Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

S . G . D . S . N
Agence nationale de la sécurité des systèmes d'information
CERT-FR

Paris, le 10 octobre 2014
N° CERTFR-2014-ACT-041

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2014-ACT-041

1 - Vulnérabilité CVE-2014-1568 concernant la bibliothèque NSS

Le 24 septembre, une vulnérabilité dans la bibliothèque NSS (Network Security Services) a été rendue publique. NSS est utilisée par de nombreux logiciels (Mozilla Firefox, Mozilla Thunderbird, Google Chrome, etc.) pour mettre en œuvre des communications sécurisées selon les standards SSL/TLS. Cette vulnérabilité peut être exploitée pour forger un certificat qui sera accepté par NSS à la place du certificat légitime du site ciblé. Ainsi, une personne malveillante pourra mettre en œuvre une attaque de type homme du milieu pour déchiffrer et intercepter les flux réseau à destination du site ciblé.

Détails de la vulnérabilité

Un débordement d'entier dans l'analyseur syntaxique ASN.1 des certificats X.509 entraîne un problème de malléabilité des signatures. Dans ce contexte, un attaquant peut forger une signature pour un certificat quelconque à condition que l'exposant public RSA de la clé dont il veut usurper la signature soit égal à 3.
La majorité des autorités de certification utilisent aujourd'hui un exposant public différent de 3 (généralement égal à 65537). Cependant, il suffit qu'une seule autorité de certification considérée comme de confiance soit concernée pour qu'une personne malveillante puisse forger une signature pour un site de son choix. En particulier, il existe de tels certificats reconnus par les produits Mozilla dans leur configuration par défaut.

Les principaux risques encourus sont :

- le déchiffrement des communications sécurisées par SSL/TLS via une attaque de type homme du milieu (MITM) ;
- la mise en œuvre d'attaques par hameçonnage visant des sites accessibles en HTTPS ;
- un contournement d'un système d'authentification par certificats utilisateur.

Logiciels affectés

Les versions de NSS antérieures à 3.16.2.1, 3.16.x avant 3.16.5 et 3.17.x avant 3.17.1 sont affectées. Ces dernières sont utilisées dans les logiciels suivants :

- Mozilla Firefox (avant 32.0.3) ;
- Mozilla Firefox ESR (24.x avant 24.8.1 et 31.x avant 31.1.1) ;
- Mozilla Thunderbird (avant 24.8.1 et 31.x avant 31.1.2) ;
- Mozilla SeaMonkey (avant 2.29.1) ;
- Google Chrome (avant 37.0.2062.124 sur Windows et OS X) ;
- Google Chrome OS (avant 37.0.2062.120).

D'autres outils utilisent NSS (par exemple Pidgin, OpenOffice.org 2.0, Opera, Sun Java System Web server, etc.) et sont susceptibles d'être vulnérables. Le CERT-FR conseille de réduire la surface d'exposition de ces outils s'ils utilisent une version de NSS vulnérable.

Conclusion

Le CERT-FR recommande de mettre à jour les logiciels affectés vers leur dernière version disponible pour bénéficier de la correction de cette vulnérabilité.

Documentation

– Rapport de sécurité Ubuntu :
http://people.canonical.com/ ubuntu-security/cve/2014/CVE-2014-1568.html

2 - Indexation de journaux d'évènements pour faciliter les recherches

Afin de permettre la détection d'incidents de sécurité et de rendre possible la recherche a posteriori des traces laissées par ceux-ci, la collecte des journaux d'évènements est primordiale.

L'ANSSI a publié en décembre 2013 une note [1] qui traite non seulement de la collecte, mais aussi du stockage centralisé, de la cohérence des formats, de l'horodatage, de l'archivage et de la protection des journaux.

En juin 2014, un article complémentaire concernant spécifiquement les journaux liés aux flux réseau a été publié en deux parties sous la forme de bulletins d'actualités [2] et [3]. Cet article précise notamment les informations nécessaires à l'identification des machines compromises. La démarche implique que chaque équipement réseau ou applicatif (routeur, serveur mandataire HTTP, etc.) procède à la journalisation des requêtes. D'autre part, en plus de la journalisation des flux réseau, il peut être intéressant de collecter les journaux des serveurs et des postes de travail, par exemple les journaux d'évènements Windows ou les journaux Syslog des machines Linux, afin d'améliorer la détection et le traitement des incidents.

Comme pour les journaux liés aux flux réseau, un travail préliminaire d'identification des informations utiles est alors nécessaire afin de configurer la verbosité des journaux de manière adéquate. Pour un réseau de grande entreprise, la collecte de ces différents types de journaux signifie rapidement le stockage de plusieurs giga-octets de données par jour.

Une fois la collecte et la centralisation des journaux mis en place, se pose le problème de l'exploitation des données à disposition, dont le volume est en général trop important pour qu'une recherche de motifs à l'aide d'outils classiques tels que `grep` soit viable d'un point de vue opérationnel.

Une pratique courante pour accélérer les recherches est de procéder à une indexation des journaux. Pour comprendre le principe, on peut faire l'analogie avec un livre, dans lequel l'index permet de trouver rapidement les pages associées à un sujet donné. Dans le cadre de cet article, nous en donnons une définition plus générale : un index est assimilé à une structure de données qui permet d'identifier rapidement l'ensemble des documents décrits par une combinaison booléenne de descripteurs.

Dans le contexte qui nous intéresse, « document » peut correspondre à une ligne de journal, à un bloc de plusieurs lignes, ou au fichier tout entier. Un exemple de combinaisons booléennes de descripteurs pourrait être `"192.168.1.128/25" AND "ssi.gouv.fr"`. L'index devra alors permettre de récupérer l'ensemble des lignes contenant la chaîne `"ssi.gouv.fr"` et une IP appartenant au sous-réseau `"192.168.1.128/25"`.

Les stratégies d'indexation suivantes sont possibles.

Indexation brute

Si les journaux ont un format de type CSV (représentation de données tabulaires en fichier texte, les valeurs étant séparées par des virgules), la méthode la plus simple à mettre en œuvre est celle qui consiste à considérer ceux-ci comme un ensemble de lignes de texte non structuré. Le type de recherche possible sera alors limité à des requêtes simples, principalement la recherche d'un motif particulier sur l'ensemble des journaux, c'est-à-dire le même type de recherche que permet un outil comme `grep`, sans le support des expressions rationnelles. Un exemple pourrait être la recherche, limitée à la journée du 30/09/2014, de l'ensemble des journaux contenant la chaîne de caractères `() {` et une adresse IP donnée. La requête pourrait alors ressembler à ceci :

```
SEARCH "() {" AND "W.X.Y.Z" FROM 2014-09-30.idx
```

Cette méthode est simple car elle ne nécessite aucune étape de pré-traitement des journaux (normalisation, conversion de format, etc.). Notons que cette façon d'indexer n'exclut pas pour autant l'identification automatique de certains types de données particulièrement intéressants (notamment les adresses IPv4), et leur indexation en tant que telle, qui sera plus efficace.

Indexation en texte structuré

Si le format des journaux est bien défini et suffisamment stable au cours du temps, il peut être intéressant d'exploiter leur structure, pour avoir, à terme, la possibilité de lancer des recherches plus complexes, proches de ce que proposent les SGDBR classiques. Par exemple, sélectionner les journaux HTTP dont le champ `user_agent` commence par les caractères `() {` et dont le champ `return_code` (code de retour HTTP) est compris entre 500 et 600. En langage pseudo-SQL, cela donnerait quelque chose comme :

```
SELECT * FROM http_index WHERE user_agent LIKE "() {%" AND 500 <= return_code
< 600
```

L'indexation en texte structuré requiert au préalable une étape de configuration de l'outil d'indexation, qui consiste en général à une description des formats des différents types de journaux. La mise au point de cette étape est d'autant plus longue que l'hétérogénéité des formats est grande, et cette configuration devra être maintenue au gré de leur évolution au fil du temps.

Par ailleurs, l'outil d'indexation peut échouer dans l'étape de pré-traitement d'un journal, qui consiste justement à procéder à son analyse en fonction d'une description de son format. Il peut y avoir plusieurs raisons à cela, notamment :

- des problèmes d'encodage utf-8 ;
- un respect partiel de la norme CSV (concernant les règles d'échappement par exemple) ;
- une description des formats au moyen d'expressions rationnelles erronées ou incomplètes (il n'est pas aisé de décrire de façon certaine, exhaustive et sans erreur l'ensemble des cas possibles).

Quelle stratégie choisir ?

Dans la grande majorité des cas, la découverte d'incidents de sécurité est le résultat d'une recherche de marqueurs (adresses IP, domaines, motifs dans des URLs, etc.) sur un ensemble de journaux. Dans ces conditions, la première stratégie présentée, l'indexation brute des journaux, est en mesure de répondre de manière satisfaisante à la plupart des besoins. Néanmoins, si les formats des journaux sont relativement homogènes, il pourra être avantageux d'opter pour une indexation en texte structuré, qui permet de réaliser des recherches plus fines et plus rapides. De plus, certaines solutions d'indexation, comme ElasticSearch, ne révèlent pleinement leurs capacités que pour ce type d'indexation.

Quel outil d'indexation ?

Plusieurs bibliothèques logicielles open source permettent de réaliser relativement aisément une solution d'indexation minimaliste, utilisable en ligne de commande, adaptée à des besoins spécifiques. Citons notamment :

- Apache Lucene (Java)
- Xapian (C++)
- Whoosh (Python)

Les performances de la bibliothèque doivent être évaluées au préalable, afin d'être compatibles avec la volumétrie des journaux à indexer. Autrement dit, il doit nécessairement être possible d'indexer les données correspondant à une période donnée (par exemple, la journée écoulée) en un temps plus court que cette période.

Par ailleurs, l'espace disque nécessaire au stockage des index doit aussi être pris en compte pendant la phase de conception. Il dépend de plusieurs paramètres, notamment la granularité (les éléments de l'index pointent-ils vers une ligne de fichier, un bloc de lignes, un fichier complet ?), et se situe le plus souvent aux alentours de 20 à 30% de celui des données indexées. Il existe par ailleurs des solutions d'indexation plus complètes, comme par exemple :

- ElasticSearch, couplé à Logstash pour la phase de pré-traitement, et Kibana pour l'interface graphique (open source)
- Splunk (propriétaire)
- Apache Solr (open source)

Ces dernières solutions offrent de nombreuses fonctionnalités intéressantes, et ont été développées dès le départ pour répondre aux problématiques de passage à l'échelle, de répartition de charge et de résilience. De plus, elles sont généralement pourvues d'interfaces graphiques qui peuvent faciliter, par une présentation astucieuse des données, la mise en évidence a priori de comportements suspects (par exemple, par une représentation graphique des volumes de données échangées).

Documentation

1 http://www.ssi.gouv.fr/IMG/pdf/NP_Journalisation_NoteTech.pdf
2 http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-024/index.html
3 http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-025/index.html

3 - Rappel des avis émis

Dans la période du 03 au 09 octobre 2014, le CERT-FR a émis les publications suivantes :

- CERTFR-2014-AVI-407 : Multiples vulnérabilités dans Squid
- CERTFR-2014-AVI-408 : Multiples vulnérabilités dans Google Chrome
- CERTFR-2014-AVI-409 : Multiples vulnérabilités dans Huawei
- CERTFR-2014-AVI-410 : Multiples vulnérabilités dans Cisco ASA
- CERTFR-2014-AVI-411 : Multiples vulnérabilités dans Huawei E5332
- CERTFR-2014-AVI-412 : Multiples vulnérabilités dans Juniper

Gestion détaillée du document

10 octobre 2014 version initiale ;
13 octobre 2014 corrections apportées à l'article Vulnérabilité CVE-2014-1568 concernant la bibliothèque NSS.