MAJOR PROJECT

-Under the Guidance of **Sheela Madam**

-Under **Adiroha|Rinex Team**

-By:- **Vasudha.G**

Problem Statements

- Broken Authentication and session management
- Open Directories Vulnerability
- Common Attacks Using Kali Linux

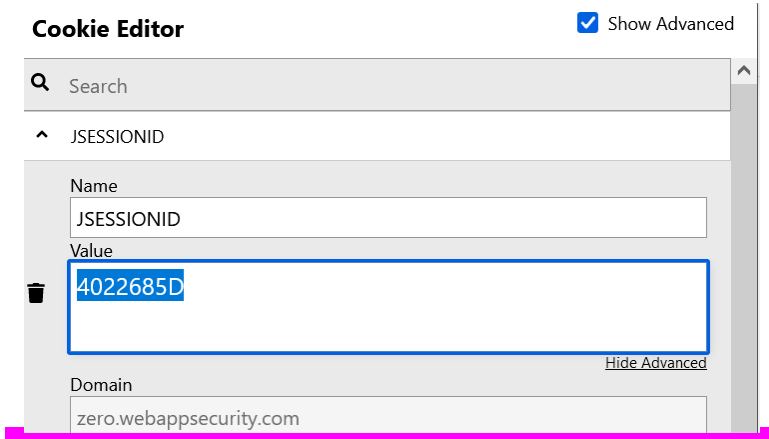
Using Zero.webappsecurity.com

Broken Authentication and Session Management

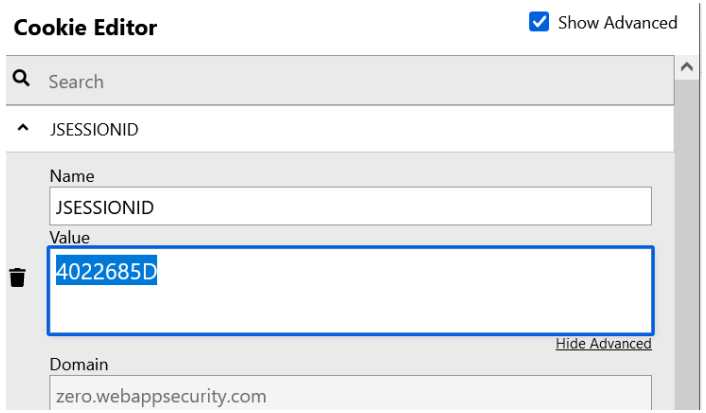
1) Session Hijacking:-

Session hijacking is an attack where a user session is taken over by an attacker. before and after login,the session id's geberated is not same. if it is same, then it becomes the session hijacking vulnerability.

Before login



After login



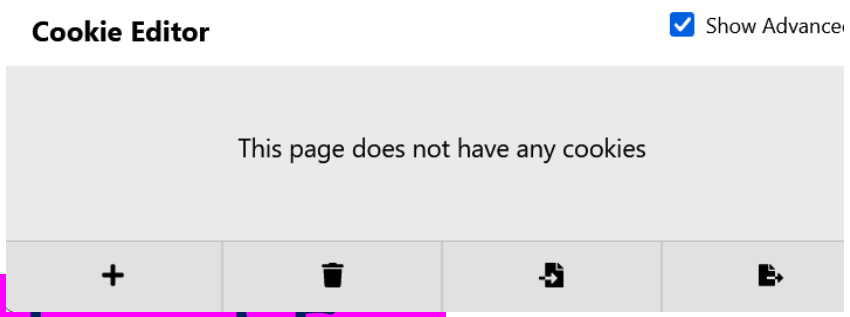
zero.webappsecurity is Vulnerable to Session Hijacking!!

1)Session Fixation

_Session fixation attacks attempt to exploit the vulnerability of a system that allows one person to fixate another person's session identifier. Before Logout and After logout, session id's should not be same.

Before Logout:-

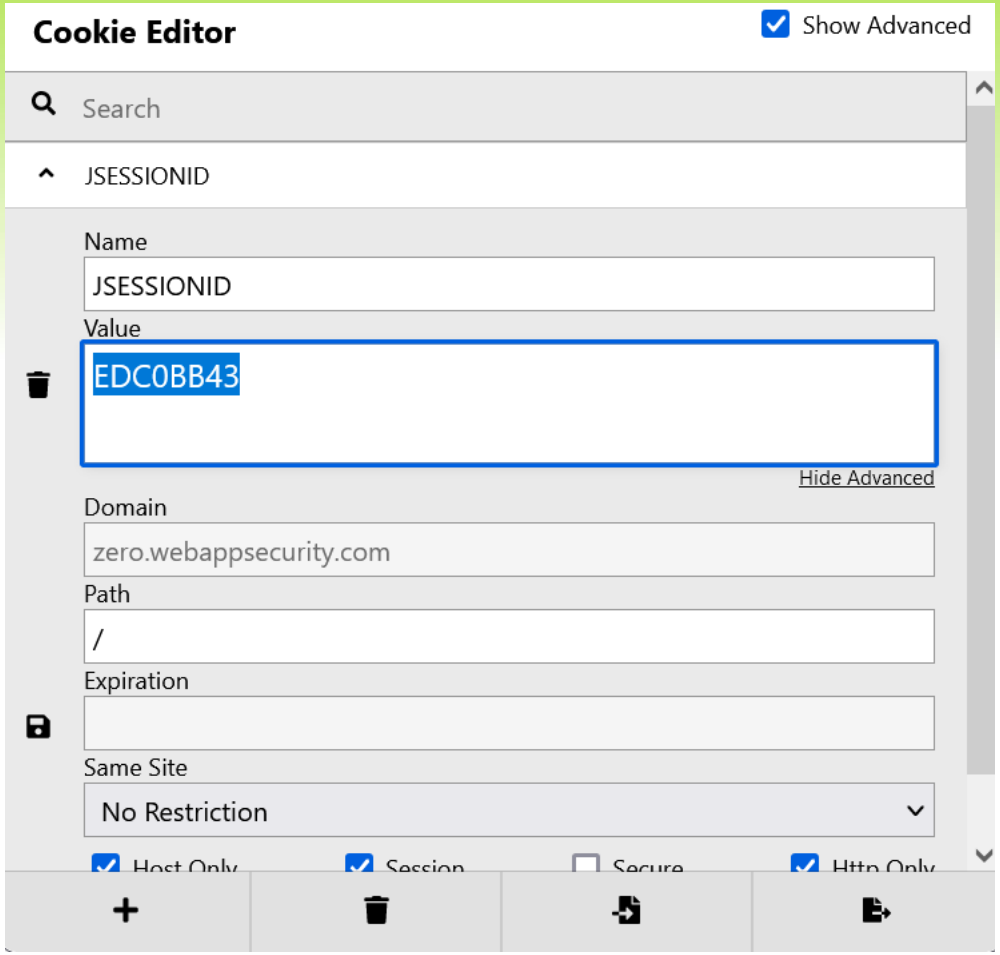
After logout:-



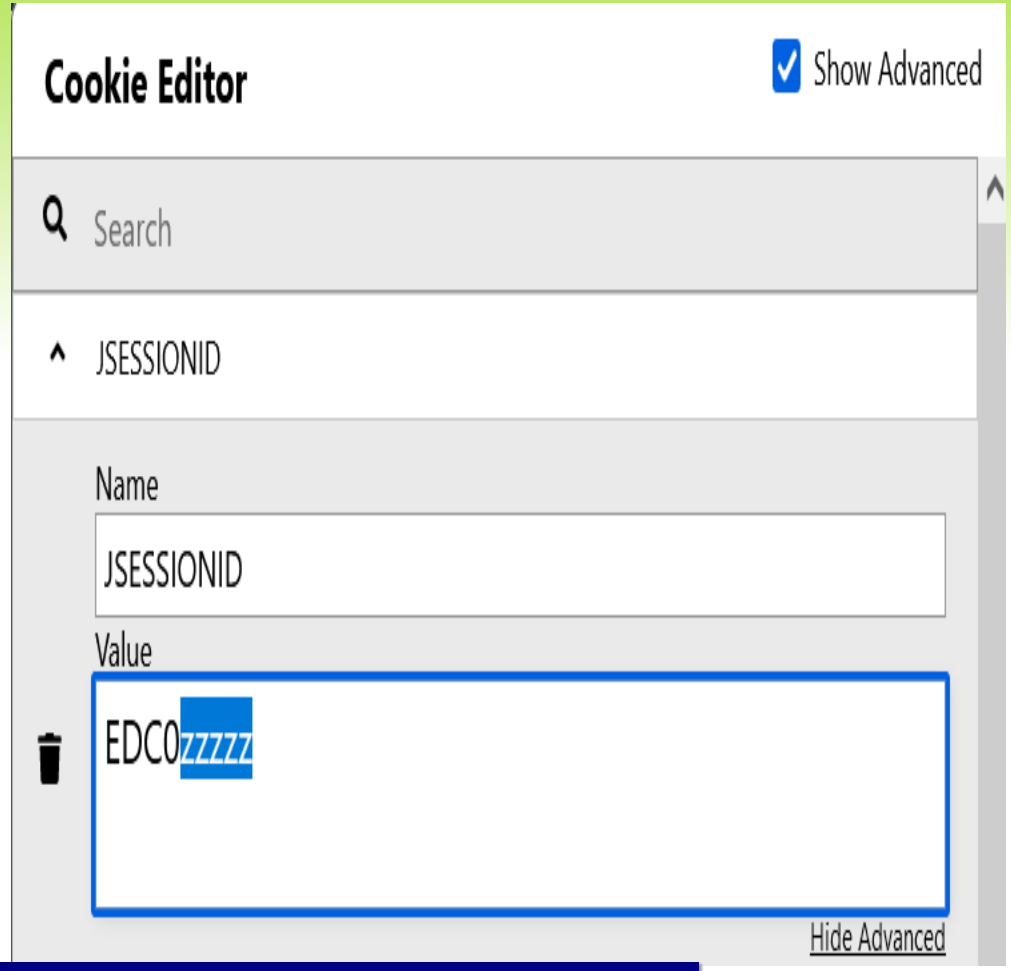
Since There Is No Session ID Seen after Logout For zero.webappsecurity.com, Hence It Is Not Vulnerable To Session Fixation!!

3) User Given Session ID and Validation

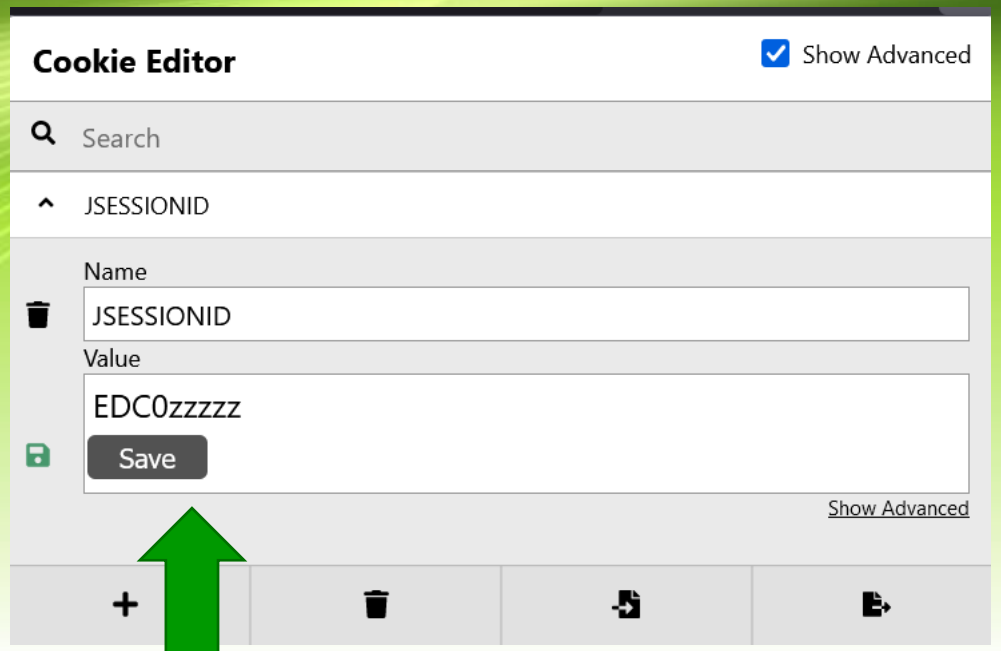
Any website should not accept the user give session id's. If it accepts then it will be the vulnerability of the paricular website.

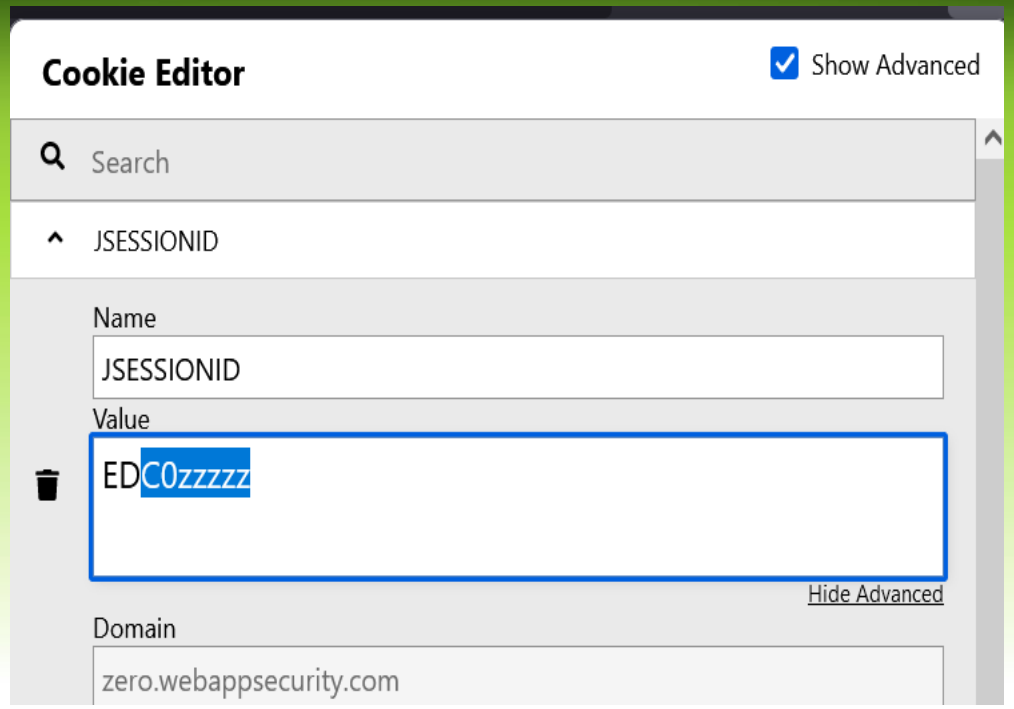


Session id generated



Giving my own session id





Saving the given Session id

Given Session id is generated after refreshing

As it accepts the User given Session Id, The website has User Given Session Id and Validation Vulnerability!!

4)HTTP and Secure Flag:-

If both HTTP and Secure flag are enabled it means that the website is safe.

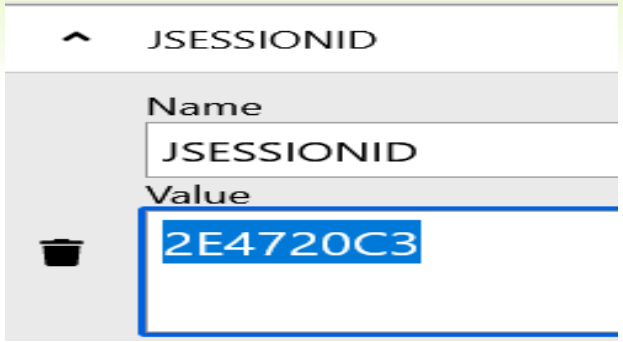
Only HTTP is enabled whereas the secure flag is disabled, hence its a Vulnearbility!!!

5) Session Id's generated should not be a random value. It should be the combination of both **Alphabets** and the **Numbers.**

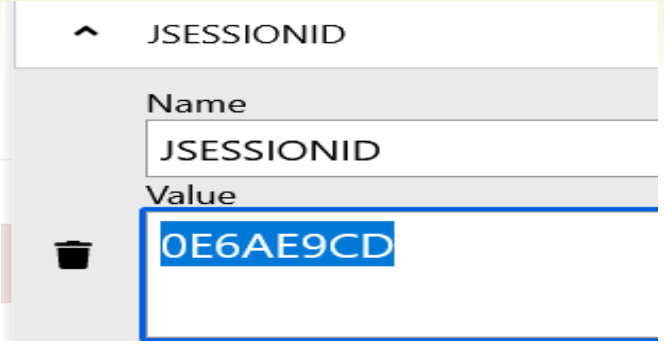
Its not a vulnearbilty as it is the combination of both alphabets and numbers!!

6) Weak Session Id Generation:-

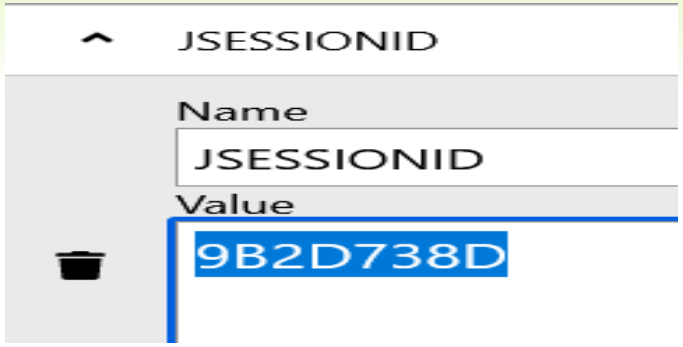
Login with the same user Id, multiple times and the check the generated Session Id's. If same session Id is generated then we can use them for life long or change the password by hacking any account and thus it becomes a vulnarbility.



1st User



2nd User

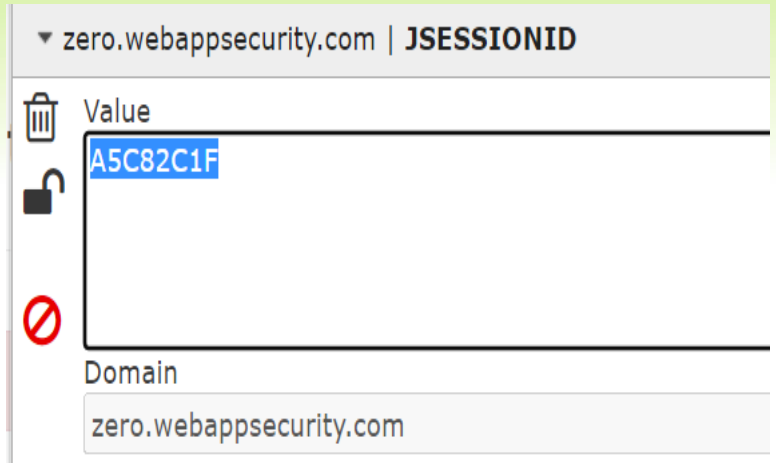


3rd User

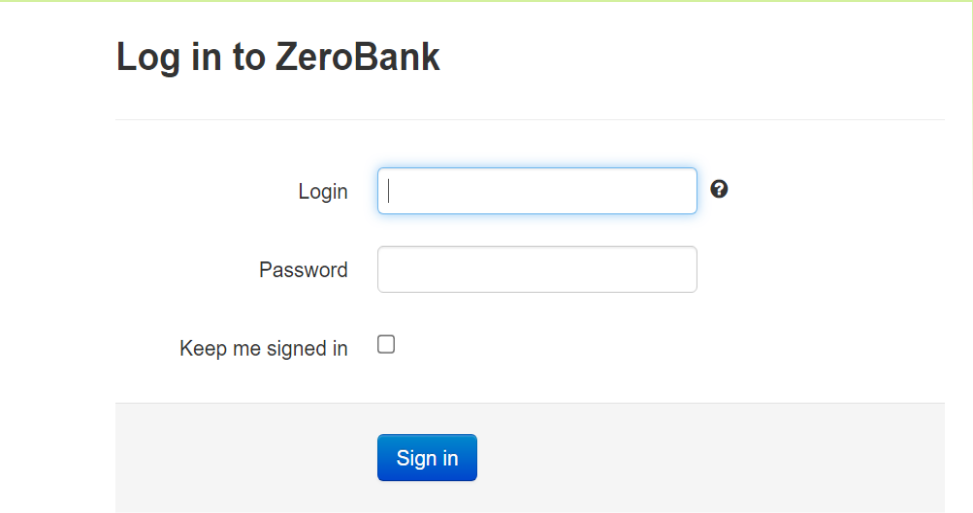
Session Id's generated are not same, hence its not a Vulnerability!!

7)Browser Cache Weakness:-

When A Browser Cache Stores The Login Details Even After Logging Out And Browser Can Access The Logged In State On Returning Back To The Site Using History. Then There's Browser Cache Weakness Seen.



Logging in to the website

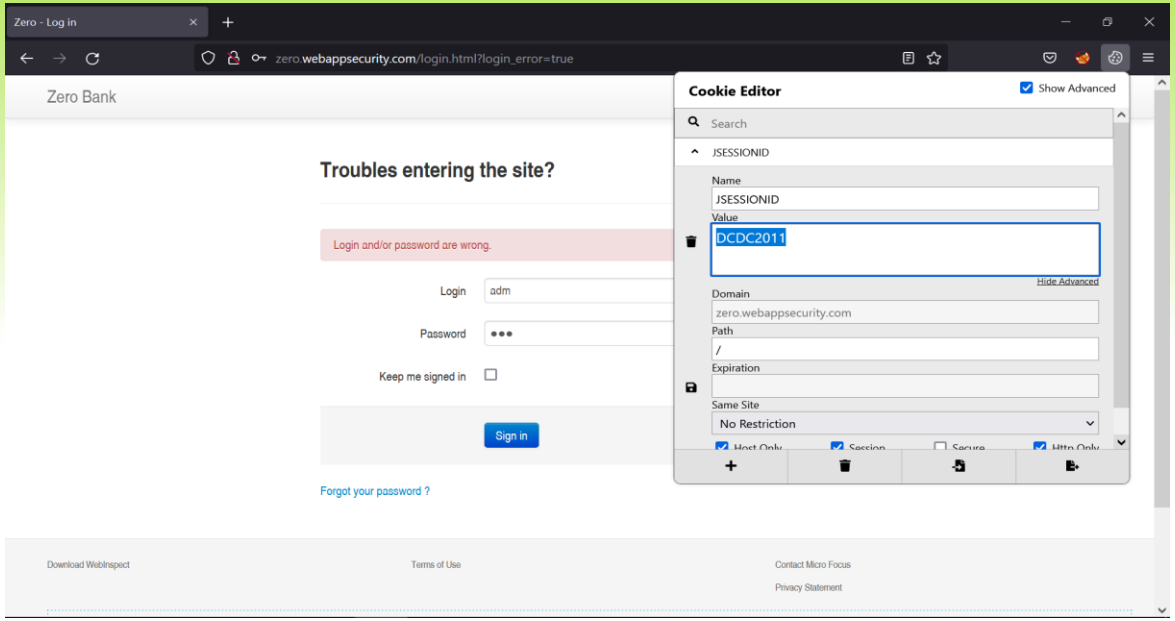


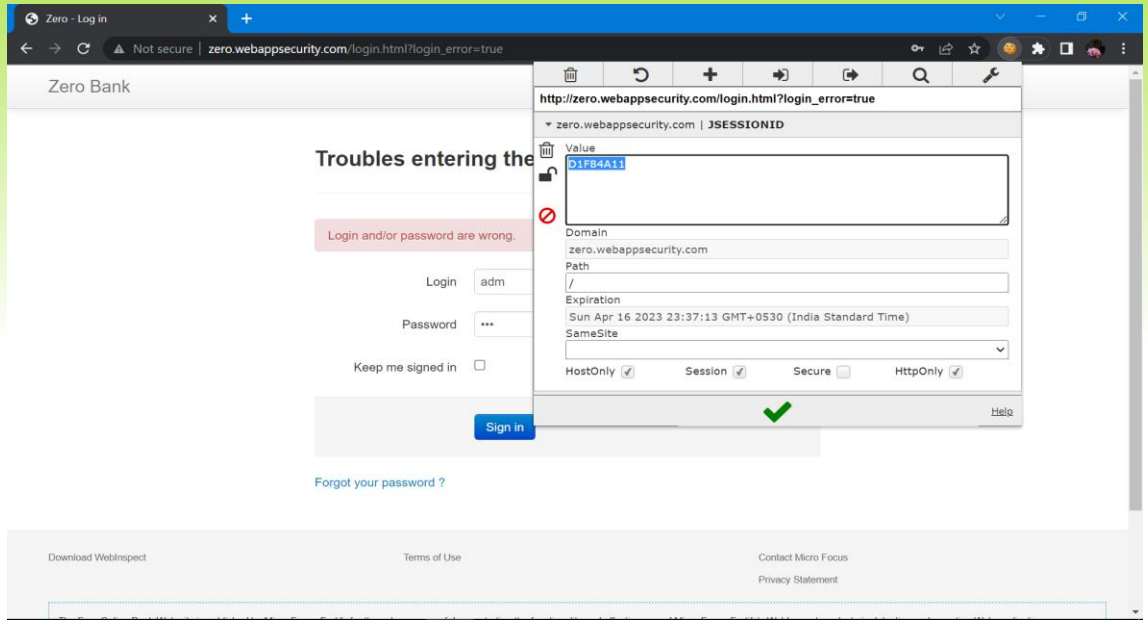
After log out

It does not support Browser Cache Weakness Vulnerability!!

8) Concurrent and Parallel Logins:-

If we log in the website with two or more different browsers simultaneously, its a vulnerability.





Loggin through Firefox

Loggin through Google Chrome

Since the website allows to log in simultaneously with different broweser, its a Vulnerability!!

Open Directories

Open Directories are basically unprotected directories that are used to share numerous types of data: documents, pictures, videos, databases, and software.These Can Be Directly Accessed From the Url Of the Webpage and Hence are Dangerous.

Common Open Directories

Common Open Dirs List:

bin , admin ,styles , css , js , log , images , readme, modules , lib, assets flags , doc , users ,auth ,forum , apache ,media ,logfiles ,authadmin ,tmp

Go to Chome and search for the Zero.webapp.security.com login users

zero.webappsecurity.com/admin/users.html

Zero Bank

Search

Settings

username

Users

Home

Users

Currencies

Name	Password	SSN
Leeroy Jenkins	VIZ10AWT8VL	536-48-3769
Stephen Bowen	OTZ07BXM0BE	607-58-7435
Linus Moran	FKO04SXA7TI	247-54-1719
Nero Chan	TXJ77CQO5EI	578-13-3713
Kadeem Higgins	MFC50OQE7VO	449-20-3206
Quinn Burks	HWZ97ZUM3NK	008-70-6738
Davis Thompson	RGD78SHB0TG	574-56-1932
Lester Keller	EIJ79NLT0TP	330-58-4012

Download WebInspect

Terms of Use

Contact Micro Focus

Privacy Statement

admin Open **Directory** accessing users.html and **currencies**.html

Docs Open Directory accessing documentation:-

Zero - Admin - Users | Logins & Passwords | Apache Tomcat 7 (7.0.70) - Docume

zero.webappsecurity.com/docs/ 67%

The Apache Tomcat Servlet/JSP Container

Apache Tomcat 7

Version 7.0.70, Jun 15 2016

Apache Logo

Links

- Docs Home
- FAQ
- User Comments

User Guide

- 1) Introduction
- 2) Setup
- 3) First webapp
- 4) Deployer
- 5) Manager
- 6) Realms and AAA
- 7) Security Manager
- 8) JNDI Resources
- 9) JDBC DataSources
- 10) Classloading
- 11) JSPs
- 12) SSL/TLS
- 13) SSI
- 14) CGI
- 15) Proxy Support
- 16) MBean Descriptor
- 17) Default Servlet
- 18) Clustering
- 19) Load Balancer
- 20) Connectors
- 21) Monitoring and Management
- 22) Logging
- 23) APR/Native
- 24) Virtual Hosting
- 25) Advanced IO
- 26) Additional Components
- 27) Mavenized
- 28) Security Considerations
- 29) Windows Service
- 30) Windows Authentication
- 31) Tomcat's JDBC Pool
- 32) WebSocket

Reference

- Release Notes
- Configuration

Documentation Index

Introduction

This is the top-level entry point of the documentation bundle for the **Apache Tomcat** Servlet/JSP container. Apache Tomcat version 7.0 implements the Servlet 3.0 and JavaServer Pages 2.2 specifications from the Java Community Process, and includes many additional features that make it a useful platform for developing and deploying web applications and web services.

Select one of the links from the navigation menu (to the left) to drill down to the more detailed documentation that is available. Each available manual is described in more detail below.

Apache Tomcat User Guide

The following documents will assist you in downloading, installing Apache Tomcat 7, and using many of the Apache Tomcat features.

1. **Introduction** - A brief, high level, overview of Apache Tomcat.
2. **Setup** - How to install and run Apache Tomcat on a variety of platforms.
3. **First web application** - An introduction to the concepts of a *web application* as defined in the Servlet Specification. Covers basic organization of your web application source tree, the structure of a web application archive, and an introduction to the web application deployment descriptor (`/WEB-INF/web.xml`).
4. **Deployer** - Operating the Apache Tomcat Deployer to deploy, precompile, and validate web applications.
5. **Manager** - Operating the **Manager** web app to deploy, undeploy, and redeploy applications while Apache Tomcat is running.
6. **Realms and Access Control** - Description of how to configure *Realms* (databases of users, passwords, and their associated roles) for use in web applications that utilize *Container Managed Security*.
7. **Security Manager** - Configuring and using a Java Security Manager to support fine-grained control over the behavior of your web applications.
8. **JNDI Resources** - Configuring standard and custom resources in the JNDI naming context that is provided to each web application.
9. **JDBC DataSource** - Configuring a JNDI DataSource with a DB connection pool. Examples for many popular databases.
10. **Classloading** - Information about class loading in Apache Tomcat, including where to place your application classes so that they are visible.
11. **JSPs** - Information about Jasper configuration, as well as the JSP compiler usage.
12. **SSL/TLS** - Installing and configuring SSL/TLS support so that your Apache Tomcat will serve requests using the `https` protocol.
13. **SSI** - Using Server Side Includes in Apache Tomcat.
14. **CGI** - Using CGIs with Apache Tomcat.
15. **Proxy Support** - Configuring Apache Tomcat to run behind a proxy server (or a web server functioning as a proxy server).
16. **MBean Descriptor** - Configuring MBean descriptors files for custom components.
17. **Default Servlet** - Configuring the default servlet and customizing directory listings.
18. **Apache Tomcat Clustering** - Enable session replication in a Apache Tomcat environment.
19. **Balancer** - Configuring, using, and extending the load balancer application.
20. **Connectors** - Connectors available in Apache Tomcat, and native web server integration.
21. **Monitoring and Management** - Enabling JMX Remote support, and using tools to monitor and manage Apache Tomcat.
22. **Logging** - Configuring logging in Apache Tomcat.
23. **Apache Portable Runtime** - Using APR to provide superior performance, scalability and better integration with native server technologies.
24. **Virtual Hosting** - Configuring virtual hosting in Apache Tomcat.
25. **Advanced IO** - Extensions available over regular, blocking IO.
26. **Additional Components** - Obtaining additional, optional components.
27. **Using Tomcat libraries with Maven** - Obtaining Tomcat jars through Maven.
28. **Security Considerations** - Options to consider when securing an Apache Tomcat installation.
29. **Windows Service** - Running Tomcat as a service on Microsoft Windows.
30. **Windows Authentication** - Configuring Tomcat to use integrated Windows authentication.
31. **High Concurrency JDBC Pool** - Configuring Tomcat to use an alternative JDBC pool.

using http://zero.webappsecurity.com/docs

Not secure | zero.webappsecurity.com/web-services/infoService

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  ▼<soap:Body>
    ▼<soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Fault occurred while processing.</faultstring>
     ▼<detail>
       <stackTrace
       xmlns="http://cxf.apache.org/fault">com.hp.webinspect.zero.ws.interceptor.SoapVulnerabilityEmulationInjector!findAndEmulateMappedVulnerabilities!SoapVulnerabilityEmula
       com.hp.webinspect.zero.ws.interceptor.SoapVulnerabilityEmulationInjector!handleMessage!SoapVulnerabilityEmulationInjector.java!46
       com.hp.webinspect.zero.ws.interceptor.SoapVulnerabilityEmulationInjector!handleMessage!SoapVulnerabilityEmulationInjector.java!23
       org.apache.cxf.phase.PhaseInterceptorChain!doIntercept!PhaseInterceptorChain.java!271
       org.apache.cxf.transport.ChainInitiationObserver!onMessage!ChainInitiationObserver.java!121
       org.apache.cxf.transport.http.AbstractHTTPDestination!invoke!AbstractHTTPDestination.java!239
       org.apache.cxf.transport.servlet.ServletController!invokeDestination!ServletController.java!218
       org.apache.cxf.transport.servlet.ServletController!invoke!ServletController.java!198
       org.apache.cxf.transport.servlet.ServletController!invoke!ServletController.java!137
       org.apache.cxf.transport.servlet.CXFNonSpringServlet!invoke!CXFNonSpringServlet.java!158
       org.apache.cxf.transport.servlet.AbstractHTTPServlet!handleRequest!AbstractHTTPServlet.java!243
       org.apache.cxf.transport.servlet.AbstractHTTPServlet!doGet!AbstractHTTPServlet.java!168 javax.servlet.http.HttpServlet!service!HttpServlet.java!624
       org.apache.cxf.transport.servlet.AbstractHTTPServlet!service!AbstractHTTPServlet.java!219
       org.apache.catalina.core.ApplicationFilterChain!internalDoFilter!ApplicationFilterChain.java!303
       org.apache.catalina.core.ApplicationFilterChain!doFilter!ApplicationFilterChain.java!208
       com.hp.webinspect.zero.web.FakeCommonFoldersEmulator!doFilter!FakeCommonFoldersEmulator.java!39
       org.apache.catalina.core.ApplicationFilterChain!internalDoFilter!ApplicationFilterChain.java!241
       org.apache.catalina.core.ApplicationFilterChain!doFilter!ApplicationFilterChain.java!208
       org.springframework.web.filter.CharacterEncodingFilter!doFilterInternal!CharacterEncodingFilter.java!88
       org.springframework.web.filter.OncePerRequestFilter!doFilter!OncePerRequestFilter.java!107
       org.apache.catalina.core.ApplicationFilterChain!internalDoFilter!ApplicationFilterChain.java!241
       org.apache.catalina.core.ApplicationFilterChain!doFilter!ApplicationFilterChain.java!208
       org.tuckey.web.filters.urlrewrite.UrlRewriteFilter!doFilter!UrlRewriteFilter.java!399
       org.apache.catalina.core.ApplicationFilterChain!internalDoFilter!ApplicationFilterChain.java!241
       org.apache.catalina.core.ApplicationFilterChain!doFilter!ApplicationFilterChain.java!208
       org.apache.catalina.core.StandardWrapperValve!invoke!StandardWrapperValve.java!218 org.apache.catalina.core.StandardContextValve!invoke!StandardContextValve.java!122
       org.apache.catalina.authenticator.AuthenticatorBase!invoke!AuthenticatorBase.java!505 org.apache.catalina.core.StandardHostValve!invoke!StandardHostValve.java!169
```


http://zero.webappsecurity.com/web-service

http://zero.webappsecurity.com/server-status

Zero - Admin - Users | Logins & Passwo | New Tab | 5. http://zero.wel | Apache Tomcat/7.0.7 | Apache Tomcat/7.0.7 | Apache Status | New Tab

zero.webappsecurity.com/server-status 67%

Apache Server Status for localhost

Server Version: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8t mod_jk/1.2.37
Server Built: Jan 28 2012 11:16:39

Current Time: Friday, 18-Jan-2013 14:55:36 GMT
Restart Time: Friday, 18-Jan-2013 14:29:04 GMT
Parent Server Generation: 0
Server uptime: 26 minutes 31 seconds
5 requests currently being processed, 59 idle workers

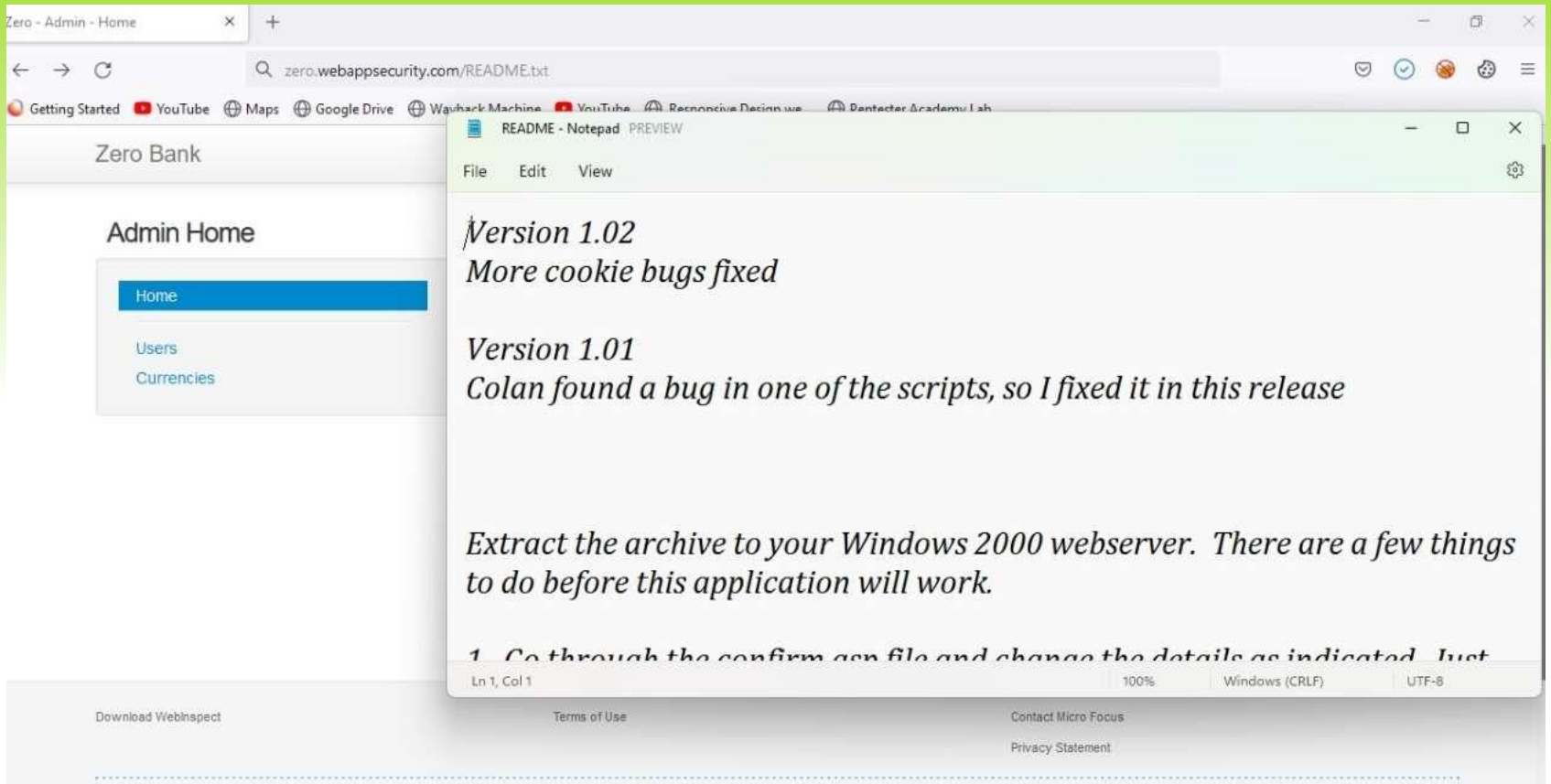
```
_____________________________________________________K_____KK_W__K__
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
.......................................................................
```

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "r" Reading Request,
"w" Sending Reply, "k" Keepalive (read), "d" DNS Lookup,
"c" Closing connection, "l" Logging, "g" Gracefully finishing,
"i" Idle cleanup of worker, "." Open slot with no current process

PID Key:

```
   11740 in state: _ ,   11740 in state: _ ,   11740 in state: _
   11740 in state: _ ,   11740 in state: _ ,   11740 in state: _
   11740 in state: _ ,   11740 in state: _ ,   11740 in state: _
   11740 in state: _ ,   11740 in state: _ ,   11740 in state: _
   11740 in state: _ ,   11740 in state: _ ,   11740 in state: _
   11740 in state: _ ,   11740 in state: _ ,   11740 in state: _
   11740 in state: _ ,   11740 in state: _ ,   11740 in state:
```

http://zero.webappsecurity.com/README.txt



readme.txt to access the Readme File

Common Attacks Using Kali Linux:-

```
┌──(root💀 kali)-[/home/kali]
└─# nslookup zero.webappsecurity.com
Server:         192.168.149.2
Address:        192.168.149.2#53

Non-authoritative answer:
Name:   zero.webappsecurity.com
Address: 54.82.22.214

┌──(root💀 kali)-[/home/kali]
└─# nmap -sS -sV 54.82.22.214

Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-11 07:16 EST
Nmap scan report for ec2-54-82-22-214.compute-1.amazonaws.com (54.82.22.214)
Host is up (0.029s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE  VERSION
80/tcp   open  http     Apache Tomcat/Coyote JSP engine 1.1
443/tcp  open  ssl/http Apache httpd 2.2.6 ((Win32) mod_ssl/2.2.6 OpenSSL/0.9.8e mod_jk/1.2.40)
8080/tcp open  http     Apache Tomcat/Coyote JSP engine 1.1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 109.20 seconds
```

Getting the open port information using nmap scanning

```
msf6 auxiliary(gather/impersonate_ssl) > set RHOST 54.82.22.214
RHOST => 54.82.22.214
msf6 auxiliary(gather/impersonate_ssl) > run
[*] Running module against 54.82.22.214

[*] 54.82.22.214:443 - Connecting to 54.82.22.214:443
[*] 54.82.22.214:443 - Copying certificate from 54.82.22.214:443
/C=US/ST=California/L=Santa Clara/O=Micro Focus LLC/CN=zero.webappsecurity.com
[*] 54.82.22.214:443 - Beginning export of certificate files
[*] 54.82.22.214:443 - Creating looted key/crt/pem files for 54.82.22.214:443
[+] 54.82.22.214:443 - key: /root/.msf4/loot/20220211062611_default_54.82.22.214_54.82.22.214_key_975104.key
[+] 54.82.22.214:443 - crt: /root/.msf4/loot/20220211062611_default_54.82.22.214_54.82.22.214_cer_663801.crt
[+] 54.82.22.214:443 - pem: /root/.msf4/loot/20220211062611_default_54.82.22.214_54.82.22.214_pem_570778.pem
[*] Auxiliary module execution completed
msf6 auxiliary(gather/impersonate_ssl) >
```

Using Metasploit Framework To Perform HTTP Trace Vulnerability Using TCP Port 443

Basic Attacks

1. Redirection And Forwards: There Are No Redirections To Different Pages And Forwards On zero.webappsecurity.com. Hence Is Not Vulnerable To Redirections And Forwards

2. HTML Injection:

Input Fields On zero.webappsecurity.com Do Not Accept HTML Code Hence Is Not Vulnerable To HTML Injections

3. SQL Injection:

Input Fields On zero.webappsecurity.com Do Not Accept SQL Code Hence Is Not Vulnerable To SQL Injections (Tested On BurpSuite)

4. XSS Cross Scripting (Stored/Reflected):

Input Fields On zero.webappsecurity.com Do Not Accept <Script>....</Script> Code Hence Is Not Vulnerable To XSS Cross Scripting