The importance of logs

You won’t see what you don’t log^Waudit

Tim (Wadhwa-)Brown
Head Of Research, CX EMEAR Security Architecture
May 2019

Introduction

Who here works in Operational Security?

TLDR

- What this talk is not about
  - Building a SOC in 30 minutes
- What this talk is about
  - Why logging goes wrong
  - Case studies
  - How to start to plan your logging requirements
  - Where to go next

whoami

- Tim (Wadhwa-)Brown
  - Background in operational security for telecoms and financial services sectors
  - 14+ years at Portcullis (and now Cisco)
  - ~12 years as a CREST consultant (Testing, Incident Response, Threat Intelligence)
- Current focus is delivering Security Engineering and operationalization of security for Cisco customers in EMEAR
- Head Of Research, CX EMEAR Security Architecture
  - >120 CVEs to my name
    - Covering Windows, Linux, AIX and Solaris platforms
    - Userland through to kernel

cat .plan

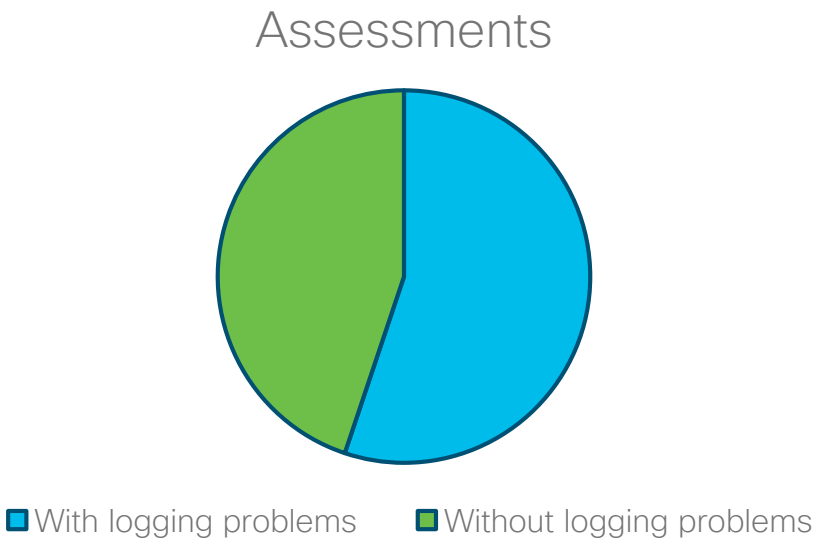
- Background
- Common failings
- Case studies
- How can I improve my telemetry?
- The “what” of auditing
- Conclusions
- Questions

Background

Slow response is expensive

- Average breach identification time is in excess of 6 months
- 50% of businesses suffered breaches with a financial impact greater than $500,000

Ineffective or missing logging is a real problem



Source: Cisco Security Advisory EMEAR assessment reports

“In over 50% of cases, logging will be insufficient to determine root cause, identify actions or attribute the actor.”

IRR consultant

Why does this matter?

- We're expected to give expert guidance from both a blue and red team perspective
- Our customers want to mature their security posture from a defensive standpoint
- The first two questions after every breach are "how?" and "when?"...
  - ...followed by "are they still active?"

Common failings

Common failings

- Unsynced time and/or multiple time zones
- Log ingestion
- Logging capacity and growth
- Poor logging capabilities
- Poorly configured logging
- Unfamiliarity with application stack
- Lack of ground truth
- Every failed security check should result in an audit event

Case studies

Case studies

- Recent incident
  - Everything was being treated as bad
- In-house development
  - No auditing in design
- Networking device
  - Logging had filled up server
- Mainframe
  - No knowledge about how to extract logs

How can I improve my telemetry?

How can I improve my telemetry?

- Ensure that you're risk focused
- Ensure that you consider your users
- Engage with the enterprise

Don’t wait for a breach!

Ensure that you're risk focused

- From an defensive standpoint, look at
  - Assets
  - Actors
  - Threats
  - Impact
- Where are the detective controls?
- Frameworks can help
  - Microsoft: STRIDE
  - MITRE: ATT&CK (TTP) and CAPEC (weaknesses)
- Does the solution help or hinder visibility?

Ensure that you consider your users

- Auditing every element of the stack could improve visibility
  - User
  - Application
  - API
  - Web Server
  - Database
  - Filesystem
  - OS
  - Network
- Get to know your SMEs

Engage with the enterprise

- With procurement
  - Build requirements into the procurement process
  - In particular, consider SaaS and PaaS vendors and their ability to service your requirements – systems you don't own are a particular pain point when collecting audit event feeds
- With platform teams
  - Ensure that the correct value of "good" is known
- With application support teams
  - Ensure auditing is switched on
- With developers
  - Ensure that detective controls are included in functional requirements
  - Check that you're not reliant on logs that are intended for debugging
  - Reject unknown exceptions

The “what” of auditing

Practice breeds confidence

- If a system is important enough to warrant a penetration test
- But you can't tell when your consultant…
  - Connected to the network
  - Began their Nessus scans
  - Ran Burp active scan against the admin interface
- You may not be collecting the right audit feeds…
  - Or you might not know where to look

Source	Category	Urgency	Events	Use case
DHCP	User/device attribution	High	IP assignments	Trace victims
VPN	User/device attribution	High	IP assignments	Trace victims
802.1x	User/device attribution	High	IP assignments	Trace victims
DNS	User/device attribution	High	DNS lookups	Identify C2
Firewall	User/device attribution	High	Blocked and successful connections	Trace victims
Email	Email activity	High	Message routing with headers and subjects	Discover campaigns
Proxy	Network activity	High	Blocked and successful connections	Identify C2
OS auditing	System activity	Medium	Authentication, configuration changes and security events	Identify breaches
AntiVirus	System activity	Medium	Malware discovery and removal	Identify contained breaches
Vulnerability scans	Vulnerability status	Medium	Vulnerability attribution	Attribute attack to vulnerability
AD authentication	User/device attribution	Low	Authentication and authorisation	Identify lateral movement
Netflow	Network activity	Low	Connections from enterprise to data center	Investigate access

Source: Aaron Varrone, Cisco Security Incident Response Services (CSIRS)

Key Considerations
Does the event meet legal requirements (no PII, etc)?
Validate that sensitive data has been randomised or removed (passwords, etc)
Ensure data is in the right format
Confirm event feed contains enough information to be useful (see Tab 4)
Event Types
Input validation failures
Change of privilege failures/successes
Authentication failure/successes
Session state changes
Suspicious behaviour and overuse
File uploads and writes
Access control failures/successes
Application and system errors
Any high-risk (those which may impact Confidentiality, Integrity or Availability (CIA) of the system) changes/administrative tasks
Sensitive Data
Personally identifiable information (PII)
Application source code
Session IDs
Access tokens
Passwords
Connection strings
Encryption keys and other master secrets
Bank account or payment card holder data
Data of a higher security classification than the logging system is allowed to store
Commercially-sensitive information
Information it is illegal to collect in the relevant jurisdictions
Information a user has opted out of collection, or not consented to e.g. use of do not track, or where consent to collect has expired
Consider replacing sensitive data with hashed equivalents in instances where these events need to be tracked

Conclusions

Conclusions

- What have we learnt?
- Next steps?

What have we learnt?

- Logging and auditing are rarely done well
  - Logging is for developers
  - Auditing should be for operators
- Everyone gets breached, plan for it
- Wouldn't it be nice to understand your environment
- Don't take my word for it...
  - https://www.ncsc.gov.uk/blog-post/learning-love-logging

Security Operations isn’t enough

(You need Security Engineering too =))

Next steps?

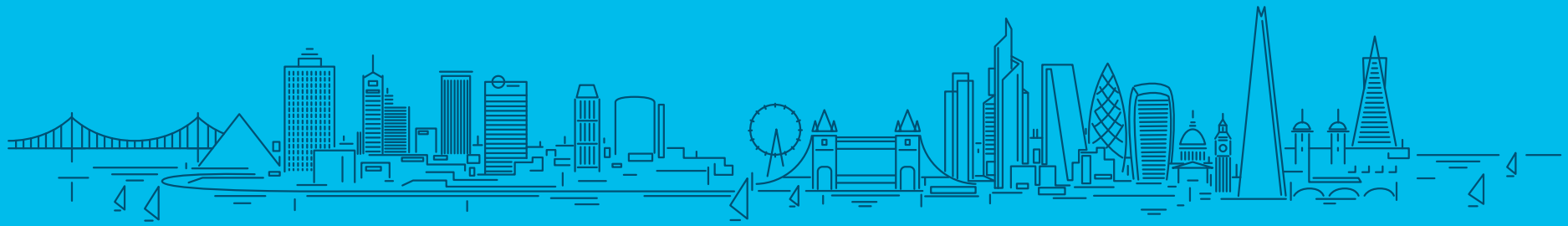
- Configure
  - Windows Event Log
    - Microsoft are obviously the canonical source
  - Linux Auditd
    - There are some great publicly shared policies on GitHub for this
- Collect the audit event feeds
  - There are open source solutions out there that DON'T use syslog but which do allow for audit event feeds to be collected in a secure fashion
- Build auditing into your SDLC
- Examine the audit events and learn what "good" looks like

Links

- NCSC
  - https://www.ncsc.gov.uk/guidance/introduction-logging-security-purposes
- NIST
  - https://csrc.nist.gov/publications/detail/sp/800-92/final
- Windows
  - https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/appendix-l--events-to-monitor
  - https://github.com/SwiftOnSecurity/sysmon-config
- Linux
  - https://github.com/bfuzzy/auditd-attack
  - https://github.com/Neo23x0/auditd

Questions?

twadhwab@cisco.com



Bonus material

Common failings

Unsynced time and/or multiple time zones

- TZ=Europe/London?
  - Ideally logs and events should always be timestamped against UTC

Log ingestion

- If we're lucky there may be remote ingestion using a SIEM agent
  - Often times there isn't
- And sometimes, the ingestion leverages syslog which is an insecure protocol
  - There's a question of integrity and attestation

Logging capacity and growth

- They may not be collected
- They almost certainly aren't processed
- You may well need to agree a suitable retention period
  - Check local regulations in case there is a legal minimum

Poor logging capabilities

- Perhaps not but…
  - Humans aren't the best at correlating ad-hoc events
  - Every attempt to brute force a vulnerability might look different but audit events tied to the root cause can be measured, benchmarked and SIEMs can be set to trigger alarms on thresholds

Poor logging configuration

- Logging often relies on defaults
  - It's really for debugging in many cases
- Auditing is rarely turned on
  - In cases where auditing is available, it's may not be ingested into the SIEM
- Configuring and enabling auditing involves thinking about TTPs and the IOCs they leave behind

Unfamiliarity with application stack

- IOCs are often missed
  - Would you spot a brute force attempt on an internal web application?
- Exceptions are left unhandled
  - Wouldn’t you want to know why a service keeps on crashing?

Lack of ground truth

- Figure out what audit events occur and when
- Benchmark them
- Institute BAU policies to check key audit events hourly, daily or weekly
  - Get into a habit
- Incidents are not the right time to be learning about your SIEM's query language

Case studies

Have we improved in 15 years?

- 15 years ago, I was sitting on the other side of the fence
  - Senior Operational Security Analyst
  - Working for a retail bank
- Problem
  - We wanted to know when people ran sudo and why
- Solution
  - HIPS & RBAC events fed into SQL Server
  - BAU processes to review events

Recent incident

- The admins have been subjected to a red team recently
- I'm there to do a penetration test
- They're all fired up watching their event logs
- STOP! What's making all those connections to "C$"
- Turned out it was cached connections being reactivated when they used the search bar
- 12 hours of my life I won't get back

In-house development

- Development house x are building a new application
- Threat modeling has identified where attacks are likely
- They didn't build auditing in
  - No way to determine what the normal cadence of password resets was and when there was a peak

Networking device

- A system has been changed and rebooted
- It's unclear by whom and under what circumstances
- Management are ready to throw a contractor under the bus
- The log server was full

Mainframe

- The box has been compromised
- Data has been wiped
- Yay! They have logs
  - Both application and OS
  - The problem is that the application logs weren't suitably granular (HH:SS)
- Boo! There is literally no documentation on what the logs actually mean
  - Reversing mainframe binaries is fun but wasteful
  - We eventually found an OS event in the logs that acted as a crib