He’s Making a List
We’re Checking it Twice

Santa for Forensic Analysis
Other Titles Considered

I’m Telling You Why: Santa as a Forensics Tool

He Sees You When You’re Hacking, He Knows Just What You Take

I Saw Badness Using Santa Logs
whoami

Gary

DFIR @ Google

All incidents Security + Insider

Formerly Detection, Google and Federal Reserve NIRT

Sugar, fast food, television enthusiast
whoami

James

DFIR @ Google

Responding to all the security and privacy things

Previous life with Mandiant, the United Nations, US Government

Lover of cheese jokes
What’s the Plan?

- What is Santa?
- Discussion: well known stuff
- Discussion: lesser known stuff
- Analysis strategies
Santa?

Knows whether your binary is naughty or nice.
What is Santa?

- macOS extensible through kernel extensions (KEXT)
- Kernel programming interfaces (KPIs) can be leveraged
- Santa uses the Kernel Authorization (Kauth) KPI that provides powerful features
- Allows Santa to listen in on most vnode and file system operations
  - Can then take direct or indirect action on operations being performed
- Open Source (has distro signed by Google)
  - [https://github.com/google/santa](https://github.com/google/santa)
  - Covers five separate binaries and related concepts
Technical details

- Santa-driver
  - KAUTH_SCOPE_VNODE listener
    - File executions
    - File writes
  - KAUTH_SCOPE_FILEOP listener
    - File executions
    - File deletions
    - File renames
    - File links
    - File exchanges
- Disk mounts handled in user-space via callbacks from the DiskArbitration framework
Technical details

- Santa-driver
  - KAUTH_SCOPE_VNODE listener
    - File executions
    - File writes
  - KAUTH_SCOPE_FILEOP listener
    - File executions
    - File deletions
    - File renames
    - File links
    - File exchanges
- Disk mounts handled in user-space via callbacks from the DiskArbitration framework

Why Both?
Why Not Both?

- **KAUTH_SCOPE_VNODE**
  - Used to make block/deny decision
- **KAUTH_SCOPE_FILEOP**
  - Second is used to track process arguments and log action taken
Binary Whitelisting

- santa-driver registers itself as a KAUTH_SCOPE_VNODE listener. This flow follows how santa-driver handles KAUTH_VNODE_EXECUTE events.
- A santa-driver Kauth callback function is executed by the kernel when a process is trying to execve(). Information on where to find the executable is provided.
- santa-driver then checks if its cache has an allow or deny entry for the vnode_id/filesystem ID. If so it returns that decision to the Kauth KPI.
- If Kauth receives a deny, it will stop the execve() from taking place.
- If Kauth receives an allow, it will defer the decision. If there are other Kauth listeners, they also have a chance deny or defer.
- If there is no entry for the vnode_id in the cache a few actions occur, santad is then called upon to make the decision and communicate back to the santa-driver and stored in the cache.
- A write to a vnode_id will also invalidate a cache entry.
File Writes and Modifications

- Santa-driver registers itself as a KAUTH_SCOPE_VNODE listener. This flow then listens for:
  - KAUTH_VNODE_WRITE_DATA events.
- Santa-driver registers itself as a KAUTH_SCOPE_FILEOP listener. This flow then listens for:
  - KAUTH_FILEOP_DELETE
  - KAUTH_FILEOP_RENAME
  - KAUTH_FILEOP_EXCHANGE
  - KAUTH_FILEOP_LINK
  - KAUTH_FILEOP_CLOSE
Logging

- Santa currently logs to in plaintext to `/var/db/santa/santa.log` by default.
- All executions and disk mounts are logged here.
- File operations can also be configured to be logged. See the `FileChangesRegex` key in the `configuration.md` document.
- macOS Unified Logging System (ULS)
  - ALS and ULS are bypassed to continue logging to `santa.log`
Upvote

- Social Whitelisting
  - Hash, Cert, Signing Cert, Package, etc ...
- Policies per user
  - No host migration
- Compatible with Bit9 and Santa
- Open Source
- https://github.com/google/upvote
## Upvote

<table>
<thead>
<tr>
<th>State</th>
<th>Default Score Threshold</th>
<th>Blockable Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANNED</td>
<td>-15</td>
<td>Globally blacklisted.</td>
</tr>
<tr>
<td>SUSPECT</td>
<td>N/A</td>
<td>(Downvoted by an elevated-privilege user.) Cannot be voted on until an elevated-privilege user upvotes it.</td>
</tr>
<tr>
<td>UNTRUSTED</td>
<td>0</td>
<td>No policy set.</td>
</tr>
<tr>
<td>APPROVED FOR LOCAL_WHITELISTING</td>
<td>5</td>
<td>Users who have upvoted it are granted local whitelist policies.</td>
</tr>
<tr>
<td>GLOBALLY_WHITELISTED</td>
<td>50</td>
<td>Globally whitelisted.</td>
</tr>
<tr>
<td>Detection</td>
<td>Details</td>
<td>Behavior</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Ad-Aware</td>
<td>Trojan:MAC:Photon.A</td>
<td>ALPac</td>
</tr>
<tr>
<td>Avast</td>
<td>MacOS:Ransom-_ (T1)</td>
<td>AVG</td>
</tr>
<tr>
<td>Avira</td>
<td>OSX/Pseudo.AI</td>
<td>Emsisoft</td>
</tr>
<tr>
<td>CavaAV</td>
<td>OSX/Malware:Photon-8399539-97</td>
<td>Comodo</td>
</tr>
<tr>
<td>DNWS</td>
<td>Mac:Black:Photo:12</td>
<td>Emisoft</td>
</tr>
<tr>
<td>Emsisoft</td>
<td>malicious (high confidence)</td>
<td>Emsisoft</td>
</tr>
<tr>
<td>ESET-NOD32</td>
<td>OSX:Photon.A</td>
<td>F-Secure</td>
</tr>
<tr>
<td>Ixion</td>
<td>Trojan:Jad077771</td>
<td>Kaspersky</td>
</tr>
<tr>
<td>MAX</td>
<td>malware (md5:4G3)</td>
<td>Malware</td>
</tr>
<tr>
<td>McAfee-OWA-</td>
<td>OSX:Photon.a</td>
<td>NANO-Antivirus</td>
</tr>
<tr>
<td>Panda</td>
<td>OSX:Photo</td>
<td>Sophos</td>
</tr>
<tr>
<td>Avg</td>
<td>Clean</td>
<td>AVG</td>
</tr>
<tr>
<td>Antip-HSV</td>
<td>Clean</td>
<td>Avast</td>
</tr>
<tr>
<td>Avast Mobile Security</td>
<td>Clean</td>
<td>Avast</td>
</tr>
<tr>
<td>RAVI</td>
<td>Clean</td>
<td>RAVI</td>
</tr>
</tbody>
</table>
Away to the Log Files, It’s Handbrake, Not Flash

Michael George at Dropbox recently blogged a cool study of Santa tracking Proton Malware in the Handbrake Supply chain issue:

It Zipped Up 1Password, and CURL’d it to Stash

Then the Elves Thought of Something They Hadn’t Before

Using logs to hunt across the fleet (look beyond a hash):

2017-05-02 14:11:44.123456 | user1-macbookpro | <removed> | Virtual Interface | /Users/user1/Downloads/HandBrake-1.0.7.dmg|/Volumes/HandBrake

Compared to the legitimate HandBrake-1.0.7:

2017-05-02 13:12:34.123456 | user2-macbookpro | <removed> | Virtual Interface | /Users/user2/Downloads/HandBrake-1.0.7.dmg|/Volumes/HandBrake-1.0.7
Santa (baby), tell me where that binary’s from

```
user$ santactl fileinfo ~/Downloads/Updater.app
Path                   : /Users/user/Downloads/Updater.app
SHA-256                : 061f056338e00d38cdefb6b1f40d8e4f8d3f1d7214f6d9a48d0d91d766b7574b7
SHA-1                  : ef5a11a1bb5b2423554309688aa7947f4afa5388
Download Referrer URL  : https://mac.eltima.com/media-player.html
Download URL           : https://mac.eltima.com/download/elmediaplayer.dmg
Download Timestamp     : 2018/06/25 17:09:47 -0700
Download Agent         : com.google.Chrome
Type                   : Executable (x86_64)
Code-signed            : No
Rule                   : Blacklisted (Unknown)
```
What About Insider (and Outsider) Exfil?
What Else is on Santa’s List?

- If you’re going to install Santa for Binary Whitelisting
  - Why not use it for file system tracking as well?
- Current methods of tracking file activity often fall short
  - HFS+ and APFS
    - fsevents
      - Inconsistent on removable media
      - Timestamps
    - Journaling
      - Quickly overwritten
    - Metadata in things like spotlight
      - Inconsistent
There's *Some* Data

- Santa-driver
  - KAUTH_SCOPE_VNODE Listener
    - File executions
    - **File Writes**
  - KAUTH_SCOPE_FILEOP
    - File executions
    - File deletions
    - File renames
    - File links
    - File exchanges
  - Disk mounts handled in user-space via callbacks from the DiskArbitration framework.
There’s *Some* Data
There's *Some* Data

| FileChangesRegex* | String                                                                 | The regex of paths to log file changes. Regexes are specified in ICU format. |

Need a well crafted regex to track the writes you want, by default:

```
<key>FileChangesRegex</key>
<string>^(?!(?:private/tmp|Library/(?:Caches|Managed Installs/Logs|(?:Managed )?Preferences)))/</string>
```
There's *Some* Data

```xml
<key>FileChangesRegex</key>
<string>^/(?!(?:private/tmp|Library/(?:Caches|Managed Installs/Logs|(?:Managed ?)Preferences)))/</string>
```

Probably want to focus a bit more:

```bash
/User/* ???
/Volumes/* ???
```

You might be surprised at what is in /private:

```
# ls -l / | grep private
lrwxr-xr-x@ 1 root wheel 11 17 May 01:49 etc -> private/etc
drwxr-xr-x@ 6 root wheel 204 5 May 08:06 private
lrwxr-xr-x@ 1 root wheel 11 17 May 01:49 tmp -> private/tmp
lrwxr-xr-x@ 1 root wheel 11 17 May 01:49 var -> private/var
```
USB Tracking

Disk mounts handled in user-space via callbacks from the DiskArbitration framework.

$cat /var/db/santa/santa.log | grep APPEAR


[2018-06-25T18:10:25.634Z] I santad: action=WRITE | path=/Volumes/NO NAME/STUFF.zip | pid=1702 | ppid=1 | process=Finder | processpath=/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder | uid=402467 | user=user1 | gid=499 | group=corp

[2018-06-25T19:50:48.962Z] I santad: action=DISKDISAPPEAR | mount= | volume=NO NAME | bsdname=disk2s1
Volume Tracking - NFS

$cat /var/db/santa/santa.log | grep APPEAR

[2018-04-02T12:12:45.876Z] I santad: action=DISKAPPEAR|mount=/Volumes/backup|volume=backup|bsdname=|fs=smbfs|model=|serial=(null)|bus=|dmgpath=|appearance=2001-01-01T00:00:00.000Z]

[2018-04-02T13:34:12.344Z] I santad: action=WRITE|path=/Volumes/backup/Backup/Corp Laptop/Secret Stuff-20180309T091234Z-001.zip|pid=1702|ppid=1|process=Finder|processpath=/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder|uid=402467|user=user1|gid=499|group=corp]
Volume Tracking - CLOUD

$cat santa_processed | grep GoogleDrive

2018-03-14T02:34:23.567Z] I santad: action=DISKAPPEAR|mount=/Volumes/GoogleDrive|volume=Google Drive|bsdname=|fs=dfsfuse.DFS|model=|serial=(null)|bus=|dmgpath=|appearance=2001-01-01T00:00:00.000Z

2018-03-14 04:29:07.819000,WRITE,/Volumes/GoogleDrive/My Drive/SecretFile1.pdf,,840,/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder

2018-03-14 04:29:33.122000,WRITE,/Volumes/GoogleDrive/My Drive/SecretFile2.pdf,,840,/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder

2018-03-14 04:31:20.986000,WRITE,/Volumes/GoogleDrive/My Drive/SecretFile3.pdf,,840,/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder
File Renames Show Original Creation


Timeline

Without any disk forensics Santa can create a very easy to follow timeline.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Source</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-06-08T12:24:34.000 santa</td>
<td>/Volumes/USB DISK - SanDisk Cruzer - SDC98374539181</td>
<td>USB Mount</td>
<td></td>
</tr>
<tr>
<td>2018-06-09T15:40:52.000 santa</td>
<td>/Users/USER/.Trash/important-docs-20180528T042312Z-001.zip</td>
<td>DELETE</td>
<td></td>
</tr>
</tbody>
</table>
Spotlight UUIDs Disambiguate Cheap USBs

2017-11-23 04:12:22.113000,USB,,action=DISKAPPEAR|mount=|volume=NO NAME|bsdname=disk2s1|fs=msdos|model=General UDisk|serial=1|bus=USB|dmgpath=|

2017-12-01 18:37:12.109000,USB,,action=DISKAPPEAR|mount=|volume=Untitled|bsdname=disk16s1|fs=exfat|model=General UDisk|serial=1|bus=USB|dmgpath=|

2017-12-04 06:18:45.005000,USB,,action=DISKAPPEAR|mount=|volume=NO NAME|bsdname=disk2s1|fs=msdos|model=General UDisk|serial=1|bus=USB|dmgpath=|

2017-12-08 18:15:52.877000,USB,,action=DISKAPPEAR|mount=|volume=Untitled|bsdname=disk16|fs=msdos|model=General UDisk|serial=1|bus=USB|dmgpath=|

2017-12-12 19:06:11.107000,USB,,action=DISKAPPEAR|mount=|volume=NO NAME|bsdname=disk16|fs=msdos|model=General UDisk|serial=1|bus=USB|dmgpath=|
Spotlight UUIDs Disambiguate Cheap USBs


2017-12-04 06:20:45.000000, mymacbookpro, WRITE, /Volumes/NO NAME/.Spotlight-V100/Store-V2/34567890-1234-5678-9012-345678901234/store.db, 211, /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/Metadata.framework/Versions/A/Support/mds_stores


2017-12-12 08:12.000060, mymacbookpro, WRITE, /Volumes/NO NAME/.Spotlight-V100/Store-V2/12345678-9012-3456-7890-123456789012/store.db, 211, /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/Metadata.framework/Versions/A/Support/mds_stores
Santa’s Workshop (Other stuff)

- Anti-Forensics
  - File Deletions
  - File executions and arguments
- Signs of intent
  - Did the user copy everything to a folder named something interesting prior to zipping it up
  - Was this a normal historical workflow?
    - E.g. Downloading documents, using a USB, etc...
Enterprise Investigations

Analysis Methods

- Raw Log Review - the hard way
- Export all the logs to a database - the easy way
- Plaso Parser is coming - the timeline way
- Santactl - the live way
Mister Grinch

- Spectre Ops Santa Bypass
- Okta Santa Bypass