strongTNC REST API
SWID Endpoints

Im Rahmen der strongTNC BA

Danilo Bargen, Christian Fässler, Jonas Furrer

6. Juni 2014

Inhaltsverzeichnis

1 Ablauf der Messung

Eine Messung läuft wie folgt ab:

1. Auf dem Client werden die Software-IDs aller installierter Pakete generiert.
2. Die Software-IDs werden als JSON-Liste an den entsprechenden API-Endpoint [4.1] gesendet.
   - Wenn von der API "`200 OK`" zurückgegeben wurde, ist alles OK und die SWID Tags wurden mit der aktuellen Session verknüpft. ENDE.
   - Wenn von der API "`404 Not Found`" zurückgegeben wurde, ist die Session ID ungültig. Problem beheben und weiter zu Punkt 2.
   - Wenn von der API "`412 Precondition Failed`" zurückgegeben wurde, sind noch nicht alle SWID Tags in der Datenbank erfasst. Weiter zu Punkt 3.
3. Die fehlenden SWID-Tags werden im Response-Body als JSON-Liste zurückgegeben. Diese werden nun über Targeted Requests generiert.
4. Die neu generierten Tags werden als JSON-Listen an den entsprechenden API-Endpoint [4.2] gesendet. Es können auch mehrere Tags in einem Request mitgesendet werden, empfehlenswert ist allerdings ein Batching in mehrere Gruppen, da die Requests sonst unter Umständen sehr lange dauern können.
   - Wenn von der API "`200 OK`" zurückgegeben wurde, ist alles OK. Weiter zu Punkt 5.
   - Wenn von der API "`400 Bad Request`" zurückgegeben wurde, stimmt etwas mit dem Request nicht. Fehler-Details werden im Response Body zurückgesendet.
5. Nachdem alle fehlenden SWID-Tags eingetragen wurden, kann der Measurement-Request erneut gesendet werden. Weiter zu Punkt 1.

2 Beispiel Szenario

Im Folgenden soll der Ablauf einer Messung anhand eines konkreten Beispiels illustriert werden. In diesem Beispiel soll eine Messung für 3 Tags mit folgenden Software-IDs erfolgen:

- `regid.2004-03.org.strongswan_Ubuntu_12.04-i686-logrotate-3.7.8-6ubuntu5`
- `regid.2004-03.org.strongswan_Ubuntu_12.04-i686-lsb-base-4.0-0ubuntu20`
- `regid.2004-03.org.strongswan_Ubuntu_12.04-i686-strongswan-4.5.2-1.5+deb7u3`

2.1 Request - SWID Measurement

2.1.1 Request

Zuerst wird versucht eine Messung für die Session mit der ID 2 zu erfassen.

```
POST /api/sessions/2/swid-measurement/ HTTP/1.1
Authorization: Basic cm9vdDpyb290
Host: tncserver:8000
Accept: application/json
Content-Type: application/json; charset=utf-8
Content-Length: 232

[
"regid.2004-03.org.strongswan_Ubuntu_12.04-i686-logrotate-3.7.8-6ubuntu5",
"regid.2004-03.org.strongswan_Ubuntu_12.04-i686-lsb-base-4.0-0ubuntu20",
"regid.2004-03.org.strongswan_Ubuntu_12.04-i686-strongswan-4.5.2-1.5+deb7u3"
]
```

Curl Befehl:

```
curl -i -X POST http://tncserver:8000/api/sessions/2/swid-measurement/ \
                -u username:password \
                -H "Accept: application/json" \
                -H "Content-Type: application/json; charset=utf-8" \
                -d "$DATA"
```

2.1.2 Antwort - Status 412 Precondition Failed

Sind noch nicht alle Tags in der strongTNC Datenbank vorhanden, werden die Software-IDs der fehlenden Tags zurückgeliefert. Der HTTP Status Code ist "`412 Precondition Failed`". Es werden zu diesem Zeitpunkt noch keine Tags mit der Session verknüpft. Eine entsprechende Antwort kann wie folgt aussehen:

```
HTTP/1.0 412 PRECONDITION FAILED
Date: Wed, 14 May 2014 15:21:45 GMT
Server: WSGIServer/0.1 Python/2.7
Vary: Accept, Cookie
Content-Type: application/json
Allow: POST, OPTIONS

["regid.2004-03.org.strongswan_Ubuntu_12.04-i686-strongswan-4.5.2-1.5+deb7u3"]
```

2.1.3 Antwort - Status 200 OK

Wären für alle Software-IDs bereits Tags in der Datenbank vorhanden, könnte die Antwort wie folgt aussehen:

```
HTTP/1.0 200 OK
Date: Wed, 14 May 2014 15:21:45 GMT
Server: WSGIServer/0.1 Python/2.7
Vary: Accept, Cookie
Content-Type: application/json
Allow: POST, OPTIONS

[]
```

2.2 Nacherfassen von SWID Tags

2.2.1 Request

War die Messung nicht erfolgreich (HTTP 412), müssen für die zurückgelieferten Software-IDs zuerst komplette SWID Tags erfasst werden.

Im swidGenerator kann ein solcher wie folgt abgefragt werden:

```
swid-generator swid \
  --software-id="regid.2004-03.org.strongswan\_Ubuntu\_12.04-i686-fortune-mod-1:1.99.1-4"
```

Ein solcher SWID Tag sieht wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<SoftwareIdentity name="strongswan"
        uniqueId="debian_7.4-x86_64-strongswan-4.5.2-1.5+deb7u3"
        version="4.5.2-1.5+deb7u3" versionScheme="alphanumeric"
        xmlns="http://standards.iso.org/iso/19770/-2/2014/schema.xsd">
  <Entity name="strongSwan" regid="regid.2004-03.org.strongswan" role="tagcreator"/>
  <Entity name="HSR" regid="regid.2004-03.org.strongswan" role="publisher"/>
  <Payload>
    <File location="/usr/sbin" name="charon-cmd"/>
```

```
    <File location="/usr/sbin" name="strongswan"/>
    <File location="/usr/lib/systemd/system" name="strongswan.service"/>
    <File location="/usr/lib64/strongswan" name="libcharon.so.0"/>
    <File location="/usr/lib64/strongswan" name="libcharon.so.0.0.0"/>
    <File location="/usr/lib64/strongswan" name="libhydra.so.0"/>
    <File location="/usr/lib64/strongswan" name="libhydra.so.0.0.0"/>
    <File location="/usr/lib64/strongswan" name="libpttls.so.0"/>
    <File location="/usr/lib64/strongswan" name="libpttls.so.0.0.0"/>
    <File location="/usr/lib64/strongswan" name="libstrongswan.so.0"/>
    <File location="/usr/share/doc/strongswan-5.1.2" name="README"/>
  </Payload>
</SoftwareIdentity>
```

Dieser Tag kann anschliessend in der strongTNC Datenbank mittels POST Request auf die Ressource `/swid/add-tags/` erfasst werden. Die Daten für diesen Request bestehen aus einer JSON Liste von XML Strings, dadurch können auch mehrere Tags auf einmal erfasst werden. Empfehlenswert ist allerdings ein Batching in mehrere Gruppen, da die Requests sonst unter Umständen sehr lange dauern können.

Falls bereits ein Tag mit der selben Software-ID existiert, wird er komplett überschrieben.

Der Request sieht folgendermassen aus:

```
POST /api/swid/add-tags/ HTTP/1.1
Authorization: Basic cm9vdDpyb290
Host: tncserver:8000
Accept: application/json
Content-Type: application/json; charset="utf-8"
Content-Length: 239

[
"<?xml version=\"1.0\" encoding=\"utf-8\"?><SoftwareIdentity name=\"fortune-mod\"...",
"<?xml version=\"1.0\" encoding=\"UTF-8\"?><SoftwareIdentity name=\"strongswan\"..."
]
```

Curl Befehl:

```
curl -i -X POST http://tncserver:8000/api/swid/add-tags/ \
                -u username:password \
                -H "Accept: application/json" \
                -H "Content-Type: application/json; charset=utf-8" \
                -d "$DATA"
```

2.2.2 Antwort - Status 400 Bad Request

Falls der Request nicht in Ordnung ist (zB fehlerhaftes oder unvollständiges XML), könnte die Antwort wie folgt aussehen:

```
HTTP/1.0 400 BAD REQUEST
Date: Thu, 15 May 2014 21:31:10 GMT
Server: WSGIServer/0.1 Python/2.7.6
Vary: Accept, Cookie
Content-Type: application/json
Allow: POST, OPTIONS

{"status": "error", "message": "version: This field cannot be blank.
package_name: This field cannot be blank."}
```

2.2.3 Antwort - Status 200 OK

Wenn mit dem Request alles in Ordnung ist, könnte die Antwort wie folgt aussehen:

```
HTTP/1.0 200 OK
Date: Thu, 15 May 2014 21:19:19 GMT
Server: WSGIServer/0.1 Python/2.7.6
Vary: Accept, Cookie
Content-Type: application/json
Allow: POST, OPTIONS

{"status": "success", "message": "Added 1 SWID tags, replaced 0 SWID tags."}
```

3 Allgemeine Hinweise

- JSON-Objekte dürfen im Gegensatz zu Python nur in doppelte Anführungszeichen eingeschlossen werden. Beispiel: {`"foo": "bar"`} statt {`'foo': 'bar'`}.
- Alle übermittelten Daten sollten UTF-8 encoded sein. Der `Content-Type` Header sollte entsprechend übermittelt werden. Beispiel: `Content-Type: application/json; charset=utf-8`
- Nichtdruckbare Zeichen sowie doppelte Anführungszeichen innerhalb eines JSON Strings (zB bei den XML Tags) müssen mit einem Backslash escaped werden (z.B newline, tab, double quotes → `\n`, `\t`, `\"`).

4 API Endpoints

4.1 SWID Tags: Messung

URI Path `/session/{id}/register-swid-measurement/`
Archetype Controller
Methods POST
Content-Type `application/json; charset=utf-8`
Request Parameter

`softwareId` Software-IDs als JSON-Liste.

Response Statuscodes

200 OK SWID Tags der übermittelten Software-IDs sind eingetragen und wurden für die übermittelte Session eingetragen

404 Not Found Session mit der spezifizierten ID wurde nicht gefunden.

412 Precondition Failed Es existieren nicht alle SWID Tags für die übertragenen Software-IDs. Als Payload werden die fehlenden Software-IDs übertragen.

JSON Format Response

```
["<str, software-id>", ...]
```

4.2 SWID Tags: Erstellung

URI Path `/swid/add-tags/`
Archetype Controller
Methods POST
Content-Type `application/json; charset=utf-8`
Request Parameter

`xmlData` SWID Tags als JSON-Liste.

Response Statuscodes

200 OK SWID Tags wurden erfolgreich verarbeitet.

400 Bad Request Fehlerhafter Request. Details zum Fehler werden im Response-Body zurückgesendet.