Eigene CA bauen und Zertifikate ausrollen

Fix Me! Grundegedanke war hier die ganze Hürde mit gekauften Zertifikaten zu umgehen, und das ganze noch einfacher zu gestalten. Für Organisationen intern, aber auch extern. Mit Installation von deinem DEB Paket in Ubuntu, und in Windows mit einem MSI. Wir erstellen eine eigene CA mit einem Wildcardzertifikat. Somit benötigen wir pro Domäne nur ein Zertifikat und können dies auf allen Geräten und Rechnern installieren. Daraus kann man natürlich auch Anmeldezertifikate (p12) erstellen, um die Sicherheit in relevanten Seiten hinauf zu schrauben.

Erstellen der eigenen CA

Der Vorgang wird von dieser Seite hier beschrieben: https://wiki.ubuntuusers.de/CA/

Fix Me!

Für's erste hier mal einige nützliche OpenSSL-Befehle:

Zertifikat anzeigen lassen:

```
openssl x509 -in /etc/ldap/ssl/01cacert.pem -noout -text
```

Auflisten aller Zertifikate im globalen CA-Speicher von Java:

```
keytool -list -keystore /etc/ssl/certs/java/cacerts
```

Firefoxcertifikate anzeigen lassen:

```
NSS_DEFAULT_DB_TYPE="sql" certutil -d ~/.mozilla/firefox/*.default -L
```

Firefoxcertifikat hinzufügen:

```
certutil -A -d sql:$HOME/.mozilla/firefox/2gjkcvvk.default  -i cacert.pem -n
"tux.at Wildcard Selfsigned from ITEAS IT Services" -t TCP,TCP,TCP
```

Firefoxcertifikat löschen:

```
certutil -D -d sql:$HOME/.mozilla/firefox/2gjkcvvk.default  -i cacert.pem -n
"tux.at Wildcard Selfsigned from ITEAS IT Services" -t TCP,TCP,TCP
```

CA Zertifikat in Chrome/Chromium importieren

```
ertutil -d sql:$HOME/.pki/nssdb -A -n '' -i cacert.pem -t TCP,TCP,TCP
```

Alle Zertifiakte des Systems anzeigen:

```
locate .pem | grep "\.pem$" | xargs -I{} openssl x509 -issuer -enddate -
```

```
noout -in {}
```

PFX Inhalt anzeigen:

```
openssl pkcs12 -info -in testcert.pfx
```

Vorbereiten des Servers

Wir verwenden hier ein Ubuntu 18.04 als CA Server. Sehr gut eigent sich auch ein LXC-Container oder Docker. Auch kann man die natürlich überall wo es passt dazu installieren. Folgende Pakete sollten installiert sein.

```
apt-get install ca-certificates ca-certificates-java ca-certificates-mono
openssl gnutls-bin libnss3-tools
```

Nun passen wir noch unsere `/etc/ssl/openssl.cnf` an. Mit diesem File kann man das meiste komplett automatisieren und vorallem für Google Chrome anpassen. Google Chrome ist der einzige Browser der spezielle Richtlinien für das Zertifikat verwendet. Das File nach eigenem Ermessen abändern.

Thread about

openssl.cnf

```
...
[ CA_default ]
...
default_days    = 3650                  # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = default               # use public key default MD
preserve        = no                    # keep passed DN ordering
...
# Extension copying option: use with caution.
# copy_extensions = copy
...
[ v3_req ]
...
subjectAltName = @alt_names
...
[ alt_names ]
DNS.1 = *.osit.cc
DNS.2 = localhost
DNS.3 = ip6-localhost
IP.1 = 127.0.0.1
IP.2 = ::1
...
[ v3_ca ]
...
subjectAltName = @alt_names
```

```
...
```

Alle anderen Anpassungen im File sind zwecks Automatisierung empfohlen. Hier das ganze File zum Download: openssl.cnf

CA global in Ubuntu einspielen

Das CA ganz einfach unter `/usr/local/share/ca-certificates/` ablegen. Mit dem nachfolgenden Befehl wird das Zertifikat in den globalen Zertifikatsspeicher des Systems importiert.

```
update-ca-certificates
```

Alternativ interaktiv: `dpkg-reconfigure ca-certificates`

Automatische Installation in Firefox (Linux und Windows)

Folgendes Tool muss in Ubuntu installiert werden.

```
apt install libnss3-tools
```

Policies in Firefox (Autoimport global)

Mit einem File ist es möglich sämtlich Dinge wie Berechtigungen, fixe Bookmarks und vieles vieles mehr dem System für alle Benutzer vor zu geben. Man kann Dinge Sperren, und auch Zertifikate ausrollen. Zertifikate funktionieren mit fully qualified path erst ab Firefox Version 65. Das gilt auch für Windows.

Beispielfile, Syntax getestet auf KDE Neon 18.04 LTS

```
cat /usr/lib/firefox/distribution/policies.json
```

policies.json

```
{

 "policies": {
   "BlockAboutAddons": true,
   "BlockAboutConfig": true
 }
}
```

Beispiel zwei, mit Zertifikat:

policies.json

```
{

 "policies": {
   "BlockAboutAddons": true,
   "BlockAboutConfig": true,
      "Certificates": {
      "ImportEnterpriseRoots": true,
      "Install": ["/usr/local/share/ca-certificates/osit.cc-wildcard-
selfsigned-cacert.crt"]
    }
 }
}
```

Und noch ein Beispiel:

policies.json

```
{
   "policies":{
      "DisableBuiltinPDFViewer":true,
      "Extensions":{
         "Install":[
"https://addons.mozilla.org/firefox/downloads/file/1672871/ublock_origi
n-*.xpi",
"https://addons.mozilla.org/firefox/downloads/file/879506/ip_address_an
d_domain_information-*.xpi",
"https://addons.mozilla.org/firefox/downloads/file/1205950/keepassxc_br
owser-*.xpi"
         ]
      },
      "Certificates":{
         "ImportEnterpriseRoots":true,
         "Install":[
            "/usr/local/share/ca-certificates/osit.cc-wildcard-
selfsigned-cacert.crt",
            "/usr/local/share/ca-certificates/osit.cc-
Fortinet_CA_SSL_deepinspection.crt"
         ]
      }
   }
}
```

Man kann auch Zertifikate in dem Verzeichnis des Benutzers ablegen `~/.mozilla/certificates`. Diese werden dann ohne den gesamten Pfad unter „Install“ vermerkt und auch automatisch installiert. Der globale Zertifikatsordner von Firefox in Ubuntu `/usr/share/ca-certificates/mozilla` funktioniert aus unverfindlichen Gründen nicht. Daher die Empfehlung

immer den fully qualified path verwenden.

Beispielfile, Syntax getestet auf Windows 10

```
cat C:\Program Files\Mozilla Firefox\distribution\policies.json
```

policies.json

```
{

 "policies": {
   "BlockAboutAddons": true,
   "BlockAboutConfig": true,
      "Certificates": {
      "ImportEnterpriseRoots": true,
      "Install": ["C:\\Company\\bla.crt", "bla.crt"]
    }
 }
}
```

Auch hier gibt es zwei Möglichkeiten. Einmal ein fully qualified path, oder im Ordner `C:\Program Files\Mozilla Firefox\distribution\certificates`. Auch hier funktioniert das erst ab Firefox Version 65.

Automatische Installation in Google Chrome, Chromium (Linux und Windows)

Policies in Chrome (Autoimport global)

Fertige Pakete für die Installation der CA am Client (Linux und Windows)

Import in Android

Fix Me!
https://forum.xda-developers.com/google-nexus-5/help/howto-install-custom-cert-network-t2533550 (Methode 1)

Links

- Komplette Doku gibt es hier
- Mozillabeitrag Windows10
- Mozillabeitrag Ubuntu Linux
- https://thomas-leister.de/ca-zertifikat-importieren-linux-windows/

In Windows muss der Ordner Stammzertifizierungsstellen für das CA manuel ausgewählt werden um in Edge oder IE zu trusten. Das trusted wie es aussieht auch Chrome und Firefox, das wird aber nochmal getestet.