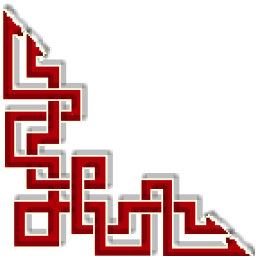
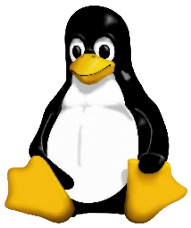
RF-232

Micronator

Serveur SME-9.1
&
Certificat Let's Encrypt





Version: 0.1.1 / mercredi 20 juillet 2016 - 21:58

AVIS DE NON-RESPONSABILITÉ

Ce document est uniquement destiné à informer. Les informations, ainsi que les contenus et fonctionnalités de ce document sont fournis sans engagement et peuvent être modifiés à tout moment. ***RF-232*** n'offre aucune garantie quant à l'actualité, la conformité, l'exhaustivité, la qualité et la durabilité des informations, contenus et fonctionnalités de ce document. L'accès et l'utilisation de ce document se font sous la seule responsabilité du lecteur ou de l'utilisateur.

RF-232 ne peut être tenu pour responsable de dommages de quelque nature que ce soit, y compris des dommages directs ou indirects, ainsi que des dommages consécutifs résultant de l'accès ou de l'utilisation de ce document ou de son contenu.

Chaque internaute doit prendre toutes les mesures appropriées (*mettre à jour régulièrement son logiciel antivirus, ne pas ouvrir des documents suspects de source douteuse ou non connue*) de façon à protéger le contenu de son ordinateur de la contamination d'éventuels virus circulant sur la Toile.

Toute reproduction interdite

Avertissement

Bien que nous utilisions ici un vocabulaire issu des techniques informatiques, nous ne prétendons nullement à la précision technique de tous nos propos dans ce domaine.

Sommaire

I- Description générale.....5
1. Introduction.....5
2. Clients Let's Encrypt.....5
3. Marche à suivre.....5
4. Particularités de ce document.....6
5. Commentaires et suggestions.....7
6. Boutique de Micronator.....7

II- Glossaire.....8

III- Let's Encrypt.....13
1. Principe de fonctionnement de Letsencrypt.....13
2. Courriels du certificat.....13
3. Transparence des certificats.....13
4. Limites.....13
5. Mode Officiel vs TEST (staging).....14
6. Clé de compte Let's Encrypt.....15
7. Aide.....16
8. En cas de trouble majeur avec un certificat.....16
9. Paramètres.....16

IV- Prérequis.....17
1. Conditions préalables.....17

V- Installation du client.....18
1. Description.....18
2. Installation.....18

VI- Création des fichiers et répertoires requis.....20
1. Répertoire des défis.....20
2. Gabarit personnalisé.....20
3. Fichiers de configuration.....21
4. Script de point d'entrée.....24
5. Sauvegarde.....25

VII- Demande d'un certificat de TEST.....27
1. Introduction.....27
2. Affichage de l'aide.....27
3. Fichier de configuration.....28
4. Sauvegarde.....28
5. Demande du certificat.....29
6. Vérification.....30
7. Conclusion.....35

VIII- Renouvellement.....36
1. Manuel.....36
2. Manuel forcé.....36
3. Automatique.....41

IX- Demande d'un certificat officiel.....46
1. Introduction.....46
2. Manuel.....46
3. Manuel forcé.....47
4. Vérification du nouveau certificat.....48
5. Conclusion.....58

X- Renouvellement.....59
1. Introduction.....59
2. Manuel.....59
3. Manuel forcé.....60
4. Automatique.....60

XI- Sauvegarde du répertoire /etc/letsencrypt.sh.....63
1. Introduction.....63
2. Création du gabarit personnalisé.....63
3. Vérification.....64

XII- Révocation.....66
1. Introduction.....66
2. Affichage des certificats actuels.....66
3. Certificat officiel.....66
4. Certificat de TEST.....68

XIII- Certificat standard SME.....70
1. Introduction.....70
2. Login.....70
3. Création d'un répertoire de sauvegarde.....70
4. Sauvegarde des fichiers du certificat actuel.....71
5. Effaçage des propriétés de modSSL.....72
6. Signalisation.....73
7. Vérification.....73
8. Crédits.....75

I- Description générale

1. Introduction

Ce document explique la marche à suivre pour installer un certificat **SSL** émis par l'autorité de certification **Let's Encrypt**.

Ce document s'est inspiré de la contribution **Letsencrypt** produite par **Flep**, **Hfwang**, **DanB35** et **Brianr**. Vous pouvez consulter cette contribution à la page **https://wiki.contribs.org/Letsencrypt**.

1.1. Version 0.1.0

Le fichier de configuration du client **letsencrypt.sh** a été changé de **config.sh** à **config**. Nous avons ajusté cette documentation pour en tenir compte.

La plupart des dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

Référence: **https://fr.wikipedia.org/wiki/Let's_Encrypt**.

Let's Encrypt est une autorité de certification lancée le 3 décembre 2015 (*Bêta Version Publique*). Cette autorité fournit des certificats gratuits **X.509** pour le protocole cryptographique **TLS** au moyen d'un processus automatisé destiné à se passer du processus complexe actuel impliquant la création manuelle, la validation, la signature, l'installation et le renouvellement des certificats pour la sécurisation des sites Internet.

2. Clients Let's Encrypt

Les deux principaux clients **Let's Encrypt** généralement utilisés:

- **letsencrypt.sh** est le client préféré de ceux qui préfèrent un client léger ne nécessitant aucune dépendance et par les utilisateurs de **Serveurs SME-8.x**.
- Le client officiel de **letsencrypt.org** est assez complet mais requiert un certain nombre de dépendances avant d'être installé. Il exige aussi une version plus récente de **Python** que celle incluse avec une installation standard d'un **Serveur SME**. Les **Serveurs SME-9.x**, dans les versions **64 bits**, résolvent ce problème par l'utilisation des logiciels **Collections** qui permettent une installation telle **Python 2.7** à côté de l'installation par défaut de **Python 2.6**. L'avantage de ce client est qu'il peut être utilisé pour demander un certificat pour un serveur privé sur un intranet.

Nous examinerons ce client dans un prochain document.

3. Marche à suivre

Utilisant un **Serveur SME-9.1**, nous allons démontrer l'utilisation du client **letsencrypt.sh**.

Tous nos essais seront effectués pour un certificat **multi-domaines**. Nous utiliserons deux domaines complètement différents: **micronator.org** et **ainesmercierouest.info**, tous deux hébergés sur notre **Serveur SME-9.1**.

- Nous allons d'abord utiliser le **client letsencrypt.sh** pour l'installation de notre premier certificat de test et son renouvellement manuel. Nous vérifierons l'installation du certificat à l'aide de différents navigateurs Web. Nous construirons une tâche **cron** de test pour le renouvellement automatique.
- Une fois ces manipulations vérifiées, nous utiliserons le client **letsencrypt.sh** pour une demande de certificat

officiel et exécuterons les mêmes procédures. Encore une fois, nous vérifierons l'installation du certificat officiel à l'aide de différents navigateurs Web.

- Nous reprendrons la création de la tâche **cron** de test et l'adapterons pour une utilisation définitive. Tous les mois et sans aucune intervention de notre part, la tâche **cron** renouvellera automatiquement le certificat s'il lui reste moins de 30 jours de validité. À chaque lancement de la tâche **cron**, un courriel sera envoyé à l'usager **admin** (***root***).
- Nous créerons un gabarit personnalisé pour indiquer à la sauvegarde standard du **Serveur SME-9.1**, d'inclure le répertoire **/etc/letsencrypt** et ses sous-répertoires.
- Nous terminerons en indiquant la marche à suivre pour recréer un certificat standard **auto-signé** par le **Serveur SME**.

4. Particularités de ce document

4.1. Notes au lecteur

* Les captures d'écrans ne sont que des références.

** Les informations écrites ont préséance sur celles retrouvées dans les captures d'écrans. Veillez vous référer aux différents tableaux lorsque ceux-ci sont présents.

4.2. Conventions

Toutes les commandes à entrer à la console sont en **gras**. Les affichages à surveiller sont en **rouge**, **bleu**, **orange** ou **magenta**.

```
# ping 192.168.1.149
192.168.1.149 is alive
#
```

Les liens de référence Internet sont en **bleu** et ceux intra document en ***bleu***.


Manipulation, truc ou ruse pour se tirer d'embarras.


Une recommandation ou astuce.


Une note.


Une étape, note ou procédure à surveiller.


Paragraphe non complété ou non vérifié.


Cette icône indique que cette commande est sur une seule ligne. Le **PDF** la mettra sur deux lignes avec un [**CR**] [**LF**] entre les deux. Il faudra donc copier la commande entière dans un éditeur de texte ASCII et la mettre sur une seule ligne avant de la copier à la console.

Certaines commandes telles celles créant un fichier et son contenu peuvent être très longues et s'étendre sur plus d'une page. Vérifiez le contenu après la copie de la commande, car à partir d'un **PDF**, **une copie inclut aussi les en-têtes et les pieds de page** qu'il faut éliminer avant de lancer la dite commande.

Une **chaîne de caractères en magenta** indique qu'il faut remplacer cette chaîne par vos propres paramètres.

5. Commentaires et suggestions

RF-232 apprécie énormément échanger avec ses internautes. Vos commentaires et suggestions sont indispensables à l'amélioration de la documentation et du site **micronator.org**.

N'hésitez pas à nous transmettre vos commentaires et à nous signaler tout problème d'ordre technique que vous avez rencontré ou n'arrivez pas à résoudre. Tous vos commentaires seront pris en considération et nous vous promettons une réponse dans les plus brefs délais.

Brancher les aînés,
encourager l'Informatique Libre
et la diffusion du savoir

6. Boutique de Micronator

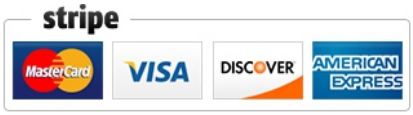
Nous sommes heureux de vous présenter notre boutique en ligne dans laquelle vous trouverez certains de nos produits qui ne sont pas disponibles sur notre site principal. Nous vous laissons le plaisir de la parcourir: **https://www.micronator.org/?post_type=product**.

Communications sécuritaires chiffrées SSL

Les communiquons avec **Stripe** et **PayPal** sont effectuées au moyen d'un **certificat SSL de 2048** bits émis par l'Autorité de Certification **Let's Encrypt**.

Faites vos achats en toute confiance, remplissez votre panier et réglez votre commande avec la carte bancaire de votre choix, **MasterCard**, **Visa**, **Discover**, **American Express**, etc.

Stripe



Vos données sont directement envoyées à **Stripe** qui s'occupe de tout et votre carte n'est pas conservée sur notre site. Les paiements sont sécurisés par le système **Stripe**. **Cliquez ici** pour voir les étapes de paiements; celles-ci sont sécurisées par le système **Stripe**.

PayPal

Il n'est pas nécessaire d'ouvrir un compte **PayPal**. Vous pouvez choisir la carte bancaire que vous désirez utiliser. **Cliquez ici** pour voir les étapes de paiements; celles-ci sont sécurisées par le système **PayPal**.

II- Glossaire

Ce chapitre rassemble quelques termes pour permettre une brève introduction à la cryptographie.

Cryptographie asymétrique

Référence: **https://fr.wikipedia.org/wiki/Cryptographie_asym%C3%A9trique**.

La cryptographie asymétrique, ou cryptographie à clé publique, est une méthode de chiffrement qui s'oppose à la cryptographie symétrique. Elle repose sur l'utilisation d'une clé publique (*qui est diffusée*) et d'une clé privée (*gardée secrète*), la première permettant de coder le message et la seconde de le décoder. Ainsi, l'expéditeur peut utiliser la clé publique du destinataire pour coder un message que seul le destinataire (*en possession de la clé privée*) peut décoder, garantissant la confidentialité du contenu.

Inversement, l'expéditeur peut utiliser sa propre clé privée pour coder un message que le destinataire peut décoder avec la clé publique de l'expéditeur; c'est le mécanisme utilisé par la signature numérique pour authentifier l'auteur d'un message.

Chiffrement

L'un des rôles de la clé publique est de permettre le chiffrement; c'est donc cette clé qu'utilisera Bob pour envoyer des messages chiffrés à Alice. L'autre clé — l'information secrète — sert à *dé*chiffrer. Ainsi, Alice, et elle seule, peut prendre connaissance des messages de Bob. La connaissance d'une clé ne permet pas de déduire l'autre.

Échange de clés Diffie-Hellman

Référence: **https://fr.wikipedia.org/wiki/%C3%89change_de_cl%C3%A9s_Diffie-Hellman**.

En cryptographie, l'échange de clés Diffie-Hellman, du nom de ses auteurs Whitfield Diffie et Martin Hellman, est une méthode par laquelle deux agents nommés conventionnellement Alice et Bob peuvent se mettre d'accord sur un nombre (*qu'ils peuvent utiliser comme clé pour chiffrer la conversation suivante*) sans qu'un troisième agent appelé Ève puisse découvrir le nombre, même en ayant écouté tous leurs échanges.

Somme de contrôle

Référence: **https://fr.wikipedia.org/wiki/Somme_de_contr%C3%B4le**.

La somme de contrôle ou checksum en anglais, parfois appelée "empreinte", est un nombre qu'on ajoute à un message à transmettre pour permettre au récepteur de vérifier que le message reçu est bien celui qui a été envoyé. L'ajout d'une somme de contrôle à un message est une forme de contrôle par redondance.

Nonce

Référence: **https://fr.wikipedia.org/wiki/Nonce_cryptographique**.

Le nonce est un nombre arbitraire, à usage unique, utilisé pour signer un ensemble de données d'une communication électronique. Il permet notamment d'éviter les attaques de type "**Attaque par rejeu**".

Condensat

Référence: **http://gdt.oqlf.gouv.qc.ca**.

Séquence de caractères alphanumériques de longueur fixe, qui représente le contenu d'un message, sans le révéler, dont la valeur unique est produite par un algorithme de hachage, et qu'on utilise pour créer une signature numérique.

Empreinte numérique

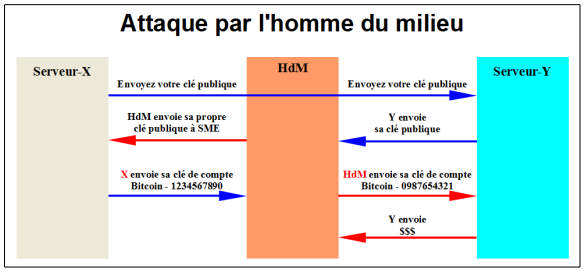
Référence: **http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=8371028**.

L'empreinte numérique sert à authentifier un message ou à vérifier l'identité de son auteur.

Une empreinte numérique a toujours la même taille, quelle que soit la longueur du message initial. À l'instar d'une empreinte digitale, deux messages différents n'ont pas la même empreinte numérique. Après avoir calculé l'empreinte de son message, l'expéditeur la chiffre avec sa clé privée. Il envoie ensuite cette signature en même temps que le reste de son message. Lorsque le destinataire reçoit cette empreinte chiffrée, il la déchiffre grâce à la clé publique de l'expéditeur. Le destinataire compare alors le résultat obtenu avec le résultat qu'il calcule lui-même à partir du message reçu. Si les deux empreintes numériques sont identiques, il est assuré à la fois de l'identité de l'expéditeur et de l'intégrité du message.

Homme-du-milieu

Référence: **https://fr.wikipedia.org/wiki/Attaque_de_l'homme_du_milieu**.



L'attaque de l'homme-du-milieu (*HDM*) ou man-in-the-middle attack (*MITM*) est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis. Le canal le plus courant est une connexion à Internet de l'internaute lambda. L'attaquant doit d'abord être capable d'observer et d'intercepter les messages d'une victime à l'autre. L'attaque "homme-du-milieu" est particulièrement applicable dans la méthode d'échange de clés Diffie-Hellman, quand cet échange est utilisé sans authentification. Avec authentification, Diffie-Hellman est en revanche invulnérable aux écoutes du canal, et est d'ailleurs conçu pour cela.

Certificat

Référence: **https://fr.wikipedia.org/wiki/Certificat_%C3%A9lectronique**.

Un certificat électronique (*aussi appelé certificat numérique ou certificat de clé publique*) peut être vu comme une carte d'identité numérique. Il est utilisé principalement pour identifier et authentifier une personne physique ou morale, mais aussi pour chiffrer des échanges.

Il est signé par un tiers de confiance qui atteste du lien entre l'identité physique et l'entité numérique (*Virtuel*).

Le standard le plus utilisé pour la création des certificats numériques est le **X.509**.

Référence: **https://fr.wikipedia.org/wiki/X.509**.

Dans le système **X.509**, une autorité de certification attribue un certificat liant une clé publique à un nom distinctif (*Distinguished Name*) dont le format est défini par la recommandation **X.500**, ou encore à un nom alternatif (*Alternative Name*) tel qu'une adresse électronique ou un enregistrement **DNS**.

Ce certificat place la signature d'une autorité de certification dans le dernier champ. Concrètement, cette signature est réalisée par un condensat de tous les champs précédents du certificat et un chiffrement de ce condensat par la clé privée de l'autorité de certification. N'importe qui possédant la clé publique de cette autorité de certification peut déchiffrer le condensat et le comparer au calcul de son propre condensat du certificat. Si les deux

condensats sont identiques, cela garantit que le certificat est intègre et qu'il n'a pas été modifié.

Chaîne de certification

Le certificat de l'autorité de certification qui contient sa clé publique peut à son tour être signé par un autre certificat de plus haut niveau, formant ainsi une chaîne. Tout en haut de la chaîne on trouve les certificats les plus importants: les certificats racines.

Certificats racines

Les certificats racines sont des clés publiques non signées, ou auto-signées, dans lesquelles repose la confiance. Les logiciels, comme les navigateurs Web ou les clients de messagerie détiennent des certificats racines de nombreuses autorités de certification commerciales ou gouvernementales. Quand un navigateur ouvre une connexion sécurisée (*TLS/SSL*) vers un site possédant un certificat émis par une autorité connue, il considère le site comme sûr dans la mesure où le chemin de certification est validé. Le passage en mode sécurisé est alors transparent.

Certificats à Validation du Domaine (DV)

Référence:
https://www.networking4all.com/fr/certificats+ssl/produits/par+validation/de+nom+de+domaine/.

Les certificats à Nom de Domaine (*DV*) sont émis très rapidement, et ce en quelques minutes. Le propriétaire ou le webmestre du nom de domaine doit confirmer la requête par courriel.

Pendant votre demande, vous devez indiquer une adresse courriel pour confirmation; vous pouvez choisir parmi une liste de différents courriel standard comme **admin@**, **root@**, **administrator@** etc... ou l'adresse courriel qui est listée dans les informations du **Whois** du nom de domaine.

Avantages

Un certificat **SSL DV** est facile à obtenir. Vous obtenez ce certificat en quelques minutes.

Bon à savoir

Ce type de **Certificat SSL** certifie seulement que le site Internet est sécurisé. Il n'y a pas d'**Autorité de Certification** qui a vérifié et validé l'identité du propriétaire du site Internet.

Pour qui?

Les certificats **DV** seront utilisés pour les sites Internet qui ont juste besoin d'une connexion **https** (*connexion sécurisée*) ou une connexion sécurisée pour webmail, intranet, citrix secure gateway, et ainsi de suite. Ce **Certificat SSL** est parfait dans le cas où quelqu'un a besoin immédiatement d'un certificat.

SAN et Wildcard

Référence: **https://www.thawte.fr/ssl/san-uc-ssl-certificates/#**.

Référence: **https://www.thawte.fr/ssl/wildcard-ssl-certificates/**.

Que signifient les termes **SAN** (*Subject Alternative Names*) et **UC** (*Unified Communications*)?

Les certificats qui utilisent les **SAN** (*Subject Alternative Names*) sont des outils puissants qui permettent de sécuriser plusieurs noms de domaines de façon efficace et économique. Les certificats **SSL** Thawte permettent de sécuriser jusqu'à 25 noms de domaines complets avec un seul certificat utilisant les **SAN**. Les noms de certificats qui utilisent les **SAN** sont également appelés certificats **UC** (*Unified Communications ou communications unifiées*) et sont utilisés avec Microsoft Exchange Server 2007, Microsoft Exchange Server 2010 et Microsoft Communications Server. L'objectif d'un certificat avec **SAN** est le même que n'importe quel autre certificat; il permet à un serveur de définir son identité et d'établir une communication sécurisée. Les certificats avec **SAN** procurent également un champ **SAN** (*Subject Alternative Name*) qui permet de protéger les noms de domaines additionnels avec un seul certificat.

Pourquoi ai-je besoin d'un **SAN?**

Au lieu d'acheter des certificats individuels pour chaque nom de domaine, vous pouvez ajouter des noms de domaines dans les champs **SAN** pour partager le même certificat. Non seulement l'entreprise économise le coût d'achat de certificats individuels, mais elle gagne également du temps en évitant d'avoir à gérer plusieurs certificats.

Par exemple, un seul certificat avec prise en charge des **SAN** serait capable de sécuriser les noms de domaines suivants:

- **www.mycompany.com**
- **mail.mycompany.com**
- **mycompany.com**
- **www.toto.net**
- **mail.toto.net**
- **toto.net**

Certificat SAN vs certificat Wildcard

Les certificats **Wildcard** sont similaires aux certificats **SAN** avec quelques restrictions. Avec un certificat **Wildcard**, vous pouvez sécuriser plusieurs sous-domaines avec un seul domaine racine. Par exemple, si vous avez un certificat **Wildcard** pour **www.mycompany.com**, il sécurise également **intranet.mycompany.com** et **email.mycompany.com** avec le même certificat.

Cependant, vous ne pourrez pas sécuriser plusieurs domaines uniques comme **www.mycompany.net** et **www.toto.org**.

Certificats SSL Wildcard

Sécurisation de plusieurs sous-domaines sur un seul serveur.

Les certificats **SSL Wildcard** Thawte sécurisent plusieurs sous-domaines avec un certificat **SSL** unique, réduisant ainsi le temps et le coût de gestion. L'utilisation de la notation **Wildcard** (*un **astérisque** et un **point** avant votre nom de domaine*) vous permet d'étendre la sécurité à différents sous-domaines, basés sur le nom de votre domaine de niveau supérieur.

CNAME

Référence: http://www.networking4all.com/fr/support/noms+de+domaine/dns/archives+cname/.

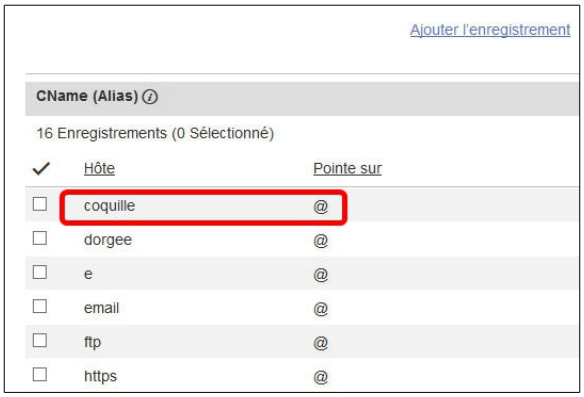
Un enregistrement **CNAME** ou enregistrement de **Nom Canonique** est un type d'enregistrement ressource dans le **Domain Name System** (*DNS*) qui spécifie que le nom de domaine est un alias d'un autre nom de domaine canonique.

1.1. Utilisation d'enregistrement CNAME

En utilisant les **CNAME**, vous rendez les données de votre **DNS** plus facile à gérer. Les enregistrements **CNAME** redirigent vers un enregistrement **A**. Par conséquent, si vous changez l'adresse **IP** d'un enregistrement **A**, tous vos enregistrements **CNAME** pointés vers cet enregistrement, suivent automatiquement le nouvel **IP** de l'enregistrement **A**. La solution alternative est d'avoir des enregistrements **A** multiples, mais alors vous aurez des places multiples pour changer l'adresse **IP** qui augmente les chances d'erreur.

L'utilisation la plus populaire d'un enregistrement **CNAME**, est de fournir un accès à un serveur Web en utilisant soit le standard **www.domain.com** ou soit **domain.com** (sans le **www**). Cette règle est généralisée en ajoutant un enregistrement **CNAME** pour le nom **www** pointant au nom court [*lors de la création d'un Enregistrement A pour le nom court* (*sans **www***)].

Exemple



Vous avez un site Web avec le nom de domaine **mywebsite.nl**. Ce nom de domaine est connecté à un enregistrement **A** qui traduit le nom de domaine à l'adresse **IP** appropriée, par exemple **11.22.33.44**.

Vous avez aussi plusieurs sous-domaines, comme **www.mywebsite.nl**, **mail.mywebsite.nl**, etc... et vous souhaitez que ces sous-domaines pointent à votre nom de domaine principal **mywebsite.nl**. Au lieu de créer des enregistrements **A** pour chaque sous-domaine et les lier à l'adresse **IP** de votre domaine principal, vous créez un **alias** (*enregistrement **CNAME***) pour chacun d'eux pour obtenir la figure ci-contre. Dans le cas où votre adresse **IP** change, vous devez seulement éditer un enregistrement **A** et tous les sous-domaines suivent automatiquement du fait des **CNAME** pointant vers le domaine principal.

Micronator a un serveur privé qui fait partie du domaine principal et dont le nom est **coquille**. Nous pouvons alors insérer un **CNAME** ***coquille*** pour ce serveur.

III- Let's Encrypt

1. Principe de fonctionnement de Letsencrypt

Référence: **https://linuxfr.org/news/reparlons-de-let-s-encrypt**.

La facilité d'utilisation promise par **Let's Encrypt** repose principalement sur le client **letsencrypt.sh** et sur l'automatisation qu'il propose.

Le client **letsencrypt.sh** s'occupe (*ou peut s'occuper*) de deux tâches distinctes:

1) obtenir un certificat pour le(*s*) domaine(*s)* souhaité(*s*), et

2) installer le certificat obtenu.

Pour obtenir un certificat, le client **letsencrypt.sh**:

- génère une paire de clefs et une demande de signature de certificat (*Certificate Signing Request, CSR*);
- envoie la demande à un serveur **ACME**;
- répond aux défis d'authentification (*challenges*) posés par la **CA**, permettant au demandeur de prouver qu'il contrôle le(*s*) domaine(*s*) demandé(*s*);
- reçoit le certificat signé en retour.

Le client installe le certificat proprement dit, la clef privée correspondante et les certificats intermédiaires là où le serveur web pourra les trouver, enfin il configure et relance ledit serveur s'il sait le faire (*si le serveur en question est Apache HTTP ou Nginx, pour l'instant*).

Le client **letsencrypt.sh** garde aussi une trace des certificats obtenus. Lancé à intervalle régulier, il répétera automatiquement la procédure s'il détecte qu'un certificat est sur le point d'expirer.

En définitive, le but est que l'administrateur puisse mettre en place **TLS** en une seule commande, avant d'oublier jusqu'à l'existence même du client **letsencrypt.sh**.

2. Courriels du certificat

Aucun courriel n'est envoyé pour confirmer le certificat mais, vous devez fournir une adresse courriel et un/des **CNAME** valides lors de l'exécution du script **letsencrypt.sh**.

3. Transparence des certificats

Une partie de la mission de transparence de la société **Let's Encrypt** comprend la divulgation publique des certificats qu'elle délivre via **Certificate Transparency**. L'adresse courriel n'est pas divulguée publiquement.

4. Limites

90 jours

Les certificats **Let's Encrypt** sont valides pour **90** jours. Elle recommande de les renouveler tous les **60** jours pour avoir une certaine marge de manoeuvre.

5/7

En date du 2015-12-03 16:46:08 UTC:

• Limite de **5** certificats par domaine, dans une fenêtre de **7** jours.

• Limite de **10** enregistrements par **IP**, toutes les **3** heures.

Référence: **https://community.letsencrypt.org/t/public-beta-rate-limits/4772**.

* Certificats par domaine signifie 5 émissions de certificat et non pas combien de domaines au sein d'un certificat multi-domaines **SAN**.

** Un certificat multi-domaines SAN ayant **domain1.com**, **www.domain1.com**, **domain2.com**, **www.domain2.com**, **toto.info**, **titi.org** est compté comme **1** certificat, mais on ne peut renouveler ce certificat multi-domaines plus de **5** fois par période de **7** jours?

*** Il n'y a pas de limites pour le nombre de domaines contenus dans un certificat multi-domaines **SAN** ou plus précisément jusqu'au maximum standard de **100**. **Let's Encrypt** a choisi cette limite de **100** sur une base de prudence, car il semble que lorsqu'on en obtient plus de **100**, certains navigateurs Web ont un comportement erratique. **Let's Encrypt** peut probablement augmenter cette limite si quelqu'un en fait la demande.

5. Mode Officiel vs TEST (*staging*[1])

Si vous voulez tester **letsencrypt.sh** et que vous n'êtes pas encore certain de vouloir l'utiliser, vous pouvez utiliser l'option **staging** (*en incluant la ligne* ***CA="https://acme-staging.api.letsencrypt.org/directory"*** *dans le fichier de configuration* ***config***.

Le principal avantage est de pouvoir demander autant de certificats que vous avez besoin pour vos tests sans vous heurter à la limite **5**/**7**.

5.1. Autorité de certification (CA)

CA officielle

Lors d'une demande de certificat officiel, le client **letsencrypt.sh** utilise la **CA** officielle, **acme-v01**.

CA de TEST

Un certificat de test ne sera pas signé directement par **Let's Encrypt** mais par sa **CA** de tests, **happy hacker fake CA**. Ce certificat de test ne sera pas reconnu par la plupart des navigateurs et affichera une erreur. La communication sera tout de même chiffrée.

• La **CA** **acme-staging** est l'émettrice pour **happy hacker fake CA**.

Il est fortement recommandé de débuter en demandant un certificat de test. Toutes les options particulières pour cette demande sont de couleur **ORANGE**.

1 **Référence**: **https://en.wikipedia.org/wiki/Staging_site**. Un site **Staging**, dans la conception de sites web, est un site utilisé pour assembler, tester et revoir une version plus récente avant de l'implanter en production.

5.2. Fichier de configuration

Certificat officiel

Le fichier de configuration pour une demande de certificat officiel ne spécifie pas la **CA** à utiliser car, par défaut, le client **letsencrypt.sh** utilise la **CA** officielle **acme-v01**.

Pour une demande de certificat, il est inutile de spécifier un fichier de configuration car par défaut c'est le fichier **config** qui est utilisé.

Fichier **config** de la configuration standard.

```
#!/bin/bash
# config
# CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@nom-de-votre-domaine"
```

La ligne **# CA="https://acme-staging.api...** est en commentaire car elle débute par le caractère "#".

Certificat de TEST

Le fichier de configuration pour une demande certificat de test doit spécifier la **CA** **acme-staging** qui seule, émet ces certificats de test pour **Let's Encrypt**.

```
#!/bin/bash
# config
CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@nom-de-votre-domaine"
```

La ligne **CA="https://acme-staging.api...** n'est pas en commentaire car elle ne débute pas par le caractère "#".

6. Clé de compte Let's Encrypt

La clé de compte est différente de la clé **privkey-nnnnnnnnn.pem** utilisée par le **Serveur SME** et qui est stockée dans le répertoire **/etc/letsencrypt.sh/certs/nom-du-domaine/**.

La clé de compte **account_key.pem** est stockée dans un sous-répertoire de **/etc/letsencrypt.sh/accounts/**.

- Si on demande un certificat de **TEST**, un sous-répertoire y est créé pour stocker la clé de compte de **TEST**.
- Si on demande un certificat **OFFICIEL**, un autre sous-répertoire y est créé pour stocker la clé de compte **OFFICIEL**.
- Le sous-répertoire de la clé de compte de **TEST** est créé lors de la première demande de certificat de **TEST** et celui de la clé de compte de **OFFICIEL** est créé lors de la première demande de certificat de **OFFICIEL**.

Le sous-répertoire et la clé utilisée dépend de la ligne **CA="https://acme-staging.api...** dans le fichier de configuration **config**.

- Si cette ligne n'est pas commentée, ce sera le sous-répertoire de la clé de compte de **TEST** qui sera utilisé.
- Si la ligne est commentée, ce sera le sous-répertoire de la clé de compte **OFFICIEL** qui sera utilisé.

La clé de compte est utilisée exclusivement par le client **letsencrypt.sh** et uniquement pour:

- créer un compte usager chez la **CA**,
- signer les domaines et
- chiffrer la communication entre le **Serveur SME** et la **CA**.

Que ce soit pour une demande de certificat de TEST ou pour un certificat **OFFICIEL**, la clé de compte est toujours appelé **account_key.pem**.

Une clé **SSL** est toujours constituée d'une partie publique et d'une partie privée.

7. Aide

Pour afficher l'aide du **client letsencrypt.sh**.

```
letsencrypt.sh --help
```

8. En cas de trouble majeur avec un certificat

Advenant un trouble majeur avec un certificat et que vous vouliez en recréer un original, émis et certifié par le **Serveur SME** lui-même, veillez consulter le paragraphe: ***Certificat standard SME*** à la page ***70***.

9. Paramètres

Une **chaîne de caractères en magenta** indique qu'il faut remplacer cette chaîne par vos propres paramètres.

IV- Prérequis

1. Conditions préalables

Le client **letsencrypt.sh** et le **Serveur SME** interagissent pour confirmer que la personne, demandant un certificat pour un nom d'hôte, contrôle réellement ce serveur.

Il existe quelques configurations préalables à une demande de certificat. Par exemple, si nous essayons d'obtenir un certificat pour www.example.com, toutes les conditions suivantes doivent être remplies:

- ☐ www.example.com est un nom de domaine valide - le nom de domaine a été enregistré et les enregistrements **DNS** sont publiés pour ce domaine.
- ☐ Le résultat d'une recherche **DNS** de www.example.com pointe vers l'adresse **IP** du **Serveur SME** – lorsqu'ils sont interrogés sur www.example.com, les enregistrements **DNS** publiés doivent donner l'adresse **IP** externe du **Serveur SME**.
- ☐ Le **Serveur SME** est connecté à l'Internet.
- ☐ Les ports **80** et **443** sur le **Serveur SME** sont ouverts à l'**Internet** – il n'est pas derrière un pare-feu ou un filtrage par le **FAI** qui bloquerait ces ports.

Le client **letsencrypt.sh** émettra un certificat qui peut comprendre plusieurs noms d'hôte (*par exemple: **www.exemple.com**, **exemple.com** et **mail.exemple.com***) qui tous, faisaient partie de la demande. Toutes les conditions ci-dessus doivent être remplies pour chacun des noms d'hôte qu'on souhaite inclure dans le certificat.

Assurez-vous que tout est correctement en place avant de continuer.

Avant de commencer l'installation, vérifiez si vous, ou une **CONTRIB** précédemment installée, avez configuré des valeurs personnalisées pour votre certificat **TLS/SSL** présentement actif.

Par défaut la commande devrait donner le résultat suivant.

```
[root@dorgee letsencrypt.sh]# config show modSSL

modSSL=service
    TCPPort=443
    access=public
    status=enabled
[root@dorgee letsencrypt.sh]#
```

Si la commande affiche les paramètres **crt**, **key**, **CertificateChainFile** et leurs valeurs, il est fortement recommandé d'en faire une ***Sauvegarde***. Ainsi, si vous rencontrez un problème avec les fichiers du certificat généré par **Let's Encrypt**, vous serez alors en mesure de revenir aux paramètres précédents.

V- Installation du client

1. Description

Plusieurs clients sont disponibles pour les services **Let's Encrypt**.

- Le client **letsencrypt.sh** est le favori de ceux qui préfèrent un client léger ne nécessitant aucune dépendance. Il peut être utilisé autant avec les **Serveurs SME-9.x** qu'avec les **Serveurs SME-8.x**.
- Liste des autres clients: **https://community.letsencrypt.org/t/list-of-client-implementations/2103**.

En date d'aujourd'hui (*juillet 2016*), une contrib, **https://wiki.contribs.org/Letsencrypt**, est en cours de développement et utilise le script **letsencrypt.sh** dans ses exemples. Pour de plus amples informations, voir **Bug 8676** et la **page GitHub** (*16 juillet 2016, cette dernière page n'est pas à date*).

2. Installation

Le script **letsencrypt.sh** est un client léger **ACME**[2] alternatif qui permet de demander/récupérer des certificats émis par les serveurs **Let's Encrypt**. Il n'est pas nécessaire d'installer de logiciels supplémentaires autres que **git**, un gestionnaire de versions de code source, pour télécharger le client et l'installer.

2.1. git

Avec **PuTTY**, on se logue à la console du serveur de notre domaine en tant qu'usager **root** et on installe **git**.

```
[root@dorgee ~]# yum -y install git

Modules complémentaires chargés : fastestmirror, smeserver
Configuration du processus d'installation
Determining fastest mirrors
...
Résolution des dépendances
...
Dépendances résolues
...
Installé:
  git.x86_64 0:1.7.1-4.el6_7.1

Dépendance(s) installée(s) :
  perl-Git.noarch 0:1.7.1-4.el6_7.1

Terminé !
[root@ dorgee ~]#
```

2 **Référence**: **https://www.metachris.com/2015/12/comparison-of-10-acme-lets-encrypt-clients/**. Le protocole **Automated Certificate Management Environment** (*ACME*) définit un moyen d'obtenir automatiquement des certificats sans intervention humaine. Tout d'abord, le contrôle d'un domaine doit être prouvée et alors l'agent peut demander, renouveler et révoquer les certificats.

2.2. Téléchargement

Un sous-répertoire **letsencrypt.sh** va être créé dans le répertoire dans lequel on lance la commande **git**.

On se rend donc dans **/etc**, car on veut que le sous-répertoire **letsencrypt.sh** y soit créé.

```
[root@dorgee ~]# cd /etc

[root@ dorgee etc]#
```

On vérifie.

```
[root@dorgee etc]# pwd

/etc
[root@dorgee etc]#
```

On télécharge le client.

```
[root@dorgee etc]# git clone https://github.com/lukas2511/letsencrypt.sh

Initialized empty Git repository in /etc/letsencrypt.sh/.git/
remote: Counting objects: 1134, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 1134 (delta 0), reused 0 (delta 0), pack-reused 1131
Receiving objects: 100% (1134/1134), 278.95 KiB, done.
Resolving deltas: 100% (687/687), done.
[root@dorgee etc]#
```

On vérifie la création du répertoire **letsencrypt**.

```
[root@dorgee etc]# ls -als /etc/letsencrypt.sh/

total 92
 4 drwxr-xr-x   4 root root  4096 15 juil. 23:51 .
12 drwxr-xr-x 100 root root 12288 15 juil. 23:51 ..
 4 -rw-r--r--   1 root root  1406 15 juil. 23:51 CHANGELOG
 4 drwxr-xr-x   3 root root  4096 15 juil. 23:51 docs
 4 drwxr-xr-x   8 root root  4096 15 juil. 23:51 .git
 4 -rw-r--r--   1 root root   108 15 juil. 23:51 .gitignore
40 -rwxr-xr-x   1 root root 37772 15 juil. 23:51 letsencrypt.sh
 4 -rw-r--r--   1 root root  1080 15 juil. 23:51 LICENSE
 4 -rw-r--r--   1 root root  3040 15 juil. 23:51 README.md
 8 -rwxr-xr-x   1 root root  8048 15 juil. 23:51 test.sh
 4 -rw-r--r--   1 root root   107 15 juil. 23:51 .travis.yml
[root@dorgee etc]#
```

On sécurise le fichier **letsencrypt.sh**.

```
[root@dorgee etc]# chmod 700 /etc/letsencrypt.sh/letsencrypt.sh

[root@dorgee etc]#
```

On vérifie.

```
[root@dorgee etc]# ls -ls /etc/letsencrypt.sh/letsencrypt.sh

40 -rwx------ 1 root root 37772 15 juil. 23:51 /etc/letsencrypt.sh/letsencrypt.sh
[root@dorgee etc]#
```


VI- Création des fichiers et répertoires requis

1. Répertoire des défis

Le greffon **webroot** fonctionne en créant un fichier temporaire incluant chacun des domaines demandés dans **${webroot-path}/.well-known/acme-challenge/**. Le serveur de validation de **Let's Encrypt** fera des requêtes **HTTP** pour valider que tous les noms de domaines/**CNAME** contenus dans ce fichier pointent vers le serveur exécutant **letsencrypt.sh**.

On se rend dans le répertoire personnel de l'usager **root**. Il sera notre répertoire de travail.

```
[root@dorgee etc]# cd

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# pwd

/root
[root@dorgee ~]#
```

On crée le répertoire des défis (*challenge*) afin de prouver que le serveur est bien celui qu'il prétend être

```
[root@dorgee ~]# mkdir -p /home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# ls -d /home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge

/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge
[root@dorgee ~]#
```

2. Gabarit personnalisé

Nous avons besoin d'un gabarit personnalisé (*custom template*) pour indiquer à **Apache** le répertoire **acme-challenge**.

On crée le répertoire pour le gabarit personnalisé.

```
[root@dorgee ~]# mkdir -p /etc/e-smith/templates-custom/etc/httpd/conf/httpd.conf

[root@dorgee ~]
```

On vérifie.

```
[root@dorgee ~]# ls -d /etc/e-smith/templates-custom/etc/httpd/conf/httpd.conf

/etc/e-smith/templates-custom/etc/httpd/conf/httpd.conf
[root@dorgee ~]#
```

Pour le gabarit personnalisé, on crée le fichier **VirtualHosts40ACME** et on y insère son contenu.

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/e-smith/templates-custom/etc/httpd/conf/httpd.conf/VirtualHosts40ACME <<'EOT'
# Alias for letsencrypt
#
Alias /.well-known/acme-challenge /home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge

EOT
```

On vérifie.

```
[root@dorgee ~]# cat  \
                 /etc/e-smith/templates-custom/etc/httpd/conf/httpd.conf/VirtualHosts40ACME

# Alias for letsencrypt
#
Alias /.well-known/acme-challenge /home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge

[root@dorgee ~]#
```

Il n'y a pas de ligne vide avant la ligne **# Alias for letsencrypt**. Ci-dessus nous avons inséré une ligne vide pour aider à la copie de la commande.

On développe le gabarit personnalisé.

```
[root@dorgee ~]# expand-template /etc/httpd/conf/httpd.conf

[root@dorgee ~]#
```

2.1. Redémarrage du service httpd-e-smith

```
[root@dorgee ~]# service httpd-e-smith restart

Restarting httpd-e-smith                                   [  OK  ]
[root@dorgee ~]#
```

3. Fichiers de configuration

Il faut créer deux fichiers de configuration: **config** (*anciennement **config.sh***) et **domains.txt**.

3.1. config

On crée le fichier **config** et on y insère son contenu. Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/letsencrypt.sh/config <<'EOT'
#!/bin/bash
# config
# CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@micronator.org"

EOT
```

- **Attention** au domaine de l'adresse courriel **CONTACT_EMAIL**="admin@micronator.org" ci-dessus. Il faut le remplacer par **votre domaine**.

On sécurise le fichier.

```
[root@dorgee ~]# chmod 700 /etc/letsencrypt.sh/config

[root@dorgee ~]#
```

On vérifie la création du fichier.

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/config

4 -rwx------ 1 root root 323 16 juil. 00:34 /etc/letsencrypt.sh/config
[root@dorgee ~]
```

On vérifie son contenu.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/config

#!/bin/bash
# config
# CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@micronator.org"

[root@dorgee ~]#
```

Il n'y a pas de ligne vide avant la ligne **#!/bin/bash**. Ci-dessus nous avons inséré une ligne vide pour aider à copier la commande.

3.2. domains.txt

Dans ce fichier, on énumère, sur une seule ligne, chacun des noms des domaines/**CNAME** qu'on veut couvrir avec notre certificat **SAN**; chaque nom doit être séparé par un espace.

Le contenu devrait ressembler à l'exemple ci-dessous:

```
domain1.com www.domain1.com mail.domain1.com domain2.net www.domain2.net domain3.org
ftp.domain3.org
```

Le certificat est toujours émis au nom du premier domaine de la ligne, ci-dessus ce sera au nom de **domain1.com**.

On demandera un certificat **SAN** pour les domaines comprenant les **CNAME** suivants:

micronator.org	*ainesmercierouest.info*
www.micronator.org	www.ainesmercierouest.info
micronator.org	ainesmercierouest.info
dorgee.micronator.org	dorgee.ainesmercierouest.info
mail.micronator.org	mail.ainesmercierouest.info
ftp.micronator.org	ftp.ainesmercierouest.info
wpad.micronator.org	wpad.ainesmercierouest.info
proxy.micronator.org	proxy.ainesmercierouest.info

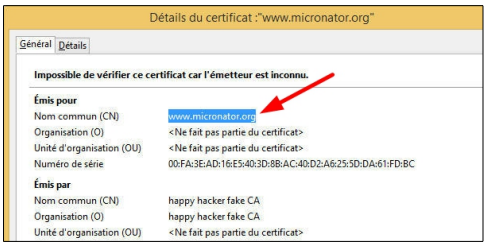
Le nom de notre serveur est **dorgee**.

Exemple de ce que devrait contenir notre fichier **domains.txt**. (***Une seule ligne***).

```
www.micronator.org micronator.org dorgee.micronator.org mail.micronator.org
ftp.micronator.org wpad.micronator.org proxy.micronator.org www.ainesmercierouest.info
ainesmercierouest.info dorgee.ainesmercierouest.info mail.ainesmercierouest.info
ftp.ainesmercierouest.info wpad.ainesmercierouest.info proxy.ainesmercierouest.info
```

Il n'y a pas de ligne vide avant la ligne **www.micronator.org micronator.org dorgee...**

Le certificat **SAN** sera émis pour le premier domaine de la ligne. Ici le certificat sera émis pour le site **www.micronator.org**.



Notre choix s'est arrêté à www.micronator.org afin de faciliter la connexion de certains rares clients de messagerie qui refusent de se connecter si le domaine ne débute pas par **www**. Ce refus est pour soi-disant limiter les pourriels provenant de sites utilisant un **DNS** dynamique (*sic*).

On crée le fichier et on y insère son contenu.

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/letsencrypt.sh/domains.txt <<'EOT'
www.micronator.org micronator.org dorgee.micronator.org mail.micronator.org
ftp.micronator.org wpad.micronator.org proxy.micronator.org www.ainesmercierouest.info
ainesmercierouest.info dorgee.ainesmercierouest.info mail.ainesmercierouest.info
ftp.ainesmercierouest.info wpad.ainesmercierouest.info proxy.ainesmercierouest.info
EOT
```

On vérifie la création du fichier.

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/domains.txt

4 -rw-r--r-- 1 root root 328 16 juil. 00:41 /etc/letsencrypt.sh/domains.txt
[root@dorgee ~]#
```

On vérifie son contenu. (*Une seule ligne*).

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/domains.txt

www.micronator.org micronator.org dorgee.micronator.org mail.micronator.org
ftp.micronator.org wpad.micronator.org proxy.micronator.org www.ainesmercierouest.info
ainesmercierouest.info dorgee.ainesmercierouest.info mail.ainesmercierouest.info
ftp.ainesmercierouest.info wpad.ainesmercierouest.info proxy.ainesmercierouest.info
[root@dorgee ~]#
```

Il n'y a pas de ligne vide avant la ligne **www.micronator.org micronator.org dor...** Ci-dessus nous avons inséré une ligne vide pour aider à copier la ligne de commande seulement.

Note importante sur le fichier domains.txt

Si tous les domaines et sous domaines sont répertoriés sur la même ligne, il en résultera un seul certificat **SAN** (*Subject-Alternative-Name*) au nom du **premier domaine** de la ligne.

```
domain1.com www.domain1.com mail.domain1.com domain2.net www.domain2.net domain3.org
```


Si chacun des domaines sont énumérés sur des lignes différentes, il en résultera autant de certificats que le nombre de lignes dans le fichier.

```
domain1.com  www.domain1.com  mail.domain1.com
domain2.net  www.domain2.net
domain3.org  ftp.domain3.org
domain4.info  www.domain4.org
```

Le résultat d'une telle énumération des domaines pourrait facilement atteindre ou dépasser la limite décrite au paragraphe *5/7* à la page *14*.

Le fichier **domains.txt** ci-dessus, indique qu'il devrait y avoir 4 certificats: **domain1.com**, **domain2.net**, **domain3.org** et **domain4.info**. Chaque certificat couvrant le domaine principal et ses **CNAME** (*sous-domaines*).

4. Script de point d'entrée[3]

Lorsqu'un certificat est émis ou renouvelé, vous aurez besoin d'un script de **point d'entrée** pour mettre à jour les paramètres de **modSSL** et déclencher le rechargement des services système.

On crée le fichier **letsencrypt-hook.sh** et on y insère son contenu.

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/letsencrypt.sh/letsencrypt-hook.sh <<'EOT'
#!/bin/bash

if [ $1 = "deploy_cert" ]; then
  KEY=$3
  CERT=$4
  CHAIN=$6
#
  /sbin/e-smith/db configuration setprop modSSL key $KEY
  /sbin/e-smith/db configuration setprop modSSL crt $CERT
  /sbin/e-smith/db configuration setprop modSSL CertificateChainFile $CHAIN
  /sbin/e-smith/signal-event domain-modify
  /sbin/e-smith/signal-event email-update
  /sbin/e-smith/signal-event ibay-modify
fi
EOT
```

4.1. Version de e-smith-base

On vérifie la version de ***e-smith-base***.

```
[root@dorgee ~]# rpm -qa | grep e-smith-base

e-smith-base-5.6.0-28.el6.sme.noarch
[root@dorgee ~]#
```

Si la version de **e-smith-base** est égale ou plus grande que **5.6.0-26** (i.e., si vous avez installé des mises à jour depuis la fin Janvier 2016), on remplace les 3 lignes de **/sbin/e-smith/signal-event...** dans le fichier **letsencrypt-hook.sh** ci dessus, par:

```
/sbin/e-smith/signal-event ssl-update
```

3 **Référence**: https://fr.wikipedia.org/wiki/Interface_%28informatique%29. Un principe clé de conception est d'interdire l'accès à toutes les ressources par défaut, en autorisant l'accès seulement à travers des points d'entrée bien définis.

Ce qui donnera:

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/letsencrypt.sh/letsencrypt-hook.sh <<'EOT'
#!/bin/bash

if [ $1 = "deploy_cert" ]; then
  KEY=$3
  CERT=$4
  CHAIN=$6
#
  /sbin/e-smith/db configuration setprop modSSL key $KEY
  /sbin/e-smith/db configuration setprop modSSL crt $CERT
  /sbin/e-smith/db configuration setprop modSSL CertificateChainFile $CHAIN
  /sbin/e-smith/signal-event ssl-update
fi
EOT
```

On vérifie.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/letsencrypt-hook.sh

#!/bin/bash

if [ $1 = "deploy_cert" ]; then
  KEY=$3
  CERT=$4
  CHAIN=$6
#
  /sbin/e-smith/db configuration setprop modSSL key $KEY
  /sbin/e-smith/db configuration setprop modSSL crt $CERT
  /sbin/e-smith/db configuration setprop modSSL CertificateChainFile $CHAIN
  /sbin/e-smith/signal-event ssl-update
fi
[root@dorgee ~]#
```

Il n'y a pas de ligne vide avant la ligne **#!/bin/bash**. Ci-dessus nous avons inséré une ligne vide pour faciliter la copie de la commande.

On sécurise et on rend le fichier du script exécutable.

```
[root@dorgee ~]# chmod 700 /etc/letsencrypt.sh/letsencrypt-hook.sh

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/letsencrypt-hook.sh

4 -rwx------ 1 root root 311 16 juil. 00:56 /etc/letsencrypt.sh/letsencrypt-hook.sh
[root@dorgee ~]#
```

5. Sauvegarde

5.1. Paramètres de modSSL

Si vous aviez déjà un certificat, vous pouvez faire une sauvegarde de ses paramètres.

```
 config show modSSL > "/root/sauvegarde_BD_params_modSSL_$(date +%Y%m%d_%H%M%S)"

[root@dorgee ~]#
```


On vérifie la création du fichier.

```
[root@dorgee ~]# ls -ls /root/sauvegarde_BD_params_modSSL*

4 -rw-r--r-- 1 root root 68 16 juil. 00:59 /root/sauvegarde_BD_params_modSSL_20160716_005904
[root@dorgee ~]#
```

On affiche le contenu.

```
[root@dorgee ~]# cat /root/sauvegarde_BD_params_modSSL_20160716_005904

modSSL=service
    TCPPort=443
    access=public
    status=enabled
[root@dorgee ~]#
```

5.2. BD du Serveur SME

Pour être encore plus sécuritaire, on peut faire une sauvegarde complète de la base de données de la configuration du **Serveur SME** (*une bonne pratique avant toute action telle que le changement manuel des valeurs de la BD ou avant l'installation d'une Contrib*).

```
[root@dorgee ~]# config show > "/root/sauvegarde_BD_Serveur_$(date +%Y%m%d_%H%M%S)"

[root@dorgee ~]#
```

On vérifie la création du fichier.

```
[root@dorgee ~]# ls -ls /root/sauvegarde_BD_Serveur_*

12 -rw-r--r-- 1 root root 10882 16 juil. 01:00 /root/sauvegarde_BD_Serveur_20160716_010022
[root@dorgee ~]#
```

On affiche le contenu du fichier. (*Plusieurs centaines de lignes.*)

```
[root@dorgee ~]# cat /root/sauvegarde_BD_Serveur_20160716_010022

AccessType=dedicated
...
yum=service
    AutoInstallUpdates=disabled
    CheckContribs=enabled
    EnableGroups=0
    GPGCheck=0
    PackageFunctions=disabled
    RandomDelay=120
    check4updates=daily
    status=enabled
[root@dorgee ~]#
```

Ces sauvegardes se restaurent manuellement, entrée par entrée.

On peut comparer deux fichiers de ces sauvegardes pour afficher leurs différences en se référant au site:

http://www.linux-france.org/article/memo/node12.html.

Nous sommes prêts à lancer le **client letsencrypt.sh** et obtenir notre premier certificat de **Let's Encrypt**.

VII- Demande d'un certificat de TEST

1. Introduction

Référence: **https://github.com/lukas2511/letsencrypt.sh**.

Let's Encrypt impose des limites strictes. Si vous commencez à tester en utilisant un certificat officiel (*la valeur par défaut*), vous allez rapidement atteindre ces limites et vous vous retrouvez en restriction de demandes/renouvellements de certificats.

C'est pour cette raison que nous recommandons fortement de commencer en mode **TEST** afin de vérifier les configurations et surtout de ne pas dépasser les ***5/7*** de **Let's Encrypt**.

2. Affichage de l'aide

Pour afficher l'aide sur la commande.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh --help

Usage: /usr/local/bin/letsencrypt.sh [-h] [command [argument]] [parameter [argument]]
[parameter [argument]] ...

Default command: help

Commands:
 --cron (-c)                      Sign/renew non-existant/changed/expiring certificates.
 --signcsr (-s) path/to/csr.pem   Sign a given CSR, output CRT on stdout (advanced usage)
 --revoke (-r) path/to/cert.pem   Revoke specified certificate
 --cleanup (-gc)                  Move unused certificate files to archive directory
 --help (-h)                      Show help text
 --env (-e)                       Output configuration variables for use in other scripts

Parameters:
 --domain (-d) domain.tld         Use specified domain name(s) instead of domains.txt entry
                                  (one certificate!)
 --force (-x)                     Force renew of certificate even if it is longer valid than
                                  value in RENEW_DAYS
 --privkey (-p) path/to/key.pem   Use specified private key instead of account key (useful
                                  for revocation)
 --config (-f) path/to/config     Use specified config file
 --hook (-k) path/to/hook.sh      Use specified script for hooks
 --challenge (-t) http-01|dns-01  Which challenge should be used? Currently http-01 and
                                  dns-01 are supported
 --algo (-a) rsa|prime256v1|secp384r1 Which public key algorithm should be used? Supported:
                                  rsa, prime256v1 and secp384r1
[root@dorgee ~]#
```


3. Fichier de configuration

3.1. Fichier config

Nous allons modifier le fichier **/etc/letsencrypt.sh/config** pour enlever le commentaire de la ligne **# CA="https://acme-staging.api...** pour demander un certificat de **TEST**.

```
[root@dorgee ~]# sed -i 's/^# CA="https:/CA="https:/'  /etc/letsencrypt.sh/config

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/config

#!/bin/bash
# config
CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@micronator.org"

[root@dorgee ~]#
```

Il n'y a pas de ligne vide avant la ligne **#!/bin/bash**. Ci-dessus nous avons inséré une ligne vide pour faciliter la copie de la commande.

Attention au domaine de l'adresse courriel **CONTACT_EMAIL**="admin@micronator.org" ci-dessus. Il faut le remplacer par **votre domaine**.

4. Sauvegarde

Si la **BD** du **Serveur SME** et les paramètres de **modSSL** n'ont pas été sauvegardés, on peut le faire maintenant. Voir le paragraphe ***Sauvegarde*** à la page ***25***.

5. Demande du certificat

On lance le **client letsencrypt.sh**. pour obtenir notre premier certificat de **Let's Encrypt**.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh -c

# INFO: Using main config file /etc/letsencrypt.sh/config
+ Generating account key...
+ Registering account key with letsencrypt...
Processing www.micronator.org with alternative names: micronator.org dorgee.micronator.org
mail.micronator.org ftp.micronator.org wpad.micronator.org proxy.micronator.org
www.ainesmercierouest.info ainesmercierouest.info dorgee.ainesmercierouest.info
mail.ainesmercierouest.info ftp.ainesmercierouest.info wpad.ainesmercierouest.info
proxy.ainesmercierouest.info
 + Signing domains...
 + Creating new directory /etc/letsencrypt.sh/certs/www.micronator.org...
 + Generating private key...
 + Generating signing request...
 + Requesting challenge for www.micronator.org...
 + Requesting challenge for micronator.org...
 + Requesting challenge for dorgee.micronator.org...
 + Requesting challenge for mail.micronator.org...
 + Requesting challenge for ftp.micronator.org...
 + Requesting challenge for wpad.micronator.org...
 + Requesting challenge for proxy.micronator.org...
 + Requesting challenge for www.ainesmercierouest.info...
 + Requesting challenge for ainesmercierouest.info...
 + Requesting challenge for dorgee.ainesmercierouest.info...
 + Requesting challenge for mail.ainesmercierouest.info...
 + Requesting challenge for ftp.ainesmercierouest.info...
 + Requesting challenge for wpad.ainesmercierouest.info...
 + Requesting challenge for proxy.ainesmercierouest.info...
 + Responding to challenge for www.micronator.org...
 + Challenge is valid!
 + Responding to challenge for micronator.org...
 + Challenge is valid!
 + Responding to challenge for dorgee.micronator.org...
 + Challenge is valid!
 + Responding to challenge for mail.micronator.org...
 + Challenge is valid!
 + Responding to challenge for ftp.micronator.org...
 + Challenge is valid!
 + Responding to challenge for wpad.micronator.org...
 + Challenge is valid!
 + Responding to challenge for proxy.micronator.org...
 + Challenge is valid!
 + Responding to challenge for www.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for dorgee.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for mail.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for ftp.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for wpad.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for proxy.ainesmercierouest.info...
 + Challenge is valid!
 + Requesting certificate...
 + Checking certificate...
 + Done!
 + Creating fullchain.pem...
[root@dorgee ~]#
```

Aucune erreur. Tout a bien fonctionné.

- Avant la demande du certificat, le script a utilisé le fichier de configuration, **# INFO: Using main config file /etc/letsencrypt.sh/config**.
- Il a créé localement "**+ Generating account key...**" une clé privée de compte pour **Let's Encrypt**; exemple: il a créé le sous-répertoire ***/etc/letsencrypt.sh/accounts***/aHR0cHM6Ly9hY21lLXN0YWdpabcuYXB-pLmxldHNlbmNyeXB0Lm9yZy9kaXJlY3Rvcncc/ et y a déposé la clé privée de compte **account_key.pem**. Les droits de cette clé sont limités à **root** seulement.
- Il a fait un échange de clés avec Let's Encrypt, **+ Registering account key with letsencrypt...**
- Il a analysé le fichier **domains.txt**, **Processing www.micronator.org**.
- Il a signé les domaines, **+ Signing domains...**
- Il a créé un répertoire utilisant le nom du premier domaine dans le fichier **domains.txt**, **+ Creating new directory /etc/letsencrypt.sh/certs/www.micronator.org...**
- Il a généré localement une clé privée **SSL** pour le serveur, **+ Generating private key...** exemple: **/etc/letsencrypt.sh/certs/www.micronator.org/privkey-1468698988.pem**.
- Il a créé une requête CSR, **+ Generating signing request...** exemple: **/etc/letsencrypt.sh/certs/www.micronator.org/cert-1468698988.csr**.
- Il a demandé les défis, attendu leurs validités et vérifié leurs réponses.
- Il a fait la demande du certificat, **+ Requesting certificate...**,
- Une fois reçu, il a vérifié le certificat, **+ Checking certificate...**,
- Le tout terminé **+ Done!**, il a alors créé la chaîne de certification et ajusté les pointeurs.
- Il a appelé le script de point d'entrée et celui-ci a modifié les propriétés de **modSSL** et signalé les changements pour activer le nouveau certificat.

6. Vérification

6.1. Console du serveur

On vérifie le répertoire **/etc/letsencrypt.sh/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/

total 112
 4 drwxr-xr-x   6 root root  4096 16 juil. 23:56 .
12 drwxr-xr-x 100 root root 12288 16 juil. 19:26 ..
 4 drwx------   3 root root  4096 16 juil. 23:55 accounts
 4 drwx------   3 root root  4096 16 juil. 23:56 certs
 4 -rw-r--r--   1 root root  1406 15 juil. 23:51 CHANGELOG
 4 -rwx------   1 root root   321 16 juil. 23:45 config
 4 drwxr-xr-x   3 root root  4096 15 juil. 23:51 docs
 4 -rw-r--r--   1 root root   328 16 juil. 00:41 domains.txt
 4 drwxr-xr-x   8 root root  4096 15 juil. 23:51 .git
 4 -rw-r--r--   1 root root   108 15 juil. 23:51 .gitignore
 4 -rwx------   1 root root   311 16 juil. 00:56 letsencrypt-hook.sh
40 -rwx------   1 root root 37772 15 juil. 23:51 letsencrypt.sh
 4 -rw-r--r--   1 root root  1080 15 juil. 23:51 LICENSE
 4 -rw-r--r--   1 root root  3040 15 juil. 23:51 README.md
 8 -rwxr-xr-x   1 root root  8048 15 juil. 23:51 test.sh
 4 -rw-r--r--   1 root root   107 15 juil. 23:51 .travis.yml
[root@dorgee ~]#
```


On vérifie le répertoire **/etc/letsencrypt.sh/certs/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/certs/

total 12
4 drwx------ 3 root root 4096 16 juil. 23:56 .
4 drwxr-xr-x 6 root root 4096 16 juil. 23:56 ..
4 drwx------ 2 root root 4096 16 juil. 23:56 www.micronator.org
[root@dorgee ~]#
```

Le script a créé un répertoire de stockage pour notre domaine **www.micronator.org** (*le premier **CNAME** de la ligne du fichier **/etc/letsencrypt.sh/domains.txt***).

On vérifie le répertoire **www.micronator.org/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/certs/www.micronator.org/

total 28
4 drwx------ 2 root root 4096 12 mars  10:28 .
4 drwx------ 3 root root 4096 12 mars  10:27 ..
4 -rw------- 1 root root 2118 12 mars  10:27 cert-1468698988.csr
4 -rw------- 1 root root 2614 12 mars  10:28 cert-1468698988.pem
0 lrwxrwxrwx 1 root root   19 12 mars  10:28 cert.csr -> cert-1468698988.csr
0 lrwxrwxrwx 1 root root   19 12 mars  10:28 cert.pem -> cert-1468698988.pem
4 -rw------- 1 root root 1123 12 mars  10:28 chain-1468698988.pem
0 lrwxrwxrwx 1 root root   20 12 mars  10:28 chain.pem -> chain-1468698988.pem
4 -rw------- 1 root root 3737 12 mars  10:28 fullchain-1468698988.pem
0 lrwxrwxrwx 1 root root   24 12 mars  10:28 fullchain.pem -> fullchain-1468698988.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1468698988.pem
0 lrwxrwxrwx 1 root root   22 12 mars  10:28 privkey.pem -> privkey-1468698988.pem
[root@dorgee ~]#
```

- La requête, **cert-1468698988.csr**.
- Le certificat, **cert-1468698988.pem**.
- La chaîne de certification, **chain-1468698988.pem**.
- La clé privée **SSL** du serveur, **privkey-1468698988.pem**.
- Les pointeurs sont ajustés en conséquence.

modSSL

Les paramètres de **modSSL** ont été modifiés.

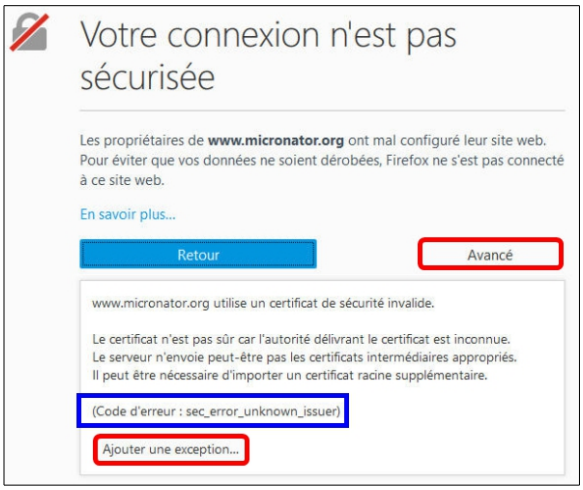
```
[root@dorgee ~]# config show modSSL

modSSL=service
    CertificateChainFile=/etc/letsencrypt.sh/certs/www.micronator.org/chain.pem
    TCPPort=443
    access=public
    crt=/etc/letsencrypt.sh/certs/www.micronator.org/cert.pem
    key=/etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem
    status=enabled
[root@dorgee ~]#
```


6.2. Navigateurs WEB

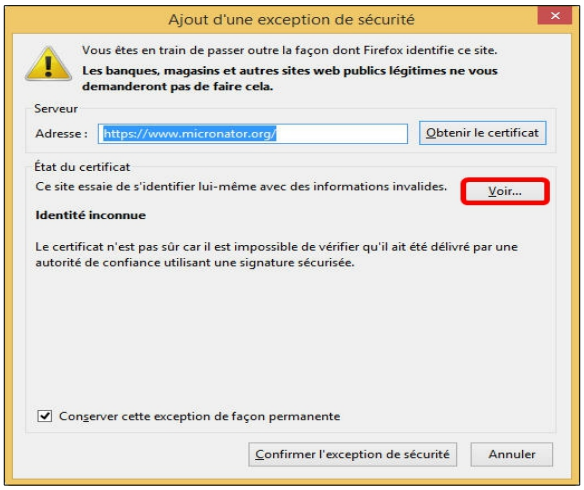
Notre certificat en est un de TEST. Pour ce genre de certificat, l'émetteur est **happy hacker fake CA** (*émetteur des certificats de test pour **Let's Encrypt***). Cet émetteur n'est pas reconnu comme une véritable **CA** et nous aurons l'erreur **sec_error_unknown_issuer**.

Firefox & le premier domaine

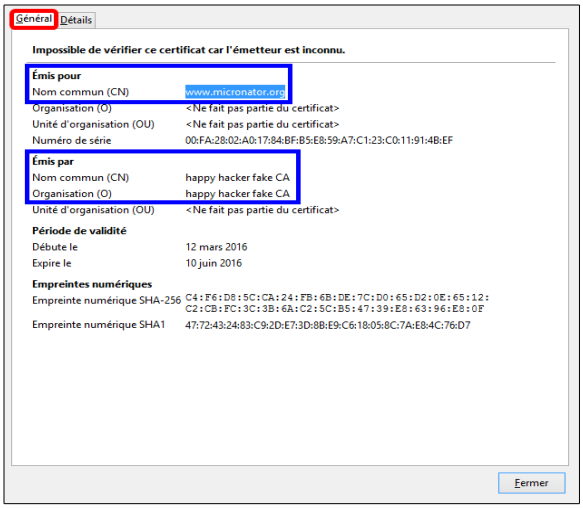
- Site: ***https://www.micronator.org/***.
- On reçoit l'erreur normale pour un certificat de TEST.
- **Avancé | Ajouter une exception...**



Voir...



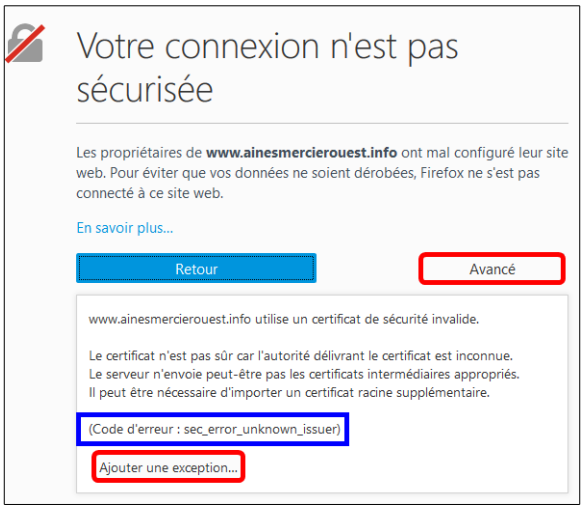
- ***Émis pour***: ***www.micronator.org***.
- L'***Émetteur*** est ***happy hacker fake CA*** et il n'est pas reconnu comme une véritable *CA*; c'est pourquoi nous avons une erreur.
- **Fermer | Confirmer l'exception de sécurité**.



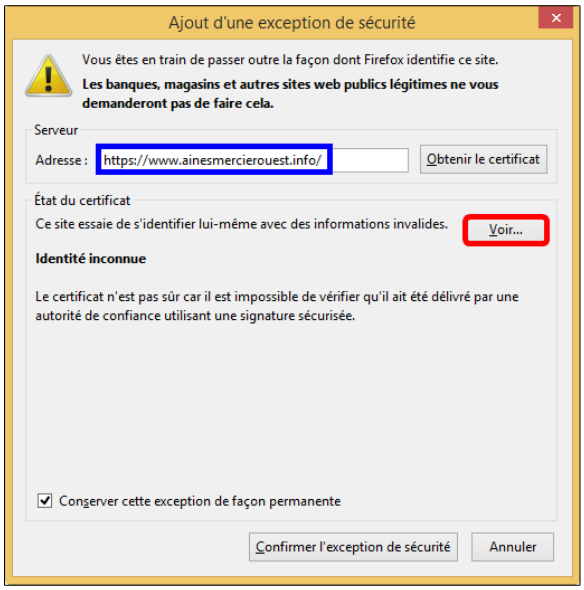
- Après la confirmation de l'exception de sécurité, la page d'acceuil de ***https://www.micronator.org*** s'affiche.
- Le cadenas est vert, on le clique.
- La connexion est sécurisée.

Firefox & le deuxième domaine

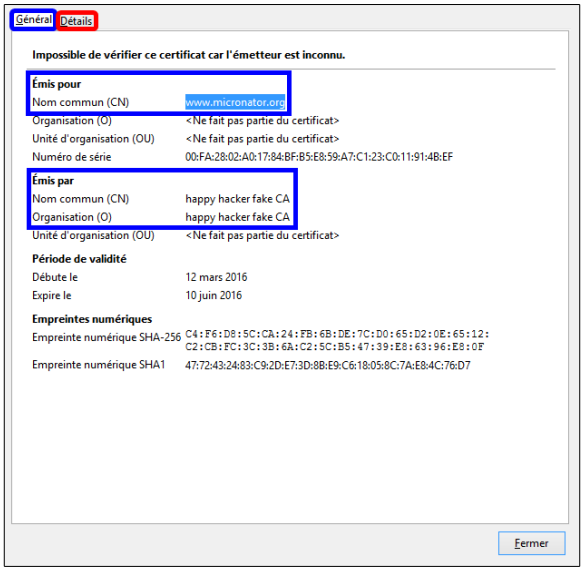
- Site: ***https://www.ainesmercierouest.info/***.
- On reçoit la même erreur.
- **Avancé | Ajouter une exception...**



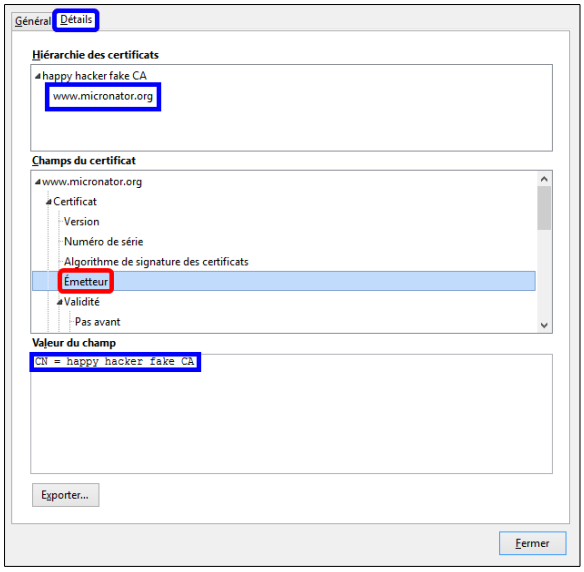
Voir.



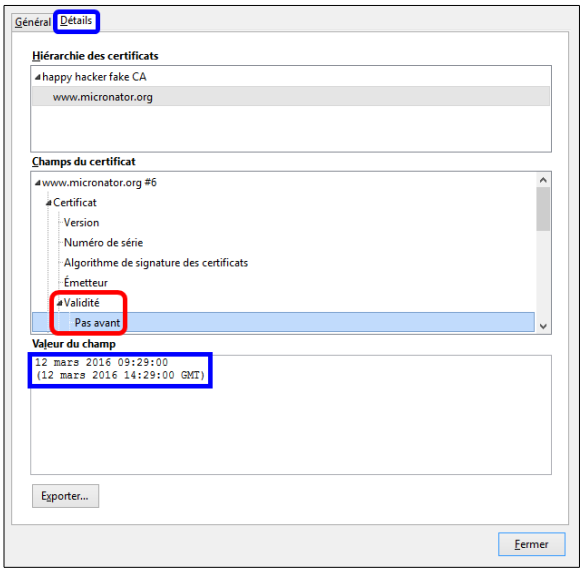
- Le même émetteur et la même date.
- Cette fois on clique **Détails** pour examiner le certificat.



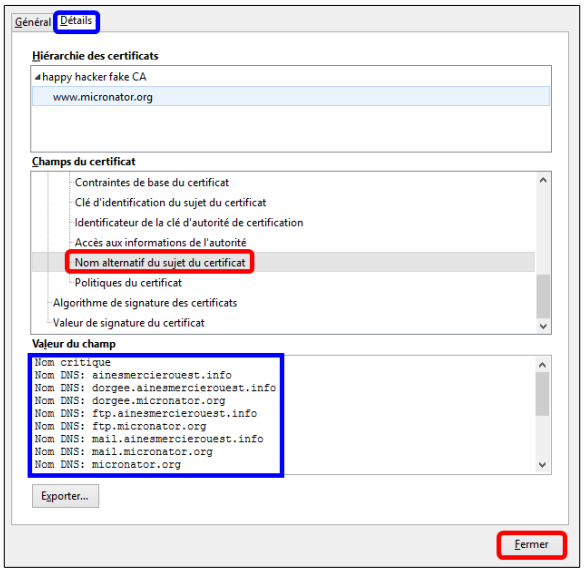
On voit que le nom de l'**Émetteur**, est bien **happy hacker fake CA**.



- **Validité | Pas avant**.
- Le certificat a été émis le **12 mars 2016 09:29:00**.

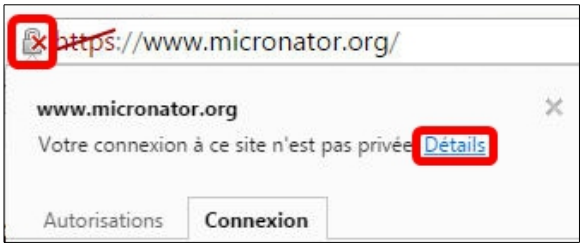


- **Nom alternatif du sujet du certificat**.
- Tous les domaines couverts par ce certificat sont affichés.
- On ferme toutes les fenêtres.

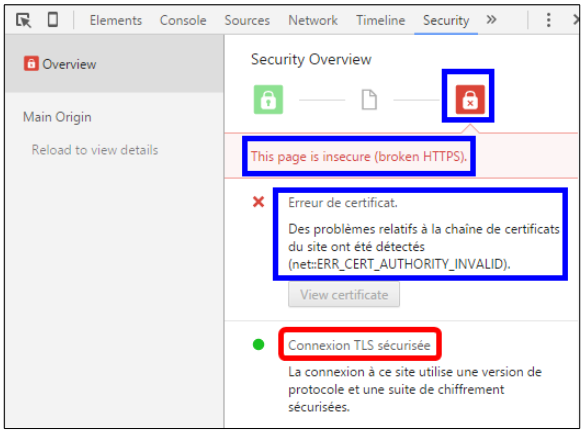


Google Chrome

- Site **https://www.micronator.org**.
- On clique le cadenas | **Détails**.



- Même erreur.
- Le site est sécurisé.
- Avec *Chrome*, le cadenas restera toujours barré avec un **X**.
- On ferme toutes les fenêtres.

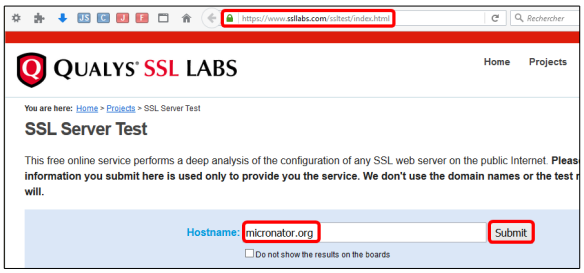


6.3. Qualys SSL LABS

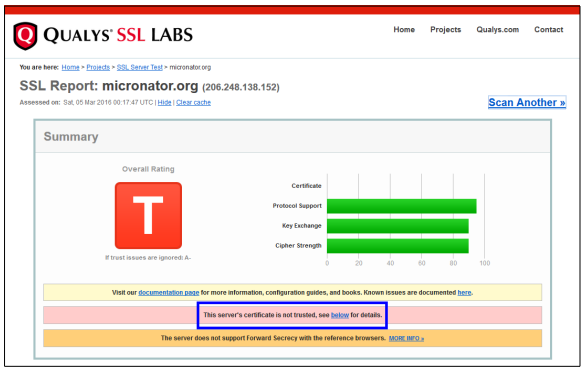
Une fois que le client **letsencrypt.sh** a obtenu notre certificat et configuré notre serveur, on roule un test avec un outil tel **SSLLabs.com**, **https://www.ssllabs.com/ssltest/**, pour nous assurer qu'il fonctionne correctement.

Il faut se souvenir que notre certificat en est un de TEST et que pour ce genre de certificat, l'émetteur est **happy hacker fake CA** qui n'est pas reconnu comme une véritable CA.

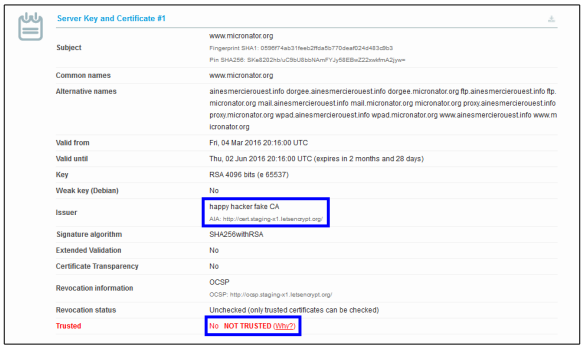
https://www.ssllabs.com/ssltest/



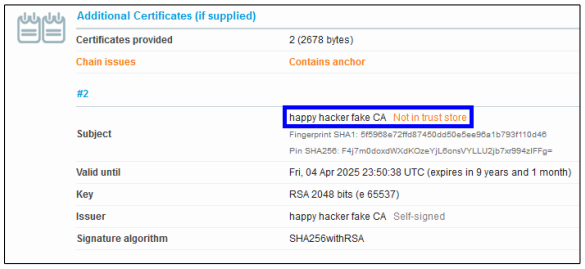
Le certificat n'est pas fiable.



L'émetteur est **happy hacker fake CA** et il n'est pas fiable car non reconnu comme une véritable *CA*.



L'émetteur **happy hacker fake CA** n'est pas parmi les émetteurs fiables.



La chaîne de certification ne pointe pas vers une **CA** fiable.

7. Conclusion

Le client **letsencrypt.sh** fonctionne bien de même que tous nos fichiers de configuration. Avec le **script de point d'entrée**: les propriétés de **modSSL** ont été modifiées et les changements signalés pour installer le nouveau certificat.

VIII- Renouvellement

1. Manuel

Situation actuelle

- Aucun ajout d'un nouveau domaine ou **CNAME**.
- Aucune modification des fichiers de configuration.
- Le premier certificat est toujours valide (*plus longtemps que 30 jours*) et il n'a pas été révoqué.

Si nous lancions le **client letsencrypt.sh**, il nous retournerait ce qui suit.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh -c

# INFO: Using main config file /etc/letsencrypt.sh/config
Processing www.micronator.org with alternative names: micronator.org dorgee.micronator.org
...
proxy.ainesmercierouest.info
 + Checking domain name(s) of existing cert... unchanged.
 + Checking expire date of existing cert...
 + Valid till Jun 10 14:29:00 2016 GMT (Longer than 30 days). Skipping!
[root@dorgee ~]#
```

Au début de l'exécution, le client **letsencrypt.sh**:

a) a examiné le fichier /etc/letsencrypt.sh/**config** et n'a vu aucune modification des paramètres,

b) a examiné le fichier **/etc/letsencrypt.sh/domains.txt** et vu qu'il n'avait pas été modifié par rapport au dernier certificat: **unchanged**,

c) a vérifié la validité du certificat et vu qu'il était encore valide pour plus de 30 jours: **Longer than 30 days**,

d) a affiché la dernière ligne du message: **Skipping!**, et il s'est arrêté sans aller plus loin,

e) arrêté, il ne s'est pas rendu au script **de point d'entrée**; les propriétés de **modSSL** sont demeurées inchangées et il n'y a eu aucun changement de signaler.

2. Manuel forcé

Nous avons plusieurs choix pour lancer le client **letsencrypt.sh** et l'obliger à renouveler notre premier certificat.

- Attendre la fin du certificat.
- Révoquer le certificat.
- Modifier un domaine du fichier **domains.txt**.
- Ajouter/enlever un domaine ou **CNAME** au fichier **domains.txt**.
- Utiliser l'option **--force** lors du lancement de la commande **letsencrypt.sh**.

On choisit de lancer le **client letsencrypt.sh** avec l'option **--force** pour imposer un renouvellement.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh -c --force

# INFO: Using main config file /etc/letsencrypt.sh/config
Processing www.micronator.org with alternative names: micronator.org dorgee.micronator.org
mail.micronator.org ftp.micronator.org wpad.micronator.org proxy.micronator.org
www.ainesmercierouest.info ainesmercierouest.info dorgee.ainesmercierouest.info
mail.ainesmercierouest.info ftp.ainesmercierouest.info wpad.ainesmercierouest.info
proxy.ainesmercierouest.info
 + Checking domain name(s) of existing cert... unchanged.
 + Checking expire date of existing cert...
 + Valid till Jun 10 14:29:00 2016 GMT (Longer than 30 days). Ignoring because renew was
forced!
 + Signing domains...
 + Generating signing request...
 + Requesting challenge for www.micronator.org...
 + Requesting challenge for micronator.org...
 + Requesting challenge for dorgee.micronator.org...
 + Requesting challenge for mail.micronator.org...
 + Requesting challenge for ftp.micronator.org...
 + Requesting challenge for wpad.micronator.org...
 + Requesting challenge for proxy.micronator.org...
 + Requesting challenge for www.ainesmercierouest.info...
 + Requesting challenge for ainesmercierouest.info...
 + Requesting challenge for dorgee.ainesmercierouest.info...
 + Requesting challenge for mail.ainesmercierouest.info...
 + Requesting challenge for ftp.ainesmercierouest.info...
 + Requesting challenge for wpad.ainesmercierouest.info...
 + Requesting challenge for proxy.ainesmercierouest.info...
 + Responding to challenge for www.micronator.org...
 + Challenge is valid!
 + Responding to challenge for micronator.org...
 + Challenge is valid!
 + Responding to challenge for dorgee.micronator.org...
 + Challenge is valid!
 + Responding to challenge for mail.micronator.org...
 + Challenge is valid!
 + Responding to challenge for ftp.micronator.org...
 + Challenge is valid!
 + Responding to challenge for wpad.micronator.org...
 + Challenge is valid!
 + Responding to challenge for proxy.micronator.org...
 + Challenge is valid!
 + Responding to challenge for www.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for dorgee.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for mail.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for ftp.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for wpad.ainesmercierouest.info...
 + Challenge is valid!
 + Responding to challenge for proxy.ainesmercierouest.info...
 + Challenge is valid!
  + Requesting certificate...
 + Checking certificate...
 + Done!
 + Creating fullchain.pem...
[root@dorgee ~]#
```

Aucune erreur, tout a bien fonctionné pour le renouvellement forcé.

Pour ce renouvellement, le client **letsencrypt.sh**:

- a commencé par analyser le fichier de configuration **config**, **# INFO: Using main config file /etc/letsencrypt.sh/config**.
- n'a pas généré une nouvelle clé de compte **Let's Encrypt**.
- n'a vu aucune modification dans les noms de domaines + Checking domain name(s) of existing cert... unchanged.
- a vérifié la date d'expiration du certificat, **+ Checking expire date of existing cert...**
- a ignoré le temps de validité restant, **Ignoring because renew was forced!**
- a signé les domaines **+ Signing domains...**
- n'a pas créé de nouveaux répertoires.
- a créé une nouvelle requête CSR, **+ Generating signing request...**
- a demandé les défis, attendu leurs validités et vérifié leurs réponses.
- a fait la demande du certificat, **+ Requesting certificate...**,
- une fois reçu, il a vérifié le certificat, **+ Checking certificate...**,
- Le tout terminé **+ Done!**, il a créé la chaîne de certification et ajusté les pointeurs.
- Il a alors appelé le **script de point d'entrée** et celui-ci a modifié les propriétés de **modSSL** et signalé les changements pour activer le nouveau certificat.

Tous les répertoires existaient, la ligne ci-dessous est donc manquante avec ce renouvellement.

```
 + Creating new directory /etc/letsencrypt.sh/certs/www.micronator.org ...
```

Avec le premier certificat, **Let's Encrypt** avait généré localement une clé **SSL** privée pour le serveur.

Avec ce renouvellement, cette ligne est manquante. La clé privée **SSL** du serveur demeure la même.

```
 + Generating private key...
```

2.1. Vérification

On vérifie le répertoire **/etc/letsencrypt.sh/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/

total 112
 4 drwxr-xr-x   6 root root  4096 17 juil. 00:44 .
12 drwxr-xr-x 100 root root 12288 17 juil. 00:26 ..
 4 drwx------   3 root root  4096 16 juil. 23:55 accounts
 4 drwx------   3 root root  4096 16 juil. 23:56 certs
 4 -rw-r--r--   1 root root  1406 15 juil. 23:51 CHANGELOG
 4 -rwx------   1 root root   321 16 juil. 23:45 config
 4 drwxr-xr-x   3 root root  4096 15 juil. 23:51 docs
 4 -rw-r--r--   1 root root   328 16 juil. 00:41 domains.txt
 4 drwxr-xr-x   8 root root  4096 15 juil. 23:51 .git
 4 -rw-r--r--   1 root root   108 15 juil. 23:51 .gitignore
 4 -rwx------   1 root root   311 16 juil. 00:56 letsencrypt-hook.sh
40 -rwx------   1 root root 37772 15 juil. 23:51 letsencrypt.sh
 4 -rw-r--r--   1 root root  1080 15 juil. 23:51 LICENSE
 4 -rw-r--r--   1 root root  3040 15 juil. 23:51 README.md
 8 -rwxr-xr-x   1 root root  8048 15 juil. 23:51 test.sh
 4 -rw-r--r--   1 root root   107 15 juil. 23:51 .travis.yml
[root@dorgee ~]#
```

Aucun nouveau répertoire n'a été créé. La clé de compte **Let's Encrypt** est la même.

On vérifie le répertoire **/etc/letsencrypt.sh/certs/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/certs/

total 12
4 drwx------ 3 root root 4096 12 mars  10:27 .
4 drwxr-xr-x 4 root root 4096 12 mars  12:02 ..
4 drwx------ 2 root root 4096 12 mars  12:02 www.micronator.org
[root@dorgee ~]#
```

Aucun nouveau répertoire n'a été créé.

On vérifie le répertoire **/etc/letsencrypt.sh/certs/www.micronator.org/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/certs/www.micronator.org/

total 48
4 drwx------ 2 root root 4096 12 mars  12:02 .
4 drwx------ 3 root root 4096 12 mars  10:27 ..
4 -rw------- 1 root root 2118 12 mars  10:27 cert-1457796476.csr
4 -rw------- 1 root root 2614 12 mars  10:28 cert-1457796476.pem
4 -rw------- 1 root root 2118 12 mars  12:01 cert-1457802076.csr
4 -rw------- 1 root root 2614 12 mars  12:02 cert-1457802076.pem
0 lrwxrwxrwx 1 root root   19 12 mars  12:02 cert.csr -> cert-1457802076.csr
0 lrwxrwxrwx 1 root root   19 12 mars  12:02 cert.pem -> cert-1457802076.pem
4 -rw------- 1 root root 1123 12 mars  10:28 chain-1457796476.pem
4 -rw------- 1 root root 1123 12 mars  12:02 chain-1457802076.pem
0 lrwxrwxrwx 1 root root   20 12 mars  12:02 chain.pem -> chain-1457802076.pem
4 -rw------- 1 root root 3737 12 mars  10:28 fullchain-1457796476.pem
4 -rw------- 1 root root 3737 12 mars  12:02 fullchain-1457802076.pem
0 lrwxrwxrwx 1 root root   24 12 mars  12:02 fullchain.pem -> fullchain-1457802076.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem_T_2016-03-12_10h27
0 lrwxrwxrwx 1 root root   22 12 mars  10:28 privkey.pem -> privkey-1457796476.pem
[root@dorgee ~]#
```

Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

- Nouvelle requête, **cert-1457802076.csr**.
- Nouveau certificat, **cert-1457802076.pem**.
- Nouvelle chaîne de certification, **chain-1457802076.pem**.
- La clé privée du serveur, **privkey-1457796476.pem** est demeurée la même.
- Les pointeurs ont été ajustés vers ***-1457802076*** sauf celle de la clé privée.

Les propriétés de **modSSL** ont été modifiées mais vu que **letsencrypt.sh** utilise des pointeurs qui ne changent jamais, il semble que les pointeurs soient demeurés les mêmes. C'est la commande ***/sbin/e-smith/signal-event ssl-update*** du fichier ***/etc/letsencrypt.sh/letsencrypt-hook.sh*** qui les a pourtant modifiés.

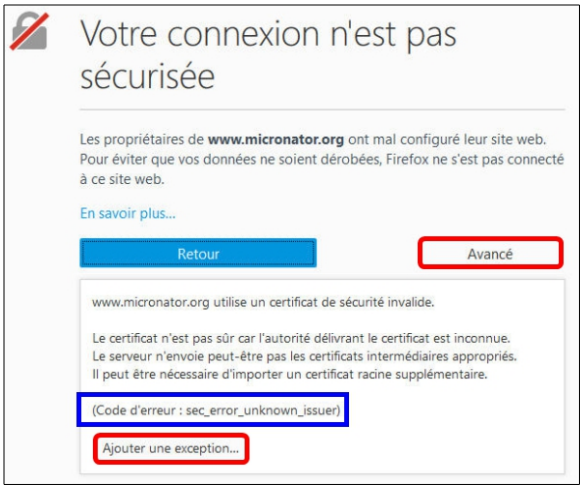
```
[root@dorgee ~]# config show modSSL

modSSL=service
    CertificateChainFile=/etc/letsencrypt.sh/certs/www.micronator.org/chain.pem
    TCPPort=443
    access=public
    crt=/etc/letsencrypt.sh/certs/www.micronator.org/cert.pem
    key=/etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem
    status=enabled
[root@dorgee ~]#
```


2.2. Navigateurs WEB

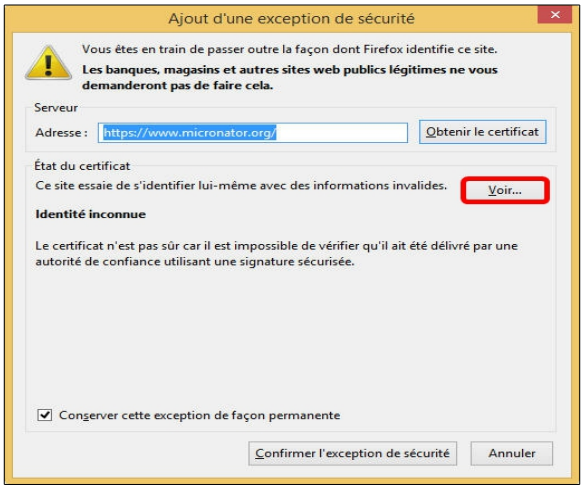
Notre certificat de TEST a été émis par un renouvellement forcé, en spécifiant le paramètre **--force**. L'émetteur est toujours **happy hacker fake CA** et nous aurons encore l'erreur **sec_error_unknown_issuer**.

Firefox & premier domaine

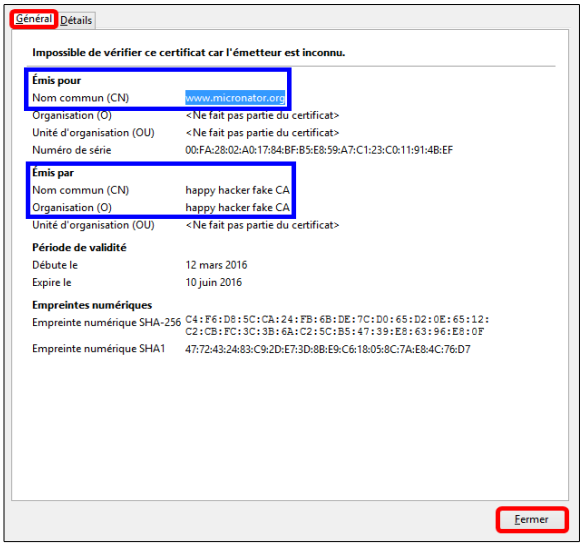
- Site: **https://www.micronator.org/**.
- On reçoit l'erreur normale pour une clé signée par une *CA* non reconnue.
- **Avancé | Ajouter une exception...**



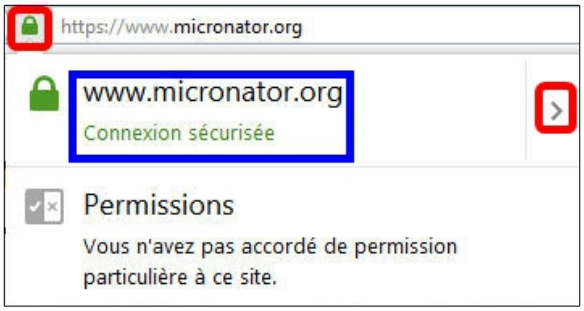
Voir...



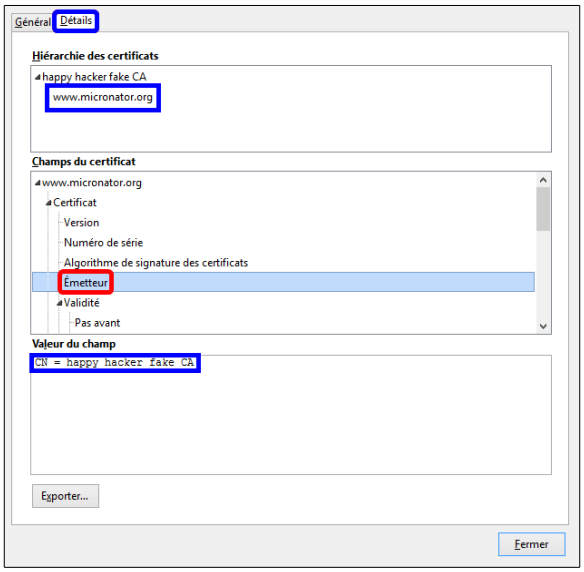
- Onglet **Général**.
- Notre certificat en est un de TEST, l'émetteur est toujours **happy hacker fake CA**.
- **Fermer | Confirmer l'exception de sécurité**.



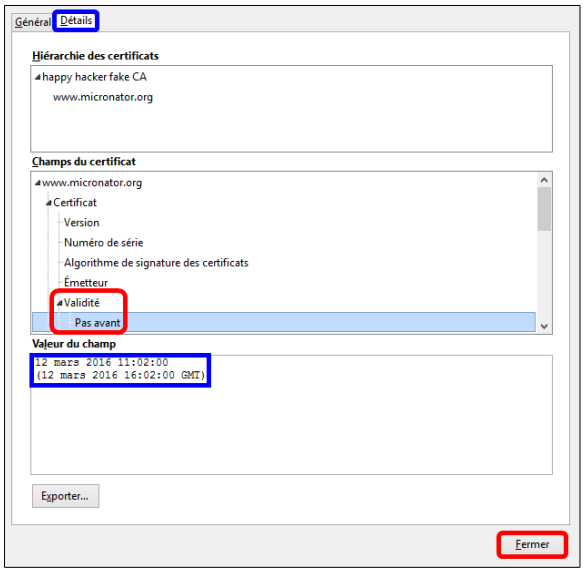
- Après la confirmation de l'exception de sécurité, la page d'acceuil de **https://www.micronator.org** s'affiche.
- Le cadenas est vert, car la connexion est sécurisée.
- On clique l'**icône** > à droite.



- Onglet **Détails**.
- Paramètres identiques à ceux du certificat précédent.



- **Validité | Pas avant**.
- Le certificat a été émis le **12 mars 2016 11:02:00**.
- Seule l'heure de validation est différente.
- On ferme toutes les fenêtres.



2.3. Conclusion

Le client **letsencrypt.sh** fonctionne correctement ainsi que tous nos fichiers de configuration. Avec le **script de point d'entrée**: les propriétés de **modSSL** ont été modifiés, les changements signalés et le certificat installé.

3. Automatique

3.1. Introduction

Dans le cadre de la sécurité de **Let's Encrypt**, les certificats doivent être renouvelés tous les 3 mois (*90 jours*).

On peut créer une tâche **cron** qui sera activée tous les mois pour vérifier si notre certificat est dû pour un renouvellement et si oui, le renouveler automatiquement.

- On peut consulter la **contrib** **Crontab Manager** pour une méthode de manipulation des tâches **cron**.
- Pour une explication complète du fonctionnement des tâches **cron**, on peut consulter l'excellent article de **Wikipédia**: **https://fr.wikipedia.org/wiki/Cron**.

3.1.1. Marche à suivre

On va créer une tâche **cron** qui s'exécutera une seule fois, **10** minutes après sa mise en place, pour vérifier qu'elle fonctionne correctement. Après ce test, on l'effacera et on la répliquera lors du mode **Officiel**.

- ☐ On crée le répertoire pour le gabarit personnalisé: **/etc/e-smith/templates-custom/etc/crontab**.
- ☐ Dans le répertoire du gabarit personnalisé, on crée le fichier cron: **renouvelerSSL**.
- ☐ On développe le gabarit personnalisé.
- ☐ On suit la tâche cron jusqu'à l'exécution de la tâche.
- ☐ On efface la tâche cron après avoir vérifié son bon fonctionnement.

3.2. Tâche cron

3.2.1. Gabarit personnalisé

Création du répertoire pour le gabarit.

```
[root@dorgee ~]# mkdir -p /etc/e-smith/templates-custom/etc/crontab

[root@dorgee ~]#
```

3.2.2. Création

On affiche l'heure.

```
[root@dorgee ~]# date

sam. mars 12 13:23:35 EST 2016
[root@dorgee ~]#
```

Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

On crée le fichier de la tâche **cron** et on y insère son contenu. La tâche sera lancée dans **10** minutes à **13:33**.

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL <<'EOT'

#
# Tâche cron qui lance le client letsencrypt.sh pour le renouvellement du certificat
#
# Si le certificat est encore valide pour plus de 30 jours, qu'il n'y a eu aucune
# modification des fichiers de configuration et aucun changement dans le fichier
# domains.txt, le client ne fera rien et attendra son prochain lancement.
#
# Si le certificat est encore valide pour moins de 30 jours, le client:
# 1) demandera un renouvellement du certificat,
# 2) ajustera les pointeurs des fichiers du certificat,
# 3) appellera le script de point d'entrée qui ajustera les paramètres de modSSL
#    et signalera les changements puis, le script letsencrypt.sh s'arrêtera.
#
# ┌──────────────── min (0 - 59)
# │ ┌────────────── heure (0 - 23)
# │ │ ┌──────────── jour du mois (1 - 31)
# │ │ │ ┌────────── mois (1 - 12)
# │ │ │ │ ┌──────── jour de la semaine (0 - 6) (0 à 6 sont de dimanche à samedi,
# │ │ │ │ │                 7 est dimanche, même que 0)
# │ │ │ │ │
# * * * * *  [usager] commande à exécuter
33 13 12 3 * root /etc/letsencrypt.sh/letsencrypt.sh -c
#
EOT
```


On vérifie le contenu de la tâche **cron**.

```
[root@dorgee ~]# cat /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL

#
# Tâche cron qui lance le client letsencrypt.sh pour le renouvellement du certificat
#
# Si le certificat est encore valide pour plus de 30 jours, qu'il n'y a eu aucune
# modification des fichiers de configuration et aucun changement dans le fichier
# domains.txt, le client ne fera rien et attendra son prochain lancement.
#
# Si le certificat est encore valide pour moins de 30 jours, le client:
# 1) demandera un renouvellement du certificat,
# 2) ajustera les pointeurs des fichiers du certificat,
# 3) appellera le script de point d'entrée qui ajustera les paramètres de modSSL
#    et signalera les changements puis, le script letsencrypt.sh s'arrêtera.
#
# ┌──────────────── min (0 - 59)
# │ ┌────────────── heure (0 - 23)
# │ │ ┌──────────── jour du mois (1 - 31)
# │ │ │ ┌────────── mois (1 - 12)
# │ │ │ │ ┌──────── jour de la semaine (0 - 6) (0 à 6 sont de dimanche à samedi,
# │ │ │ │ │                 7 est dimanche, même que 0)
# │ │ │ │ │
# * * * * *  [usager] commande à exécuter
33 13 12 3 * root /etc/letsencrypt.sh/letsencrypt.sh -c
#
[root@dorgee ~]#
```

On développe le gabarit personnalisé.

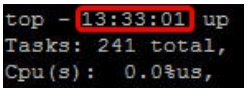
```
[root@dorgee ~]# expand-template /etc/crontab

[root@dorgee ~]#
```

On redémarre le service **crontab**.

```
[root@dorgee ~]# service crond restart

Arrêt de crond :                                           [  OK  ]
Démarrage de crond :                                       [  OK  ]
[root@dorgee ~]#
```

3.2.3. Suivi

Avec **PuTTY**, on lance un nouvel écran de connexion au serveur et on se logue en **root**. Avec cet écran, dans le coin supérieur gauche, on pourra suivre l'heure sur le serveur en lançant la commande **top -d 1**.



Sur l'écran **PuTTY** original, on lance la commande ci-dessous pour suivre toutes les tâches **cron** et on filtre avec **grep** celle qui contiendra **letsencrypt.sh**.

```
tail -F /var/log/cron | grep letsencrypt.sh
```

Après quelques minutes, à **13H33**, on verra notre tâche s'afficher.

```
Mar 12 13:33:01 dorgee CROND[8430]: (root) CMD (/etc/letsencrypt.sh/letsencrypt.sh -c)
```

On arrête la commande **tail** avec [**CTL - c**] et celle de **top** avec [**q**].

```
Mar 12 13:33:01 dorgee CROND[8430]: (root) CMD (/etc/letsencrypt.sh/letsencrypt.sh)
^C
[root@dorgee ~]#
```


3.2.4. Élimination

La vérification du fonctionnement de la tâche **cron** étant terminée, on l'élimine.

```
[root@dorgee ~]# rm /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL

rm : supprimer fichier « /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL  » ? y
[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# ls -als /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL

ls: impossible d'accéder à /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL: Aucun
fichier ou dossier de ce type
[root@dorgee ~]#
```

On développe le gabarit **crontab** pour qu'il se mette à jour.

```
[root@dorgee ~]# expand-template /etc/crontab

[root@dorgee ~]#
```

On redémarre le service **crontab**.

```
[root@dorgee ~]# service crond restart

Arrêt de crond :                                           [  OK  ]
Démarrage de crond :                                       [  OK  ]
[root@dorgee ~]#
```

3.3. Vérification

Il ne devrait y avoir aucun changement, car le certificat est encore valide.

On affiche tous les certificats émis à date.

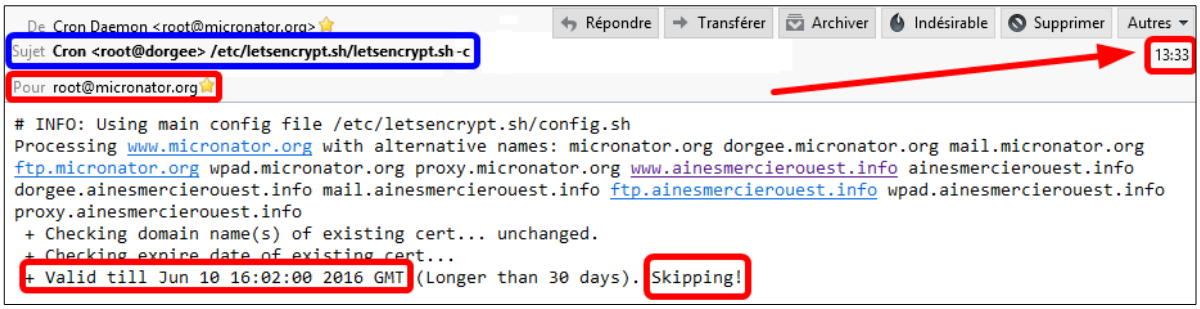
```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/certs/www.micronator.org/

total 40
4 -rw------- 1 root root 2118 12 mars  10:27 cert-1457796476.csr
4 -rw------- 1 root root 2614 12 mars  10:28 cert-1457796476.pem
4 -rw------- 1 root root 2118 12 mars  12:01 cert-1457802076.csr
4 -rw------- 1 root root 2614 12 mars  12:02 cert-1457802076.pem
0 lrwxrwxrwx 1 root root   19 12 mars  12:02 cert.csr -> cert-1457802076.csr
0 lrwxrwxrwx 1 root root   19 12 mars  12:02 cert.pem -> cert-1457802076.pem
4 -rw------- 1 root root 1123 12 mars  10:28 chain-1457796476.pem
4 -rw------- 1 root root 1123 12 mars  12:02 chain-1457802076.pem
0 lrwxrwxrwx 1 root root   20 12 mars  12:02 chain.pem -> chain-1457802076.pem
4 -rw------- 1 root root 3737 12 mars  10:28 fullchain-1457796476.pem
4 -rw------- 1 root root 3737 12 mars  12:02 fullchain-1457802076.pem
0 lrwxrwxrwx 1 root root   24 12 mars  12:02 fullchain.pem -> fullchain-1457802076.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem_T_2016-03-12_10h27
0 lrwxrwxrwx 1 root root   22 12 mars  10:28 privkey.pem -> privkey-1457796476.pem
[root@dorgee ~]#
```

Rien n'a changé dans le répertoire des certificats, les cibles des pointeurs sont toujours les mêmes.

Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

3.4. Courriel de notification

Réception d'un courriel par **root**.



À **13h33**, **root** (*admin*) a reçu un courriel du **daemon cron** disant qu'une tâche avait exécuté le programme **/etc/letsencrypt.sh/letsencrypt.sh -c**.

Notre certificat en était un de **TEST**, il n'y a pas eu de nouveaux certificats d'émis, car le certificat est encore valide pour plus de **30** jours et nous n'avons pas utilisé l'option **--force**.

Le renouvellement a donc été outrepassé mais, on a démontré que la tâche **cron** fonctionnait.

Notre procédé fonctionne parfaitement,

il pourra être étendu au **client letsencrypt.sh** en mode **Officiel**.

IX- Demande d'un certificat officiel

1. Introduction

Nous avons vérifié nos fichiers de configuration, obtenu deux certificats, vérifié le renouvellement manuel et automatique de notre certificat de TEST et nous n'avons pas encore utilisé un seul certificat de notre limite **5/7**; nous pouvons donc maintenant demander un certificat **officiel**.

1.1. Marche à suivre

- Demande manuel.
- Demande forcée.
- Automatique.
- Vérification du bon fonctionnement du certificat.

2. Manuel

Situation présente

- Nous n'avons ajouté aucun nouveau domaine ou **CNAME**.
- Nous n'avons pas révoqué notre dernier certificat.
- Notre dernier certificat est toujours valide.
- La ligne spécifiant la **CA** de TEST *acme-staging* n'est pas commentée dans le fichier **config**.

2.1. Modification du fichier /etc/letsencrypt.sh/config

On commente la ligne de la **CA** de TEST dans le fichier **config**.

```
[root@dorgee ~]# sed -i 's/^CA="https:/# CA="https:/' /etc/letsencrypt.sh/config

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/config

#!/bin/bash
# config
# CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@micronator.org"

[root@dorgee ~]#
```

⚠ Il n'y a pas de ligne vide avant la ligne **#!/bin/bash**. Ci-dessus nous avons inséré une ligne vide pour faciliter la copie de la commande.

Si nous demandons un certificat **OFFICIEL** à la **CA**, (*L'Émetteur officiel de Let's Encrypt est acme-v01*) le client **letsencrypt.sh** vérifiera si le certificat est encore valide pour plus de 30 jours et si oui, il affichera le message ci-dessous.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh -c

# INFO: Using main config file /etc/letsencrypt.sh/config
Processing www.micronator.org with alternative names: micronator.org dorgee.micronator.org
mail.micronator.org ftp.micronator.org wpad.micronator.org proxy.micronator.org
www.ainesmercierouest.info ainesmercierouest.info dorgee.ainesmercierouest.info
mail.ainesmercierouest.info ftp.ainesmercierouest.info wpad.ainesmercierouest.info
proxy.ainesmercierouest.info
 + Checking domain name(s) of existing cert... unchanged.
 + Checking expire date of existing cert...
 + Valid till Jun 10 16:02:00 2016 GMT (Longer than 30 days). Skipping!
[root@dorgee ~]#
```

⚠ Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

3. Manuel forcé

3.1. Lancement de la demande

Nous avons plusieurs choix pour forcer le renouvellent d'un certificat.

- Attendre la fin du certificat
- Révoquer le certificat
- Modifier un domaine du fichier **/etc/letsencrypt.sh/domains.txt**.
- Ajouter/enlever un domaine ou **CNAME** au fichier **/etc/letsencrypt.sh/domains.txt**.
- Utiliser l'option **--force** lors du lancement de la commande **letsencrypt.sh**.

On utilise l'option **--force**, car la clé est encore active pour plus de **30** jours.

```
root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh -c --force

# INFO: Using main config file /etc/letsencrypt.sh/config
+ Generating account key...
+ Registering account key with letsencrypt...
Processing www.micronator.org with alternative names: micronator.org
dorgee.micronator.org ...
proxy.ainesmercierouest.info
 + Checking domain name(s) of existing cert... unchanged.
 + Checking expire date of existing cert...
 + Valid till Jun 10 16:02:00 2016 GMT (Longer than 30 days). Ignoring because renew was
forced!
 + Signing domains...
 + Generating private key...
 + Generating signing request...
 + Requesting challenge for www.micronator.org...
...
 + Responding to challenge for www.micronator.org...
 + Challenge is valid!
...
 + Requesting certificate...
 + Checking certificate...
 + Done!
 + Creating fullchain.pem...
[root@dorgee ~]#
```

Il n'y a pas eu d'erreur.

- Le client **letsencrypt.sh** a utilisé de fichier de configuration par défaut /etc/letsencrypt.sh/**config**.
- Il a créé une clé de compte pour la **CA** officielle, **+ Generating account key...**, créé un nouveau sous-répertoire dans **/etc/letsencrypt.sh/accounts/** pour le compte **OFFICIEL** et y a déposé la clé de compte pour la **CA** officielle;
  exemple: ***/etc/letsencrypt.sh/accounts/****aHR0cHM6Ly9hY21lAAYwMS5hcGkubGV0c2VuY3J5-cHQub3JnL2RpcmVjdG9yeTT/***account_key.pem**. Les droits de cette clé sont limités à **root** seulement,
- Il a enregistré la clé de compte chez la CA officielle, **+ Registering account key with letsencrypt...**
- La **CA acme-v01**, l'**Émettrice officielle**, a créé un compte-usager au nom de la nouvelle clé de compte.
- Le client a ignoré la validité du certificat **Ignoring because renew was forced!**
- La suite est la même que pour le certificat précédent.

4. Vérification du nouveau certificat

4.1. Console du serveur

On vérifie le répertoire **/etc/letsencrypt.sh/**.

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/

total 112
 4 drwxr-xr-x   6 root root  4096 17 juil. 01:41 .
12 drwxr-xr-x 100 root root 12288 17 juil. 01:26 ..
 4 drwx------   4 root root  4096 17 juil. 01:40 accounts
 4 drwx------   3 root root  4096 16 juil. 23:56 certs
 4 -rw-r--r--   1 root root  1406 15 juil. 23:51 CHANGELOG
 4 -rwx------   1 root root   323 17 juil. 01:31 config
 4 drwxr-xr-x   3 root root  4096 15 juil. 23:51 docs
 4 -rw-r--r--   1 root root   328 16 juil. 00:41 domains.txt
 4 drwxr-xr-x   8 root root  4096 15 juil. 23:51 .git
 4 -rw-r--r--   1 root root   108 15 juil. 23:51 .gitignore
 4 -rwx------   1 root root   311 16 juil. 00:56 letsencrypt-hook.sh
40 -rwx------   1 root root 37772 15 juil. 23:51 letsencrypt.sh
 4 -rw-r--r--   1 root root  1080 15 juil. 23:51 LICENSE
 4 -rw-r--r--   1 root root  3040 15 juil. 23:51 README.md
 8 -rwxr-xr-x   1 root root  8048 15 juil. 23:51 test.sh
 4 -rw-r--r--   1 root root   107 15 juil. 23:51 .travis.yml
[root@dorgee ~]#
```

Aucun nouveau répertoire n'a été créé.

On vérifie le répertoire **/etc/letsencrypt.sh/certs/**.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/certs/

total 4
4 drwx------ 2 root root 4096 17 juil. 01:40 www.micronator.org
[root@dorgee ~]#
```

Aucun nouveau répertoire n'a été créé.

On vérifie le répertoire **/etc/letsencrypt.sh/certs/www.micronator.org/**.

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/certs/www.micronator.org/

total 64
4 -rw------- 1 root root 2118 12 mars  10:27 cert-1457796476.csr
4 -rw------- 1 root root 2614 12 mars  10:28 cert-1457796476.pem
4 -rw------- 1 root root 2118 12 mars  12:01 cert-1457802076.csr
4 -rw------- 1 root root 2614 12 mars  12:02 cert-1457802076.pem
4 -rw------- 1 root root 2118 12 mars  14:57 cert-1457812624.csr
0 -rw------- 1 root root    0 12 mars  14:57 cert-1457812624.pem
4 -rw------- 1 root root 2118 12 mars  15:59 cert-1457816391.csr
4 -rw------- 1 root root 2598 12 mars  16:00 cert-1457816391.pem
0 lrwxrwxrwx 1 root root   19 12 mars  16:00 cert.csr -> cert-1457816391.csr
0 lrwxrwxrwx 1 root root   19 12 mars  16:00 cert.pem -> cert-1457816391.pem
4 -rw------- 1 root root 1123 12 mars  10:28 chain-1457796476.pem
4 -rw------- 1 root root 1123 12 mars  12:02 chain-1457802076.pem
4 -rw------- 1 root root 1675 12 mars  16:00 chain-1457816391.pem
0 lrwxrwxrwx 1 root root   20 12 mars  16:00 chain.pem -> chain-1457816391.pem
4 -rw------- 1 root root 3737 12 mars  10:28 fullchain-1457796476.pem
4 -rw------- 1 root root 3737 12 mars  12:02 fullchain-1457802076.pem
8 -rw------- 1 root root 4273 12 mars  16:00 fullchain-1457816391.pem
0 lrwxrwxrwx 1 root root   24 12 mars  16:00 fullchain.pem -> fullchain-1457816391.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem_T_2016-03-12_10h27
0 lrwxrwxrwx 1 root root   22 12 mars  10:28 privkey.pem -> privkey-1457796476.pem
[root@dorgee ~]#
```

Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

- Nouvelle requête, **cert-1457816391.csr**.
- Nouveau certificat, **cert-1457816391.pem**.
- Nouvelle chaîne de certification, **chain-1457816391.pem**.
- La clé privée du **Serveur SME**, **privkey-1457796476.pem** est demeurée la même depuis le tout début.
- Les pointeurs ont été ajustés vers ***-1457816391*** sauf celle de la clé privée.

modSSL

```
[root@dorgee ~]# config show modSSL

modSSL=service
    CertificateChainFile=/etc/letsencrypt.sh/certs/www.micronator.org/chain.pem
    TCPPort=443
    access=public
    crt=/etc/letsencrypt.sh/certs/www.micronator.org/cert.pem
    key=/etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem
    status=enabled
[root@dorgee ~]#
```

Les propriétés de **modSSL** ne semblent pas avoir été modifiées, mais elles le furent par les commandes du script de point d'entrée. Le client **letsencrypt.sh** utilise des pointeurs qui eux, ne changent jamais.

Ce sont les cibles des pointeurs que **letsencrypt.sh** a modifiées. Si, précédemment, nous avions eu un certificat d'une autre **CA**, nous aurions vu les modifications apportées aux propriétés de **modSSL**.

Fichier pem

On affiche la date de création du fichier **pem**.

```
[root@dorgee ~]# ls -ls /home/e-smith/ssl.pem/dorgee.micronator.org.pem

8 -rw-r--r-- 1 root root 7897 12 mars  16:00 /home/e-smith/ssl.pem/dorgee.micronator.org.pem
[root@dorgee ~]#
```

Le fichier **pem** vient tout juste d'être recréé; le nouveau certificat a été installé et il est fonctionnel.

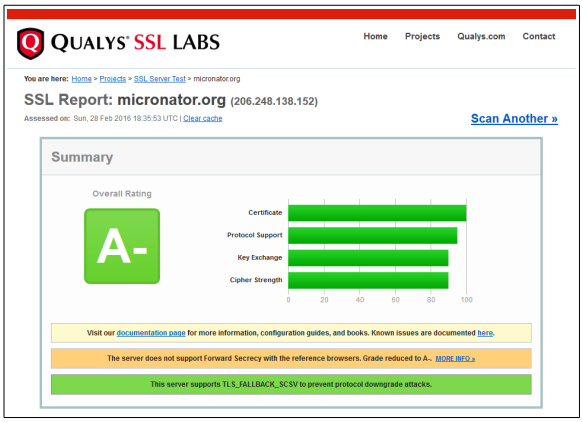
⚠ Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

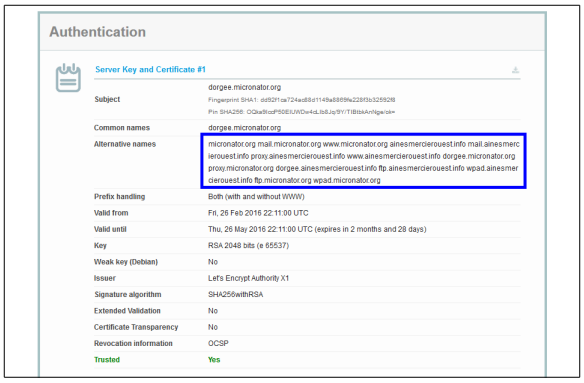
4.2. Qualys SSL Lab

⚠ Les dates, heures et certificats sont ceux de la version 0.0.1 de ce document.

Sur le site **https://www.ssllabs.com/ssltest/**, on entre le **FQDN** de notre premier domaine | **Submit**.

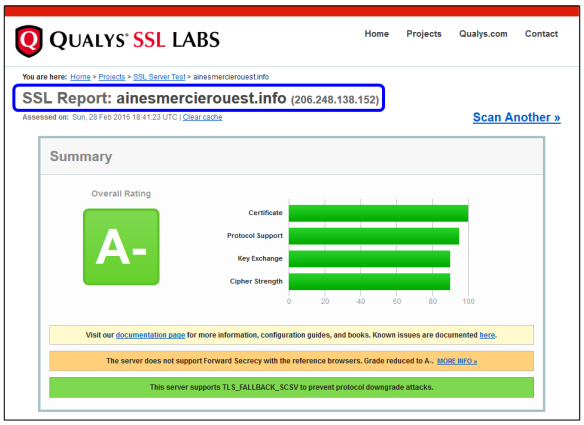
L'analyse peut prendre plusieurs minutes, il faut être patient. Les résultats sont très complets.

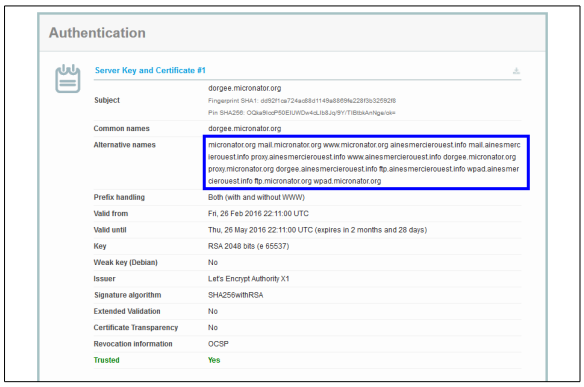




Le **Serveur SME** ne supporte pas **Forward Secrecy**.

On retourne sur le site de **Qualys SSL Lab** et on entre le **FQDN** de notre deuxième domaine | **Submit**.





4.3. Navigateur Web

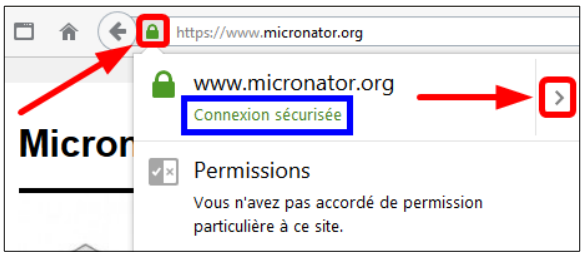
Firefox & premier domaine

Dans le site **micronator.org**, nous avons ajouté l'extension **WordPress HTTPS** qui est destinée à être une solution tout-en-un pour l'utilisation de **SSL** sur les sites **WordPress**. Cette extension oblige toutes les pages à utiliser **https** et ainsi tout leur contenu est sécurisé.

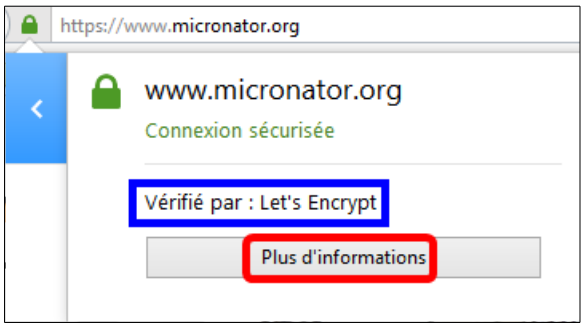
Si cette extension n'est pas présente, il se pourrait que certaines images n'apparaissent pas.

La **CA acme-v01** étant dans le magasin des **CA** et de ce fait reconnue, le certificat est automatiquement accepté.

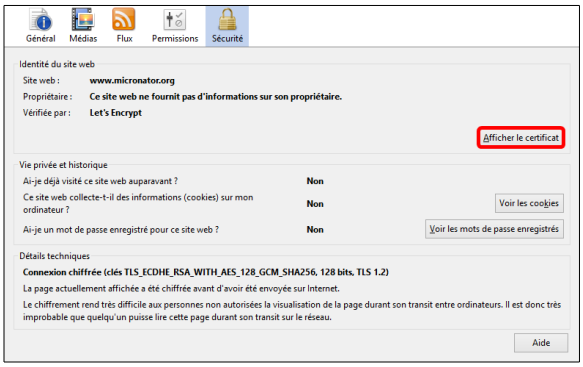
- Site: **https://www.micronator.org/**.
- Le cadenas est vert, on le clique.
- On voit que la connexion est sécurisée.
- On clique l'icône > à droite.



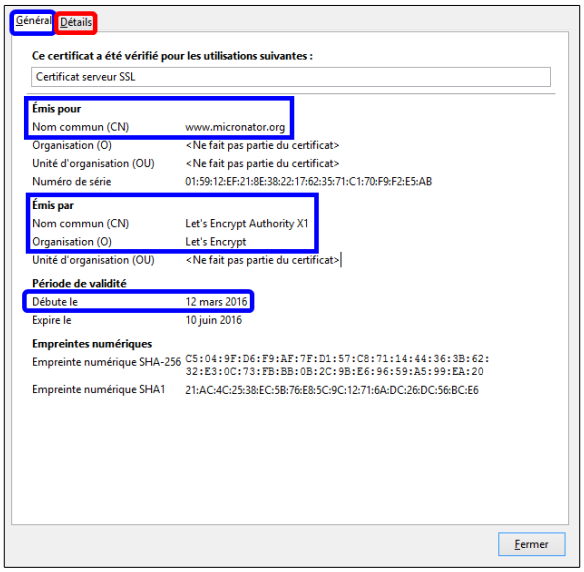
- Plus d'informations.



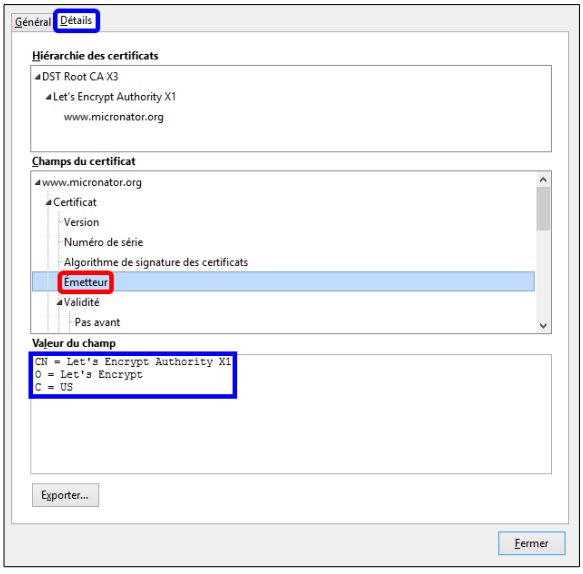
Afficher le certificat.



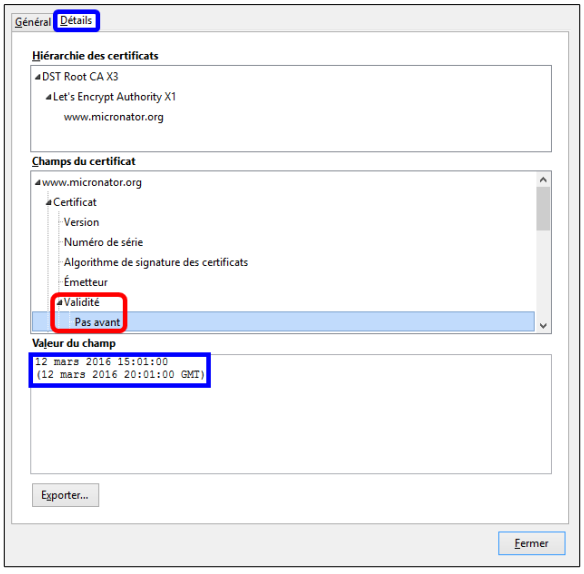
- Onglet Général.
- ***Émis pour www.micronator.org***.
- ***Émis par Let's Encrypt Authority X1***.
- On clique l'onglet **Détails**.



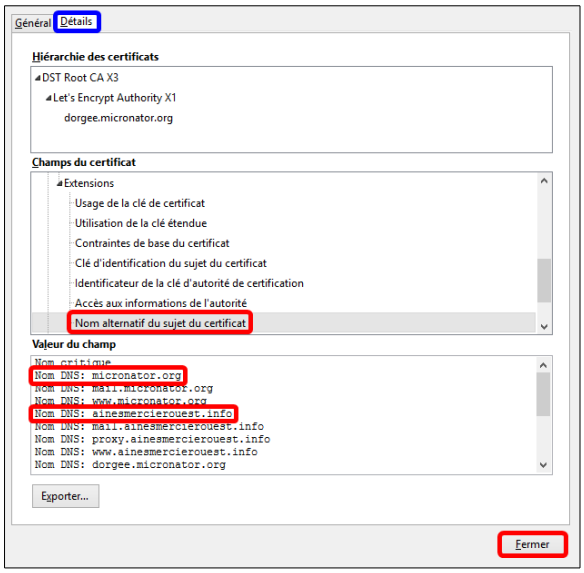
- **Émetteur**.
- Les détails de **Let's Encrypt Authority X1** sont affichés.



- **Validité | Pas avant**.
- La date et l'heure de l'émission sont affichées.

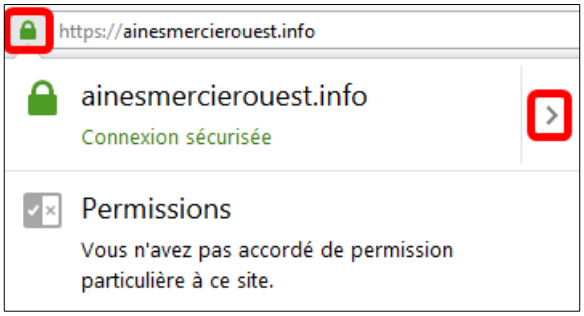


- **Nom alternatif du sujet du certificat**.
- Tous les domaines couverts sont affichés.
- On ferme toutes les fenêtres.

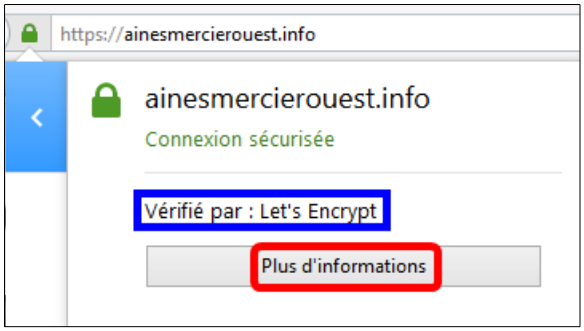


Firefox & deuxième domaine

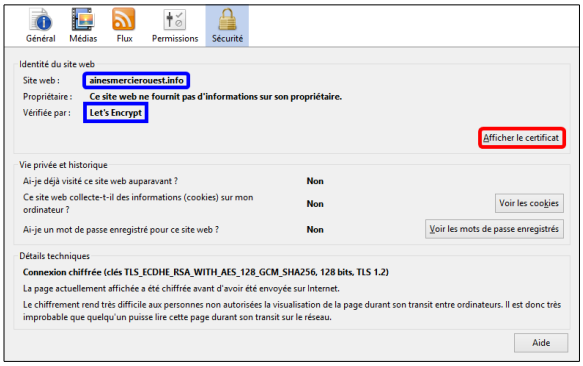
- Site: **https://www.ainesmercierouest.info/**.
- Le cadenas est vert, on le clique.
- On voit que la connexion est sécurisée.
- On clique l'icône > à droite.



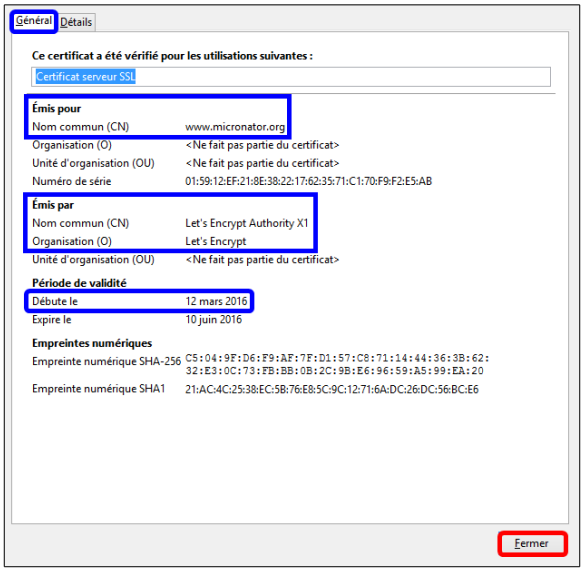
- **Plus d'informations**.



- ***Site web:*** **ainesmercierouest.info**.
- **Afficher le certificat**.

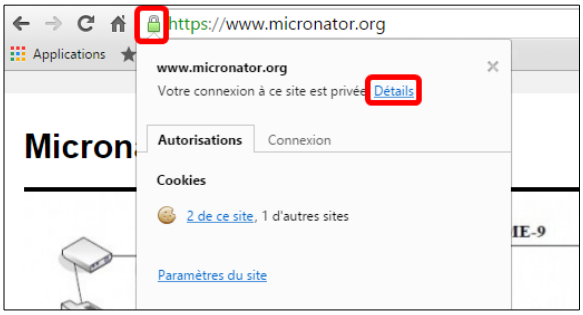


- Les informations du **certificat** sont affichées.
- On ferme toutes les fenêtres.

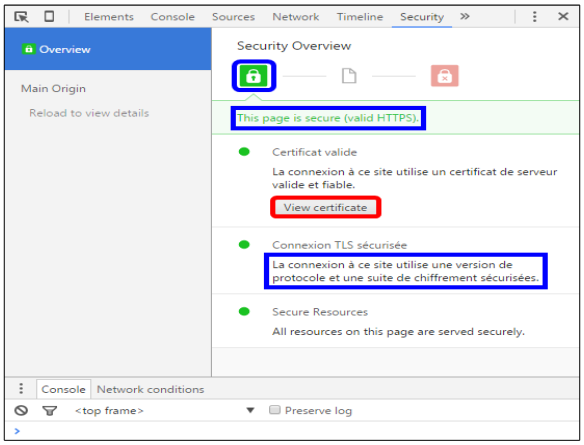


Google Chrome

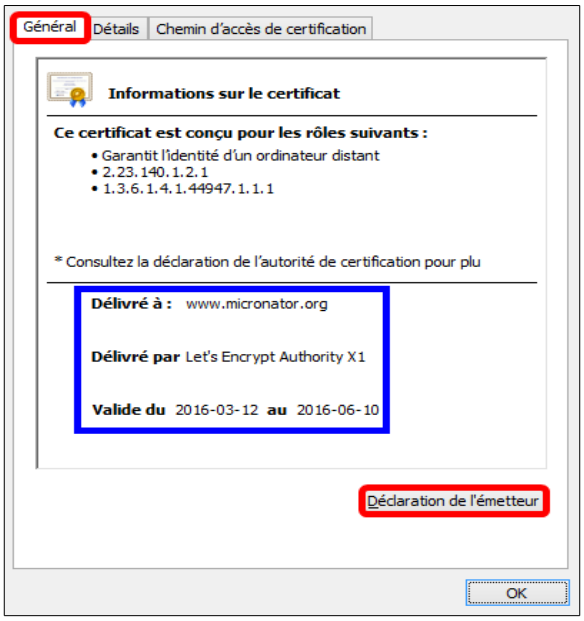
- Site: **https://www.micronator.org/**.
- Le cadenas est vert, on le clique.
- **Détails**.



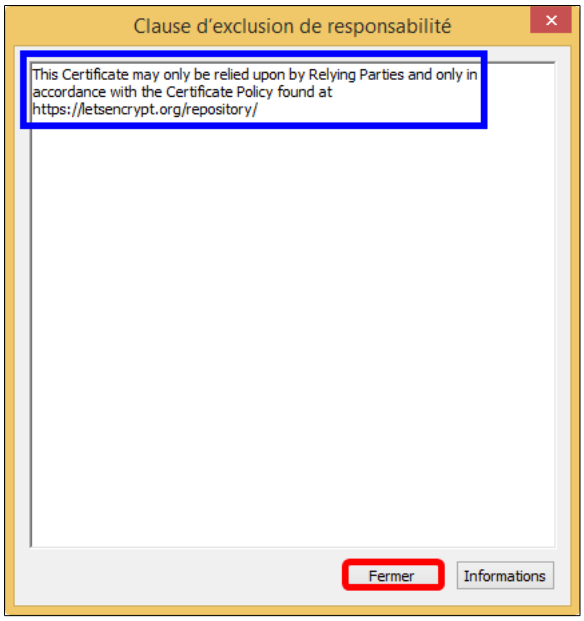
- La page est sécurisée.
- **View certificate**.



- Les informations du **certificat** sont affichées.
- **Déclaration de l'émetteur**.

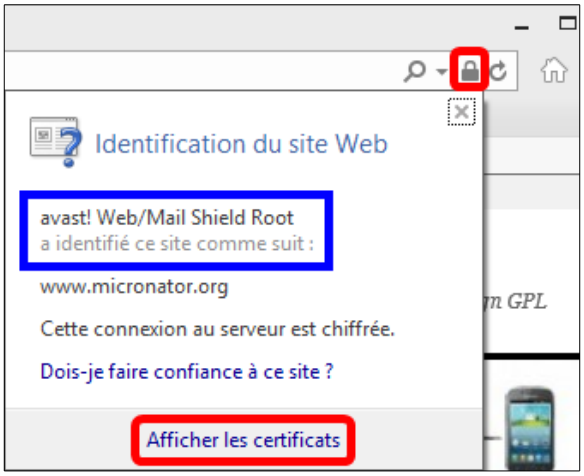


- La clause d'exclusion de responsabilité est affichée.
- On ferme toutes les fenêtres.

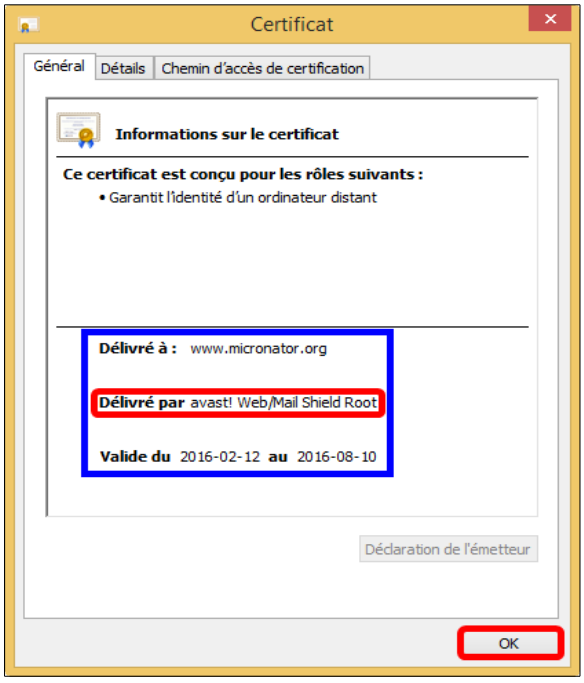


Internet Explorer

- Site: **https://www.micronator.org/**.
- Il semble que c'est *avast!* qui a émis le certificat?
- On clique **Afficher le certificat**.

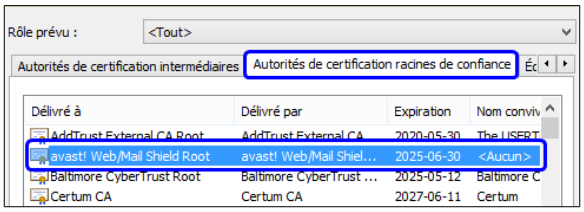


C'est bien *avast!* qui a émis le certificat.



Les résultats qu'on voit ici sont dûs au module de l'antivirus **Avast!** pour **IE**. Ce module d'**Avast!** en est un du genre homme-du-milieu (*man in the middle*) qui intercepte toutes les requêtes **https** et émet son propre certificat.

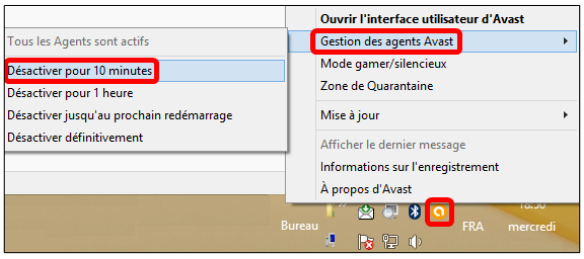
La protection **Courriel/Web** d'**Avast!** doit être capable de balayer votre trafic Web avant de l'envoyer au fureteur. Le balayage d'un connecteur logiciel (*socket*) chiffré **TLS** exige qu'**Avast!** puisse déchiffrer la connexion. Il n'y a pas d'autre moyen pour **Avast!**, de déchiffrer la connexion, que de générer son propre certificat et de le signer avec un certificat racine d'**Avast** déjà installé sur le système. De cette façon, **Avast!** peut vérifier la connexion.



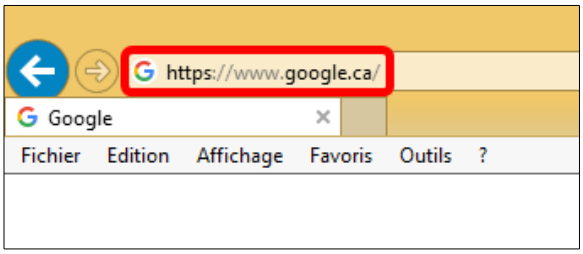
Même si on importe le fichier du certificat de **Let's Encrypt** dans **IE**, ce dernier persiste à utiliser celui d'**Avast!**.

Contournement du problème

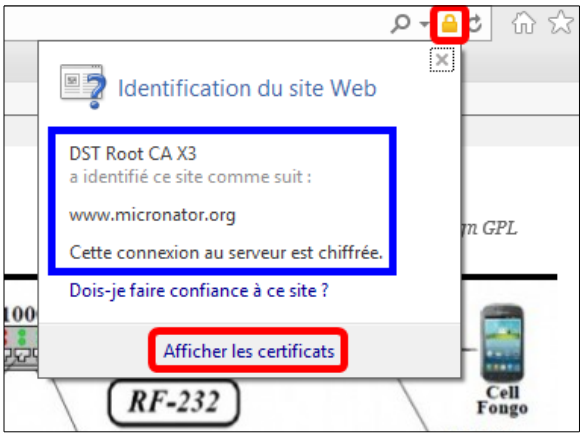
On arrête les agents d'*Avast!*.
- Clac (*clic droit*) sur l'icône **Avast!** sur la barre de notification | **Gestion des agents Avast** | **Désactiver pour 10 minutes**.
- À l'écran qui s'affiche, on clique **Oui** pour l'***Arrêt d'un composant***.



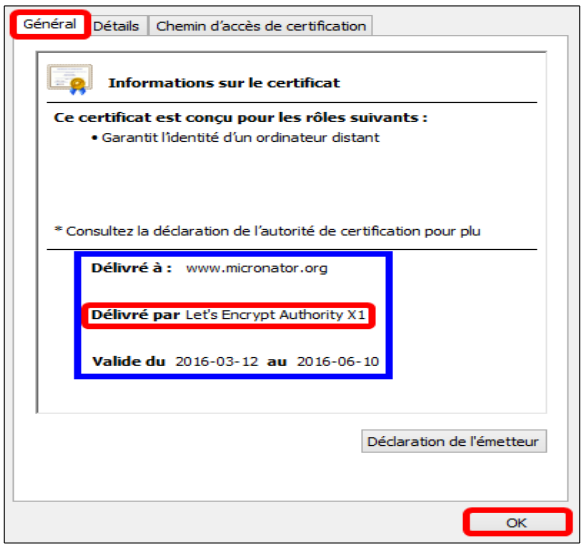
On se rend à une connexion sécurisée quelconque.



- Site: **https://www.micronator.org/**.
- On clique le cadenas.
- **Afficher les certificats**.

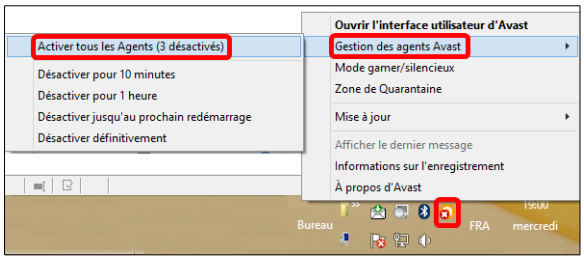


- Les informations du **certificat** sont affichées.
- On ferme toutes les fenêtres.



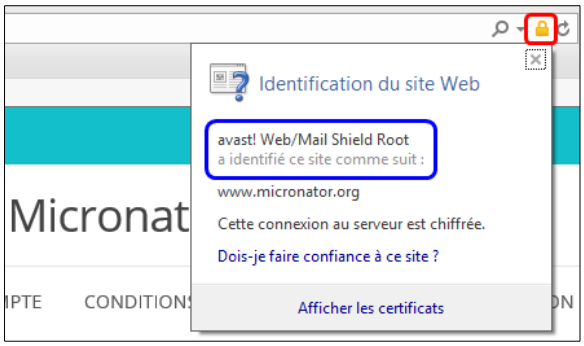
Il faut réactiver les agents **Avast**.

Clac (*clic droit*) sur l'icône **Avast!** dans la barre de notifications | **Gestion des agents Avast** | **Activer tous les Agents (3 désactivés)**.



On ferme **Internet Explorer**, on le relance et on se rend encore une fois à **https://www.micronator.org/**; le certificat retourne à celui d'**Avast!**.

Vu que chez **Micronator**, il est formellement interdit d'utiliser un navigateur **Microsoft**, de quelle que version que ce soit, on n'a pas de tels problèmes.



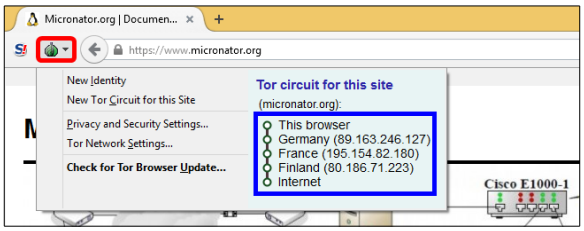
TOR

Le navigateur **TOR** est très utile pour vérifier les certificats, car il envoie la requête **HTTP** à **Privoxy** et non à votre serveur passerelle, il agit comme un fureteur provenant directement de l'**Internet** et non de votre réseau local. **TOR** fonctionne exactement comme **Firefox**.

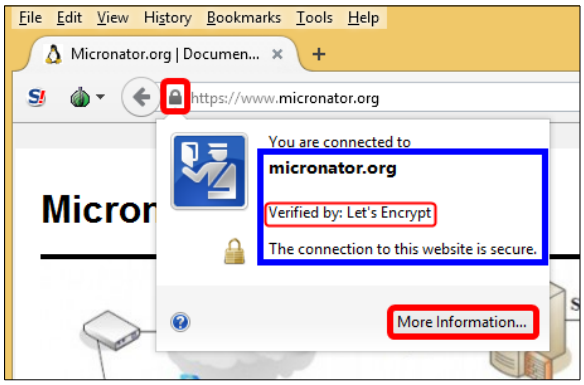
Site de téléchargement: **https://www.torproject.org/download/download.html.en**.

Il faut absolument télécharger **TOR** du site original seulement.

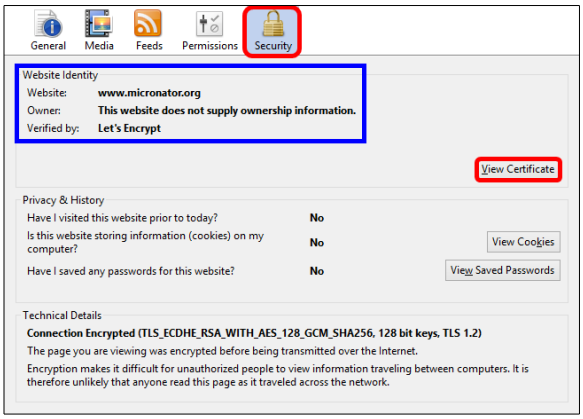
Même si nous avons lancé *TOR* sur notre station de travail, la requête pour notre site a passé par l'Allemagne, la France et la Finlande.



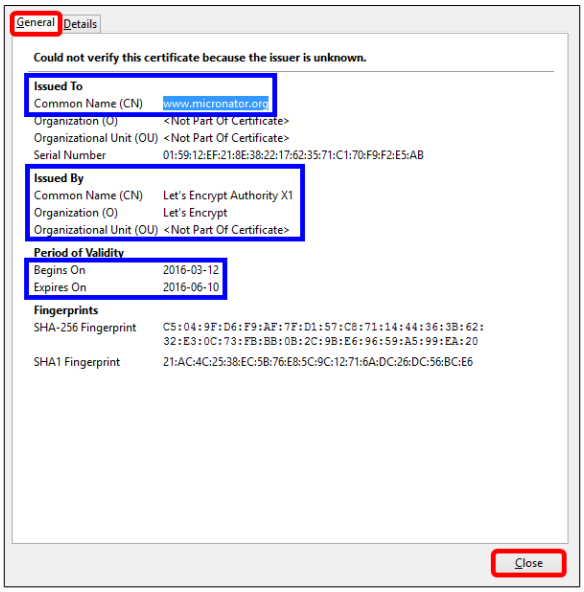
- On clique le cadenas et le nom de la *CA* émettrice s'affiche.
- **More information...**



- Onglet **Security**.
- Le vérificateur (***Verified by:***) est affiché.
- **View Certificate**.



- Les mêmes informations que celles obtenues par *Firefox*.
- Fermer toutes les fenêtres.



5. Conclusion

Le client **letsencrypt.sh** fonctionne très bien pour un demande de certificat **officiel**. Il en va de même pour tous nos fichiers de configuration. Avec le **script de point d'entrée**: les propriétés de **modSSL** ont été modifiées, les changements signalés et le certificat installé.

X- Renouvellement

1. Introduction

1.1. Limite de 90 jours

Les certificats **Let's Encrypt** sont valides pour **90** jours. Il est recommandé de les renouveler à tous les **60** jours afin d'avoir une certaine marge de manoeuvre.

1.2. Limite 5/7

En date du 2015-12-03 16:46:08 UTC:

• Limite de **5** certificats par domaine dans une fenêtre de **7** jours.

2. Manuel

Cette méthode de renouvellement se comporte exactement comme celle de **TEST**.

Situation actuelle

• Aucun ajout/retrait d'un nouveau domaine ou **CNAME**.

• Aucune modification des fichiers de configuration.

• Le dernier certificat officiel est toujours valide (*plus longtemps que 30 jours*) et il n'a pas été révoqué.

Si nous lancions le **client letsencrypt.sh**, il nous retournerait ce qui suit.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh -c

# INFO: Using main config file /etc/letsencrypt.sh/config
Processing www.micronator.org with alternative names: micronator.org dorgee.micronator.org
mail.micronator.org ftp.micronator.org wpad.micronator.org proxy.micronator.org
www.ainesmercierouest.info ainesmercierouest.info dorgee.ainesmercierouest.info
mail.ainesmercierouest.info ftp.ainesmercierouest.info wpad.ainesmercierouest.info
proxy.ainesmercierouest.info
 + Checking domain name(s) of existing cert... unchanged.
 + Checking expire date of existing cert...
 + Valid till Jun 10 20:01:00 2016 GMT (Longer than 30 days). Skipping!
[root@dorgee ~]#
```

Au début de l'exécution, le client **letsencrypt.sh**:

a) a examiné le fichier **config** et n'a vu aucune modification,

b) a examiné le fichier **/etc/letsencrypt.sh/domains.txt** et vu qu'il n'avait pas été modifié par rapport au dernier certificat: **unchanged**,

c) a vérifié la validité du certificat et vu qu'il était encore valide pour plus de 30 jours: **Longer than 30 days**,

d) a affiché la dernière ligne du message: **Skipping!**, et il s'est arrêté sans aller plus loin,

e) arrêté, il ne s'est pas rendu au **script de point d'entrée**; les propriétés de **modSSL** sont demeurées inchangées et il n'y a eu aucun changement de signaler.

3. Manuel forcé

Encore une fois, cette méthode de renouvellement se comporte exactement comme celle de **TEST**.

Ne voulant pas atteindre la limite **5/7**, nous laissons le renouvellement de votre dernier certificat **officiel** et valide, en utilisant le paramètre **--force**, à votre entière discrétion.

4. Automatique

Exactement le même comportement que celui de **TEST**.

4.1. Répertoire pour le gabarit personnalisé

Création du répertoire.

```
[root@dorgee ~]# mkdir -p /etc/e-smith/templates-custom/etc/crontab

[root@dorgee ~]#
```

4.2. Création de la tâche cron

Comme nous l'avons vu précédemment, lorsque le certificat est encore valide pour plus de 30 jours, le client **letsencrypt.sh** s'arrête avant de demander un renouvellement et ainsi, il n'incommode pas inutilement les serveurs de **Let's Encrypt**.

On crée le fichier de la tâche **cron** et on y insère son contenu. Il s'exécutera quotidiennement à **02H15**.

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL <<'EOT'

#
# Tâche cron qui lance le client letsencrypt.sh pour le renouvellement du certificat
# Elle s'exécutera quotidiennement à 02H15
#
# Si le certificat est encore valide pour plus de 30 jours, qu'il n'y a eu aucune
# modification des fichiers de configuration et aucun changement dans le fichier
# domains.txt, le client ne fera rien et attendra son prochain lancement.
#
# Si le certificat est encore valide pour moins de 30 jours, le client:
# 1) demandera un renouvellement du certificat,
# 2) ajustera les pointeurs des fichiers du certificat,
# 3) appellera le script de point d'entrée qui ajustera les paramètres de modSSL
#    et signalera les changements puis, le script letsencrypt.sh s'arrêtera.
#
# ┌──────────────── min (0 - 59)
# │ ┌────────────── heure (0 - 23)
# │ │ ┌──────────── jour du mois (1 - 31)
# │ │ │ ┌────────── mois (1 - 12)
# │ │ │ │ ┌──────── jour de la semaine (0 - 6) (0 à 6 sont de dimanche à samedi,
# │ │ │ │ │                 7 est dimanche, même que 0)
# │ │ │ │ │
# * * * * *  [usager] commande à exécuter
#
15 02 * * * root /etc/letsencrypt.sh/letsencrypt.sh -c

EOT
```

L'heure et le jour du mois peuvent être choisis à votre discrétion – nous avons choisi un moment qui ne devrait pas en être un de pointe (*tel le premier du mois*) dans l'espoir de réduire la charge des serveurs de **Let's Encrypt**. Étant donné que les certificats ont une durée de vie limitée à 90 jours, ce script aura amplement le temps pour renouveler notre certificat.

On vérifie le contenu du fichier de la tâche.

```
[root@dorgee ~]# cat /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL

#
# Tâche cron qui lance le client letsencrypt.sh pour le renouvellement du certificat
# Elle s'exécutera quotidiennement à 02H15
#
# Si le certificat est encore valide pour plus de 30 jours, qu'il n'y a eu aucune
# modification des fichiers de configuration et aucun changement dans le fichier
# domains.txt, le client ne fera rien et attendra son prochain lancement.
#
# Si le certificat est encore valide pour moins de 30 jours, le client:
# 1) demandera un renouvellement du certificat,
# 2) ajustera les pointeurs des fichiers du certificat,
# 3) appellera le script de point d'entrée qui ajustera les paramètres de modSSL
#    et signalera les changements puis, le script letsencrypt.sh s'arrêtera.
#
# ┌──────────── min (0 - 59)
# │ ┌────────── heure (0 - 23)
# │ │ ┌──────── jour du mois (1 - 31)
# │ │ │ ┌────── mois (1 - 12)
# │ │ │ │ ┌──── jour de la semaine (0 - 6) (0 à 6 sont de dimanche à samedi,
# │ │ │ │ │                 7 est dimanche, même que 0)
# │ │ │ │ │
# * * * * *  [usager] commande à exécuter
#
15 02 * * * root /etc/letsencrypt.sh/letsencrypt.sh -c

[root@dorgee ~]#
```

On sécurise le fichier.

```
[root@dorgee ~]# chmod 600 /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL

[root@dorgee ~]#
```

On vérifie

```
[root@dorgee ~]# ls -ls /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL

4 -rw------- 1 root root 1332 12 mars  20:19 /etc/e-smith/templates-custom/etc/crontab/renouvelerSSL
[root@dorgee ~]#
```

On développe le gabarit personnalisé.

```
[root@dorgee ~]# expand-template /etc/crontab

[root@dorgee ~]#
```

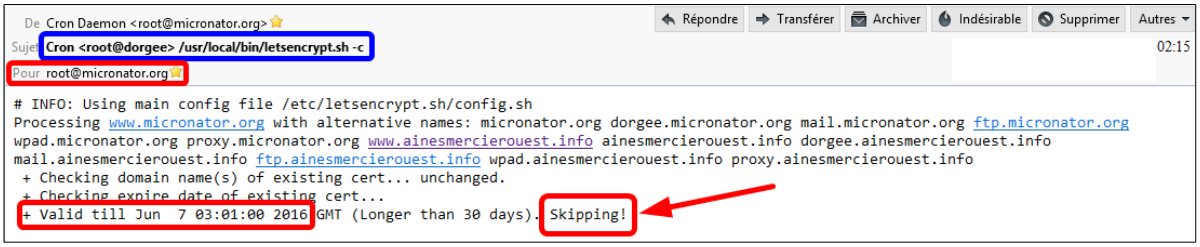
On redémarre le daemon **crond**.

```
[root@dorgee ~]# service crond restart

Arrêt de crond :                                           [  OK  ]
Démarrage de crond :                                       [  OK  ]
[root@dorgee ~]#
```


4.3. Courriel de notification

Lors d'un lancement de la tâche **cron**, **root** recevra un courriel, semblable à celui reçu ci-dessous lors de la vérification, à exactement **02H15**.

- Skipping! indiquera que le renouvellement n'a pas eu lieu car, le certificat est encore valide pour plus de **30** jours.
- Un renouvellement indiquera quand il a été effectué et la date limite de validité du nouveau certificat.



Notre tâche **cron** fonctionne parfaitement.

XI- Sauvegarde du répertoire /etc/letsencrypt.sh

1. Introduction

Le répertoire **/etc** et ses sous-répertoires ne sont pas tous inclus dans la sauvegarde standard.

Nous allons créer un gabarit personnalisé pour inclure le répertoire **/etc/letsencrypt.sh** et ses sous-répertoires dans la sauvegarde standard.

Référence: **https://wiki.contribs.org/Backup_with_dar**

Paragraphe: *Adding/Excluding Directories and Files from the backup list*.

2. Création du gabarit personnalisé

On crée le répertoire pour le gabarit personnalisé.

```
[root@dorgee letsencrypt.sh]# mkdir -p /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf

[root@dorgee letsencrypt.sh]#
```

On sécurise.

```
[root@dorgee ~]# chmod 600 /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# ls -lsd /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf

4 drw------- 2 root root 4096 13 mars  16:25 /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf
[root@dorgee ~]#
```

On crée le fichier **41go-into** et on y insère son contenu pour indiquer d'inclure le répertoire **/etc/letsencrypt.sh**.

Prendre tout le contenu de l'encadré pour la commande.

```
 cat > /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf/41go-into <<'EOT'
#
# Indique à la sauvegarde d'inclure le répertoire /etc/letsencrytp et tous ses
# sous-répertoires dans la sauvegarde standard.
--go-into etc/letsencrypt.sh
EOT
```

Il n'y a pas de caractère "**/**" avant **etc**.

On vérifie.

```
[root@dorgee ~]# cat /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf/41go-into

#
# Indique à la sauvegarde d'inclure le répertoire /etc/letsencrytp et tous ses
# sous-répertoires dans la sauvegarde standard.
--go-into etc/letsencrypt.sh
[root@dorgee ~]#
```

On développe le gabarit personnalisé.

```
[root@dorgee ~]# expand-template /etc/dar/DailyBackup.dcf

[root@dorgee ~]#
```

On vérifie que le fichier a bien été incorporé dans **DailyBackup.dcf**.

```
[root@dorgee ~]# cat /etc/dar/DailyBackup.dcf | grep letsencrypt

--go-into etc/letsencrypt.sh
[root@dorgee ~]#
```

3. Vérification

Le lendemain de ces procédures, avec **VirtualBox**, nous avons créé un serveur semblable à celui de notre serveur passerelle et nous y avons restauré la sauvegarde de la nuit passée.

```
...
Restoring file's data: /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf
Restoring file's data: /etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf/41go-into
...
Restoring file's data: /etc/letsencrypt.sh
Restoring file's data: /etc/letsencrypt.sh/domains.txt.example
Restoring file's data: /etc/letsencrypt.sh/hook.sh.example
Restoring file's data: /etc/letsencrypt.sh/letsencrypt.sh
...
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/chain-1457816391.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/chain.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457812624.csr
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/chain-1457802076.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert.csr
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.csr
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457802076.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/privkey-1457796476.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457802076.csr
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/fullchain-1457816391.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/fullchain-1457796476.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457812624.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457796476.csr
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457796476.pem
Restoring file's data: /etc/letsencrypt.sh/certs/www.micronator.org/fullchain.pem
Restoring file's data: /etc/letsencrypt.sh/config
Restoring file's data: /etc/letsencrypt.sh/LICENSE
Restoring file's data: /etc/letsencrypt.sh/config.example
Restoring file's data: /etc/letsencrypt.sh/domains.txt
...
```

Nous voyons que l'inclusion du répertoire et de ses sous-répertoires de **/etc/letsencrypt.sh** a fonctionné.

3.1. Gabarit personnalisé httpd.conf

Précédemment, au paragraphe ***Gabarit personnalisé*** à la page ***20***, nous avons créé un gabarit personnalisé contenant le fichier **VirtualHosts40ACME** pour indiquer à **Apache** un alias pour le répertoire **acme-challenge**.

On vérifie si ce gabarit est bien dans les répertoires de la sauvegarde standard du **Serveur SME**.

À la console du nouveau serveur virtuel, on affiche le répertoire de ce gabarit pour vérifier s'il a été sauvegardé.

```
[root@dorgee ~]# ls -als /etc/e-smith/templates-custom/etc/httpd/conf/httpd.conf

total 12
4 drwxr-xr-x 2 root root 4096 14 mars  05:09 .
4 drwxr-xr-x 3 root root 4096  4 mars  16:02 ..
4 -rw-r--r-- 1 root root  127 12 mars  08:58 VirtualHosts40ACME
[root@dorgee ~]#
```

Le gabarit personnalisé **httpd.conf** a été sauvegardé et restauré, car il fait partie des répertoires de la sauvegarde standard des **Serveurs SME**.

XII- Révocation

1. Introduction

À certaines occasions, telle la mise au rancart d'un serveur, on devrait révoquer le certificat du serveur.

2. Affichage des certificats actuels

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/certs/www.micronator.org/

total 64
4 -rw------- 1 root root 2118 12 mars  10:27 cert-1457796476.csr
4 -rw------- 1 root root 2614 12 mars  10:28 cert-1457796476.pem
4 -rw------- 1 root root 2118 12 mars  12:01 cert-1457802076.csr
4 -rw------- 1 root root 2614 12 mars  12:02 cert-1457802076.pem
4 -rw------- 1 root root 2118 12 mars  14:57 cert-1457812624.csr
0 -rw------- 1 root root    0 12 mars  14:57 cert-1457812624.pem
4 -rw------- 1 root root 2118 12 mars  15:59 cert-1457816391.csr
4 -rw------- 1 root root 2598 12 mars  16:00 cert-1457816391.pem
0 lrwxrwxrwx 1 root root   19 12 mars  16:00 cert.csr -> cert-1457816391.csr
0 lrwxrwxrwx 1 root root   19 12 mars  16:00 cert.pem -> cert-1457816391.pem
4 -rw------- 1 root root 1123 12 mars  10:28 chain-1457796476.pem
4 -rw------- 1 root root 1123 12 mars  12:02 chain-1457802076.pem
4 -rw------- 1 root root 1675 12 mars  16:00 chain-1457816391.pem
0 lrwxrwxrwx 1 root root   20 12 mars  16:00 chain.pem -> chain-1457816391.pem
4 -rw------- 1 root root 3737 12 mars  10:28 fullchain-1457796476.pem
4 -rw------- 1 root root 3737 12 mars  12:02 fullchain-1457802076.pem
8 -rw------- 1 root root 4273 12 mars  16:00 fullchain-1457816391.pem
0 lrwxrwxrwx 1 root root   24 12 mars  16:00 fullchain.pem -> fullchain-1457816391.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem_T_2016-03-12_10h27
0 lrwxrwxrwx 1 root root   22 12 mars  10:28 privkey.pem -> privkey-1457796476.pem
[root@dorgee ~]#
```

3. Certificat officiel

3.1. Prérequis

On a décidé de retirer notre serveur pour le mettre au rancart et au lieu d'attendre la fin de vie du certificat, on choisit de le révoquer immédiatement.

Nous allons révoquer le dernier certificat obtenu, **cert-1457816391.pem**.

```
4 -rw------- 1 root root 2598 12 mars  16:00 cert-1457816391.pem
```

Ce certificat a été obtenu avec la clé de compte O**fficiel**.

Clé de compte Let's Encrypt

Pour révoquer un certificat, il faut avoir la même clé de compte que celle utilisée lors de l'obtention du certificat.

On vérifie le mode de fonctionnement.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/config

#!/bin/bash
# config
# CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@micronator.org"

[root@dorgee ~]#
```

Nous somme bien en mode **Officiel** car la ligne pour le mode TEST est en commentaire, elle débute par un #.

Il n'y a pas de ligne vide avant la ligne **#!/bin/bash**. Ci-dessus nous avons inséré une ligne vide pour faciliter la copie de la commande.

Attention au domaine de l'adresse courriel **CONTACT_EMAIL**="admin@micronator.org" ci-dessus. Il faut le remplacer par **votre domaine**.

3.2. Commande de révocation

On vérifie le chemin du certificat.

```
[root@dorgee ~]# ls -als /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.pem

4 -rw------- 1 root root 2598 12 mars  16:00
/etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.pem
[root@dorgee ~]
```

La commande générale pour la révocation d'un certificat.

```
/etc/letsencrypt.sh/letsencrypt.sh  --revoke  /chemin/du/cert-numéro.pem
```

On révoque le certificat.

```
[root@dorgee ~]#  /etc/letsencrypt.sh/letsencrypt.sh     \
                  --revoke                                \
                  /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.pem

# INFO: Using main config file /etc/letsencrypt.sh/config
Revoking /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.pem
 + Done.
 + Renaming certificate to /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457816391.pem-
revoked
[root@dorgee ~]#
```

Le certificat a été révoqué et aucune erreur n'a été reçue.

3.3. Vérification

On affiche tous les certificats.

```
[root@dorgee ~]# ls -ls /etc/letsencrypt.sh/certs/www.micronator.org/

total 60
4 -rw------- 1 root root 2118 12 mars  10:27 cert-1457796476.csr
4 -rw------- 1 root root 2614 12 mars  10:28 cert-1457796476.pem
4 -rw------- 1 root root 2118 12 mars  12:01 cert-1457802076.csr
4 -rw------- 1 root root 2614 12 mars  12:02 cert-1457802076.pem
4 -rw------- 1 root root 2118 12 mars  14:57 cert-1457812624.csr
0 -rw------- 1 root root    0 12 mars  14:57 cert-1457812624.pem
4 -rw------- 1 root root 2118 12 mars  15:59 cert-1457816391.csr
4 -rw------- 1 root root 2598 12 mars  16:00 cert-1457816391.pem-revoked
0 lrwxrwxrwx 1 root root   19 12 mars  16:00 cert.csr -> cert-1457816391.csr
0 lrwxrwxrwx 1 root root   19 12 mars  16:00 cert.pem -> cert-1457816391.pem
4 -rw------- 1 root root 1123 12 mars  10:28 chain-1457796476.pem
4 -rw------- 1 root root 1123 12 mars  12:02 chain-1457802076.pem
4 -rw------- 1 root root 1675 12 mars  16:00 chain-1457816391.pem
0 lrwxrwxrwx 1 root root   20 12 mars  16:00 chain.pem -> chain-1457816391.pem
4 -rw------- 1 root root 3737 12 mars  10:28 fullchain-1457796476.pem
4 -rw------- 1 root root 3737 12 mars  12:02 fullchain-1457802076.pem
8 -rw------- 1 root root 4273 12 mars  16:00 fullchain-1457816391.pem
0 lrwxrwxrwx 1 root root   24 12 mars  16:00 fullchain.pem -> fullchain-1457816391.pem
4 -rw------- 1 root root 3243 12 mars  10:27 privkey-1457796476.pem
0 lrwxrwxrwx 1 root root   22 12 mars  10:28 privkey.pem -> privkey-1457796476.pem
[root@dorgee ~]#
```

On essaie d'afficher le certificat révoqué.

```
[root@dorgee ~]# cat ls -ls /etc/letsencrypt.sh/certs/www.micronator.org/cert.pem

cat: cert.pem: Aucun fichier ou dossier de ce type
[root@dorgee ~]#
```

Le système nous informe que le fichier n'existe plus. La cible du lien a été renommée cert-1457816391.pem-**revoked** donc, le lien pointe vers une cible qui n'existe plus.

4. Certificat de TEST

4.1. Basculement du mode OFFICIEL au mode TEST

Avant de manipuler un certificat de **TEST**, il faut basculer en mode **TEST**.

Nous allons modifier le fichier **/etc/letsencrypt.sh/config** pour enlever le signe de commentaire **#** du début de la ligne **CA="https://acme-staging.api...**

```
[root@dorgee ~]# sed -i 's/^# CA="https:/CA="https:/'  /etc/letsencrypt.sh/config

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/config

#!/bin/bash
# config
CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
WELLKNOWN="/home/e-smith/files/ibays/Primary/html/.well-known/acme-challenge"
HOOK="/etc/letsencrypt.sh/letsencrypt-hook.sh"
# E-mail to use during the registration (default: <unset>)
CONTACT_EMAIL="admin@micronator.org"

[root@dorgee ~]#
```

Nous sommes maintenant en mode **TEST**.

Il n'y a pas de ligne vide avant la ligne **#!/bin/bash**. Ci-dessus nous avons inséré une ligne vide pour faciliter la copie de la commande.

Attention au domaine de l'adresse courriel **CONTACT_EMAIL**="admin@micronator.org" ci-dessus. Il faut le remplacer par **votre domaine**.

4.2. Révocation

On procède de la même manière que pour révoquer un certificat **OFFICIEL**.

On peut alors révoquer notre premier certificat de **TEST**: **cert-1457796476.pem** qui est datée **2016-03-12 à 10:28**.

```
[root@dorgee ~]# /etc/letsencrypt.sh/letsencrypt.sh --revoke           \
                    /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457796476.pem

# INFO: Using main config file /etc/letsencrypt.sh/config
Revoking /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457796476.pem
 + Done.
 + Renaming certificate to /etc/letsencrypt.sh/certs/www.micronator.org/cert-1457796476.pem-
revoked
[root@dorgee ~]#
```

4.3. Basculement du mode TEST au mode OFFICIEL

Après une manipulation en mode **TEST**, il faut revenir au mode **OFFICIEL** pour le bon fonctionnement de la tâche **cron**.

On commente la ligne de la CA de **TEST** dans le fichier **config**.

```
[root@dorgee ~]# sed -i 's/^CA="https:/# CA="https:/' /etc/letsencrypt.sh/config

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# cat /etc/letsencrypt.sh/config | grep http

# CA="https://acme-staging.api.letsencrypt.org/directory"  # CA pour mode TEST.
[root@dorgee ~]#
```

Nous sommes bien en mode **Officiel**.

XIII- Certificat standard SME

1. Introduction

On veut recréer un certificat original émis et certifié par le **Serveur SME-9.x** lui-même. Si le certificat présentement actif a été émis par une autre **CA** que **Let's Encrypt**, il suffit de suivre les instructions ci-dessous.

2. Login

Après s'être logué avec l'usager **root**, on devrait être dans le répertoire personnel de ce dernier.

On vérifie.

```
[root@dorgee ~]# pwd

/root
[root@dorgee ~]#
```

3. Création d'un répertoire de sauvegarde

On crée un répertoire de sauvegarde.

```
[root@dorgee ~]# mkdir letsencrypt.sh

[root@dorgee ~]#
```

On vérifie.

```
[root@dorgee ~]# ls -alsd letsencrypt.sh/

4 drwxr-xr-x 2 root root 4096 11 mars 14:25 letsencrypt.sh/
[root@dorgee ~]#
```

On se rend dans le répertoire de sauvegarde.

```
[root@dorgee ~]# cd letsencrypt.sh/

[root@dorgee letsencrypt.sh]#
```

On vérifie.

```
[root@dorgee letsencrypt.sh]# pwd

/root/letsencrypt.sh
[root@dorgee letsencrypt.sh]#
```

4. Sauvegarde des fichiers du certificat actuel

Recherche des chemins des fichiers originaux du certificat présentement actif.

```
[root@dorgee letsencrypt.sh]# cat /etc/httpd/conf/httpd.conf  | grep SSLCertificate

SSLCertificateChainFile /etc/letsencrypt.sh/certs/www.micronator.org/chain.pem
SSLCertificateFile /etc/letsencrypt.sh/certs/www.micronator.org/cert.pem
SSLCertificateKeyFile /etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem
[root@dorgee letsencrypt.sh]#
```

On sauvegarde le fichier de la chaîne de certification.

```
[root@dorgee letsencrypt.sh]# cp /etc/letsencrypt.sh/certs/www.micronator.org/chain.pem .

[root@dorgee letsencrypt.sh]#
```

On sauvegarde le fichier du certificat.

```
[root@dorgee letsencrypt.sh]# cp /etc/letsencrypt.sh/certs/www.micronator.org/cert.pem .

[root@dorgee letsencrypt.sh]#
```

On sauvegarde le fichier de la clé privée.

```
[root@dorgee letsencrypt.sh]# cp /etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem .

[root@dorgee letsencrypt.sh]#
```

On affiche le fichier **pem**.

```
[root@dorgee letsencrypt.sh]# ls -ls /home/e-smith/ssl.pem/

total 8
8 -rw-r--r--  1 root  root  7901 10 mars  11:11 dorgee.micronator.org.pem
[root@dorgee letsencrypt.sh]#
```

Il n'est pas nécessaire de le faire, mais on sauvegarde quand même le fichier **pem**.

```
[root@dorgee letsencrypt.sh]# cp /home/e-smith/ssl.pem/dorgee.micronator.org.pem .

[root@dorgee letsencrypt.sh]#
```

On vérifie les sauvegardes.

```
[root@dorgee letsencrypt.sh]# ls -ls

total 20
4 -rw-------  1 root root 2598 10 mars  11:10 cert.pem
4 -rw-------  1 root root 1675 10 mars  11:10 chain.pem
8 -rw-r--r--  1 root root 7901 10 mars  11:11 dorgee.micronator.org.pem
4 -rw-------  1 root root 3247 10 mars  11:10 privkey.pem
[root@dorgee letsencrypt.sh]#
```


5. Effaçage des propriétés de modSSL

On affiche les propriétés de **modSSL**.

```
[root@dorgee letsencrypt.sh]# config show modSSL

modSSL=service
    CertificateChainFile=/etc/letsencrypt.sh/certs/www.micronator.org/chain.pem
    CommonName=www.micronator.org
    TCPPort=443
    access=public
    crt=/etc/letsencrypt.sh/certs/www.micronator.org/cert.pem
    key=/etc/letsencrypt.sh/certs/www.micronator.org/privkey.pem
    status=enabled
[root@dorgee letsencrypt.sh]#
```

On efface la propriété de la chaîne de certificat.

```
[root@dorgee letsencrypt.sh]# config delprop modSSL CertificateChainFile

[root@dorgee letsencrypt.sh]#
```

On efface la propriété **Nom Commun**.

```
[root@dorgee letsencrypt.sh]# config delprop modSSL CommonName

[root@dorgee letsencrypt.sh]#
```

On efface la propriété du certificat.

```
[root@dorgee letsencrypt.sh]# config delprop modSSL crt

[root@dorgee letsencrypt.sh]#
```

On efface la propriété de la clé.

```
[root@dorgee letsencrypt.sh]# config delprop modSSL key

[root@dorgee letsencrypt.sh]#
```

On vérifie les propriétés de **modSSL**.

```
[root@dorgee letsencrypt.sh]# config show modSSL

modSSL=service
    TCPPort=443
    access=public
    status=enabled
[root@dorgee letsencrypt.sh]#
```

6. Signalisation

6.1. Signalisation des changements

Il faut signaler les changements avec un des blocs de commandes ci-dessous:

☐ Si on ne veut pas réamorcer, on lance le bloc de commandes suivant.

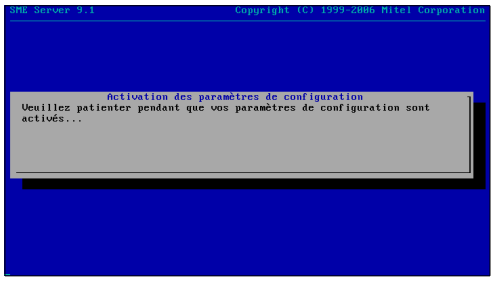
Sans réamorçage, le serveur ne prendra que quelques secondes pour effectuer les modifications nécessaires.

```
[root@dorgee letsencrypt.sh]# signal-event domain-modify;  \
                              signal-event email-update;   \
                              signal-event ibay-modify

[root@dorgee letsencrypt.sh]#
```

☐ Si on veut réamorcer, on applique les changements en signalant une mise à jour et un réamorçage.

```
[root@dorgee letsencrypt.sh]# signal-event post-upgrade  ;  signal-event reboot

Broadcast message from root@dorgee
        (/dev/pts/0) at 14:47 ...

The system is going down for reboot NOW!
[root@dorgee letsencrypt.sh]#
```

Si on choisit le bloc de commandes avec réamorçage, la commande **signal-event post-upgrade** mettra à jour les paramètres de configuration de **modSSL**, le serveur réamorcera pour activer tous les nouveaux paramètres et prendra une minute ou deux pour redevenir actif.



7. Vérification

7.1. Console du serveur

On affiche la date.

```
[root@dorgee letsencrypt.sh]# date

ven. mars 11 14:53:40 EST 2016
[root@dorgee letsencrypt.sh]#
```

Fichier pem.

```
[root@dorgee letsencrypt.sh]# ls -ls /home/e-smith/ssl.pem/

total 4
4 -rw-r--r--  1 root  root  3596 11 mars  14:48 dorgee.micronator.org.pem
[root@dorgee letsencrypt.sh]#
```

Le fichier **pem** vient tout juste d'être recréé.

7.2. Navigateur Web

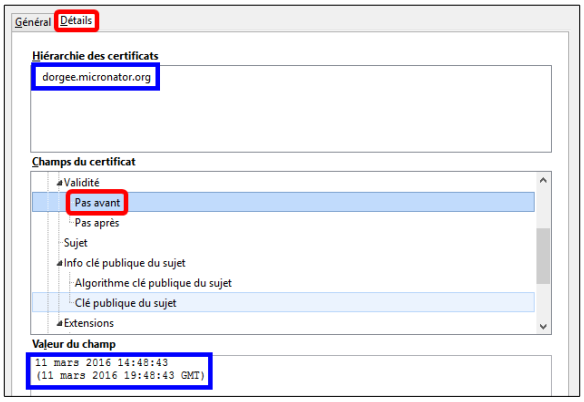
On se rend au site **https://www.micronator.org**.

Firefox affiche un écran d'avertissement. **Avancé** | **Ajouter une exception...** | **Voir...** | onglet **Détails** | **Pas avant**.

Le certificat est au nom de **dorgee.micronator.org**.

On voit qu'il vient d'être émis en même temps que la création du fichier **pem**.

Fermer toutes les fenêtres.



Le nouveau certificat émis et certifié par le **Serveur SME-9.1** lui-même est fonctionnel.

Victoire totale, hissons la bannière de la victoire.

8. Crédits

Auteur: **Michel-André Robillard CLP**

Remerciement: **Tous les contributeurs GNU/GPL**.

Intégré par: **Michel-André Robillard CLP**

Contact: **michelandre at micronator.org**

Répertoire de ce document: E:\000_DocPourRF232_general\RF-232_SME-9.1_LetsEncrypt\RF-232_SME-9.1_LetsEncrypt.sh_2016-04-20_21h58.odt

Historique des modifications**:**

Version	*Date*	*Commentaire*	*Auteur*
0.0.1	2016-01-26	Début.	M.-A. Robillard
0.0.2	2016-03-21	Ajustement de la tâche cron, pour qu'elle roule quotidiennement à 02h15, à cause de la possibilité d'un "effet de bord" du programme calculant le nombre de jours restants pour la validité du certificat.	M.-A. Robillard
0.0.3	2016-04-03	Corrections orthographiques.	M.-A. Robillard
0.1.0	2016-04-03	- Mise à jour à cause du changement du nom du fichier de configuration de config.sh à config. - La majorité des dates, heures et certificats sont ceux de la version 0.0.1 de ce document.	M.-A. Robillard
0.1.1	2016-07-18	Correction pour le fichier **/etc/e-smith/templates-custom/etc/dar/DailyBackup.dcf/41go-into**.	M.-A. Robillard

Index

0

02H15.....60

1

10 enregistrements par IP.....14
15 02 * * *.....60, 61

2

2048 bits.....7

3

3 mois (90 jours).....41
33 13 12 3 *.....42, 43

4

41go-into.....63

5

5 certificats par domaine.....14
5/7.....14

7

7 jours.....14

9

90 jours.....13

A

ACME.....18
acme-challenge.....65
acme-challenge/.....20
acme-v01.....47
Activer tous les Agents.....57
Affichage de l'aide.....27
Afficher le certificat.....52, 54, 55
Aide.....16
Ajouter une exception.....33
American Express.....7
Apache.....65
Arrêt d'un composant.....56
Arrêt de crond.....43, 61
ASCII.....6
astuce.....6
aucune dépendance.....18
Automatique.....60
Autorité de Certification Let's Encrypt.....7
Avancé.....40
Avast.....57
Avast! pour IE.....56
Avertissement.....2

B

BD du Serveur SME.....26
Bêta Version Publique.....5
bleu.....6
Boutique de Micronator.....7
Brancher les aînés.....7
Broadcast message.....73

C

CA acme-staging.....14
CA acme-v01.....51
CA de TEST.....14
CA="https:.....28, 68
cadenas.....40
cat >.....23
cert.pem.....68, 71
certificat multi-domaines.....5
certificat officiel.....46
Certificat SAN.....11
certificat SSL.....5, 7
certificat SSL DV.....10
certificat standard auto-signé.....6
Certificat standard SME.....70
certificat TLS/SSL.....17
certificat Wildcard.....11
CertificateChainFile.....17, 31, 72
certificats actuels.....66
Certificats racines.....10
Certificats SSL Wildcard.....11
certs/.....39
chain.pem.....71
Chaîne de certification.....10
challenge.....20
Chiffrement.....8
chmod 600.....61
chmod 700.....19
citrix secure gateway.....10
clause d'exclusion.....55
Clé de compte.....66
Clients Let's Encrypt.....5
CMD.....43
CNAME.....11
Commande de révocation.....67
Commentaire.....75
Commentaires et suggestions.....7
CommonName.....72
Condensat.....9
Conditions préalables.....17
config.....21
config delprop.....72
config show modSSL.....17, 72
CONTACT_EMAIL.....21
Conventions.....6
Courriel de notification.....45, 62
Courriels du certificat.....13
CR.....6
Création de la tâche.....60
Création des fichiers.....20
Création du gabarit.....63
Crédits.....75
CROND.....43
crt.....17, 31, 72

D

daemon crond.....61
DailyBackup.dcf.....63
DanB35.....5
date.....73
delprop.....72
Démarrage de crond.....43, 61
Désactiver pour 10 minutes.....56
Description générale.....5
Détails.....33, 54
deuxième domaine.....53
Diffie-Hellman.....8, 9
Discover.....7
DNS.....11
Domain Name System.....11
domain-modify.....24
domains.txt.....22, 23
DV.....10

E

Échange de clés.....8
écoutes du canal.....9
Élimination.....44
email-update.....24

Émetteur....32, 52
Émetteur officiel....47
Émis par....52
Émis pour....32, 52
empreinte....8
Empreinte numérique....9
Enregistrement A....11
enregistrement CNAME....11
enregistrement DNS....9
enregistrements DNS....17
EOT....21
étape....6
expand-template....21

F

FAI....17
Fichier de configuration....28
fichier de la tâche cron....42
fichier pem....71
Fichier pem....50
fichiers du certificat actuel....71
Firefox & premier domaine....40
Flep....5
Forward Secrecy....50
FQDN....51

G

gabarit personnalisé....60
Gabarit personnalisé....20, 42
Generating account key....48
Gestion des agents Avast....56, 57
git....18
git clone....19
Glossaire....8
going down for reboot NOW!....73
Google Chrome....54
greffon webroot....20

H

happy hacker fake CA....14
HDM....9
heure (0 - 23)....42
Hfwang....5
homme-du-milieu....56
Homme-du-milieu....9
httpd.conf....65
https....51

I

ibay-modify....24
icône >....40, 51, 53
Ignoring....47
informations du certificat....55
Installation du client....18
Internet Explorer....55, 57
intranet....10

J

jour du mois....42

K

key....17, 31, 72

L

Lancement de la demande....47
Let's Encrypt....13
Let's Encrypt Authority X1....52
letsencrypt.sh --revoke....67
letsencrypt.sh --help....16
letsencrypt.sh --revoke....69
letsencrypt.sh -c....36
letsencrypt.sh -c --force....37, 47
LF....6
Limite 5/7....59
limite de 100....14
Limite de 90 jours....59
Login....70
Longer than 30 days....59

M

magenta....6
man in the middle....56
Manipulation....6
Manuel forcé....60
Mar 12 13:33:01....43
Marche à suivre....5
MasterCard....7
maximum standard de 100....14
Micronator....57
micronator.org....7
min (0 - 59)....42
MITM....9
mkdir -p....20
modSSL....31, 49, 72
moins de 30 jours....42, 60, 61
mois (1 - 12....42

N

Navigateur Web....51
Navigateurs WEB....32
Nom alternatif....34, 53
Nom Canonique....11
non vérifié....6
NON-RESPONSABILITÉ....2
Nonce....8
notation Wildcard....11
note....6
Notes au lecteur....6

O

onglet Détails....52
Onglet Général....40
orange....6

P

page GitHub....18
Paramètres....16
Paramètres de modSSL....25
pare-feu....17
Particularités de ce document....6
Pas avant....34, 41, 52, 74
PayPal....7
PDF....6
Plus d'informations....53
pointeurs....39, 49
ports 80 et 443....17
premier domaine....23, 32, 51
Prérequis....17, 66
privkey.pem....71
Privoxy....57
procédure....6
propriétés de modSSL....72
protocole cryptographique TLS....5
Python....5
Python 2.6....5
Python 2.7....5

Q

Qualys SSL Lab....50
Qualys SSL LABS....35

R

recommandation....6
référence Internet....6
Registering account....48
renew was forced!....47
renouvelerSSL....44
Renouvellement....36, 59
répertoire /etc....63
répertoire de sauvegarde....70

Répertoire des défis.....20
responsabilité.....55
Révocation.....66
révocation d'un certificat.....67
RF-232.....7
rouge.....6
rw-------.....61

S

SAN.....10
SAN et Wildcard.....10
Sauvegarde.....25
Sauvegarde du répertoire.....63
Serveurs SME-8.x.....5
service httpd-e-smith.....21
setprop.....24, 25
signal-event.....24
signal-event domain-modify;.....73
signal-event email-update;.....73
signal-event ibay-modify.....73
signal-event post-upgrade.....73
signal-event reboot.....73
signal-event.....24
Signalisation.....73
Situation présente.....46
Skipping!.....36, 47, 59
Somme de contrôle.....8
SSL.....7
ssl.pem.....71
Staging.....14
Stripe.....7
Subject Alternative Names.....10
Submit.....50, 51
Suivi.....43

T

TEST acme-staging.....46
TLS/SSL.....10
top -d 1.....43
TOR.....57
Transparence des certificats.....13
trouble majeur avec un certificat. .16

U

UC.....10
unchanged.....47, 59
Unified Communications.....10

V

valeur par défaut.....27
Validation du Domaine.....10
Validité.....34, 52
Vérification du nouveau certificat 48
Victoire.....74
View certificate.....54
View Certificate.....58
VirtualBox.....64
VirtualHosts40ACME.....21, 65
Visa.....7
Voir.....40

W

webmail.....10
WordPress.....51
WordPress HTTPS.....51

X

X.509.....5, 9

Y

yum -y install git.....18

-

--cleanup.....27
--cron (-c).....27
--force.....27, 36, 60
--revoke.....27, 67

.

.well-known.....20

"

"/".....63

(

(root).....43

[

[CTL - c].....43
[q].....43
[usager].....42

/

/etc.....63
/etc/crontab.....44, 61
/etc/dar/.....63
/home/e-smith/ssl.pem/.....73

#

INFO.....38
#!/bin/bash.....22

©

© RF-232.....2

+

+ Challenge is valid!.....29
+ Checking certificate.....29
+ Creating fullchain.pem.....29
+ Done!.....29
+ Generating private key.....29
+ Requesting certificate.....29
+ Requesting challenge.....29
+ Signing domains.....29

$

$CERT.....24, 25
$CHAIN.....24, 25
$KEY.....24, 25

