Logging and Audit

Intro to log management and audit strategies

Stefan Kelm, DFN-CERT Services GmbH

WP8-T1

5 August 2020

www.geant.org
Logging is fairly simple, isn’t it?

Jul 04 14:10:59 shadow su:(to root) jones on /dev/pts2
Logging is fairly simple, isn’t it?

• No, it isn’t...
  − No standard format
  − No standard transport
  − No guidance on what to log and how
  − No guidance for developers

• But you have to do it!
Definitions / Terminology

• In general
  – **Logging** is commonly referred to as the act of keeping a log, a recording of **events** as well as the storage and analysis of these events
  – The **purpose of logging** is to have a record of events that happened in order to resolve
    • Errors / safety issues / security incidents / ...

• From NIST (National Institute of Standards and Technology)
  – “**A log** is a record of the events occurring within an organization’s systems and networks. Logs are composed of log entries; each **entry** contains information related to a specific **event** that has occurred within a system or network.”
  – “**Log management** is [...] the process for generating, transmitting, storing, analysing, and disposing of log data.”
Logs, where art thou?

- Logs are everywhere
  - Operating systems
    - Linux Syslog, Windows Event Log, ...
  - Server logs (HTTP, SMTP, SNMP, DNS, SQL, etc.) and many, many more...
    - Application logs were designed for troubleshooting/debugging, not investigating
  - Device logs
    - Routers, Switches, Firewalls, IDS, EDR, AV, ...
    - Smartphones
    - Did anyone say “IoT”?

- All these logs will make our jobs easier, no?
Why is logging important?

• Logs assist us in
  – Detecting all things security
    • Intrusion Detection
    • Incident Containment and Response
    • Forensic Analysis / e-Discovery ("super timeline all the things")
    • Real Time Alerting
  – Providing a Network Baseline
  – Determining the Health of the Network
    • Operational issues
    • Performance issues
  – Detecting policy breaches: Audit Trail
  – Achieving
    • $$$ regulatory compliance (GDPR, PCI DSS, ISO 27001, HIPAA, SOX, ...) $$$
    • (Internal) policies compliance
Who wants logs?

• Who are typical log file users/consumers?
  – System and network administrators
  – Security administrators
  – Computer Security Incident Response Teams (CSIRT, CERT)
  – Application developers
  – Chief information security officers (CISO, CIO)
  – (External) auditors
  – Others in your org?

• Do all consumers want the same logs?
  – Do they all need the same logs?
  – Are they all allowed to look at the same logs?
Let’s have a look...
Windows Event Logs

- **The dark ages (up to Windows XP)**
  - Binary Event Log file format
  - **Location:** %SystemRoot%\System32\Config
  - Mainly 3 categories:
    - **Security:** secevent.evt
    - **System:** sysevent.evt
    - **Application:** appevent.evt

- **Beginning with Vista**
  - New binary XML format, new extension: .evtx
  - **Location:** \Windows\System32\winevt\Logs\n  - Many more files:
    - Security.evtx, System.evtx, Application.evtx
    - → 120 files ++
Example: logon event (Event Viewer)
What am I looking for? (=> check the Appendix for more)

<table>
<thead>
<tr>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Lockouts</td>
<td>4740</td>
<td>Informational</td>
<td>Security</td>
</tr>
<tr>
<td>User Added to Privileged Group</td>
<td>4728, 4732, 4756</td>
<td>Informational</td>
<td>Security</td>
</tr>
<tr>
<td>Successful User Account Login</td>
<td>4624</td>
<td>Informational</td>
<td>Security</td>
</tr>
<tr>
<td>Failed User Account Login</td>
<td>4625</td>
<td>Informational</td>
<td>Security</td>
</tr>
<tr>
<td>Account Login with Explicit Credentials</td>
<td>4648</td>
<td>Informational</td>
<td>Security</td>
</tr>
</tbody>
</table>
**Linux Syslog: Facility (source) vs. Severity level**

<table>
<thead>
<tr>
<th>Facility code</th>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>kern</td>
<td>Kernel messages</td>
</tr>
<tr>
<td>1</td>
<td>user</td>
<td>User-level messages</td>
</tr>
<tr>
<td>2</td>
<td>mail</td>
<td>Mail system</td>
</tr>
<tr>
<td>3</td>
<td>daemon</td>
<td>System daemons</td>
</tr>
<tr>
<td>4</td>
<td>auth</td>
<td>Security/authentication messages</td>
</tr>
<tr>
<td>5</td>
<td>syslog</td>
<td>Messages generated internally by syslogd</td>
</tr>
<tr>
<td>6</td>
<td>lpr</td>
<td>Line printer subsystem</td>
</tr>
<tr>
<td>7</td>
<td>news</td>
<td>Network news subsystem</td>
</tr>
<tr>
<td>8</td>
<td>uucp</td>
<td>UUCP subsystem</td>
</tr>
<tr>
<td>9</td>
<td>cron</td>
<td>Clock daemon</td>
</tr>
<tr>
<td>10</td>
<td>authpriv</td>
<td>Security/authentication messages</td>
</tr>
<tr>
<td>11</td>
<td>ftp</td>
<td>FTP daemon</td>
</tr>
<tr>
<td>12</td>
<td>ntp</td>
<td>NTP subsystem</td>
</tr>
<tr>
<td>13</td>
<td>security</td>
<td>Log audit</td>
</tr>
<tr>
<td>14</td>
<td>console</td>
<td>Log alert</td>
</tr>
<tr>
<td>15</td>
<td>solaris-cron</td>
<td>Scheduling daemon</td>
</tr>
<tr>
<td>16–23</td>
<td>local0 – local7</td>
<td>Locally used facilities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Severity</th>
<th>Keyword</th>
<th>Deprecated keywords</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Emergency</td>
<td>emerg</td>
<td>panic [7]</td>
<td>System is unusable</td>
</tr>
<tr>
<td>1</td>
<td>Alert</td>
<td>alert</td>
<td></td>
<td>Action must be taken immediately</td>
</tr>
<tr>
<td>2</td>
<td>Critical</td>
<td>crit</td>
<td></td>
<td>Critical conditions</td>
</tr>
<tr>
<td>3</td>
<td>Error</td>
<td>err</td>
<td>error [7]</td>
<td>Error conditions</td>
</tr>
<tr>
<td>4</td>
<td>Warning</td>
<td>warning</td>
<td>warn [7]</td>
<td>Warning conditions</td>
</tr>
<tr>
<td>5</td>
<td>Notice</td>
<td>notice</td>
<td></td>
<td>Normal but significant conditions</td>
</tr>
<tr>
<td>6</td>
<td>Informational</td>
<td>info</td>
<td></td>
<td>Informational messages</td>
</tr>
<tr>
<td>7</td>
<td>Debug</td>
<td>debug</td>
<td></td>
<td>Debug-level messages</td>
</tr>
</tbody>
</table>

```bash
# print some stuff on tty10
# kern.warning;*.err;authpriv.none    /dev/tty10

# Forward all messages to central loghost
*.* @loghost

# Backup file of all messages
*.* -/var/log/messages
```
Audit
Definition: Audit

- NIST
  
  "Audit records contain security event information such as successful and failed authentication attempts, file accesses, security policy changes, account changes (e.g., account creation and deletion, account privilege assignment), and use of privileges."

- "OSs typically permit system administrators to specify which types of events should be audited and whether successful and/or failed attempts to perform certain actions should be logged."

- In other words
  
  - "An audit policy determines which type of information about the system you’ll find in the logs."
Windows Audit: auditpol

- 9 categories / many sub-categories
  - Audit **account logon** events
  - Audit account management
  - Audit directory service access
  - Audit **logon** events
  - Audit object access
  - Audit policy change
  - Audit privilege use
  - Audit process tracking
  - Audit system events

- Log: success, failure, both, none
Linux Audit (audit.rules)

- Does this look familiar?
  - Watching file access
  - Monitoring system calls
  - Recording commands run by a user
  - Recording security events
  - Searching for events
  - Running summary reports
  - Monitoring network access
  - Changes to any trusted database such as /etc/passwd
  - ...

```bash
-a exit,always -F arch=b64 -S execve -F path=/bin/rm  -k Delete
-a exit,always -F arch=b64 -S execve -F path=/bin/vi   -k Create_Edit_View_File

# Audit shutdown & Reboot command
-a exit,always -F arch=b64 -S execve -F path=/sbin/reboot -k Reboot
-a exit,always -F arch=b64 -S execve -F path=/sbin/init  -k Reboot
-a exit,always -F arch=b64 -S execve -F path=/sbin/poweroff -k Reboot
-a exit,always -F arch=b64 -S execve -F path=/sbin/shutdown -k Reboot

# Audit mount unmount commands
-a exit,always -F arch=b64 -S execve -F path=/bin/mount  -k mount_device
-a exit,always -F arch=b64 -S execve -F path=/bin/umount  -k unmount_device

# Kill Process
-a exit,always -F arch=b64 -S kill   -k Kill_Process

# Important files
-w /etc/passwd  -p wa  -k passwd_changes
-w /etc/group  -p wa  -k group_changes
```
However...

“The company's server logs recorded only unsuccessful log-in attempts, not successful ones, frustrating a detailed analysis.”
You really want to have log management

- Log management 101
  - More logs don’t make you more secure
  - Better management does
Challenges in log management

- Log generation and storage
  - Many log sources: log data is scattered (sometimes even crossing borders) and has become **big data**
  - Inconsistent log contents
  - **Inconsistent timestamps / time zones / lots of different time formats**
  - Different vendors: different log formats
  - Different character sets
  - Log data is (sometimes) ephemeral
Challenges in log management

• Log analysis
  – boring, but that’s what it’s all about, no?
    • Log analysis is a unique skill
    • Log analysis takes time
  – **Correlation** of log entries
    • Multiple lines belonging to a single event
    • Similar log entries showing up in different log files
    • Timestamps
    • Probes over time

• Without sound processes for analyzing logs, their value is significantly reduced
Best Practices

- Start small
- Start small
- Start small
- Start small
- Start really small
Best Practices

• First steps
  - Develop logging/audit policy – what to log and why
    • Don’t fall for the recommendation to enable only *Failure* events for audit categories
  - Simple use cases: determine what information is relevant to you
    • What devices/events are important?
    • What reports do you and the org want/need?
  - Create a baseline
    • Determine “normal” behaviour (systems and network)
    • Repeat at regular intervals
  - Ensure all devices use the same time source (if possible)
    • Use NTP from a secure source
    • Use UTC, especially if operating in more than one time zone

• Don’t forget to enable logging ;-)
Best Practices

• Log security / log protection
  - Limit access to log files
  - Secure the processes that generate the log files
  - Configure log sources appropriately
    • What if logging fails?
    • What about “full” log files/partitions?
  - Implement secure mechanisms for transporting log data from the system(s) to the centralized log management server(s)
    • Consider using not only syslog via UDP (e.g., syslog-ng)
  - Are there regulatory requirements?
  - Anonymisation / Pseudonymisation?
  - Disk space / Storage / hardware requirements
    • Compression is very useful
Best Practices: Central log management

Devices

Collector

Storage

Analyzer
Best Practices: Central log management

• Log host (Collector) requirements
  - (High) Availability / Scalability
    - Redundancy necessary?
  - Watch out for single point of failures
    - Network, routing
    - Power supply, environmental conditions (fire, water, ...)
  - Log hosts are **highly critical systems**!
    - Just logging, no other services running
    - No connections from the outside
    - Will usually not “talk” to other systems
    - Harden the system, apply patches quickly
    - No general accounts, separation of duty!
  - Configure all devices to send logs to the dedicated log host
  - Check the license(s)!
Best Practices: Central log management

● Normalise the logs
  - Event Logs, syslog, SNMP, etc. need to be converted into the same format

● Log rotation
  - Determine time schedule, based on volume of data
  - Develop meaningful naming convention

● Log retention
  - Based on disk space
  - Based on regulatory requirements: Why? How? What? For how long?
  - Archive in secure area / external storage / offline backup?

● Visualise the logs (one interface to rule them all)

● Fortunately, you don’t have to do the above manually
Example log mgmt: NXLog

Linux
- RHEL
- CentOS
- Debian
- Ubuntu
- SUSE Linux

Unix
- AIX
- Solaris
- macOS

Windows
- Win Server
- Win Nano

BSD
- FreeBSD
- OpenBSD

Input modules
- NXLog
- Agent, agentless or hybrid
- Parallel Input and output
- Parsing, Filtering & Conversion
- Scalable & modular architecture
- Message buffering
- Prioritized processing
- Over 100,000 EPS handling

Log formats
- Syslog (BSD, IETF, CEF, LEEF and Snare)
- Windows Event Log, XML, CSV and JSON

SIEM
- IBM QRadar
- RSA NetWitness
- Rapid7 InsightIDR
- Splunk Enterprise
- FireEye Helix
- MicroFocus ArcSight
- McAfee ESM
- Securonix
So many tools to choose from

- From **Log** management to **Security** Information and Event Management (**SIEM**)
Pro tips: for future use

- Eventually you may want to do things like
  - File and folder auditing
    - file/folder creation/deletion, permission changes, ownership changes, ...
  - Windows Registry Auditing
    - Forensicators love this!
  - Process command line logging
    - May catch fileless malware
  - Windows PowerShell Logging
    - a.k.a. “command line 2.0”
  - Windows Sysmon Logging
    - Greatly enhances the Windows Event Log
- Note: the above are all very noisy
Legal and Regulatory Compliance

- Not part of this talk (IANAL :-) )

- But always **do** check with your data protection officer (DPO)!
Summary

• Log management usually is funded by compliance budgets!
  – “Buy for compliance, use for security and operations.”

• There’s no: one size fits all
  – Each network is unique
  – Keep it simple, stupid (KISS) --- Start really small!

• Ask yourself...
  – What logs am I going to collect locally?
  – What logs am I going to send to the central log host/SIEM and how?
  – What logs am I going to analyse centrally?

➲ Start logging
  – then start collecting logs
  – then start reviewing and analyzing logs
Thank you

Any questions?

www.geant.org

© GÉANT Association on behalf of the GN4 Phase 2 project (GN4-2). The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).
References (1)

• Useful web sites
  - Guide to Computer Security Log Management (NIST)
  - Microsoft TechNet
  - Ultimate Windows Security
    • https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/
  - SolarWinds
    • https://www.loggly.com/ultimate-guide/centralizing-windows-logs/
References (2)

- **Useful web sites**
  - EventID.Net
    - [https://eventid.net/](https://eventid.net/)
  - Security Event Log Collection from a Domain Controller
  - Microsoft TechNet
  - RHEL Audit System Reference
    - [https://access.redhat.com/articles/4409591](https://access.redhat.com/articles/4409591)
  - Linux Audit Documentation Project
    - [https://github.com/linux-audit/audit-documentation/wiki](https://github.com/linux-audit/audit-documentation/wiki)
References (3)

• Useful web sites
  – Spotting the Adversary with Windows Event Log Monitoring
  – Centralizing Windows Logs
    • https://www.loggly.com/ultimate-guide/centralizing-windows-logs/
  – Windows Logging Cheat Sheets
    • https://www.malwarearchaeology.com/cheat-sheets
  – OWASP Logging Cheat Sheet
    • https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html
References (4)

• Useful web sites
  - Processing Data to Protect Data: Resolving the Breach Detection Paradox
Further reading

  
  - Chris Phillips, Kevin Schmidt, Anton Chuvakin
  - Syngress 2013
  - ISBN: 9781597496353
## General logging configuration recommendations (NIST)

### Table 4-1. Examples of Logging Configuration Settings

<table>
<thead>
<tr>
<th>Category</th>
<th>Low Impact Systems</th>
<th>Moderate Impact Systems</th>
<th>High Impact Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long to retain log data</td>
<td>1 to 2 weeks</td>
<td>1 to 3 months</td>
<td>3 to 12 months</td>
</tr>
<tr>
<td>How often to rotate logs</td>
<td>Optional (if performed, at least every week or every 25 MB)</td>
<td>Every 6 to 24 hours, or every 2 to 5 MB</td>
<td>Every 15 to 60 minutes, or every 0.5 to 1.0 MB</td>
</tr>
<tr>
<td>If the organization requires the system to transfer log data to the log management infrastructure, how frequently that should be done</td>
<td>Every 3 to 24 hours</td>
<td>Every 15 to 60 minutes</td>
<td>At least every 5 minutes</td>
</tr>
<tr>
<td>How often log data needs to be analyzed locally (through automated or manual means)</td>
<td>Every 1 to 7 days</td>
<td>Every 12 to 24 hours</td>
<td>At least 6 times a day</td>
</tr>
<tr>
<td>Whether log file integrity checking needs to be performed for rotated logs</td>
<td>Optional</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Whether rotated logs need to be encrypted</td>
<td>Optional</td>
<td>Optional</td>
<td>Yes</td>
</tr>
</tbody>
</table>
4.1 Application Whitelisting

Application whitelisting events should be collected to look for applications that have been blocked from execution. Any blocked applications could be malware or users trying to run unapproved software. Software Restriction Policies (SRP) is supported on Windows XP and above. The AppLocker feature is available for Windows 7 and above Enterprise and Ultimate editions only. Application Whitelisting events can be collected if SRP or AppLocker are actively being used on the network.

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppLocker Block</td>
<td>8003, 8004</td>
<td>Error</td>
<td>Microsoft-Windows-AppLocker/EXE and DLL</td>
<td>Microsoft-Windows-AppLocker</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Warning</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Warning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRP Block</td>
<td>865, 866, 867, 868, 882</td>
<td>Warning</td>
<td>Application</td>
<td>Microsoft-Windows-SoftwareRestrictionPolicies</td>
</tr>
</tbody>
</table>

Table 2: Whitelisting Events
4.2 Application Crashes

Application crashes may warrant investigation to determine if the crash is malicious or benign. Categories of crashes include Blue Screen of Death (BSOD), Windows Error Reporting (WER), Application Crash and Application Hang events. If the organization is actively using the Microsoft Enhanced Mitigation Experience Toolkit (EMET), then EMET logs can also be collected.

<table>
<thead>
<tr>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Error 1000</td>
<td>Error</td>
<td>Application</td>
<td>Application Error</td>
</tr>
<tr>
<td>App Hang 1002</td>
<td>Error</td>
<td>Application</td>
<td>Application Hang</td>
</tr>
<tr>
<td>BSOD 1001</td>
<td>Error</td>
<td>System</td>
<td>Microsoft-Windows-WER-SystemErrorReporting</td>
</tr>
<tr>
<td>WER 1001</td>
<td>Informational</td>
<td>Application</td>
<td>Windows Error Reporting</td>
</tr>
<tr>
<td>EMET 1</td>
<td>Warning</td>
<td>Application</td>
<td>EMET</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>Application</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Application Events
### 4.3 System or Service Failures

System and Services failures are interesting events that may need to be investigated. Service operations normally do not fail. If a service fails, then it may be of concern and should be reviewed by an administrator. If a Windows service continues to fail repeatedly on the same machines, then this may indicate that an attacker is targeting a service.

<table>
<thead>
<tr>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Service Fails</td>
<td>Error</td>
<td>System</td>
<td>Service Control Manager</td>
</tr>
<tr>
<td>or Crashes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7022, 7023, 7024, 7026,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7031, 7032, 7034</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: System Events
4.4 Windows Update Errors
A machine must be kept up to date to mitigate known vulnerabilities. Although unlikely, these patches may sometimes fail to apply. Failure to update issues should be addressed to avoid prolonging the existence of an application issue or a vulnerability in the operating system or an application.

<table>
<thead>
<tr>
<th>Windows Update Failed</th>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotpatching Failed</td>
<td>1009</td>
<td>Informational</td>
<td>Setup</td>
<td>Microsoft-Windows-Servicing</td>
</tr>
</tbody>
</table>

Table 5: Windows Update Failed Events
4.5 Windows Firewall
If client workstations are taking advantage of the built-in host-based Windows Firewall, then there is value in collecting events to track the firewall status. For example, if the firewall state changes from on to off, then that log should be collected. Normal users should not be modifying the firewall rules of their local machine.

<table>
<thead>
<tr>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
</table>

Table 6: Firewall Events

The above events for the listed versions of the Windows operating system are only applicable to modifications of the local firewall settings.
4.6 Clearing Event Logs
It is unlikely that event log data would be cleared during normal operations and it is likely that a malicious attacker may try to cover their tracks by clearing an event log. When an event log gets cleared, it is suspicious. Centrally collecting events has the added benefit of making it much harder for an attacker to cover their tracks. Event Forwarding permits sources to forward multiple copies of a collected event to multiple collectors thus enabling redundant event collection. Using a redundant event collection model can minimize the single point of failure risk.

<table>
<thead>
<tr>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Log was Cleared</td>
<td>104</td>
<td>Informational</td>
<td>System</td>
</tr>
<tr>
<td>Audit Log was Cleared</td>
<td>1102</td>
<td>Informational</td>
<td>Security</td>
</tr>
</tbody>
</table>

Table 7: Log Activity Events
4.7 Software and Service Installation
As part of normal network operations, new software and services will be installed, and there is value in monitoring this activity. Administrators can review these logs for newly installed software or system services and verify that they do not pose a risk to the network.

<table>
<thead>
<tr>
<th>Event Description</th>
<th>ID</th>
<th>Level</th>
<th>Event Log</th>
<th>Event Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Kernel Filter Driver</td>
<td>6</td>
<td>Informational</td>
<td>System</td>
<td>Microsoft-Windows-FilterManager</td>
</tr>
<tr>
<td>New Windows Service</td>
<td>7045</td>
<td>Informational</td>
<td>System</td>
<td>Service Control Manager</td>
</tr>
<tr>
<td>New MSI File Installed</td>
<td>1022, 1033</td>
<td>Informational</td>
<td>Application</td>
<td>MsilInstaller</td>
</tr>
<tr>
<td>Updated Application</td>
<td>905, 906</td>
<td>Informational</td>
<td>Microsoft-Windows-Application-Experience/Program-Inventory</td>
<td>Microsoft-Windows-Application-Experience</td>
</tr>
<tr>
<td>Removed Application</td>
<td>907, 908</td>
<td>Informational</td>
<td>Microsoft-Windows-Application-Experience/Program-Inventory</td>
<td>Microsoft-Windows-Application-Experience</td>
</tr>
<tr>
<td>Summary of Software Activities</td>
<td>800</td>
<td>Informational</td>
<td>Microsoft-Windows-Application-Experience/Program-Inventory</td>
<td>Microsoft-Windows-Application-Experience</td>
</tr>
<tr>
<td>Update Packages Installed</td>
<td>2</td>
<td>Informational</td>
<td>Setup</td>
<td>Microsoft-Windows-Servicing</td>
</tr>
<tr>
<td>Windows Update Installed</td>
<td>19</td>
<td>Informational</td>
<td>System</td>
<td>Microsoft-Windows-WindowsUpdateClient</td>
</tr>
</tbody>
</table>

[^46]: Microsoft-Windows-Application-Experience/Program-Inventory
[^47]: Microsoft-Windows-Application-Experience/Program-Inventory

Table 8: Software and Service Events