Minting and collecting SWID tags

Jan Pazdziora
Sr. Principal Software Engineer
Security Engineering Special Projects, Red Hat
jpazdziora@redhat.com

DEVCONF.cz

26th January 2019

Problem space

- What software is installed on machine X, VM Y, or in container Z?
  - `rpm -qa`?
- The reason for the question
  - Troubleshooting and debugging
  - Vulnerability scanning; application whitelisting
  - System management, inventory, accounting, entitlements
- Why is `rpm -q` not enough?
  - `dpkg-query -W -f='${binary:Package} ${Version}\n'`
  - `.zip`/`.jar` Java applications, single static binary in a container image
  - Higher-level collections and products and relationships among them
- XKCD 927, FTW!

What is SWID?

- SWID stands for Software Identification
- ISO/IEC 19770-2:2015 standard
  - Text is not available free of charge
  - But you are not missing much
  - It mostly just describes XML schema for XML namespace `http://standards.iso.org/iso/19770/-2/2015/schema.xsd`
- Quiz: Where would you look for that XML schema definition?

XML schema definition

- `xsi:schemaLocation=`
  `"http://standards.iso.org/iso/19770/-2/2015/schema.xsd`
  `http://standards.iso.org/iso/19770/-2/2015-current/schema.xsd"`
- It includes decent `xs:documentation`, containing most of the facts from the ISO standard
- XML file with `<SoftwareIdentity />` root element in SWID XML namespace shall be called **SWID tag**
  - Primary purpose is to describe installed software
  - And relationships
  - Also distribution/installation media (corpus tags)

Example SWID tag for distribution

- /usr/lib/swidtag/fedoraproject.org/org.fedoraproject.Fedora-29.swidtag

```
<?xml version="1.0" encoding="utf-8"?>
<SoftwareIdentity
  xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://standards.iso.org/iso/19770/-2/2015/schema.xsd http://standards.
  xml:lang="en-US"
  tagId="org.fedoraproject.Fedora-29" tagVersion="1"
  name="Fedora" version="29" versionScheme="unknown" media="(OS:linux)">
  <Entity name="Fedora Project" regid="fedoraproject.org"
                    role="tagCreator softwareCreator aggregator distributor licensor" />
  <Link rel="license" href="https://fedoraproject.org/wiki/Legal:Licenses/LicenseAgreement" /
  <Meta product="Fedora" colloquialVersion="29"
    summary="Linux distribution developed by the community-supported Fedora Project and spons
    entitlementDataRequired="false"
    unspscCode="43233004" unspscVersion="20.0601" />
</SoftwareIdentity>
```

Possible SWID tag for package

```
<SoftwareIdentity xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"
  xmlns:sha256="http://www.w3.org/2001/04/xmlenc#sha256"
  xmlns:n8060="http://csrc.nist.gov/ns/swid/2015-extensions/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://standards.iso.org/iso/19770/-2/2015/schema.xsd http://standards.
    http://csrc.nist.gov/ns/swid/2015-extensions/1.0 https://csrc.nist.gov/schema/swid/2015-e
    name="bash" tagId="org.fedoraproject.bash-4.4.23-5.fc29.x86_64"
    version="4.4.23-5.fc29.x86_64" versionScheme="rpm">
  <Entity name="Fedora Project" regid="fedoraproject.org" role="tagCreator softwareCreator"/>
  <Meta product="bash" colloquialVersion="4.4.23" revision="5.fc29" arch="x86_64"
    summary="The GNU Bourne Again shell"/>
  <Payload n8060:pathSeparator="/" n8060:envVarPrefix="$" n8060:envVarSuffix="">
    <File size="312" name=".bashrc" location="/etc/skel" n8060:mutable="true" key="true"
      sha256:hash="30a80bfce3d108d6878cf13dfb1f3a1ea15b141dbdc5bc5803f4ab40a2a39f9c"/>
    <File size="33" name="alias" location="/usr/bin" key="true"
      sha256:hash="c277897660adddce26a75871188bdae7ffa73f571a6fb779090a4c92df33988d"/>
    <File size="1190216" name="bash" location="/usr/bin" key="true"
      sha256:hash="4dbd988966d44069f3502927866825a1de1df39f814c60af32b10e06e6b7c8ea"/>
[...]
```

Additional XML namespaces

- http://www.w3.org/2001/04/xmlenc#sha256
- xsi:schemaLocation=
  "http://csrc.nist.gov/ns/swid/2015-extensions/1.0
  https://csrc.nist.gov/schema/swid/2015-extensions/swid-2015-extensions-1.0.xsd"
  - Guidelines for the Creation of Interoperable Software Identification (SWID) Tags: National Institute of Standards and Technology (NIST) Internal Report (NIST IR) 8060
    - Practical steps for implementing SWID tags
    - Still some missing areas, sometimes internally contradicting
- http://www.w3.org/2006/12/xml-c14n11
- http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
- http://www.w3.org/2000/09/xmldsig#enveloped-signature

Let's make a SWID tag

```
# dnf copr enable adelton/swid
# dnf install -y rpm2swidtag
[...]
# rpm2swidtag glibc
<?xml version="1.0" encoding="utf-8"?>
<SoftwareIdentity xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd" xmlns:sha256="http
  name="glibc" tagId="unavailable.invalid.glibc-2.28-26.fc29.x86_64"
  version="2.28-26.fc29.x86_64" versionScheme="rpm">
  <Entity name="" regid="invalid.unavailable" role="tagCreator"/>
  <Entity name="Fedora Project" regid="fedoraproject.org" role="softwareCreator"/>
  <Meta product="glibc" colloquialVersion="2.28" revision="26.fc29" arch="x86_64"
    summary="The GNU libc libraries"/>
  <Evidence date="2019-01-26T12:05:22Z" deviceId="test.example.com" n8060:pathSeparator="/" n8060:
    <File size="0" name="gai.conf" location="/etc" n8060:mutable="true"/>
    <File size="0" name="ld.so.cache" location="/etc" n8060:mutable="true"/>
    <File size="28" name="ld.so.conf" location="/etc"
      sha256:hash="239c865e4c0746a01f82b03d38d620853bab2a2ba8e81d6f5606c503e0ea379f"
      n8060:mutable="true" key="true"/>
    <Directory name="ld.so.conf.d" location="/etc"/>
    <File size="1498" name="nsswitch.conf" location="/etc"
      sha256:hash="2005fea527b6203e9e1f3f8ce5290ffa4df8ebffd50e5b592cb4502d5de2dbd3"
[...]
```

Any SWID tags around?

- After all, I have software installed on my Fedora machine

```
# swidq -a
org.fedoraproject.Fedora-29 /usr/lib/swidtag/fedoraproject.org/org.fedoraproject.Fedora-29.swidtag
+ org.fedoraproject.Fedora-29-Container /usr/lib/swidtag/fedoraproject.org/org.fedoraproject.Fedor
```

- But we can generate SWID tag inventory for installed rpms

```
# dnf rpm2swidtag regen
# swidq -a
[ ... try it and see ... ]
```

- Changes from subsequent DNF operations on rpms will be reflected
- The `rpm2swidtag --repo ...` can generate `swidtags` data for YUM/DNF repository
  - DNF `rpm2swidtag` plugin can deploy SWID tags from `swidtags` repository metadata upon package installations

XML is hard (on eyes)

- How do you make sense of that XML content?

```
$ swidq -i -n Fedora
Tag id                [org.fedoraproject.Fedora-29]
Tag version           [1]
File                  [/usr/lib/swidtag/fedoraproject.org/org.fedoraproject.Fedora-29.swidtag]
Name                  [Fedora]
Version               [29] version scheme [unknown]
Colloquial version    [29]
XML language          [en-US]
Edition               [Container] (+)
Product               [Fedora]
Entitlement required [false]
Summary               [Linux distribution developed by the community-supported Fedora Project and s
United Nations Standard Products and Services Code [43233004]
Media                 [(OS:linux)]
Entity [tagCreator softwareCreator aggregator distributor licensor] regid [fedoraproject.org]
                                                                     name [Fedora Project]
Link [license] to [https://fedoraproject.org/wiki/Legal:Licenses/LicenseAgreement]
Link [component] to [swid:com.example.test.mailcap-2.1.48-4.fc29.noarch] (*)
Link [component] to [swid:com.example.test.mutt-5:1.10.1-1.fc29.x86_64] (*)
[...]
```

XML is hard (on eyes) (cont'd)

```
$ swidq -i -n mutt
Tag id                [com.example.test.mutt-5:1.10.1-1.fc29.x86_64]
File                  [/var/lib/swidtag/rpm2swidtag-generated/com.example.test.mutt-5:1.10.1-1.fc2
Name                  [mutt]
Version               [5:1.10.1-1.fc29.x86_64] version scheme [rpm]
Colloquial version    [1.10.1]
Revision              [1.fc29]
Architecture          [x86_64]
[...]
RPM resource          [mutt-5:1.10.1-1.fc29.x86_64]
Entity [tagCreator] regid [test.example.com] name []
Entity [softwareCreator] regid [fedoraproject.org] name [Fedora Project]
Evidence gathered at [2019-01-26T12:08:15Z] from [test.example.com]
---
Tag id                [com.example.test.mutt-5:1.10.1-1.fc29.x86_64-component-of-org.fedoraproject
Tag is supplemental
Supplemental to       [swid:org.fedoraproject.Fedora-29]
File                  [/var/lib/swidtag/rpm2swidtag-generated/com.example.test.mutt-5:1.10.1-1.fc2
Name                  [mutt]
Entity [tagCreator] regid [test.example.com] name []
Entity [softwareCreator] regid [fedoraproject.org] name [Fedora Project]
Link [component] to [swid:com.example.test.mutt-5:1.10.1-1.fc29.x86_64]
```

Some implementation notes

- The `@tagId` could be just UUID but I like to know what I'm looking at
  - On the other hand, the attribute value should bear no semantics
- Tag creator vs. software creator
- We use `@arch` for architecture (`<xs:anyAttribute processContents="lax"/>`)
- `Link` elements can reference other SWID tags, as well as generic URIs
  - Used for relationships
  - Should package point to higher-level component, or vice versa?
- Special relationship type is `supplemental`, for amending other tag(s)
  - Powerful but potentially confusing
  - Is the `@name` value from the `com.example.test.mutt-5:1.10.1-1.fc29.x86_64-component-of-org.fedoraproject.Fedora-29` example correct?

Listing SWID tags on system

- The standard and NIST IR 8060 assume /swidtag or

```
# find / -name '*.swidtag'
```

- To be reasonably efficient, we propose

```
/etc/swid/swidtags.d/*/*.swidtag
```

  - Entries in /etc/swid/swidtags.d are symlinks to directories where .swidtag files are expected to live
  - With the goal of supporting non-rpm (.zip, .jar) installations easily
    - While not scanning all the disks to list SWID tags on system
  - This is where swidq looks
    - But -p can point it to any location

Signed SWID tags

- Enveloped XML signatures
  - `rpm2swidtag's --sign-pem ...` option invokes `xmlsec1 --sign ...`
- TagVault.org's SWID Tag Signing Guidelines seem to require full secure-timestamped XAdES-T signatures
  - But what will be the root of trust?
- We propose `/etc/pki/swid/CA/<tag-creator-regid>` as location for CA certificates for local validation
  - Scanning tools can of course use their own trust database
  - Red Hat published its code signing key at `access.redhat.com/security/team/key`

Near-term plans

- The XML namespace for the DNF/YUM repository `swidtags` metadata
- Publishing our practices (like the use of `/etc/swid/swidtags.d` or supplemental `component`) in repo under `github.com/swidtags`
  - To make discussion easy, via issues or pull requests
- Implementing libdnf plugin as well
- Getting the tools (`rpm2swidtag` and plugins, `swidq`) to Fedora 30
- Signature validation, probably in `swidq`
- Relationship tree display, probably in `swidq`
- Fetching SWID tags from YUM/DNF repo metadata for already installed packages

Welcoming contributions

- Opinions of people who looked at SWID are valuable
  - Really, we'd love to discuss the details with people with strong opinions
    - Should the `@tagId` for rpms be `NEVRA` or `NEVRA.rpm`?
    - The use of `@arch`
    - Symlinks in `Payload/Evidence`, file ownership and mode
    - Our use of `Resource` for `rpm` and `rpm-signature` types
    - ...
- Real-life use-cases
- Extending the tools to produce SWID tags for other package formats, including `.zip` or unpacked directories

References

- github.com/swidtags/rpm2swidtag
- copr.fedorainfracloud.org/coprs/adelton/swid/
- standards.iso.org/iso/19770/-2/2015-current/schema.xsd
- csrc.nist.gov/publications/detail/nistir/8060/final
- csrc.nist.gov/Projects/Software-Identification-SWID/resources#swid-validation-tool
- Repo documenting practices, under github.com/swidtags (to be created)

- Feedback time: how useful was this session? devconfcz2019.sched.com/event/Jcil/minting-and-collecting-swid-tags