



Command injection

[What is it?](#)

[Attack strategy](#)

[Separators](#)

[Commands](#)

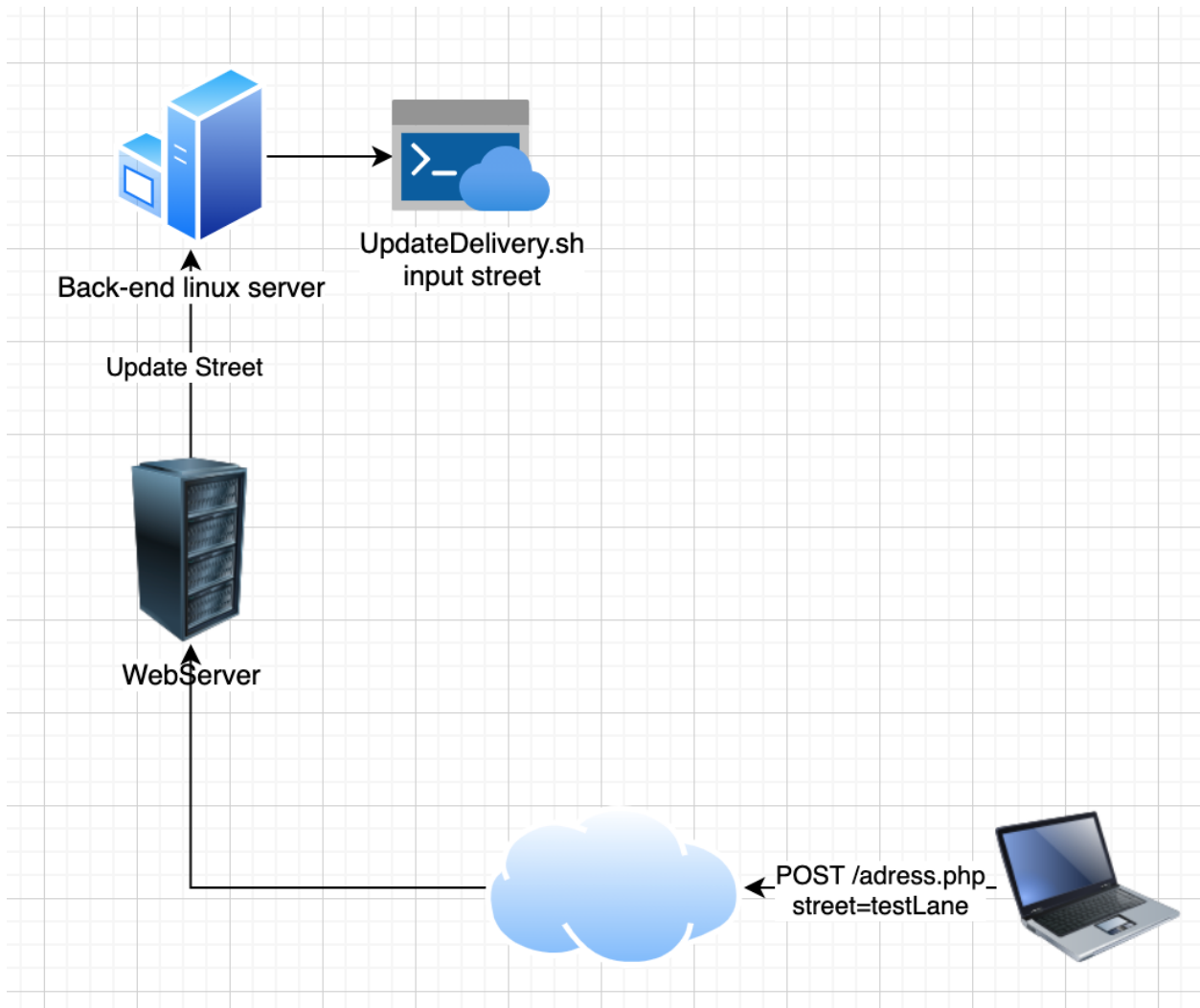
[fuzzing list](#)

[Blind command injection](#)

What is it?

Command injection happens when we can control a parameter that gets passed into a shell. If that input is not handled safely and sanitised properly, we can insert a command into our input and have that executed by the shell. Depending on the capabilities and priviledges of that shell, we can execute various commands.

The theory sounds very simple however it's not simple at all to find this kind of vulnerability.



Attack strategy

The reason command injection is so hard to find is because we never really know which of our processes will trigger a back-end shell to execute. This means we will need to fuzz every parameter we find but you might be wondering what characters to fuzz with. To determine this, we first need to talk about which command separators can possibly be used and also which commands.

Separators

The following command separators work on both Windows and Unix-based systems:

- `&`

- `&&`
- `|`
- `||`

The following command separators work only on Unix-based systems:

- `;`
- Newline (`0x0a` or `\n`)

On Unix-based systems, you can also use backticks or the dollar character to perform inline execution of an injected command within the original command:

- `` injected command ``
- `$(injected command)`

Commands

Below is a summary of some commands that are useful on Linux and Windows platforms:

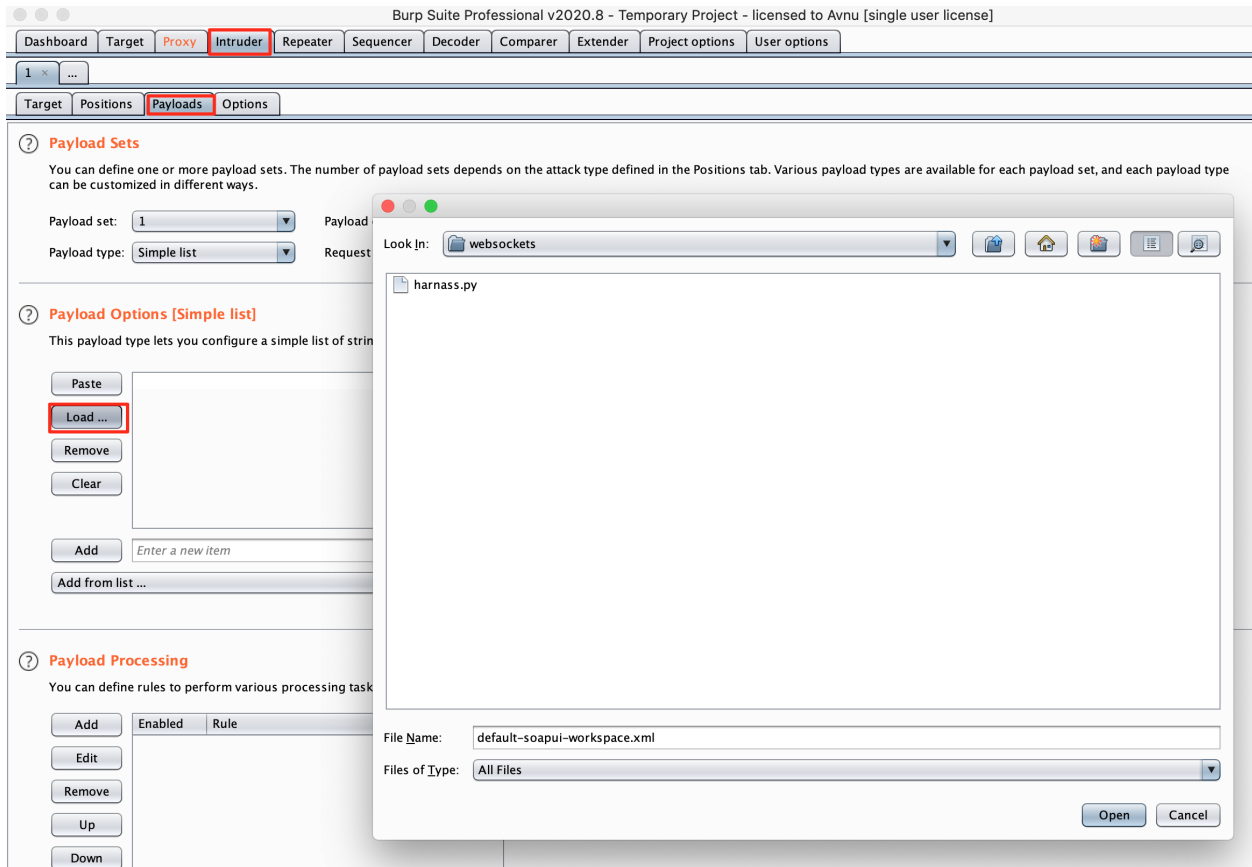
Linux and windows commands

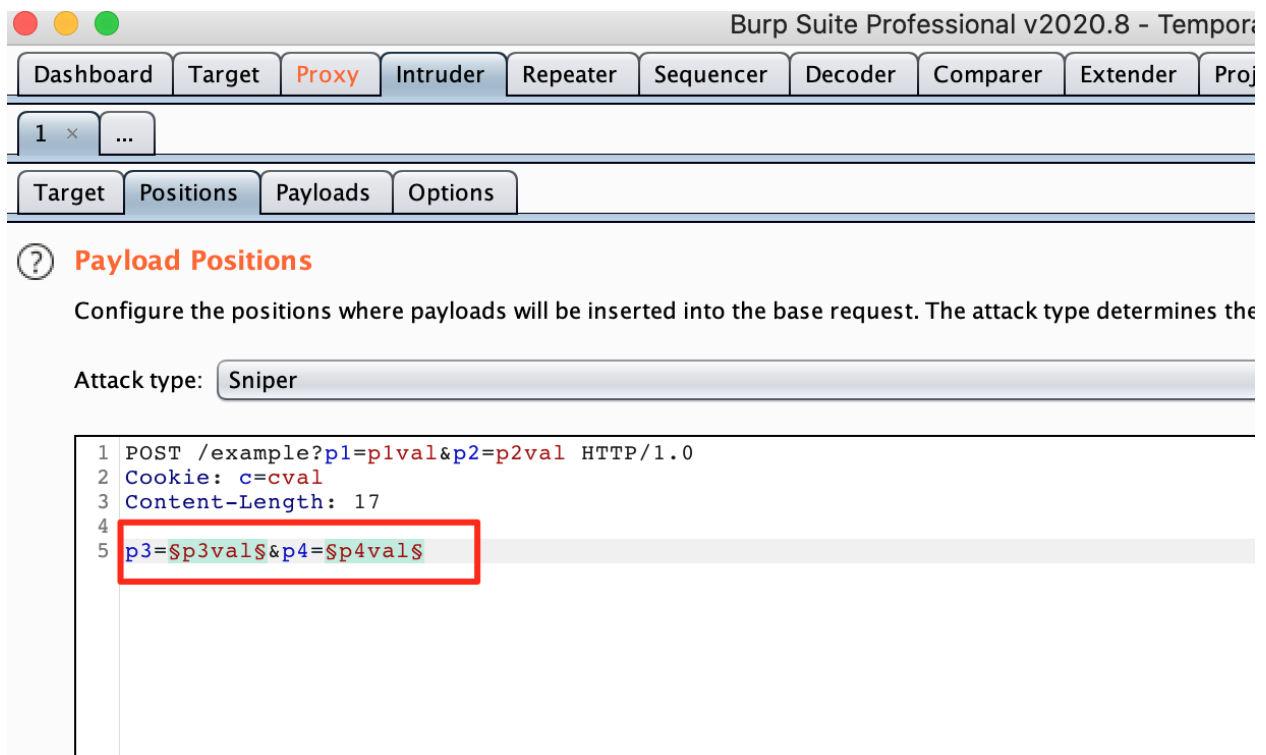
<u>Purpose of command</u>	<u>Linux</u>	<u>Windows</u>
<u>Name of current user</u>	<code>whoami</code>	<code>whoami</code>
<u>Operating system</u>	<code>uname -a</code>	<code>ver</code>
<u>Network configuration</u>	<code>ifconfig</code>	<code>ipconfig /all</code>
<u>Network connections</u>	<code>netstat -an</code>	<code>netstat -an</code>
<u>Running processes</u>	<code>ps -ef</code>	<code>tasklist</code>

fuzzing list

Based on this, we can create a fuzzing list that contains all the separators together with all the possible commands. I will leave this up to you as it should be a good exercise. Feel free to contact me on discord if you have issues with this however.

Fuzz every single parameter you can find with this worldlist by using burp intruder.





As you can see in the screenshot, we load in our fuzzing list and mark the parameters we want to test.

Blind command injection

We can test for blind command injection by launching a request that will execute a ping command to the loopback address.

```
& ping -c 10 127.0.0.1 &
```

Again, add this to your fuzzing list and mind the response time for this attack vector. If it's longer than 10 seconds, we probably have a blind command injection but be mindful as lag might occur and give a false positive.