Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

intro :

Qu'est ce que le XML :

XML signifie « langage de balisage extensible ». XML est un langage conçu pour stocker et transporter des données. Comme HTML, XML utilise une structure arborescente de balises et de données. Contrairement à HTML, XML n'utilise pas de balises prédéfinies, et les balises peuvent donc recevoir les noms que l'on souhaite .

Que sont les entités XML ?

Les entités XML sont une façon de représenter un élément de données dans un document XML, au lieu d'utiliser les données elles-mêmes.Par exemple, les entités **<**
et **>** représentent les caractères **<**et **>.**

Type d'entité : les entitées externe :

Les entités externes XML sont un type d'entité personnalisée dont la définition est située en dehors de la DTD où elles sont déclarées.
La déclaration d'une entité externe utilise le SYSTEM et doit spécifier une URL à partir de laquelle la valeur de l'entité doit être chargée.

Attaque XXE :

Cette attaque se produit lorsque la requete XML contient une entité externe .Cette attaque pourrait conduire à la divulgation de données confidentielles,etc…
qu'est ce qu'une entité :
En XML, une entité est comme un raccourci ou un alias pour un texte ou des données.
qu'est ce qu'une entité externe :
Ce sont des entités qui font référence à des ressources externes, comme des fichiers. Elles peuvent être utilisées pour inclure le contenu de fichiers externes dans un document XML.

Objectif en tant que personne malveillante :

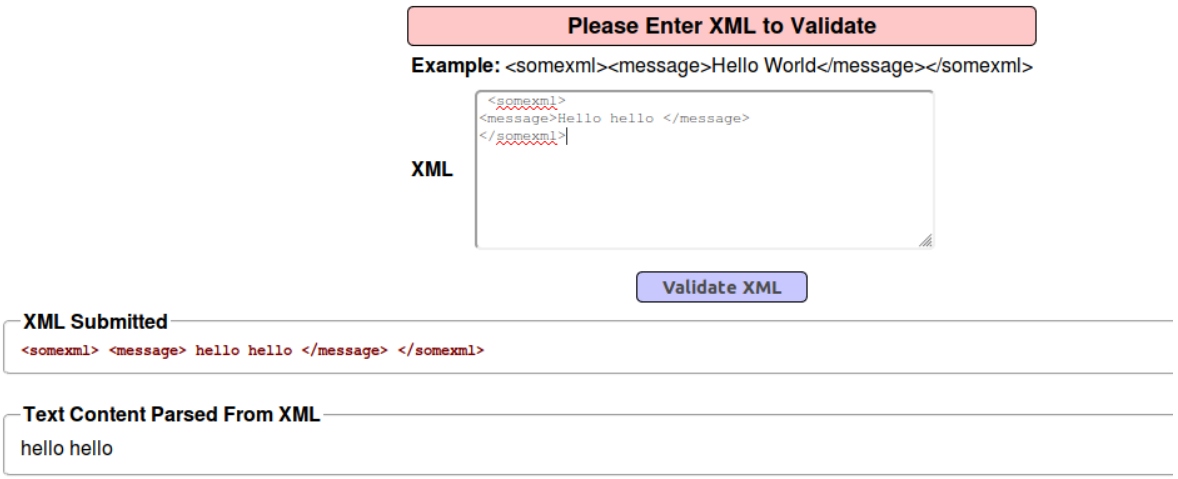
→ afficher le contenu du fichier /etc/passwd qui contient la liste de tous les utilisateurs systèmes présents sur le serveur

Etape 1 : Mise en place de l'attaque XXE :
mettre en œuvre l'attaque permettant d'afficher le fichier contenant tous les utilisateurs sur le serveur cible.

Petit test afficher un message xml non malveillant :

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd



Insertion de code malveillant :

requete :

<!DOCTYPE foo [<!ENTITY xxe SYSTEM “file:///etc/passwd”>]>
<somexml>
<message> &xxe; </message>
</somexml>

explication de la requête :
<!DOCTYPE foo> : TDT definie element “foo”.

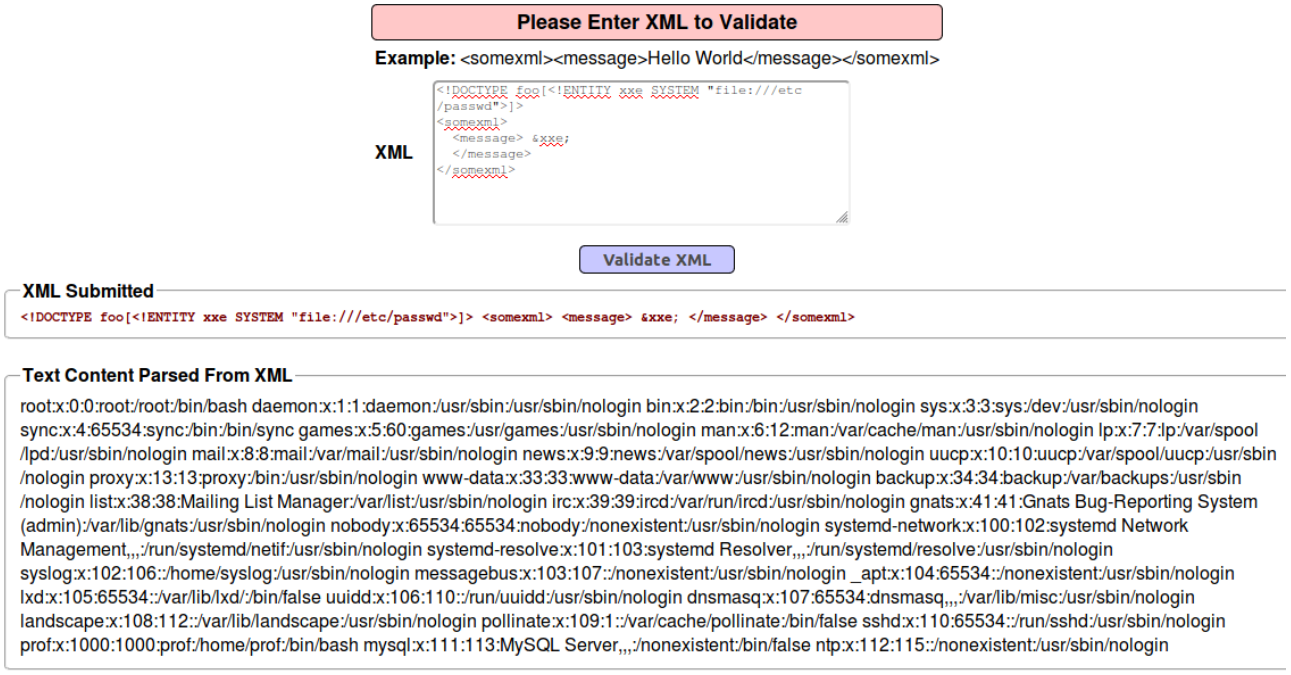
<!ENTITY xxe SYSTEM "file:///etc/passwd"> : definition de l’entité externe ‘xxe’ qui fait référence au fichier “/etc/passwd”.

<somexml> : balise racine
<message> : balise ‘enfant’ contenu dans la balise racine

&xxe; : Utilise l'entité externe définie dans la DTD pour inclure le contenu du fichier /etc/passwd situé dans la balise enfant <message>.

Maïssane El Mjidi
BTS SIO 2

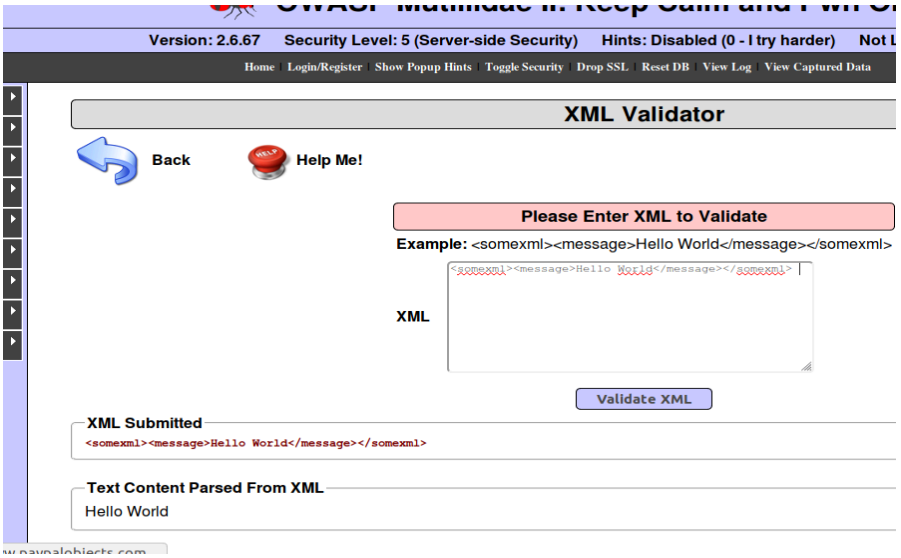
Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd



L'attaque est un succès , nous avons bien récupéré le fichier de login ✅

Nous allons maintenant monter le niveau de sécurité de owasp à 5 .

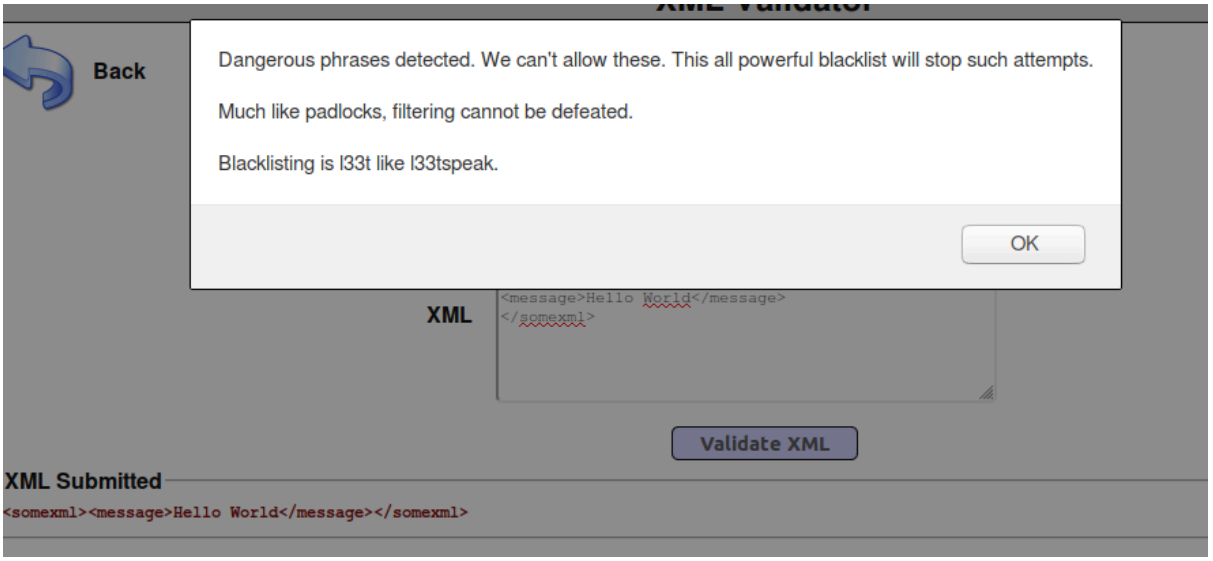
- petit test non malfaisant :



Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

- test avec le code malfaisant :

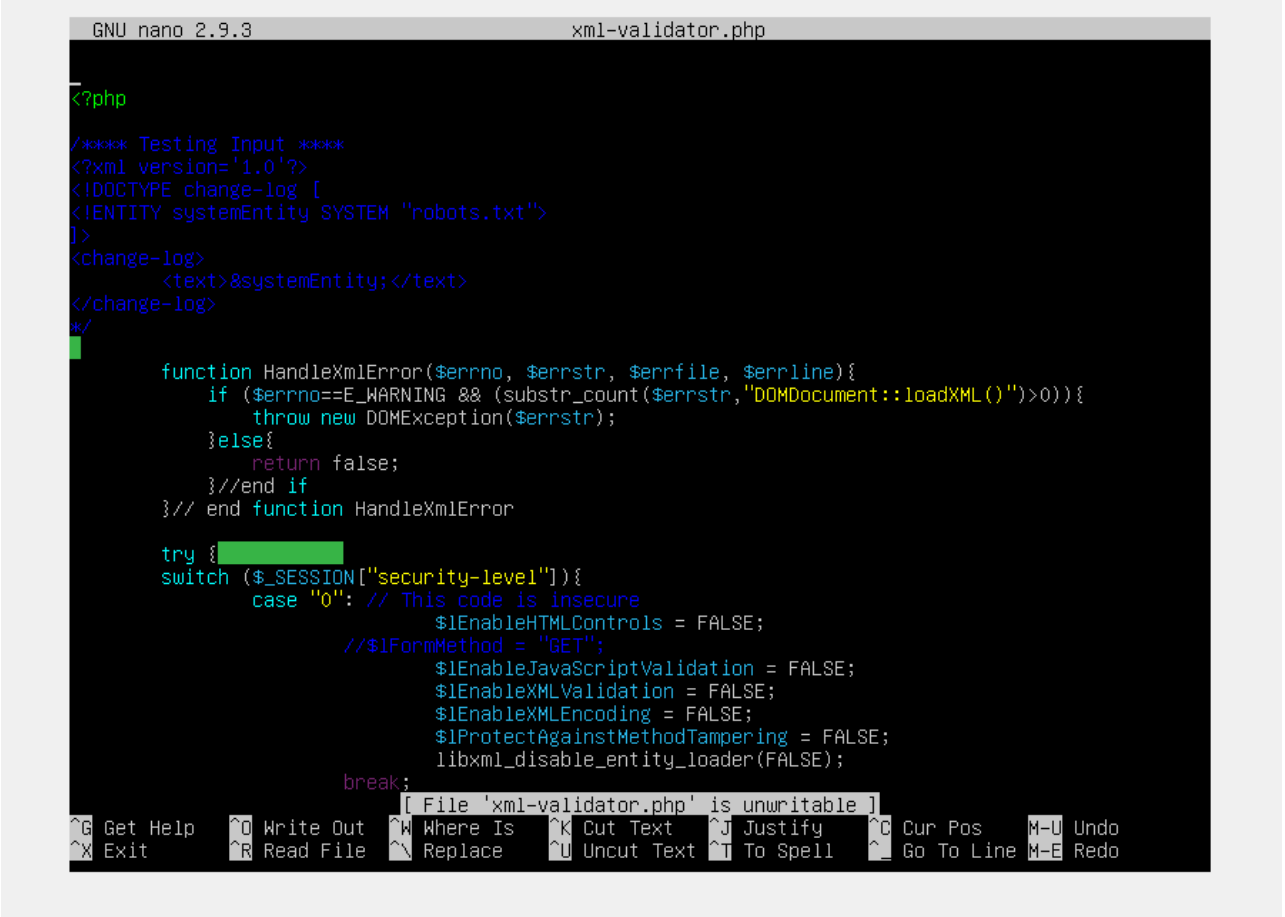


On constate que l'attaque à échoué et un message d'erreur s'affiche . Dans ce cas la, il faut modifier le code source de
protection du niveau 5 dans le fichier php xml-validator.php
(serveur ubuntu : cd mutillidae , nano xml-validator.php)

image lorsqu'on se trouve dans le fichier :

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd



modifier le level 5 : mettre à FALSE la méthode $EnableJavaScriptValidation :

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

```
            case "1": // This code is insecure
                        $lEnableHTMLControls = TRUE;
                    //$lFormMethod = "GET";
                        $lEnableJavaScriptValidation = TRUE;
                        $lEnableXMLValidation = FALSE;
                        $lEnableXMLEncoding = FALSE;
                        $lProtectAgainstMethodTampering = FALSE;
                        libxml_disable_entity_loader(FALSE);
                    break;

                    case "2":
                    case "3":
                    case "4":
            case "5": // This code is fairly secure
                        $lEnableHTMLControls = TRUE;
                    //$lFormMethod = "POST";
                        $lEnableJavaScriptValidation = FALSE;
                        $lEnableXMLValidation = TRUE;
                        $lEnableXMLEncoding = TRUE;
                        $lProtectAgainstMethodTampering = TRUE;
                        libxml_disable_entity_loader(TRUE);
                    break;
    }//end switch

    if ($lEnableHTMLControls) {
            $lHTMLControlAttributes='required="required"';
```

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

Travail à faire 2 - Nouvelle tentative en mode sécurisé et analyse du code source

En mettant le niveau de sécurité à 1 .

Q1. Est-ce que le niveau de sécurité 1 permet d'éviter l'attaque XXE ? Si oui, est-ce dû à une protection spécifique contre le XXE ?

En effet, le niveau de sécurité 1 permet d'éviter l'attaque, grace au code sécurisé , situé dans le fichier xml-validator.php :

la méthode EnableJavaScriptValidation est basé sur 'TRUE'

```
case "1": // This code is insecure
                $lEnableHTMLControls = TRUE;
        //$lFormMethod = "GET";
                $lEnableJavaScriptValidation = TRUE;
                $lEnableXMLValidation = FALSE;
                $lEnableXMLEncoding = FALSE;
                $lProtectAgainstMethodTampering = FALSE;
                libxml_disable_entity_loader(FALSE);
        break;
```

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

Questions - Niveau 5

-Est-ce que le niveau de sécurité 5 permet d'éviter l'attaque XXE ?

Oui, en arrivant au niveau 5 on constate que le code malveillant n'a pas fonctionné . Il est toutefois détecté : "*Possible XML external entity injection attack detected.Support has been notified"*

-Expliquer le rôle d'une expression régulière (motif).

une expression régulière est une suite de caractères normaux et de caractères spéciaux qui permet de trouver un mot .Cette suite de caractère va ensuite etre appliquer à une chaîne pour trouver les occurences correspondant a la place de l'expression , par exemple :

`^blob` Une chaîne de caractère commençant par blob (utilisation du ^)

`gulp$` Une chaîne de caractère se terminant par gulp (utilisation de $)

`^texte$` La chaîne globale " texte"

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

-afficher le code source de la page xml-validator.php sur le serveur à l'aide de la commande more/var/www/html/mutillidae/xml-validator.php :

```
<?php

/**** Testing Input ****
<?xml version='1.0'?>
<!DOCTYPE change-log [
<!ENTITY systemEntity SYSTEM "robots.txt">
]>
<change-log>
        <text>&systemEntity;</text>
</change-log>
*/

        function HandleXmlError($errno, $errstr, $errfile, $errline){
            if ($errno==E_WARNING && (substr_count($errstr,"DOMDocument::loadXML()")>0)){
                throw new DOMException($errstr);
            }else{
                return false;
            }//end if
        }// end function HandleXmlError

        try {
        switch ($_SESSION["security-level"]){
                case "0": // This code is insecure
                                $lEnableHTMLControls = FALSE;
                        //$lFormMethod = "GET";
                                $lEnableJavaScriptValidation = FALSE;
                                $lEnableXMLValidation = FALSE;
                                $lEnableXMLEncoding = FALSE;
                                $lProtectAgainstMethodTampering = FALSE;
                                libxml_disable_entity_loader(FALSE);
                        break;

                case "1": // This code is insecure
                                $lEnableHTMLControls = TRUE;
                        //$lFormMethod = "GET";
--More--(13%)
```

-Toujours dans la même page, expliquer le rôle de la fonction preg_match.

```
            try{
                    if(!($lEnableXMLValidation && (preg_match(XML_EXTERNAL_ENTITY_REGEX_PATTERN$
```

```
                try{
$ATTERNS, $lXML) || !preg_match(VALID_XML_CHARACTERS, $lXML))){
```

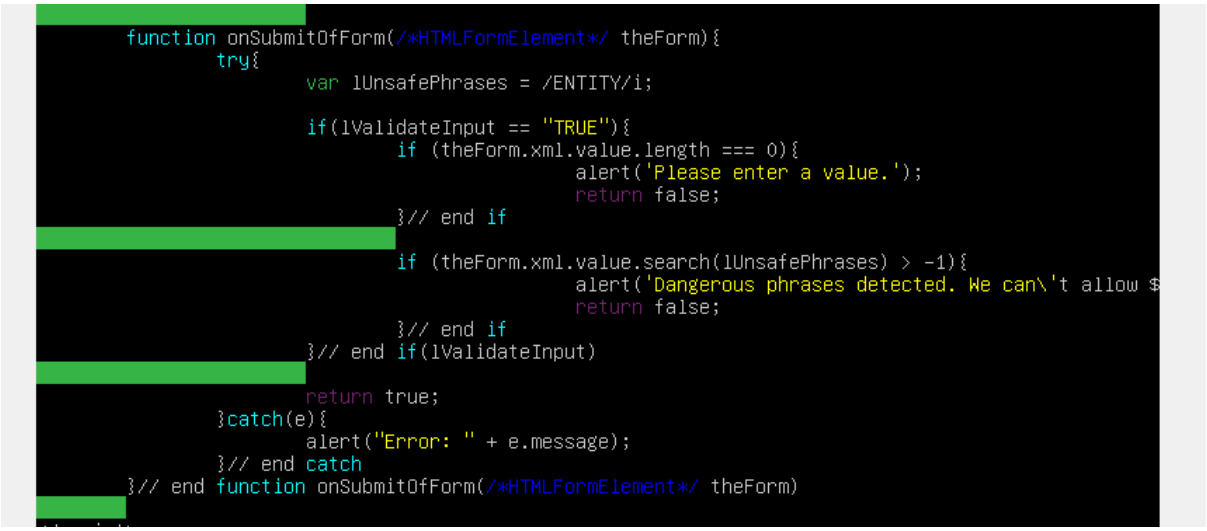
preg_match est une fonction php qui recherche le motif de la chaîne, et retourne true si le motif existe autrement retourne faux.Dans ce bout de code, la fonction

Maïssane El Mjidi
BTS SIO 2

Cybersécurité - Activité 3 : Récupération du fichier système /etc/passwd

Conclure sur la méthode de codage sécurisée utilisée pour empêcher l’attaque XXE.

fonction : onSubmitOfForm



```
function onSubmitOfForm(/*HTMLFormElement*/ theForm){
        try{
                var lUnsafePhrases = /ENTITY/i;

                if(lValidateInput == "TRUE"){
                        if (theForm.xml.value.length === 0){
                                alert('Please enter a value.');
                                return false;
                        }// end if

                        if (theForm.xml.value.search(lUnsafePhrases) > -1){
                                alert('Dangerous phrases detected. We can\'t allow $
                                return false;
                        }// end if
                }// end if(lValidateInput)

                return true;
        }catch(e){
                alert("Error: " + e.message);
        }// end catch
}// end function onSubmitOfForm(/*HTMLFormElement*/ theForm)
```