I Le format XML

XML (eXtensible Markup Language) est un format de fichier conçu pour transmettre des informations entre des applications. Un document XML est composé d'éléments (blocs) qui représentent sa structure. Les principales caractéristiques de XML sont :

- langage de balisage, proche du HTML ;
- langage de description de contenus, décrivant la structure du document et non son affichage ;
- un document XML peut être associé à un schéma (p. ex. à l'aide d'une DTD) définissant les règles à respecter pour que le document soit bien formé et valide ;
- facile à lire pour un être humain.

La **DTD** (Document Type Definition) est un document qui permet de valider un document XML en définissant un ensemble de règles décrivant la structure du document. Ces règles comprennent notamment le nom des éléments pouvant apparaître dans le document XML et leur contenu ainsi que des attributs possibles précisant l'organisation hiérarchique des éléments entre eux et leur nombre d'occurrences. Cette description DTD peut être interne ou externe au document XML.

Exemple avec DTD interne

```
<?xml version="1.0">
<!DOCTYPE chercheur [
   <!ELEMENT chercheur (livre)+>
   <!ELEMENT livre (titre, auteur, ref)>
   <!ELEMENT titre (#PCDATA)>
   <!ELEMENT auteur (#PCDATA)>
   <!ELEMENT ref (#PCDATA)>
]>
<chercheur>
   <livre>
      <titre>Big data</titre>
      <auteur>Jean Dupont</auteur>
      <ref>Bases de données</ref>
   </livre>
   <livre>
      <titre>Debian 10 administration</titre>
      <auteur>Léo Quinville</auteur>
      <ref>Tech</ref>
   </livre>
   <livre>
      <titre>Mathématiques financières</titre>
      <auteur>Franck Paris</auteur>
      <ref>Finance</ref>
   </livre>
</chercheur>
```

Exemple avec DTD externe

Fichier externe nommé univ.dtd :

```
   <!ELEMENT chercheur (livre)+>
   <!ELEMENT livre (titre, auteur, ref)>
   <!ELEMENT titre (#PCDATA)>
   <!ELEMENT auteur (#PCDATA)>
   <!ELEMENT ref (#PCDATA)>
```

Puis injection dans le document XML :

```
<?xml version="1.0"?>
<!DOCTYPE chercheur SYSTEM "univ.dtd">
<chercheur>
<livre>
      <titre>Big data</titre>
      <auteur>Jean Dupont</auteur>
      <ref>Bases de données</ref>
  </livre>
  <livre>
      <titre>Debian 10 administration</titre>
      <auteur>Léo Quinville</auteur>
      <ref>Tech</ref>
  </livre>
  <livre>
      <titre>Mathématiques financières</titre>
      <auteur>Franck Paris</auteur>
      <ref>Finance</ref>
  </livre>
</chercheur>
```

Cas d'usages :

Le XML est utilisé pour structurer des informations dans des fichiers textes. On peut par exemple l'utiliser dans les cas suivants :

- fichiers de configuration ;
- fichiers de sauvegarde de configurations ;
- fichiers de journalisation ;
- enregistrement de résultats (mesures, liste de personnes…).

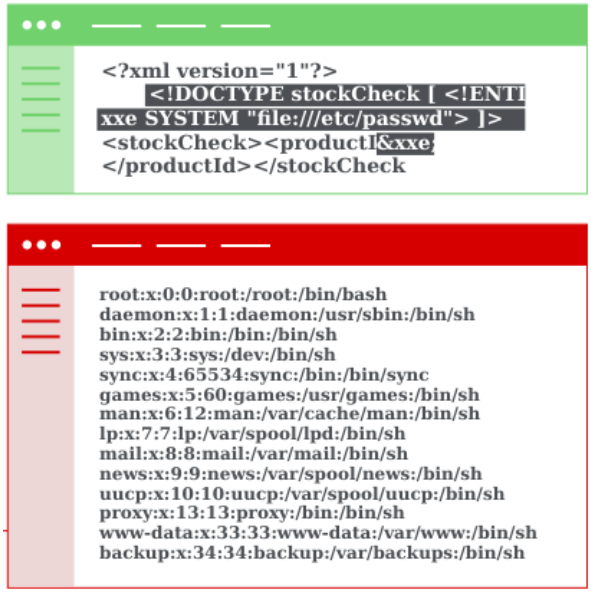
II Présentation générale des vulnérabilités XXE

1 Présentation de XXE

L'injection d'entités externes XML (également appelée XXE) est une vulnérabilité de sécurité Web qui permet à un attaquant **d'interférer avec le traitement des données XML par une application**. Il permet souvent à un attaquant **d'afficher des fichiers** sur le système de fichiers du serveur d'applications et **d'interagir** avec tous les systèmes dorsaux ou externes auxquels l'application elle-même peut accéder.

Exemple :

D'après portswigger.net.



Un classique du genre pour obtenir un fichier de configuration

Dans certaines situations, un attaquant peut intensifier une attaque XXE pour compromettre le serveur sous-jacent ou une autre infrastructure dorsale, en exploitant la vulnérabilité XXE pour effectuer des **attaques de falsification de demande côté serveur** (SSRF[1] - Server Side Request Forgery).

Le plus grand risque avec XXE est la **grande variété** de façons dont il peut être exploité. Qu'il soit simple ou complexe, si un morceau de code externe peut se frayer un chemin sur un document XML, le système est compromis. **L'omniprésence de XML** signifie que les applications utilisant XML sont susceptibles de manipuler un grand nombre de données sensibles.

[1] *SSRF (attaque de falsification de demande côté serveur) : attaque visant à abuser des fonctionnalités d'un serveur pour lire ou mettre à jour des ressources internes. L'attaquant peut fournir ou modifier une URL utilisée par le code exécuté sur le serveur pour lire ou mettre à jour des données. Avec des URL bien choisies, l'attaquant peut ainsi découvrir la configuration du serveur, se connecter à des services internes comme http, activer des bases de données ou effectuer des demandes de publication auprès de services internes qui ne sont pas destinés à être exposés. Définition d'après owasp.org.*

La forme la plus connue d’attaque XXE est connue sous le nom d’attaque **« Billion Laughs »** ou « XML Bomb ». Il s’agit d’une attaque par **déni de service** simple mais efficace utilisée pour surcharger et arrêter un serveur cible. En définissant une entité – généralement quelque chose de petit et insensé, comme « lol » ou « haha » – comme une chaîne imbriquée d’autres entités, un attaquant peut rapidement surcharger les ressources d’un système.

2 Panorama des attaques XXE

Il existe différents types d’attaques XXE :

- Récupération de fichiers. Une entité externe est définie avec le contenu d’un fichier et est renvoyée dans la réponse de l’application.
- Attaques SSRF.
- Attaques XXE en aveugle. Des vulnérabilités aveugles XXE surviennent lorsque l’application est vulnérable à l’injection XXE mais ne renvoie pas les valeurs des entités externes définies dans ses réponses. Cela signifie que la récupération directe des fichiers côté serveur n’est pas possible, et donc une vulnérabilité XXE aveugle est généralement plus difficile à exploiter qu’une vulnérabilité XXE classique.

III Exemples d’interactions XML/XXE

Cas d’usage	Requête	Réponse
Usage standard XML :	<foo> Hello World </foo>	HTTP/1.0 200 OK Hello World
Usage avec du code malveillant : Il s'agit d'une DTD appelée foo avec un élément appelé bar, qui est maintenant un alias pour le mot World. Par conséquent, à chaque fois que &bar; est utilisé, le parseur XML remplace cette entité par le mot World.	<?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE foo [<!ELEMENT foo ANY> <!ENTITY bar "World">]> <foo> Hello &bar; </foo>	HTTP/1.0 200 OK Hello World
Déni de service : Un attaquant peut utiliser des entités XML pour provoquer un déni de service en incorporant des entités au sein d’entités. Cette attaque est communément appelée l’attaque Billion Laughs.	<?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE foo [<!ELEMENT foo ANY> <!ENTITY bar "World "> <!ENTITY t1 "&bar;&bar;"> <!ENTITY t2 "&t1;&t1;&t1;&t1;"> <!ENTITY t3 "&t2;&t2;&t2;&t2;&t2;">]> <foo> Hello &t3; </foo>	HTTP/1.0 200 OK Hello World

IV

V Bonnes pratiques

Les contre-mesures suivantes sont à envisager :

- Le moyen le plus simple et le plus sûr d'éviter les attaques XXE consiste à désactiver complètement les définitions de type de document (entités externes).

  - Exemple en PHP :

  libxml_disable_entity_loader (true); (pour les versions de PHP < 8.0).

  - Exemple en Java :

  factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true);
  Remarque: factory est une instance de DocumentBuilderFactory.

- Dans la mesure du possible, utiliser des formats de données moins complexes tels que JSON et éviter la sérialisation des données sensibles.

- Corriger ou mettre à niveau tous les processeurs et bibliothèques XML utilisés par l'application ou sur le système d'exploitation sous-jacent. Utiliser des vérificateurs de dépendances. Mettre à jour SOAP[2] vers SOAP 1.2 ou supérieur.

- Mettre en œuvre une validation, un filtrage des entrées XML côté serveur (« liste blanche ») pour empêcher les données hostiles dans les documents, en-têtes ou nœuds XML.

- Utiliser un pare-feu d'application (WAF[3]) ou des outils de tests de la sécurité des applications web tels que des scanners de vulnérabilités.

[2] *SOAP : **SOAP** (ancien acronyme de Simple Object Access Protocol) est un protocole d'échange d'information structurée dans l'implémentation de services web basés sur XML.*
[3] WAF : Web application Firewall, pare-feu d'applications web.