

Compte rendu

Q1) Commencer par installer l'extension Wsdler en réalisant les manipulations décrites dans l'étape n°1. Puis, positionner le niveau de sécurité à 0. :

- L'extension Wsdler a été installée dans BurpSuite via le Bapp Store. Cette extension permet d'intercepter les requêtes SOAP et de manipuler les fichiers WSDL, qui sont des descriptions des services web disponibles

- Le proxy de BurpSuite a été configuré pour intercepter les requêtes HTTP du navigateur Firefox. Les requêtes et réponses de l'application web sont redirigées à travers BurpSuite.

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Wsdler

Installed BApp Store APIa BChecks Extensions settings

Total estimated system impact: **Low**

BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Popularity	Last updated	System im...	Detail
Timestamp Editor		☆☆☆☆☆	18 Mar 2021	Low		
Token Extractor		☆☆☆☆☆	10 Feb 2022	Low		
Token Incrementor		☆☆☆☆☆	27 Nov 2020	Low		
TokenJar		☆☆☆☆☆	09 Jun 2022	Low		
Turbo Data Miner		☆☆☆☆☆	20 Apr 2022	Low		
Turbo Intruder		☆☆☆☆☆	07 Aug 2024	Medium		
Type Confusion Scan...		☆☆☆☆☆	11 Sep 2023	Low	Requires Burp...	
Upload Scanner		☆☆☆☆☆	21 Feb 2022	Low	Requires Burp...	
UPnP Hunter		☆☆☆☆☆	06 Dec 2021	Low		
URL Fuzzer - 401/403 ...		☆☆☆☆☆	09 Jan 2024	Low	Requires Burp...	
UUID Detector		☆☆☆☆☆	23 Feb 2017	Low		
ViewState Editor		☆☆☆☆☆	10 Mar 2021	Low		
WAF Bypassd		☆☆☆☆☆	07 Sep 2023	Low		
WAF Cookie Fetcher		☆☆☆☆☆	16 Jan 2018	Low		
WAFDetect		☆☆☆☆☆	25 Aug 2021	Low	Requires Burp...	
Wayback Machine		☆☆☆☆☆	18 Jun 2018	Low		
WCF Deserializer		☆☆☆☆☆	15 Jun 2017	Low		
Web Cache Deception ...		☆☆☆☆☆	23 Nov 2017	Low	Requires Burp...	
WebAuthn CBOR Dec...		☆☆☆☆☆	09 Dec 2022	Low		
WebInspect Connector		☆☆☆☆☆	10 Aug 2016	Low	Requires Burp...	
WebSocket Turbo Intr...		☆☆☆☆☆	14 Feb 2024	Low		
WebSphere Portlet Sta...		☆☆☆☆☆	17 Feb 2015	Low		
Wordlist Extractor		☆☆☆☆☆	20 Apr 2017	Low		
WordPress Scanner		☆☆☆☆☆	25 Feb 2022	Low		
WS Security		☆☆☆☆☆	10 Feb 2022	Medium		
WSDL Wizard		☆☆☆☆☆	01 Jul 2014	Low		
Wsdler	✓	☆☆☆☆☆	01 Nov 2016	Low		
XChromeLogger Deco...		☆☆☆☆☆	15 Dec 2021	Low		
XSS Cheatsheet		☆☆☆☆☆	17 Oct 2023	Low		
XSS Validator		☆☆☆☆☆	10 Feb 2022	High	Requires Burp...	
Yara		☆☆☆☆☆	25 Jan 2017	Low		
YesWeBurp		☆☆☆☆☆	25 Feb 2022	Low		

Refresh list Manual install ...

Wsdler

This extension takes a WSDL request, parses out the operations that are associated with the targeted web service, and generates SOAP requests that can then be sent to the SOAP endpoints.

To use this extension, select a suitable item in Burp, and choose "Parse WSDL" from the context menu.

The extension builds upon the work done by Tom Bujok and his soapws project which is essentially the WSDL parsing portion of SoapUI without the UI.

Requires Java version 8

Estimated system impact

Overall: **Low**

Memory: Low CPU: Low Time: Low Scanner: Low

Author: Eric Gruber
Version: 2.0.12
Source: <https://github.com/portswigger/wsdler>
Updated: 01 Nov 2016

Rating: ☆☆☆☆☆ Submitting
Popularity: ☆☆☆☆☆

Reinstall

Event log (2) All issues Memory: 120.8MB

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy Port

☐ Also use this proxy for HTTPS

HTTPS Proxy Port

SOCKS Host Port

☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

No proxy for

Q2) Tester un et de réponse à non valide en

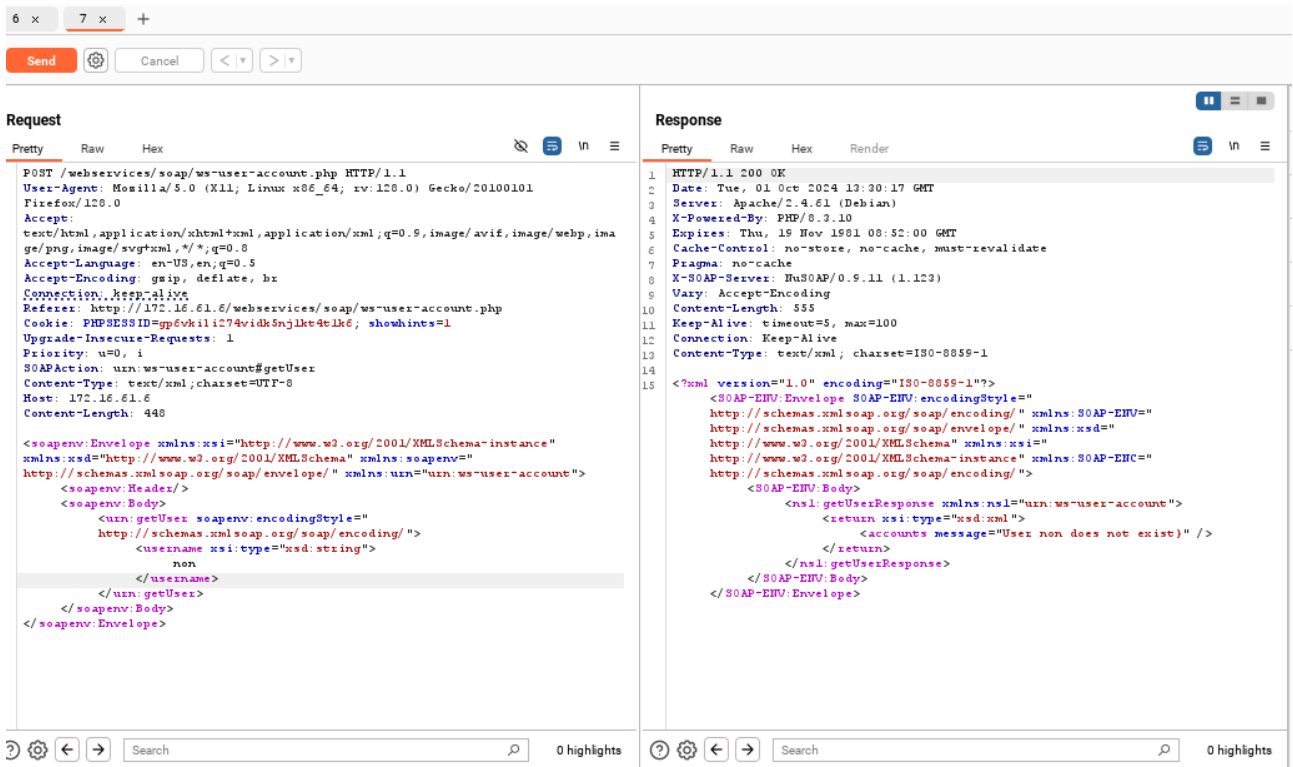
manipulations décrites dans l'étape n°2 (parse de la page wsdl, envoi au répéteur, génération de la réponse et envoi au comparateur) :

exemple de requête l'aide d'un login réalisant les

L'application Mutillidae a été configurée pour tester l'énumération de logins. En accédant au service web SOAP proposé par l'application, la requête SOAP a été modifiée pour utiliser un login non valide, ici « gero et ».

-La requête a été envoyée au serveur, et la réponse obtenue est visible dans BurpSuite : "User gero et does not exist". Cette réponse confirme que le serveur renvoie un message explicite lorsqu'un login non valide est utilisé.

-Cette réponse a été envoyée au Comparer de BurpSuite, qui sera utilisé plus tard pour analyser les différences avec une réponse pour un login valide.

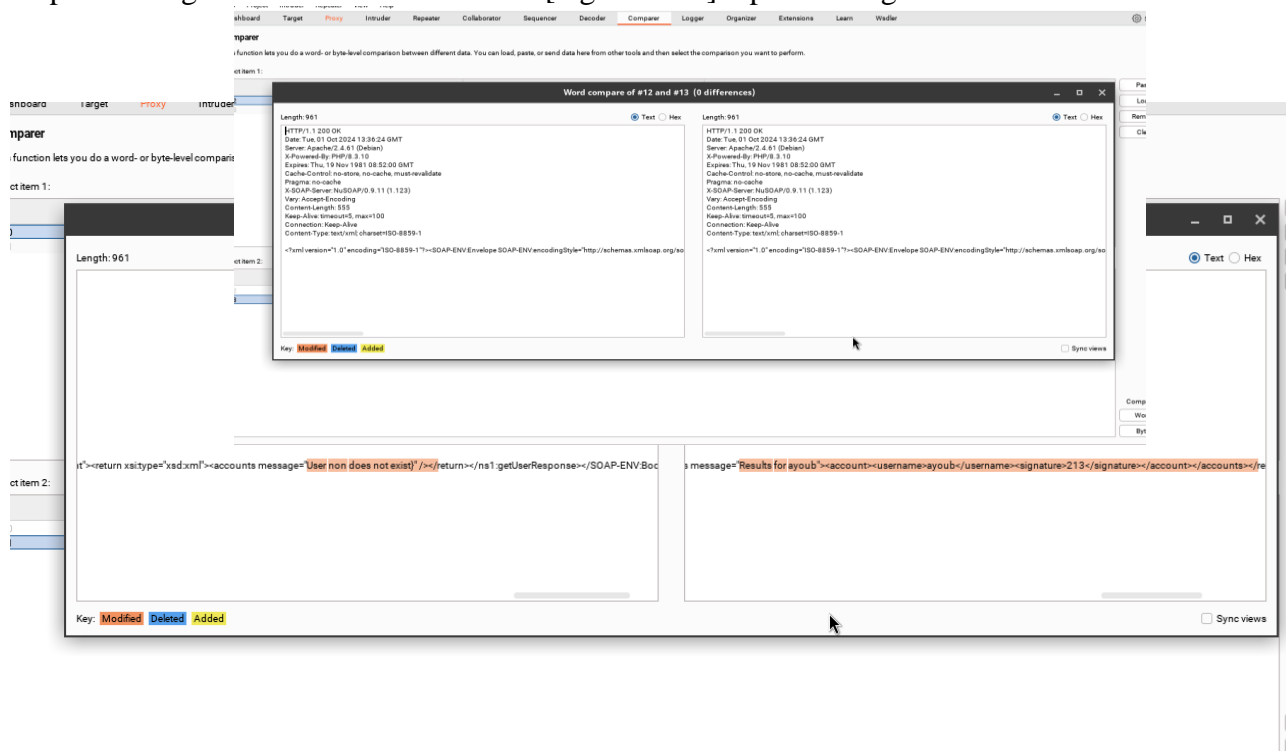


Q3) Tester un exemple de requête et de réponse à l'aide d'un login valide en réalisant les manipulations décrites dans l'étape n°3 (parse de la page wsdl, envoi au répéteur, modification avec un login valide, génération de la réponse et envoi au comparateur) :

Un login valide a été utilisé pour tester l'application. Dans ce cas, le login ayoub a été testé. La requête SOAP a été envoyée avec ce login, et la réponse renvoyée par le serveur indique un succès : "Results for ayoub".

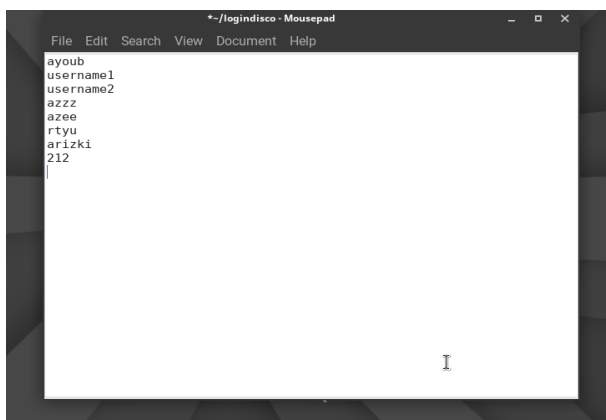
-Cette réponse a été également envoyée au Comparer de BurpSuite pour comparaison avec celle obtenue dans la question 2.

-Les réponses avec un login valide et un login non valide ont des différences notables. Le Comparer de BurpSuite met en évidence les lignes spécifiques, notamment les messages « User does not exist » pour un login invalide et « Results for [login valide] » pour un login valide.



Q4) Créer un dictionnaire de login sur votre machine cliente. Pour cela, ouvrir un éditeur de texte et saisir des logins les uns en dessous des autres et enregistrer votre fichier :

-Un fichier texte a été créé avec différents logins potentiels (par exemple, ayoub, username1, username2, etc.). Chaque login a été placé sur une ligne distincte dans un éditeur de texte.



Q5)Lancer l'énumération et relever les logins valides en réalisant les manipulations décrites dans l'étape n°4.

La requête SOAP a été envoyée au module Intruder de BurpSuite pour automatiser l'énumération des logins à partir du fichier dictionnaire créé en Q4.

-Dans l'onglet Payload, le fichier dictionnaire a été chargé, et dans l'onglet Positions, le champ du login a été marqué comme variable à remplacer lors de l'attaque.

-Un filtre a été ajouté pour repérer les réponses avec le texte « Results for », permettant ainsi de détecter automatiquement les logins valides.

-L'attaque a été lancée, et les résultats montrent que les logins « ayoub », « username1 », et « username2 » sont valides.

4. Intruder attack of http://172.16.61.6

Attack Save

Results Positions Payloads Resource pool Settings

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Results for	Comment
0		200	71			1034	ayoub">=account=<username...	
1	ayoub	200	16			1033	ayoub">=account=<username...	
2	username1	200	35			1042	username1">=account=<usern...	
3	username2	200	14			1041	username2">=account=<usern...	
4	moh	200	25			961		
5	arizki	200	14			963		

Target: http://172.16.61.6 Update Host header to match target

Ad Cle Au Ref

```
POST /webservices/soap/ws-user-account.php HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://172.16.61.6/webservices/soap/ws-user-account.php
Cookie: PHPSESSID=gg6vkilic74vdk5njlkt4t1k6; showhins=1
Upgrade-Insecure-Requests: 1
Priority: u=0, i
SOAPAction: urn:ws-user-account#getUser
Content-Type: text/xml; charset=UTF-8
Host: 172.16.61.6
Content-Length: 450

<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:xi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:ws-user-account">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getUser soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <username xsi:type="xsd:string">ayoub</username>
    </urn:getUser>
  </soapenv:Body>
</soapenv:Envelope>
```

Payload settings [Simple list]

This payload type lets you configure a simple list of strings

Paste

Load ...

Remove

Clear

Deduplicate

Add

Enter a new item

Add from list ... [Pro version only]

ayoub

username1

username2

moh

arizki

Music

Pictures

Public

Templates

Videos

logindisco

Define the location of the item to be extracted. Selecting the item in automatically. You can also modify the configuration manually to e

☒ Define start and end

☒ Start after expression: Results for

☐ Start at offset: 877

☒ End at delimiter:

☐ End at fixed length:

☐ Exclude HTTP headers ☒ Update config based on selection

Q6) A l'aide du comparateur, expliquer quelles sont les lignes de la réponse sur lesquelles l'attaquant a pu s'appuyer pour lancer l'attaque ?

L'outil Comparer de BurpSuite a été utilisé pour comparer les réponses du serveur entre un login valide et un login non valide :

-Les lignes pertinentes sont celles qui diffèrent dans les deux réponses. Pour un login invalide, le message est « User does not exist », tandis que pour un login valide, on obtient « Results for [login] ».

-Cette différence dans les réponses permet à un attaquant de déterminer facilement quels logins sont valides dans l'application.

Travail à faire 2 :

Q1) Fermer puis relancer BurpSuite. Positionner le niveau de sécurité à 5 et relancer l'attaque en suivant les étapes 2 à 4:

-Redémarrage de BurpSuite : Après avoir testé l'application en mode non sécurisé, BurpSuite a été relancé et reconfiguré pour l'attaque en mode sécurisé.

-Changement du niveau de sécurité : L'application web Mutillidae a été configurée avec un niveau de sécurité de 5, ce qui active les protections contre les attaques par énumération de logins.

⑩ Lancement de l'attaque avec le mode sécurisé :

-L'attaque précédente a été relancée en suivant les mêmes étapes que dans le Travail à faire 1, c'est-à-dire en interceptant les requêtes SOAP et en utilisant un dictionnaire de logins pour tester les réponses.

-Les requêtes SOAP ont été envoyées en utilisant à nouveau le service getUser de l'API SOAP de Mutillidae.

<pre> user-agent: Mozilla/5.0 (AAA; Linux x86_64; rv:100.0) Gecko/20100101 Firefox/120.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima ge/png,image/svg+xml,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate, br </pre>	<pre> 3 Server: Apache/2.4.61 (Debian) 4 X-Powered-By: PHP/8.2.10 5 Expires: Thu, 19 Nov 1981 08:52:00 GMT 6 Cache-Control: no-store, no-cache, must-revalidate 7 Pragma: no-cache 8 X-SOAP-Server: NuSOAP/0.9.11 (1.123) 9 Vary: Accept-Encoding </pre>
---	--

[Home](#) | [Login/Register](#) | [Toggle Security](#) | [Enforce TLS](#) | [Reset DB](#) | [View Log](#) | [View Captured Data](#)

```
<?xml version='1.0' encoding='utf-8' ?>
<SOAP-ENV:Body>
  <nsl:getUserResponse xmlns:nsl="urn:ws-user-account">
    <return nsl:type="xsd:xm1">
      <accounts message="Results for ayoub">
        <account>
          <username>
            ayoub
          </username>
          <signature>
            213
          </signature>
        </account>
      </accounts>
    </return>
  </nsl:getUserResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

⑩ Vérification des informations dans le comparateur :

-Contrairement au mode non sécurisé, en mode sécurisé, les réponses sont moins explicites.

L'application ne renvoie rien n indiquant pas si le login existe.

10 Résultat de l'attaque en mode sécurisé :

- Les informations obtenues dans le comparateur ne permettent plus de distinguer facilement les logins valides des non valides. L'application sécurisée renvoie un message identique pour les deux cas, empêchant ainsi l'énumération des logins à l'aide des différences dans les réponses du serveur.

[illegible]

Q3) Chercher dans le code source de la page `ws-user-account.php` (située dans `/var/www/html/mutillidae/webservices/soap/`) le codage mis en place permettant d'obtenir un encodage sécurisé. Expliquer le rôle de l'instruction `EncodeforHTML`.

-Le code implémente un service web SOAP sécurisé pour la gestion des comptes utilisateurs, il comprend des fonctionnalités telles que la création, la mise à jour, la suppression et la récupération d'informations utilisateurs. Le niveau de sécurité, défini dans la session (de 0 à 5), détermine si les données sont encodées avant d'être renvoyées dans les réponses SOAP. En mode non sécurisé (niveaux 0 et 1), le code est vulnérable aux attaques comme les injections SQL car les données ne sont pas encodées, laissant potentiellement des failles exploitables. En revanche, aux niveaux de sécurité 2 à 5, la méthode `encodeForHTML()` est activée pour encoder les données sensibles telles que les noms d'utilisateur et les signatures avant leur affichage ou envoi dans les réponses. Cette fonction transforme les caractères spéciaux (comme `<`, `>`, et `&`) en équivalents HTML sécurisés, empêchant ainsi l'injection de scripts malveillants dans les pages web. Par exemple, un utilisateur malveillant qui tenterait d'injecter un script via un champ de formulaire verrait ce script converti en texte inoffensif, empêchant son exécution. Chaque méthode du service (`getUser`, `createUser`, `updateUser`, `deleteUser`) suit une logique sécurisée : les paramètres sont validés avec la fonction `assertParameter()`, les erreurs sont traitées de manière générique pour ne pas révéler d'informations sensibles aux attaquants (comme l'existence d'un utilisateur), et les sorties sont encodées pour éviter les injections. De plus, les messages d'erreur restent toujours génériques (par exemple, « User does not exist » ou « Inserted account »), évitant les fuites d'informations qui pourraient permettre des attaques d'énumération de logins.


```

/* Example SQL injection: jeremy' union select username,password from accounts -- */

f (session_status() == PHP_SESSION_NONE){
    session_start();
// end if

f (!isset($_SESSION["security-level"])){
    $_SESSION["security-level"] = 0;
// end if

/* -----
 * Constants used in application
 * ----- */
require_once('.../includes/constants.php');
require_once('.../includes/minimum-class-definitions.php');

try{
    switch ($_SESSION["security-level"]){
        case "0": // This code is insecure
        case "1": // This code is insecure
            $lEncodeOutput = FALSE;
            break;

        case "2":
        case "3":
        case "4":
        case "5": // This code is fairly secure
            $lEncodeOutput = TRUE;
            break;

    }//end switch

} catch (Exception $e) {
    echo $CustomErrorHandler->FormatError($e, "ws-user-account.php: Unable to parse session");
}
}

```