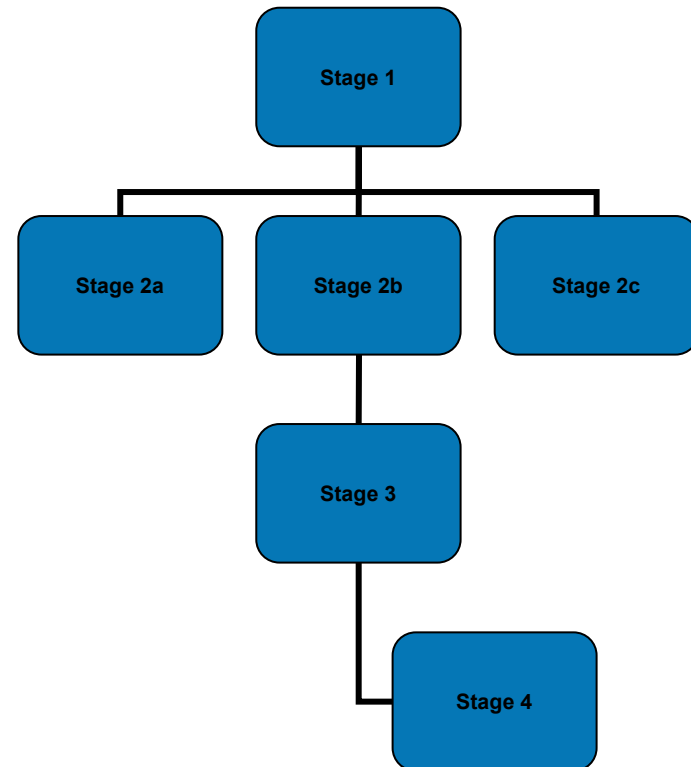
Incident Response Methodologies: NIST

- NIST Incident Response 800-61
 1. Preparation
 2. Detection and Analysis
 3. Containment , Eradication and Recovery
 4. Post-incident Activity



Why Have a Process?

- It **helps**...
 - Predictability
 - Efficiency
 - *Auditability*
 - Constant Improvement
- It **shrinks**...
 - Indecision
 - Uncertainty
 - Panic! ☹️



Example: Worm “Mitigation” in a Large Company...

... circa 2002 AD ☺

- Worm hits
- Panic + initial response in parallel (urgh! ☺)
- Mitigation + investigation at the *same* time
- Two walking steps forward and 10 running steps back...



From Incident Response to Logs

From Incident Response to Logs

Definitions

- **Log** = record related to whatever activities occurring on an information system, event record

...standard definitions are coming soon:
CEE standard by MITRE
(<http://cee.mitre.org>)



Terms and Definitions

- Logging
- Auditing
- Monitoring
- Event reporting
- Log analysis
- Alerting
- **Message** – some system indication that an event has transpired
- **Log** or **audit record** – recorded message related to the event
- **Log file** – collection of the above records
- **Alert** – a message usually sent to notify an operator
- **Device** – a source of security-relevant logs

Login? Logon? Log in?



<18> Dec 17 15:45:57 10.14.93.7 ns5xp: NetScreen device_id=ns5xp system-warning-00515: Admin User netscreen has **logged on** via Telnet from 10.14.98.55:39073 (2002-12-17 15:50:53)



<57> Dec 25 00:04:32:%SEC_LOGIN-5-LOGIN_SUCCESS:**Login Success** [user:yellowdog] [Source:10.4.2.11] [localport:23] at 20:55:40 UTC Fri Feb 28 2006



<122> Mar 4 09:23:15 localhost sshd[27577]: **Accepted password** for kyle from ::ffff:192.168.138.35 port 2895 ssh2



<13> Fri Mar 17 14:29:38 2006 680 Security SYSTEM User Success Audit ENTERPRISE **Account Logon**
Logon attempt by: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Logon account: POWERUSER Source Workstation: ENTERPRISE Error Code: 0xC000006A 4574

Log Data Overview

What logs?

- Audit logs
- Transaction logs
- Intrusion logs
- Connection logs
- System performance records
- User activity logs
- Various alerts and other messages

From Where?

- Firewalls/intrusion prevention
- Routers/switches
- Intrusion detection
- Servers, desktops, mainframes
- Business applications
- Databases
- Anti-virus
- VPNs



Devices that Log: An *Attempt* at a Comprehensive List

- Network gear: routers, switches,
- Security gear: firewall, IDS, VPN, IPS, etc
- Access control: RAS, AD, directory services
- Systems: OS (Unix, Windows, VMS, i5/OS400, etc)
- Applications: databases, email, web, client applications
- Misc: physical access, other non-IT technologies
- **Other:** just about everything with the CPU...

CONFIGURE LOGGING

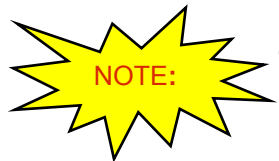
Configure Logging

Guidance

- Firewalls and network gear
 - Connections, access, firewall health
- Unix
 - Syslog, PS accounting, binary audit
- Windows
 - Windows event logs
- Mail servers
 - Email traffic, errors, access
- Web servers
 - Access, errors

Cisco IOS Boxes

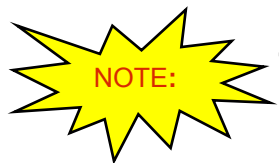
```
#config term  
#logging 10.1.1.1  
#write mem
```



There are more options available, but this gets the default log setting

Cisco Cats 😊

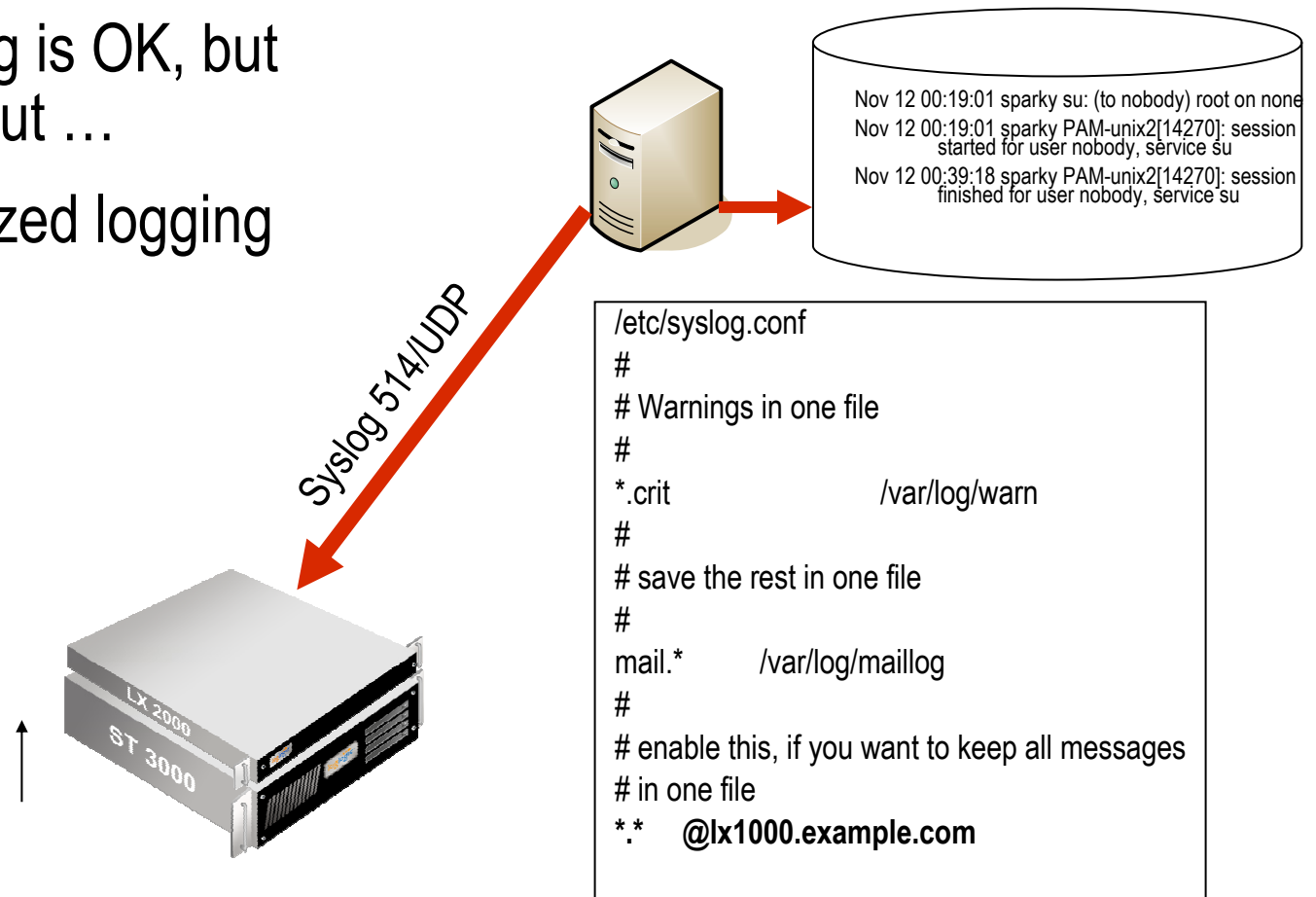
```
#set logging server 10.1.1.1  
#set logging server enable
```



There are more options available, but this gets the default log setting

Unix/Linux Syslog

- Default syslog is OK, but leaves a lot out ...
- Easy centralized logging



Unix/Linux: Other Logs

- Process accounting
 - Install *psacct* package
 - create a file e.g. */var/logs/audit*
 - start accounting e.g. *chkconfig psacct on* and */etc/init.d/psacct start*
- Detailed kernel audit
 - Solaris BSM, HP-UX, AIX Audit, SELinux
 - *complex!*

Windows Logging

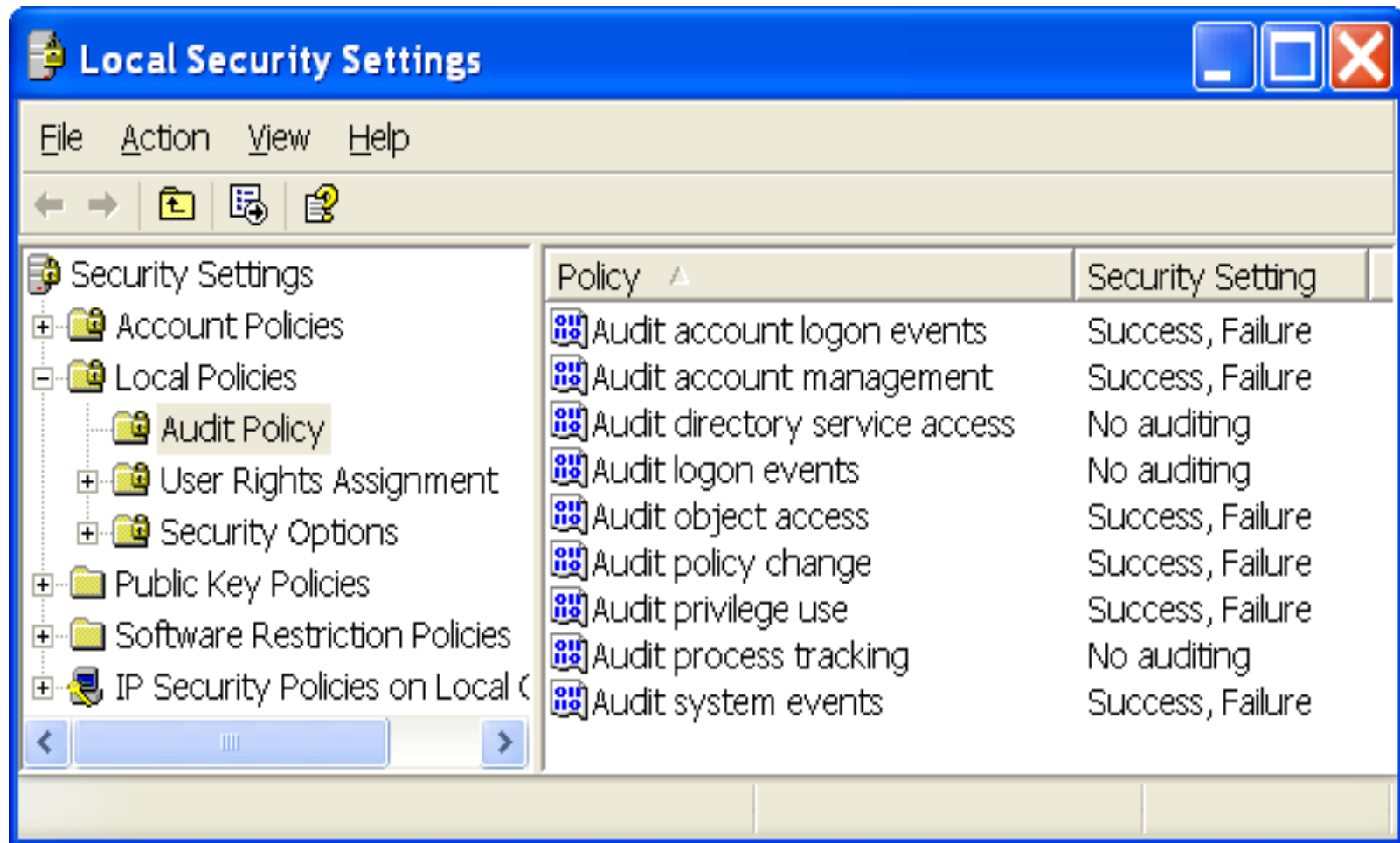
Main Windows event logs:

- **Application** log
- **Security** log
- **System** log

Domain controllers have two extra logs

- **File Replication** service log
- **DNS Server** logs

Windows Audit Policy



Web Servers

Web servers ship with sensible logging defaults

- **Apache**

- *access_log, error_log, SSL logs*

- **MS IIS**

- **W3C Extended files in**
c:\win\system32\logfiles\extXXXXXX.log

- **Errors in Windows Event logs**

Email Servers

- **Sendmail**
 - Syslog into */var/log/maillog*
- **MS Exchange**
 - Errors in Windows Event logs
 - Plus, file-based logs (SMTP, diagnostics, message tracking, subject, etc) – use *Exchange Manager*

Databases

○ Oracle

- Change *init.ora* file to have *audit_trail = db*
- Restart the database
- Run audit statements: *audit {statement|privilege} [by user] [by {session|access}] [whenever {successful|unsuccessful}];*

Oh, Horror! – Musings on Logging Defaults

- **Server OS** (Unix, Windows): authentication – yes, file access - no
- **Databases**: authentication – yes, changes – no, data access – no
- **Firewalls** - connection blocked – yes, connections allowed – sometimes, configuration changes - no

Natural Flow of Log Management: How People Enable Logs

1. Firewalls, network gear
2. Other network security gear
3. Servers (Unix, then Windows)
4. Other server applications (web, mail)
5. Databases
6. Applications
7. Desktops

LOG ANALYSIS

Log Analysis

Log Analysis: Why

- Situational **awareness** and new threat **discovery**
 - Who is doing what on a server
- Getting more value out of the network and security **infrastructure**
 - Firewall logs for ID
- **Measuring** security (metrics, trends, etc)
 - Top users by bandwidth from firewall logs
- **Compliance** and regulations (oh, my!)
 - Report on access to credit card data in a database
- **Incident** response (last, but not least!)

Log Analysis: Why NOT

- “*Real hackers don’t get logged!*” 😊
- Why bother? No, really ...
- Too much data (>10 GB per day)
- Too hard to do
- Is this device lying to me? 😊
- No tools “that do it for you”
 - Or: tools too expensive

Log Analysis Basics: Summary

1. **Manual**
2. **Filtering**
3. **Summarization and reports**
4. **Simple visualization**
5. **Log searching**
6. **Correlation**
7. **Log Data mining**

Log Analysis Basics: Manual

- **Manual log review**
 - Just fire your trusty ‘tail’, ‘more’, “notepad”, ‘vi’, Event Viewer, etc and get to it! 😊
- **Pros:**
 - Easy, no tools required (neither build nor buy)
- **Cons:**
 - Try it with 10GB log file one day 😊
 - Boring as Hell! 😊

Log Analysis Basics: Filtering

○ Log Filtering

- Just show me the **bad** stuff; here is the list (positive)
- Just ignore the **good** stuff; here is the list (negative or “Artificial Ignorance”)

○ Pros:

- Easy result interpretation: see->act
- Many tools or write your own

○ Cons:

- Patterns beyond single messages?
- Neither good nor bad, but interesting?

Log Analysis Basics: Summary

- **Summarization and reports**
 - Top X Users, Connections by IP,
- **Pros:**
 - Dramatically reduces the size of data
 - Suitable for high-level reporting
- **Cons:**
 - Loss of information by summarizing
 - Which report to pick for a task?

Log Analysis Basics: Visualization

- **Visualization**, from simple to 4D
 - *A pie chart* worth a thousand words?
- **Pro**
 - You just look at it – and know what it means and what to do
- **Con**
 - You just look at it – and hmmm.....

Log Analysis Basics: Search

○ Search

- User specifies a time period, a log source or all, and an expression; gets back logs that match (regex vs Boolean)

○ Pro

- Easy to understand
- Quick to do

○ Con

- What do you search for?
- A LOT of data back, sometimes

Log Analysis Basics: Correlation

○ Correlation

- Rule-based and other “correlation” and “Correlation” algorithms

○ Pro

- Highly automated

○ Con

- Needs rules written by experts
- Needs tuning for each site

Log Analysis Basics: Log Data Mining

- **Log mining**
 - Algorithms that extract meaning from raw data
- **Pro**
 - Promises fully-automated analysis
- **Con**
 - Still research-grade technology

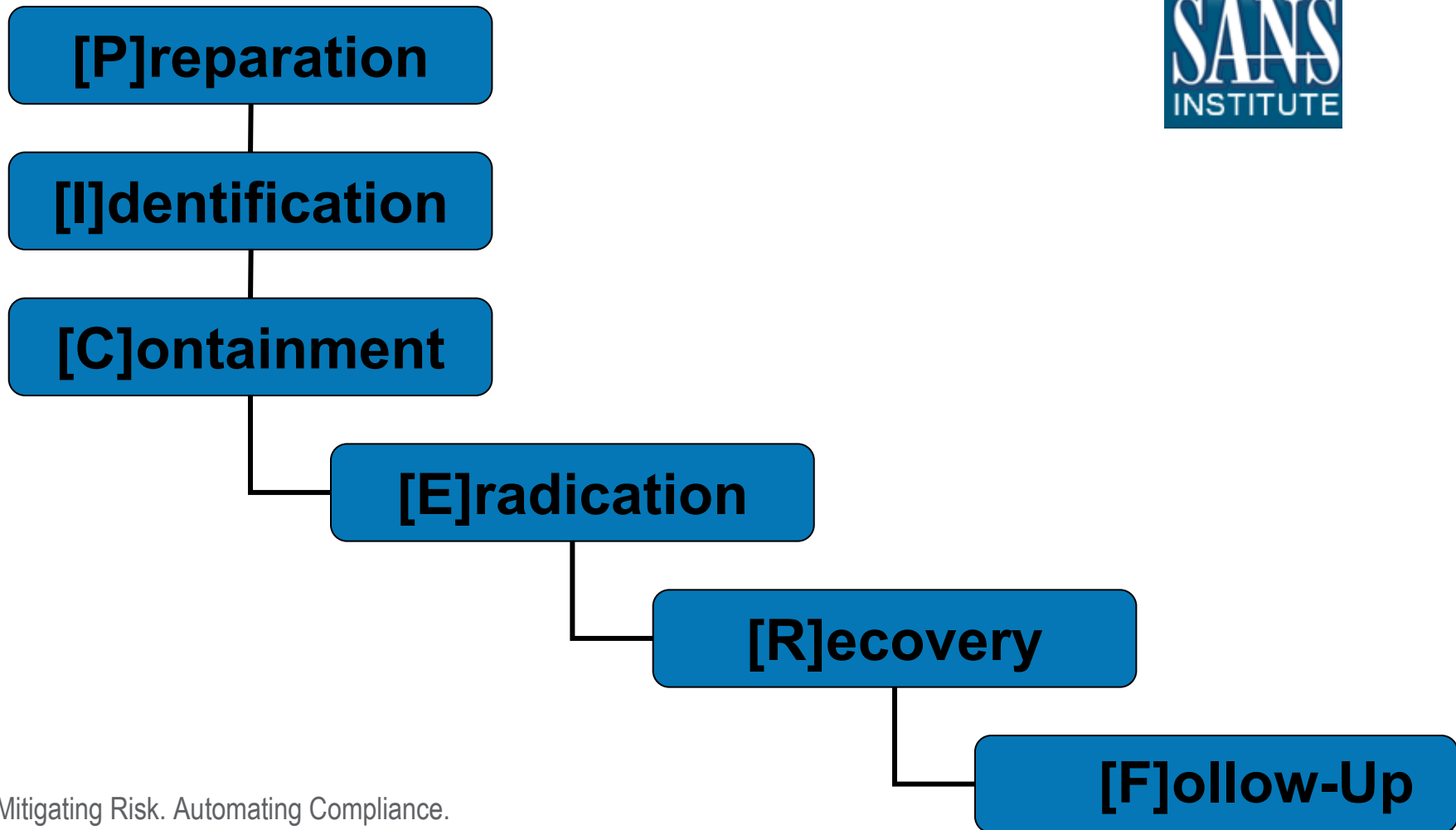
Select Log Analysis Tools

- **Log collection**
 - Syslog-ng, kiwi, ntsyslog, LASSO, DAD, Apache2syslog, etc
- **Secure centralization**
 - Stunnel, ssh, free IPsec VPNs
- **Pre-processing**
 - LogPP, MS LogParser (Windows -> text)
- **Storage**
 - MySQL or design your own
- **Analysis** – ooh, a tough one! 😊
 - SEC for rule-based correlation
 - SLCT and loghound for simple clustering
 - OSSEC, OSSIM, Prelude for [some] intelligence
 - Swatch, logwatch, logsend, other *match-n-bug* 😊 scripts

Back to Incident Response

Back to Incident Response: How Logs Help

Reminder: SANS Incident Response Model



Logs at Various Stage of Incident Response

- **Preparation:** verify controls, collect normal usage data, baseline, etc
- **Identification:** detect an incident, confirm incident, etc
- **Containment:** scope the damage, learn what else is “lost”, what else the attacker visited/tried, etc
- **Eradication:** preserving logs for the future, etc
- **Recovery:** confirming the restoration, etc
- **Follow-Up:** logs for “peaceful” purposes (training, etc) as well as preventing the recurrence

Using Logs at Preparation Stage

- Verify Controls
- Ongoing Monitoring
- Change Management Support
- “If you know the cards, you’d live on an island” 😊
- In general, verifying that you have control over your environment

1: P

Example 1 Logging Infrastructure for Optimum Response

- Monitoring infrastructure based on NSM philosophy: *netflow + packet content + logs (NIDS, etc)*
- Pre- and post-incident monitoring
- Useful even if *deployed after* the incident, but most useful if *deployed prior* to it

Using Logs at Identification Stage

- Detect Intrusion, Infections and Attacks
- Observe Attack Attempts, Recon and Suspicious Activity
- Perform Trend Analysis and Baselining for Anomaly Detection
- Mine the Logs for Hidden Patterns, Indicating Incidents in the Making...
- “What is Out There?”

2:1

Example 2 FTP Hack Case

- Server stops
- Found 'rm-ed' by the attacker
- What logs do we have?
- Forensics on an image to undelete logs
- Client FTP logs reveals...
- Firewall confirms!

Sidetrack: What if Not Prepared – Logging Defaults

- **Unix (typical):** system messages, login/logout, failures
- **Windows:** system messages, login/logout, failures
- **Web servers:** access (some details), errors
- **Databases:** errors, restarts, NO access or changes
- **Firewalls:** varies (denied, NO allowed)
- **Proxies:** access, caching
- **VPNs:** connections, login/logouts, errors
- **NIDS/NIPS:** alerts, failures

Sidetrack: What if Not Prepared – Local Retention

- **Unix (typical):** varies – weeks to days
- **Windows:** days to hours (!)
- **Web servers:** varies – weeks to days
- **Databases:** retained
- **Firewalls:** no data (or days to hours)
- **Proxies:** no data
- **VPNs:** no data
- **NIDS/NIPS:** no data (or weeks to days)

Using Logs at Containment Stage

- Assess Impact of the Infection, Compromise, Intrusion, etc
- Correlate Logs to Know What You Can [Still] Trust
- Verify that Containment Measures Are Working
- “What Else is Hit?”

3 : C

Example 3 But Did It Spread?

- “A classic”: regular desktop starts scanning internally
- Cut from the network soon after: an incident is declared
- An impressive array of malware is discovered; AV is dead
- Problem solved? Did it infect anybody else?!
- Logs from firewalls and flow to the rescue...

Using Logs at Eradication Stage

- Preserving the Log Evidence from Previous Stages
 - Especially if court action is likely or possible (see **Forensics**)
- Confirming that Backups are Safe
 - Using Logs, How Else? 😊
- “Is it Gone?”

4: E

Example 4 Logs for [Possible] Litigation

- Deliberations on the log retention (and destruction!) policy: IDS, VPN, firewalls, servers – oh, my!
- Decided: IDS – longest; server – next; firewalls, VPN – shortest
- Case: financial information leaked to the media
- Investigation points to a specific user
- Did he do it?!!
- Well, the answer *died* with 6-mo old VPN logs...

Using Logs at Recovery Stage

- Increased Post-Incident Monitoring
- Watch for Recurrence
- Watch for Related Incidents Elsewhere
- “Better Safe than Sorry”

5: R

Example 5 When They Come Back...

- Password guessing hack: non-root account password guessed
- IRC bot, scanning, phishing site setup, etc
- Password changed; attacker files cleaned
- More guessing attempts across the network– are those the same folks?
- *Will they succeed again?*

Using Logs at Follow-Up Stage

- Train Analysts, Responders and Administrators
- Create Management Reports
- Verify and Audit Newly Implemented Controls
- “We know we are OK”

6: F

Example 6 Logs for Responder Training

<http://www.honeynet.org/scans/scan34/>

a. honeynet IPs sanitized to: 11.11.11.*
b. Our DNS server IPs sanitized to: 22.22.22.* and 23.23.23.*
c. Some other sensitive IPs are sanitized to: 10.22.*.*

Download the Images
[SotM34-anton.tar.gz](#)
b23755326714e39cac91ca881f1ca668 SotM34-anton.tar.gz [MD5]
5e006e9503801dde2d57e44281d784c516601c35 SotM34-anton.tar.gz [SHA1]

The evidence includes:

- ◆ Apache logs
- ◆ Linux syslogs
- ◆ Snort NIDS logs
- ◆ iptables firewall logs

Questions

1. What are the significant events that happened on the honeypot in the time period covered by the logs? Show how you analyzed the data to paint the picture of those events.
2. Was the system compromised? How do you know? If yes, how many times and by how many attackers? What would you consider the most compelling evidence of the compromise available if you find that the system was indeed compromised?
3. If this were the evidence from a production system, how would you learn that the machine was compromised, given the data available? For this question, assume you do not have the honeynet-specific data streams, such as sebek2 or bash logger, just like in this challenge.
4. What else was going on at the system at the same time? What times of "Internet noise" can you categorize, given the data? Is there anything out of the ordinary with the noise levels? What attack and probe types observed actually had a chance of affecting the target?
5. Do you think that the time was synchronized between the various monitoring systems (where Snort and iptables logs were collected) and a victim system (where syslog and Apache logs were collected)?

Bonus Question:

- ◆ Describe the procedures and tools of that you used to analyzed all the distinct log sources together.

The Results:

Anton Chuvakin has provided an [official writeup here](#).

Sidetrack: Incident Record Keeping and Log Retention

- Retention policy for routine and incident logs
- #1: Human action logs – the longest!
 - Logs *created* during incident response
- Before planning any log retention policy changes – define incident and routine log retention
- Specifically ...

LOG RETENTION – A TRIVIAL MATTER?

Log Retention

What is Log Retention?

Q: When is “log storage” considered “log retention”?

A: Log retention =

Log storage +

Accessibility +

Log destruction

What is NOT Retention?

- A database that stores a few fields from each log
- A tape closet with log data tapes that were never verified
- A syslog server that just spools logs into files

Sidetrack: Why Destroy Log Data?

- **Log destruction?** You've got to be kidding ... 😊
- Why you need to destroy logs ... sometimes?
- How to destroy them?

Retention Time Question

- I have the answer! ☺ No, not really.
- Regulations?
 - Unambiguous: PCI – keep'em for **1 year**
 - 1 yr + 1 mo is also common (and so it 39 mos)
- Tiered retention strategy
 - **Online**
 - **Nearline**
 - **Offline/tape**

Example: Retention Strategy

Type + network + storage tier

- IDS + DMZ + online = 90 days
- Firewall + DMZ + online = 30 days
- Servers + internal + online = 90 days
- ALL + DMZ + archive = 3 years
- Critical + internal + archive = 5 years
- OTHER + internal + archive = 1 year

Retention Strategy HOWTO

1. Assess applicable compliance requirements
2. Look at risk posture
3. Look at various log source and their log volumes
4. Review available storage options
5. Decide on tiers

Log Storage Options

1. RDBMS
 - Oracle, MySQL, etc
2. Flat files
 - “Files+”: Compressed, indexed, etc
3. Hybrid
 - Combine #1 and #2
4. Proprietary datastore
 - Build from scratch to store logs

What Makes It “Accessible?”

Why store logs? Duh, so you can get to them later!

Use case for logs	Time frame of logs needed	Response time to get logs
Incident response	Weeks to months	Seconds to minutes
Audit	Months to years	Minutes to hours
Regulatory	Up to years	Any
E-Discovery	Up to years	Hours to days

Sidetrack: Log Sizing

Vendors	Product	Message Size (Bytes)	Typical Rate (Msgs/sec)
Cisco	PIX 515	150	100
Cisco	All Routers	150	< 10
Cisco	VPN 3000	150	50
Check Point	FW1 & VPN1	250	250
Juniper NetScreen	520	200	500
Windows	2003 Servers	1000	< 10
IBM	AIX	250	< 10
Red Hat	Linux	250	< 10
BlueCoat	Proxy	500	100

Mitigating Risk. Automating Compliance.

Example: How to Deal with A Trillion Log Messages?

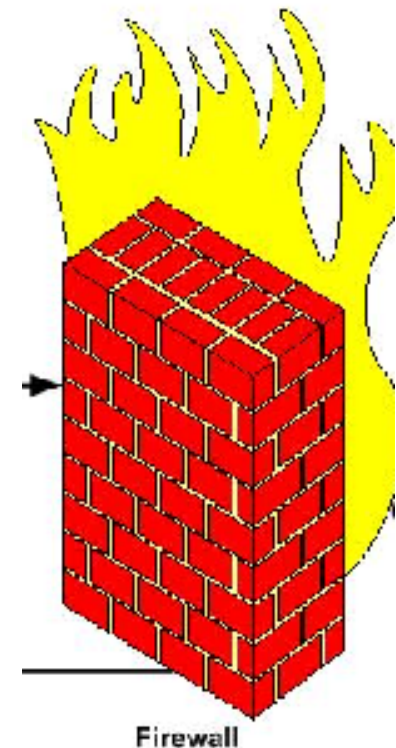
- How to manage a trillion (~1000 billions) of log messages?
- Hundreds of terabytes (1/2 of a petabyte ...) of data
- **Which tool to pick?**
- “Sorry, buddy, you are writing your own code here!”

So, What Logs are Useful for Incident Response?

- **Security Logs vs “Non-Security” Logs**
 - Witness confusion in the NIST 800-92 Guide on log management
- **Let’s quickly go through various logs and see how they help (and helped in specific cases!)**
 - Looking at some specifics in the process

Firewall Logs in Incident Response

- Proof of Connectivity
- Proof of NO Connectivity
- Scans
- Malware: Worms, Spyware
- Compromised Systems
- Misconfigured Systems
- Unauthorized Access and Access Attempts
- Spam (yes, even spam!)



Example: Firewall Logs in Place of Netflow

- Why Look at Firewall Logs During Incident Investigation?
- 1990-2001 – to see what *external* threats got **blocked** (in, failure)
- 2002-2006 – to see what *internal* system got **connected** (out, success)
- Thus, firewall logs is “poor man’s” *netflow*...

NIDS Logs in Incident Response

- Attack, Intrusion and Compromise Detection
- Malware Detection: Worms, Viruses, Spyware, etc
- Network Abuses and Policy Violations
- Unauthorized Access and Access Attempts
- Recon Activity
- [NIPS] Blocked Attacks



Example 8 Zero-Day Discovery with NIDS

- Can I discover undiscoverable?
- [Mostly] Signature NIDS is still king! But what about those pesky 0days?
- NIDS log pattern discovery to the rescue!
- Samba hack case: 3-4 of the same semi-suspicious signatures firing in the same time sequence => 0day in action

Server Logs in Incident Response

- Confirmed Access by an Intruder
- Service Crashes and Restarts
- Reboots
- Password, Trust and Other Account Changes
- System Configuration Changes
- *A World of Other Things* 😊



Example: “Irrelevant, You Say”

- Using disk failures for IDS 😊
- What is really there?
- Is this OUR server? Well ...
- “Detection by catastrophe”
- Is *CNN* your *IDS*?

Database Logs in Incident Response

- Database and Schema Modifications
- Data and Object Modifications
- User and Privileged User Access
- Failed User Access
- Failures, Crashes and Restarts



T.J. maxx[®]

you should go™ **LOOK AT LOGS!** 😊

Example: And What is NOT Stolen?

- *Supposedly*, all of ChoicePoint 40 mil cards were not stolen... They just **couldn't prove they weren't**.
- Database logs as a way of *non-intrusion detection* (or, rather, confirmation)

- On the other hand, TJMaxx 47 mil cards **were** stolen 😞

Example: Oracle Logging

- **Defaults:**
 - minimum system logging
 - minimum database server access
 - no data access logging
- **So, where is ...**
 - data access audit
 - schema and data change audit
 - configuration change audit

Proxy Logs in Incident Response

- Internet Access Patterns
- IP theft and/or disclosure
- Policy violations
- Malware: Spyware, Trojans, etc

Example: Proxy Logs vs Uploads

How to look for data uploads?

- **HTTP method** (logged as "cs-method" by BlueCoat) = **POST**
- For information uploads: content type (logged as "RS(content-type)" by BlueCoat) = **anything but "html/text"** (which is the type used for uploading web form contents) - especially try content types **"application/octet-stream"**, **"application/msword"**, **"application/powerpoint"**, **"application/vnd.ms-excel"**,
- **User-agent** set to **anything but the common ones** (i.e. not *Mozilla*, *iTunes*, *LiveUpdate*, etc) or even to "unknown" (also check agent names matching messaging applications)

Web Server Logs in Incident Response

○ Errors

- Errors – Why Are They There?
- Server Restarts (e.g. SIGHUPs)

○ Access

- Study “Weird” Response Codes: 205 Anybody?
- Study 20x on “Interesting” Documents
- Watch for Fun Methods: POST, even PUT, OPTIONS, etc

Client Logs in Incident Response

- FTP client: remote connections and file transfers
- IRC client logs
- Other client software: usually no logs, but usually leave other traces
 - E.g. web browser cache (OK, these are not logs)

Antivirus Logs in Incident Response

- Virus Detection and Clean-up (or lack thereof!)
- Failed and Successful Antivirus Signature Updates
- Other Protection Failures and Issues
- Antivirus Software Crashes and Terminations

Example: Sorry, I AM A Failure 😊

- System is compromised
- Analysis reveals major vendor AV was running
- AV logs shows that that there was a “detection-only” event with “left alone” as an action
- Malware is tracked through other logs as well, including those on and off the Owned box

Business Application Logs

- “A whole world .. with not map in sight” 😊
- **Lowest common denominator**
 - Logins/logouts
 - Critical errors
 - Starts/stops/restarts
 - “Important” operations
- Standards (CEE!) will help, just not now...

Example: Jumbled Mess of SAP Logging

|22:01:40|BTC| 7|000|DDIC | |LC2|Systemerror when
executing external command DB6_DATA_COLLECTOR on
gneisenau ()

|22:02:32|BTC| 7|000|DDIC | |R49|Communication error,
CPIC return code 020, SAP return code 456

|22:02:32|BTC| 7|000|DDIC | |R5A|> Conversation ID:
38910614

|22:02:32|BTC| 7|000|DDIC | |R64|> CPI-C function:
CMSEND(SAP)

|22:02:32|BTC| 7|000|DDIC | |LC2|Systemerror when
executing external command DB6_DATA_COLLECTOR on
gneisenau ()

Back to the Process

“Back to the Process II” 😊

Log Management Process for IR

- **Collect** the log data
- **Convert** to a common format
- **Reduce** in size, if possible
- **Transport** securely to a central location
- **Process** in real-time
- **Alert** on when needed
- **Store** securely
- **Report** on trends
- **Share** logs

Logging Process for IR Review

Log **everything**

Retain **most everything**

Analyze **enough**

Summarize and report on a **subset**

Monitor **some**

Act in real-time on a **few**

Value of Logging and Monitoring

Logging

- Audit
- Forensics
- Incident response
- *Compliance*

Monitoring

- Incident detection
- Loss prevention
- *Compliance*

Analysis and Mining

- Deeper insight
- Internal attacks
- Fault prediction

“Real-Time” Tasks

- **Malware** outbreaks
- Convincing and reliable **intrusion** evidence
- Serious **internal** network abuse
- **Loss of service** on critical assets

Daily Tasks

- Unauthorized configuration changes
- Disruption in other services
- Intrusion evidence
- Suspicious login failures
- Minor malware activity
- Activity summary

Weekly Tasks

- Review inside and perimeter log trends and activities
- Account creation/removal
- Other host and network device changes
- Less critical attack and probe summary

Monthly Tasks

- Review long-term network and perimeter trends
- Minor policy violation summary
- Incident team performance measurements
- Security technology performance measurements

Logs and Laws, Rules, Standards, Frameworks

Logs and Laws, Rules, Standards, Frameworks

Logs and Laws



Regulations Require LMI

- SOX
- GLBA
- FISMA

○ NIST 800-53

- Capture audit records
- Regularly review audit records for unusual activity and violations
- Automatically process audit records
- Protect audit information from unauthorized deletion
- Retain audit logs



Mandates Demand It

- PCI
- HIPAA
- EU

○ PCI : Requirement 10 and beyond

- Logging and user activities tracking are critical
- Automate and secure audit trails for event reconstruction
- Review logs daily
- Retain audit trail history for at least one year



Controls & SLAs Require it

- COBIT
- ISO
- ITIL

○ COBIT 4

- Provide audit trail for root-cause analysis
- Use logging to detect unusual or abnormal activities
- Regularly review access, privileges, changes
- Verify backup completion

○ ISO17799

- Maintain audit logs for system access and use, changes, faults, corrections, capacity demands
- Review the results of monitoring activities regularly and ensure the accuracy of logs

“Get fined, Go To Jail”

“Get fined, Get Sanctioned”

“Lose Customers, Reputation, Revenue or Job”

In Detail: NIST 800-92

“This publication seeks to assist organizations in understanding the need for sound computer security log management. It provides **practical, real-world guidance on developing, implementing, and maintaining effective log management practices** throughout an enterprise. “

NIST
National Institute of
Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-92

Guide to Computer Security Log Management

Recommendations of the National Institute
of Standards and Technology

More Laws! Privacy Laws: A Mess!

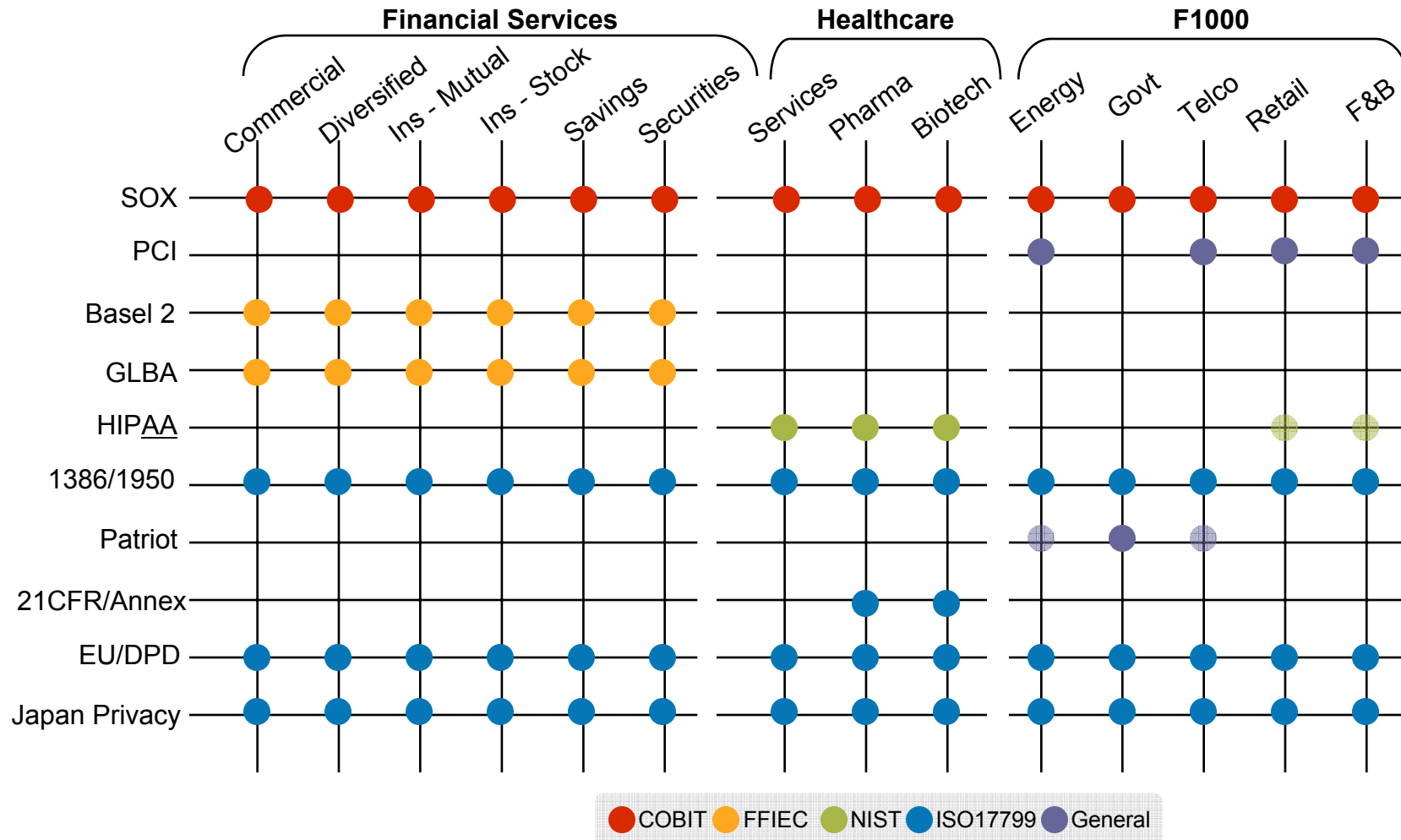
- What **MUST** be logged vs what **MUST NOT** be logged!
- Example: multinational telecom firm
 - Country 1: cannot log email headers
 - Country 2: must log all user information
 - Country 3: cannot retain for more than 6 months
 - Country 4: must retain for at least 3 months
 - Country 5: cannot retain for more than “needed”
- What’s Next?
- **Legislators, beware of the “HIPAA escape”** (aka “screw the law, we’ll pay the fine”) 😊

More Laws! Breach Laws Affected IR

- **Yesterday** CA 1386
- **Today** 35 US States
- **Tomorrow** the world

- Laws that control that consumer notification in case of a security breach
- **Incident response** is key:
 - **200,000** vs **40,000,000** notifications? Major \$\$\$ in play!

Compliance Drives ... Drives ... Drives ...



Mitigating Risk. Automating Compliance.

From Incident Response to Forensics

From Incident Response to Forensics

Logs and Forensics

- What Makes Your Incident Investigation a “Forensic” Investigation?
- Incident Response vs Forensics
- ... *and is the ‘vs’ really appropriate?*

So, What is “Log Forensics”

- **Log analysis** is trying to make sense of system and network logs
- “**Computer forensics** is application of the scientific method to digital media in order to establish factual information for judicial review.”

So....

- **Log Forensics** = trying to make sense of system and network logs + in order to establish factual information for judicial review
- Overall, a *bizarro* mix of **science, technology** and **law**

How Logs Help... Sometimes

If logs are *there*, we can *try* to

- ... figure out **who, where, what, when, how, etc**

but

- **Who** as a person or a system?
- Is **where** spoofed?
- **When?** In what time zone?
- **How?** More like 'how'd you think'...
- **What** happened or what got recorded?

Who?

Man vs machine: identity fight

- Just **who is 10.1.1.2**? Do you know *him*?
- Is **jsmith.example.com** a *who*?
- Is **JSMITH** at **\\JSMITH\EXAMPLE**?
- Is **JSMITH authenticated** by an RSA token at **\\JSMITH\EXAMPLE** and also logged to another system as “jsmith” a *who*?
- Is **JSMITH authenticated** by a fingerprint reader at **\\JSMITH\EXAMPLE** a *who*?

When?

Got timestamp? 😊 - challenges to log timing!

1. Completely **false timestamp** in logs (BTW, *today is Jan 1, 1969*)
2. It's **always 5PM somewhere**: which timezone are your logs in?
3. Are you in **drift? Your clock** might be (*those pesky seconds turn into minutes...*)
4. **Syslog forwarder** mysteries: “his” time vs “my” time
5. So, which one is right? Systems with **two timestamps!**
6. It got logged at 5:17AM. When did it **happen?** Log lag!

When? II

- **Sequence** of events is a critical fact! Miss the sequence and the whole “house of cards” goes ...
- But! **Absolute** time is also important! How to find it?
- More “*fun*” time issues to watch for:
 - Process leaves a log records when it *exits* (log lag)
 - Stuff missing from logs altogether (a rootkit?)
 - Logs with out of sequence time stamps (*WTF?* 😊)

Directory of C:\Windows\System32\LogFiles\W3SVC1

09/27/2003	07:00 PM	1,461,370	ex030927.log
09/29/2003	07:00 PM	629,006	ex030929.log
09/30/2003	07:00 PM	1,134,950	ex030930.log
		3 File(s)	3,331,151 bytes
		2 Dir(s)	7,842,717,696 bytes free

Where?

- The attack came from 10.1.1.2 which belongs to Guanjou Internet Alliance, Beijing China
- The stolen data then went to 10.2.2.1 which belongs to PakNet ISP in Karachi, Pakistan
- **Result:** *Romanian* hackers attack! 😊

or

- **Result:** it was a guy from the office on the 3rd floor?

or ...

How?

- **Interpretation is inherent** in answering the “how” question! Information is always incomplete!
- What is “how”?
 - How it got recorded [in logs]?
 - How we think it happened?
 - How we believe it happened?
 - How it happened? <- **only this is forensics**

What?

- “**False positives**” – need I say more? 😊
- Other “**lying logs**” include:
 - System errors that are corrected automatically
 - Misinterpreted log messages (*Oh my, error 4632 strikes again!*)
 - Artificially inserted logs (ever used `/usr/bin/logger`?)
 - Finally, logs that *someone edited*
 - Added (log flooding)
 - Removed (log “cleaning”)
 - Changed (log corruption)

So, How to Secure Logs?

Key theme: “**Good enough**” log security

- **Transmission** security (SSL, SSH, etc)
- **Storage** security (hashing -> signing -> encryption)
- No **changes!** After all, *why change logs?* 😊 WORM!
- **Access control** – “need to know” basis
- Access and *process logging* - log who saw the logs! [**]
- **Last resort**: printer + safe + armed guard

Overall, **no holes** from **log birth** to **analyst conclusion**


Logs Forensics Challenges: From US DoJ

“Computer Records and the Federal Rules of Evidence“

- “**First**, parties may **challenge the authenticity** of both computer-generated and computer-stored records by questioning whether the records were altered, manipulated, or damaged after they were created.
- **Second**, parties may question the authenticity of computer-generated records by **challenging the reliability** of the computer program that generated the records.
- **Third**, parties may challenge the authenticity of computer-stored records by **questioning the identity** of their author.”

Case Study: Horror of 1016 - Setup

- Employee Complains About Sexual Harassment by IT Manager
- Employee is FIRED for “hacking” in a few days
- Logs used as evidence
- Retaliation firing OR insider attack?
- **Let's find out!**



But I didn't do it!!!!!!

The Answer!

A 1016 event entry appears in the application event log after you upgrade to Outlook 2002 or to a later version of Outlook

[View products that this article applies to.](#)

This article was previously published under Q301328

Article ID : 301328
Last Review : January 31, 2007
Revision : 4.1

SUMMARY

The following event entry may be logged in the Application event log after you upgrade to Microsoft Outlook 2002 or to a later version of Outlook:

Event ID 1016 reports:

NT User [DOMAIN\User] logged on to [User] mailbox, and is not the primary Windows NT account on this mailbox.

This is expected behavior when you use Outlook.

NOTE: The Event ID 1016 message indicates a MapiLogon to the root of the mailbox. This event message occurs even if you have permission to access the mailbox, and whether you access it or not. Do not use this event message as an auditing event to monitor the logon activities to a mailbox.

“Case 1016” Lessons

- Log reading IS **NOT** log analysis
- Another angle to logs in court: **log misinterpretation**
- A question that will never be answered: **were they stupid or were they evil!?**

Bonus: Insider-proofing Your Logs

Logs vs Insiders: Insider-proofing Your Logs

Insider Threat?

Everybody, everybody talks ...

- The famous 80%-20% threat direction MYTH
- Occurrence vs damage from insider incidents?
- Solved problem of “external” threats?
- Stopping a dedicated insider: odds?
- Insider “hacking” vs crime?

Defining “Insider Threat”

Our **definition of “insider threat”** = any **threat actor** with **level of access** to your organization’s resources **beyond** that of the **general public**.

Example: partner’s engineer, janitor, window cleaner, CEO, CEO’s kid, etc

Choices, Choices

1. Tolerate?
2. Prevent
3. Block
4. Detect
5. Investigate

Repeat After Me ...

- You **cannot** prevent insider attacks!
- You **cannot** block many insider attacks!
- You *sometimes* **can** detect insider attacks!
- You **must** investigate insider attacks!

What About Access Controls?

- MYTH: **Stringent access controls** will stop insiders!



- What about those insiders that *have legitimate access*?

So, How Do You Fight!?

1. **Administrative/legal:** policy, awareness, inevitability of punishment, background check, etc
2. **Psychological:** profiling, behavioral monitoring, risky trait detection, etc
3. **Technical:** access controls, IDS/IPS, logging, honeypots, encryption, etc

But will it work?

In general, NO!

Why Logs vs Insiders?

- **Everybody leaves traces** in logs!
 - Potentially, every action could be logged!
- **Control doesn't scale**, accountability (=logs!) does!
 - More *controls* -> more complexity -> less *control*!
- The only technology that helps vs **insider with legitimate access**: logging!
 - Provided legit actions are logged...

Assumptions!

1. Insider actually touches a computer system
2. Logging is present and logs are collected
3. Insider cannot modify the logs!
4. Incident loss might grow if the insider is not stopped

Sidetrack: Log Trustworthiness Hierarchy

1. Compromised system logs
2. Desktop / laptop OS and application logs (possibly changed by users, legitimate systems owners, etc)
3. All logs from others systems where 'root'/Admin access is not controlled (e.g. test servers, etc)
4. Unix application logs (file-based)
5. Local Windows application logs
6. Local Unix OS syslogs
7. Unix kernel audit logs, process accounting records
8. Local Windows server OS (a little harder to change)
9. Database logs (*more trusted since DBA cannot touch them, while 'root' can*)
10. Other security appliance logs (located on security appliances)
11. Various systems logs centralized to a syslog server
12. Network device and firewall logs (centralized to syslog server)
13. Logs centralized to a log management system via a real-time feed (obviously, transport encryption adds even more trust)

What Logs Are Most Useful for Insider Investigations?

#1 The ones that you **actually have!**

#2 Logs from systems where the “**crown jewels**” are

#3 Logs that are associated with **user identity**

#4 Logs that cover **system and application activity**

Example: Firewall/Network Logs

Main: proof of connectivity (in and out of the company)

- Where did the data go?
- What did the system connect to?
- Who connected to the system and who didn't?
- How many bytes were transferred out?
- Who was denied when trying to connect to the system?

Firewall/Network Logs Als

Action items – to make these logs more useful for insider threat:

1. Enable logging of allowed connections
2. Enable logging for outbound connections, success and failed
3. Monitor unusual traffic from the inside out, especially successful and with large data transfers to unusual sites

Example: VPN Logs

Main: evidence on insider threats via remote access

- Network login success/failure
- Network logout
- Connection session length and the number of bytes moved

VPN Logs Als

Action items – to make these logs more useful for insider threat:

1. Retain these logs for longer
2. Retain DHCP server logs in combination with VPN logs
3. Watch for large data transfers from corporate to remote sites

Example: System Logs

Main: key logs on most activities

- Login success/failure
- Account creation
- Account deletion
- Account settings and password changes
- (On Windows) Various group policy and registry changes
- File access (read/change/delete)

Sidetrack: Protecting Logs from Admins (See Paper)

	"C" - prevent admins from reading logs	"I" - prevent admins from changing logs	"A" - prevent admin from disabling logging
Standard Unix	<i>Forget it!</i> Maybe stealth logging	Remote logging via syslog to another server, append-only log files (via RBAC)	<i>Forget it!</i> But this is logged and thus can be detected (also: stealth logging)
Windows server	<i>Forget it!</i> Maybe stealth logging	Pull the logs ASAP to a central server	<i>Forget it!</i> But this is logged and can be detected (also stealth logging)
Databases	DBA activity log stored outside the database (append-only access)	DBA activity log stored outside the database (append-only access)	DBA activity log stored outside the database
Firewalls and network gear	Remote logging via syslog to another server - no local logging	Remote logging via syslog to another server	<i>Forget it!</i> But this is logged and can be detected
IDS/IPS boxes	Remote logging to another server - no local logging	Remote logging to another server, inaccessible to admin	<i>Forget it!</i> But this is logged and can be detected
Misc enterprise applications	App admin log outside the app (not readable to application user)	App admin log outside the app (only appendable by the application user)	<i>Forget it!</i> But this is logged and can be detected

Example: Web Logs

Main: extrusion, data theft/loss records

- Connection to a specific website
- Data uploads
- Webmail access
- Some types of HTTP tunneling for data theft
- Spyware activities

Example: Database Audit

Main: database logs record access to crown jewels

- Database data access
- Data change
- Database structures and configuration change
- Database starts, stops, and other administration tasks

Database Audit AIs

Action items – to make these logs more useful for insider threat:

1. Enable data access logging
2. Enable database change logging
3. Enable backup, export and other data-intensive procedure logging
4. Enable DBA action logging
5. Preserve logs from DBAs

Example: Honeypot Logs

Main: evidence of actual insider threat activity

- Active recon by malicious insiders
- Record *only* malicious insider actions
- Can provide a complete recording of “a crime” (such as data theft)
- Needs other logs to build a case!

What You MUST Do Then?!

1. Have logs
2. Collect logs
3. Retain logs for longer
4. Review logs to learn the normal
5. Analyze logs for deviations

Conclusions: How to optimize logging for insider IR?

1. **Longer log retention: 1 year and more**
 - Might not be discovered for a while
2. **Broad range of log sources**
 - Insiders can do anything!
3. **Higher emphasis on log protection**
 - If **you** get sued (or intend to sue)
4. **More analysis of stored data**
 - Real-time won't cut it!

Conclusion

- **Turn ON Logging!!!**
- Make Sure Logs Are There When You Need Them (and need them you will 😊)
- Include Log Analysis into the IH Process
- Prepare and Learn the Analysis Tools
- When Going Into the Incident-Induced Panic Think *'Its All Logged Somewhere – We Just Need to Dig it Out'*
😊

Anton's Five Log Mistakes

How many have **you** committed? 😊

1. Not logging!
2. Not looking at logs
3. Not retaining long enough
4. Deciding what's relevant before collection
5. Ignoring application logs
6. Only looking at known bad

Anton's Five Incident Response Mistakes

How many have **you** committed? 😊

1. Not having a plan
2. Failing to increase monitoring and surveillance
3. Being unprepared for a court battle
4. “Putting it back the way it was”
5. Not learning from mistakes

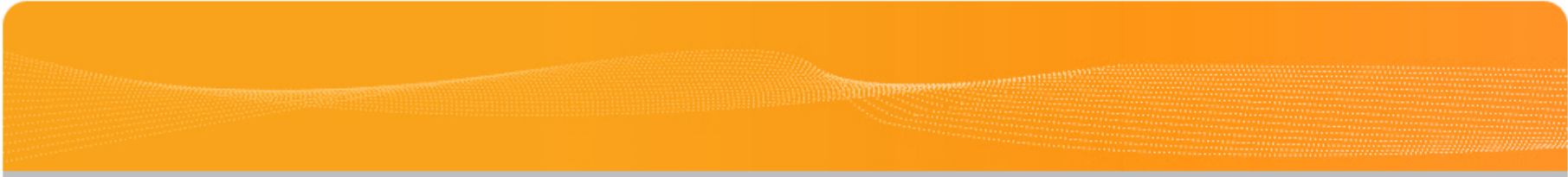
Thanks for Attending!!!

Dr Anton Chuvakin, GCIA, GCIH, GCFA
Chief Logging Evangelist

<http://www.chuvakin.org>

Author of “Security Warrior” (O’Reilly, 2004) –
<http://www.securitywarrior.org>

See **<http://www.info-secure.org>** for my
papers, books, reviews and other security
resources related to logs.



Extra Examples

Extra Examples

Example 12 Sysadmin Gone Bad

- Service Restarts Out of Maintenance Windows
- Correlated with Some Personnel Departures
- Information Leaks Start
- Log Analysis Reveals Unauthorized Software Installation

Example 13 Spyware Galore!

- System Seen Scanning – Firewall Logs
- Analysis of Logs Shows Antivirus Failures
- VPN Logs Help Track the Truth
- Full Forensic Investigation Confirms the Results of Log Analysis

Example 14 Compromise Detection (See My Paper)

Security technology/resource	Method	Example	Reliability
NIDS	Compromise signature	Shell commands on SSL port TCP 443	Medium
NIDS	Post exploit activity	'whoami' in command flow	Medium
NIDS	Volume of outbound exploits (same or different)	Lots of SSL hits out	Medium
NIDS	Volume of outbound exploits after a similar inbound exploit	Lots of SSL hits out after the system is hit by SSL exploit	High
NIDS, firewall	Outbound massive port scanning, DoS, etc	Many connections to port 1434 UDP from a single system	Medium
HIDS	Abuse-related system log records	New account created	Medium
HIPS	Application behaving significantly different from known good	Connections, registry access, file replacements	Medium