Unified Logging

Matteo Bolognini

Product Specialist

skartek.dev

Agenda

- Intro to Unified Logging
- Get started
- Examples
- Resources

jamf

What is Unified Logging?

- A comprehensive and performant API to capture telemetry across all levels of the system
- Logs to memory and datastore, persistence varies depending importance
- Design privacy into the system

Availability of Unified Logging

- iOS 10.0 and later
- macOS 10.12 and later
- tvOS 10.0 and later
- watchOS 3.0 and later

jamf

Where are logs?

- Logs are stored in tracev3 formatted files
- /var/db/diagnostic
- compressed binary format

How to read the logs?

- Console.app
- Ulbow.app (https://eclecticlight.co/consolation-t2m2-and-log-utilities/)
- log command

log command

- Open Terminal
- Type log
- Type log stream
- Both are not good approach, we need filters!

jamf

Filtering the log

- Simple comparison such as *process == “osinstallersetupd”* or process contains “bluetooth”
- Case or diacritic insensitive lookups, like process contains[cd] “BlueTooTH”
- Logical operations, like process beginswith[cd] “A” and message contains “sharingd”

Filtering the log

- Filtering is done using predicates
- Predicates are defined as parameters of the log command
- Example:

```
log stream --predicate '(process == "Safari")'
```

jamf

Subsystem and Categories

Subsystem	Category
com.your-company.your-app	Setup. In progress, teardown
com.apple.mac.install	OSPersonalization, BridgeOSInstall
com.apple.unc	Dnd, AppInfo
com.apple.sharing	AutoUnlock, AirDrop
com.apple.appstored	Install, Download, Purchase, Update

Identify the Subsystem

```
defaults read /System/Applications/Maps.app/Contents/Info.plist CFBundleIdentifier

com.apple.Maps
```

Log Level: Default

Used to capture information about things that might result in a failure

Stored in memory buffers

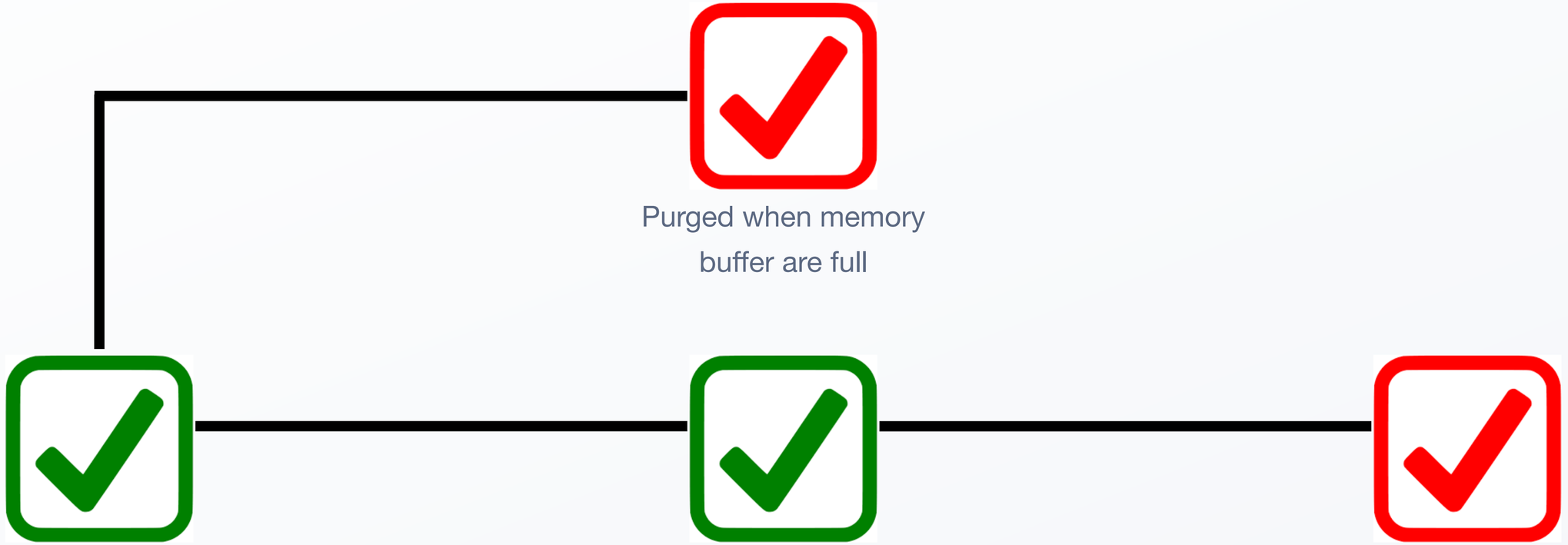
Compressed and moved to data store once memory buffers are full

Oldest purged from data store once the storage quota is exceeded

jamf

Log Level: Info

Used to capture information that may be helpful but not essential



Purged when memory buffer are full

Stored in memory buffers

Compressed and moved to data store if the coed logs are error or fault

Oldest purged from data store once the storage quota is exceeded

jamf

Log Level: Debug

Used to capture verbose information

Discarded by default

jamf

Specify the log level

```
log (stream/show) --predicate (--debug/--info)
```

Set the log level

```
log config --subsystem "com.jamf.management.binary"--mode "level:debug,persist:debug"
```

Reset the log level

```
log config --subsystem "com.jamf.management.binary" --reset"
```

jamf

AirDrop

```
log show --predicate '(
subsystem == "com.apple.sharing"
AND process == "AirDrop"
AND processImagePath BEGINSWITH "/System/Library"
AND eventMessage BEGINSWITH "Successfully issued sandbox extension for"
)'
--last 10m
--info
```

AirDrop

```
Filtering the log data using "subsystem == "com.apple.sharing" AND process == "AirDrop" AND
processImagePath BEGINSWITH "/System/Library" AND composedMessage BEGINSWITH "Successfully issued
sandbox extension for""
Skipping debug messages, pass --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2023-03-22 10:58:18.425566+0000 0x4d4d     Default     0x18008              2650   3    AirDrop:
(Sharing) [com.apple.sharing:Framework] Successfully issued sandbox extension for /Users/matteo/Desktop
/Screenshot 2023-03-22 at 10.58.02.png
---------------------------------------------------------------------------------------------
-------------
Log      - Default:          1, Info:                0, Debug:            0, Error:          0, Fault:
         0
Activity - Create:           0, Transition:          0, Actions:          0
```

```
log show --predicate '(
subsystem == "com.apple.sharing"
AND process == "AirDrop"
AND processImagePath BEGINSWITH "/System/Library"
AND eventMessage BEGINSWITH "Successfully issued sandbox extension for"
)'
--last 10m
--info
```

jamf

Security - sudo

```
log show --predicate '(process == "sudo") && (eventMessage CONTAINS[cd] "TTY")'
```

Security - sudo

```
Filtering the log data using "process == "sudo" AND composedMessage CONTAINS[cd] "TTY""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL

2023-03-22 14:31:46.443281+0000 0xf7b1     Default     0x0                  3050   0    sudo:   matteo : 3 incorrect password
attempts ; TTY=ttys002 ; PWD=/Users/matteo ; USER=root ; COMMAND=/usr/bin/su

---------------------------------------------------------------------------------------------------------------
Log      - Default:          1, Info:               0, Debug:            0, Error:          0, Fault:          0
Activity - Create:           0, Transition:         0, Actions:          0
```

```
log show --predicate '(process == "sudo") && (eventMessage CONTAINS[cd] "TTY")'
```

jamf

LoginWindow login with TouchID

```
log show --predicate '(process == "loginwindow" AND eventMessage CONTAINS[c]
"APEventTouchIDMatch")' --last 30m --info
```

jamf

LoginWindow login with TouchID

```
Filtering the log data using "process == "loginwindow" AND composedMessage CONTAINS[c] "APEventTouchIDMatch""
Skipping debug messages, pass --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2023-03-22 12:36:52.739142+0000 0x3366c    Default     0x0                  391    5    loginwindow:
[com.apple.loginwindow.logging:Standard] -[LWAuthServiceManager event:eventHints:reply:] | ======== SCREENLOCK ====
APEventTouchIDMatch =========
```

```
log show --predicate '(process == "loginwindow" AND eventMessage CONTAINS[c]
"APEventTouchIDMatch")' --last 30m --info
```

jamf

Configuration Profile installed manually by user

```
log show --predicate '(subsystem == "com.apple.ManagedClient" AND process == "mdmclient" AND category == "MDMDaemon" and eventMessage
CONTAINS "Installed configuration profile:" AND eventMessage CONTAINS "Source: Manual")' --last 10m --debug
```

jamf

Configuration Profile installed manually by user

```
Filtering the log data using "subsystem == "com.apple.ManagedClient" AND process == "mdmclient" AND category == "MDMDaemon" AND
composedMessage CONTAINS "Installed configuration profile:" AND composedMessage CONTAINS "Source: Manual""
Skipping info messages, pass --info to include.
Timestamp                       Thread     Type        Activity            PID    TTL
2023-03-22 12:46:36.522121+0000 0xb712     Default     0x270b0             2983   0    mdmclient:


[com.apple.ManagedClient:MDMDaemon] [0:MDMDaemon:<0xb712>] Installed configuration profile: Finder  68EA6B84-F4CE-4A88-B826-
DD9F831E5173:68EA6B84-F4CE-4A88-B826-DD9F831E5173) for <Computer> (Source: Manual)
```

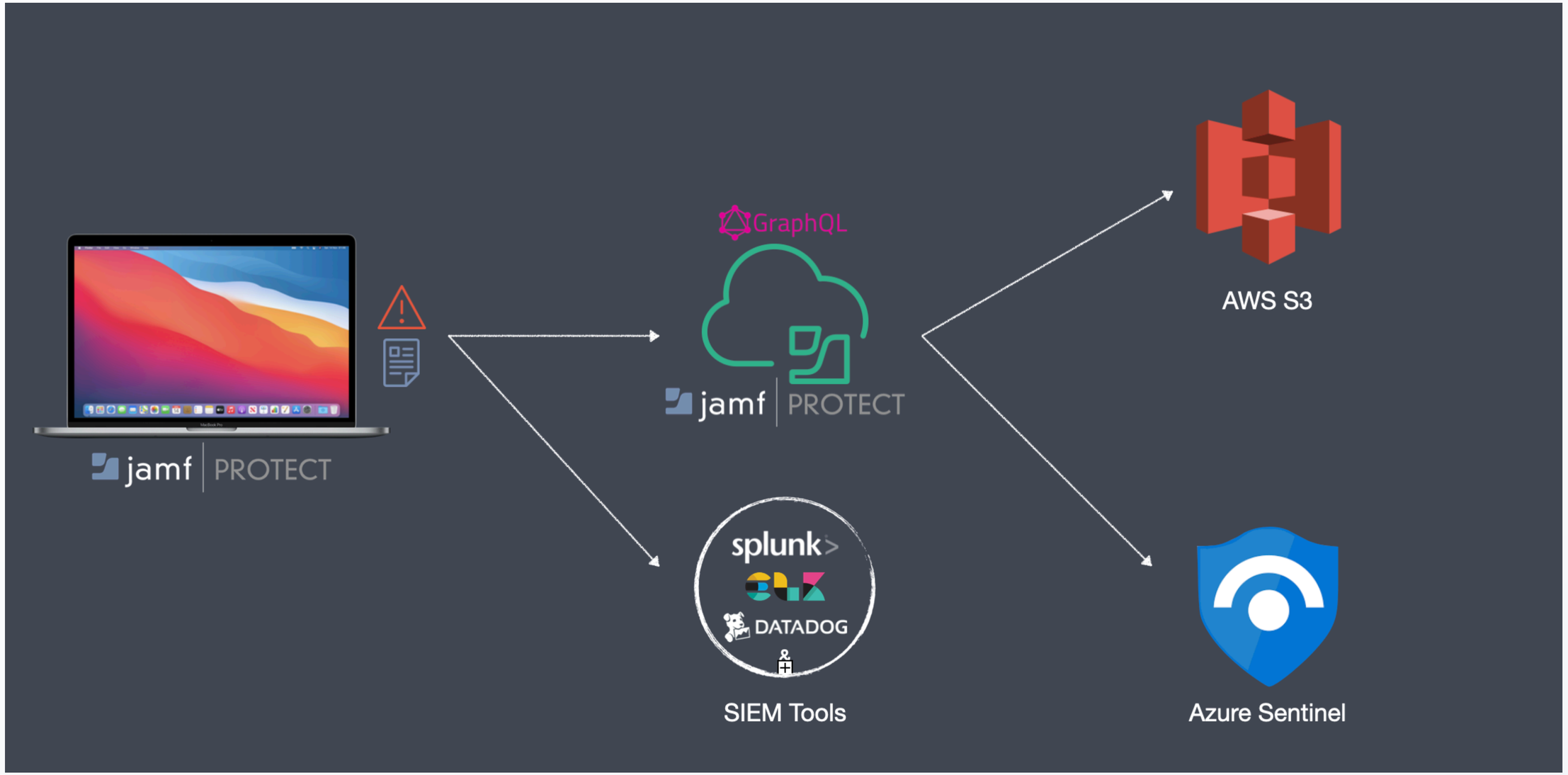
```
log show --predicate '(subsystem == "com.apple.ManagedClient" AND process == "mdmclient" AND category == "MDMDaemon" and eventMessage
CONTAINS "Installed configuration profile:" AND eventMessage CONTAINS "Source: Manual")' --last 10m --debug
```

jamf

How to make use of UL?

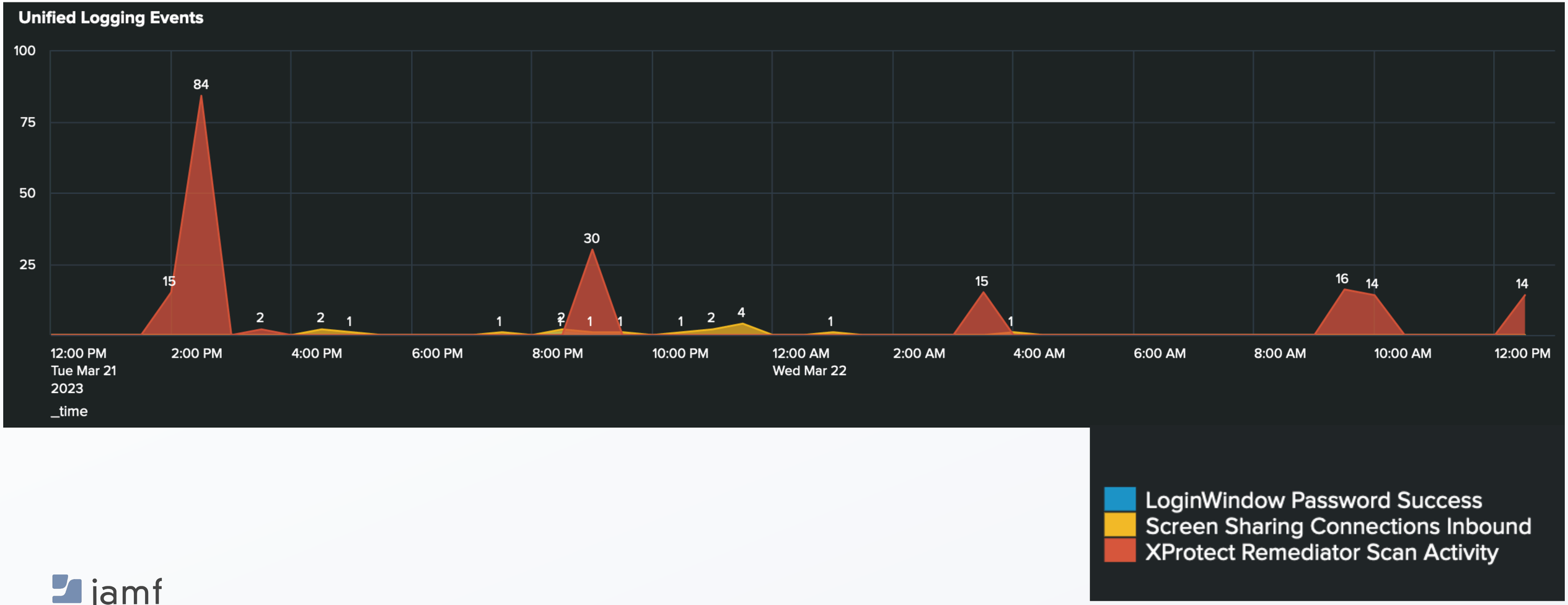
Jamf | Protecc

Jamf Protect & Unified Logs



jamf

Visualize data



jamf

Resources

Unified Log examples:

jamf

Resources

Slides:

jamf


That's all Folks!