Test HTTP Methods (HTTP Yöntemlerini Test Etme)

Summary (Özet)

HTTP, web sunucusunda eylemleri gerçekleştirmek için kullanılabilecek bir dizi yöntem sunar (HTP 1.1 standardı bunları ifade eder gibi ifade eder `methods` ama aynı zamanda yaygın olarak da tanımlanırlar. `verbs`). GET ve POST, bir web sunucusu tarafından sağlanan bilgilere erişmek için kullanılan en yaygın yöntemler olsa da, HTTP diğer (ve biraz daha az bilinen) yöntemlere izin verir. Bunlardan bazıları, web sunucusu yanlış yapılandırılırsa, kötü amaçlar için kullanılabilir.

RFC 7231 – Hipermet Aktarım Protokolü (HTTP/1.1): Semantik ve İçerik aşağıdaki geçerli HTTP istek yöntemlerini veya fiilleri tanımlar:

- `GET`
- `HEAD`
- `POST`
- `PUT`
- `DELETE`
- `CONNECT`
- `OPTIONS`
- `TRACE`

Bununla birlikte, çoğu web uygulamasının yalnızca GET ve POST isteklerine yanıt vermesi, URL sorgu dizisinde kullanıcı verilerini alması veya sırasıyla talebe eklenmesi gerekir. Standart `<a href=""></a>` Bir yöntem olmadan tanımlanan formlar bir GET talebini tetikler; aracılığıyla gönderilen veri `<form method='POST'></form>` POST taleplerini tetikleyin. JavaScript ve AJAX aramaları GET ve POST dışındaki yöntemler gönderebilir, ancak genellikle bunu yapmanız gerekmez. Diğer yöntemler çok nadiren kullanıldığından, birçok geliştirici web

sunucusunun veya uygulama çerçevesinin bu yöntemlerin uygulanmasının uygulamanın güvenlik özelliklerini nasıl etkilediğini bilmiyor veya dikkate almıyor.

Test Objectives (Test Hedefleri)

- Enumerate destekli HTTP yöntemleri.
- Erişim kontrolü bypass için test.
- XST güvenlik açıklarını test edin.
- HTTP yöntemini geçersiz kılma tekniklerini test edin.

How to Test (Nasıl Test Edilir)

Discover the Supported Methods (Desteklenen Yöntemleri Keşfedin)

Bu testi yapmak için testçinin, incelenmekte olan web sunucusu tarafından hangi HTTP yöntemlerinin desteklendiğini anlamanın bir yoluna ihtiyacı vardır. Oysa o sırada `OPTIONS` HTTP
yöntemi bunu yapmanın doğrudan bir yolunu sağlar, farklı yöntemler kullanarak istekler yayınlayarak sunucunun yanıtını doğrular. Bu, manuel test veya benzeri bir şey ile başarılabilir.
`http-methods` Nmap senaryosu.

Kullanmak için `http-methods` Nimap senaryosu bitiş noktasını test etmek için `/index.php` Sunucuda `localhost` HTTPS kullanarak komutu yayınlayın:

```
nmap -p 443 --script http-methods --script-args http-methods.url-path='/index.php' localhost
```

Diğer yöntemleri kabul etmek zorunda olan bir uygulamayı test ederken, örneğin bir RESforma Web Hizmeti, tüm uç noktaların yalnızca ihtiyaç duydukları yöntemleri kabul ettiğinden emin olmak için iyice test edin.

Testing the PUT Method (PUT Yöntemini Test Edin)

1. Hedefin temel talebini bir web proxy ile yakalayın.
2. Talep yöntemini değiştirmek için değiştirin `PUT` ve ekleyin `test.html` Dosyalayın ve talebi uygulama sunucusuna gönderin.

```
PUT /test.html HTTP/1.1
Host: testing-website
```

```
<html>
HTTP PUT Method is Enabled
</html>
```

3. 2XX başarı kodları veya 3XX yönlendirmeleri ile sunucu yanıtı ve ardından onaylanırsa `GET` talep etmek `test.html` Dosya. Uygulama savunmasız.

HTTP ise `PUT` Temel URL veya istek üzerine yönteme izin verilmez, sistemdeki diğer yolları deneyin.

> NOT: Bir web kabuğu yüklemede başarılıysanız, bunun üzerine yazmalı veya hedefin güvenlik ekibinin farkında olduğundan emin olmalısınız ve konsept kanıtınızdan hemen sonra bileşeni derhal kaldırın.

Kaldıraçlar `PUT` Bir saldırganın, uzaktan kod yürütülmesine, siteyi tahrip etmesine veya
hizmet reddine yol açabilecek sisteme keyfi ve potansiyel olarak kötü amaçlı içerik koyabileceği yöntem.

Testing for Access Control Bypass (Erişim Kontrolü Bypass Testi)

Bir GET talebinin normalde 302 yönlendirmeyi sayfadaki bir kütüğe zorlaması veya doğrudan bir kütüğü zorlaması için güvenlik kısıtlaması olan bir ziyaret için bir sayfa bulun. HEAD, POST, PUT vb. gibi çeşitli yöntemlerin yanı sıra BILBAO, FOOBAR, CATS vb. gibi keyfi olarak oluşturulmuş yöntemleri kullanarak sorun taleplerinin. Web uygulaması bir ile yanıt verirse `HTTP/1.1 200 OK` Bu
 bir sayfadaki bir gün değil, kimlik doğrulamayı veya yetkilendirmeyi atlamak mümkün olabilir. Aşağıdaki örnek Nmap'ınkini kullanır
_`ncat`. .

```
$ ncat www.example.com 80
HEAD /admin HTTP/1.1
Host: www.example.com
```

```
HTTP/1.1 200 OK
Date: Mon, 18 Aug 2008 22:44:11 GMT
Server: Apache
Set-Cookie: PHPSESSID=pKi...; path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-chec
k=0
Pragma: no-cache
Set-Cookie: adminOnlyCookie1=...; expires=Tue, 18-Aug-2009 22:44:31 GMT;
domain=www.example.com
Set-Cookie: adminOnlyCookie2=...; expires=Mon, 18-Aug-2008 22:54:31 GM
T; domain=www.example.com
Set-Cookie: adminOnlyCookie3=...; expires=Sun, 19-Aug-2007 22:44:30 GM
T; domain=www.example.com
Content-Language: EN
Connection: close
Content-Type: text/html; charset=ISO-8859-1
```

Sistem savunmasız görünüyorsa, konuyu daha tam olarak istismar etmek için aşağıdakiler gibi CSRF benzeri saldırılar düzenleyin:

- `HEAD /admin/createUser.php?member=myAdmin`
- `PUT /admin/changePw.php?member=myAdmin&passwd=foo123&confirm=foo123`
- `CATS /admin/groupEdit.php?group=Admins&member=myAdmin&action=add`

Yukarıdaki üç komutu kullanarak, test ve test gereksinimleri kapsamında uygulamaya uyacak şekilde değiştirilen yeni bir kullanıcı oluşturulacak, bir şifre atanacak ve kullanıcı, hepsi kör istek gönderimini kullanarak bir yönetici yaptı.

Testing for Cross-Site Tracing Potential (Sitelerarası İzleme Potansiyeli için Test)

Not: Bir çapraz izleme (XST) saldırısının mantığını ve hedeflerini anlamak için, siteler arası komut dosyası saldırılarına aşina olmalıdır.

The (İngilizce) `TRACE` Test ve hata ayıklama için tasarlanan yöntem, web sunucusuna alınan mesajı müşteriye geri yansıtması için talimat verir. Bu yöntem,

görünüşte zararsız olsa da, meşru kullanıcıların kimlik bilgilerini çalmak için bazı senaryolarda başarılı bir şekilde kullanılabilir. Bu saldırı tekniği, 2003 yılında Jeremiah Grossman tarafından, çerezleri JavaScript tarafından erişilmekten korumayı amaçlayan HttpOnly özelliğini atlamak amacıyla keşfedildi. Bununla birlikte, TRACE yöntemi bu korumayı atlamak ve bu özellik ayarlandığında bile çereze erişmek için kullanılabilir.

Aşağıdaki gibi bir talepte bulunarak siteler arası izleme potansiyeli için test:

```
$ ncat www.victim.com 80
TRACE / HTTP/1.1
Host: www.victim.com
Random: Header

HTTP/1.1 200 OK
Random: Header
...
```

Web sunucusu 200'ü iade etti ve yerine yerleştirilen rastgele başlığı yansıttı. Bu konuyu daha da kullanmak için:

```
$ ncat www.victim.com 80
TRACE / HTTP/1.1
Host: www.victim.com
Attack: <script>prompt()</script>
```

Yukarıdaki örnek, yanıt HTML bağlamına yansıtılıyorsa çalışır.

Eski tarayıcılarda, sunucu onları yansıttığında başlıkları sızdıran XHR teknolojisi kullanılarak saldırılar yapıldı *(*örneğin. Çerezler, Yetkilendirme belirteçleri vb.) ve HttpOnly özelliği gibi bypassed güvenlik önlemleri. Bu saldırı, son tarayıcılarda ancak uygulama Flash'a benzer teknolojilerle bütünleşirse çekilebilir.

Testing for HTTP Method Overriding (HTTP Yöntemi Gezginlik Testi)

Bazı web çerçeveleri, isteklerde bazı özel başlıkları geçen kayıp HTTP fiillerini taklit ederek talepteki gerçek HTTP yöntemini geçersiz kılmanın bir yolunu sağlar.

Bunun temel amacı, izin verilen yöntemlerin genellikle olduğu gibi fiilleri kapsamadığı bazı ara yazılım (örneğin vekalete, güvenlik duvarı) sınırlamalarını önlemektir. `PUT` ya da `DELETE` . . Aşağıdaki alternatif başlıklar bu tür fiil tünelleme yapmak için kullanılabilir:

- `X-HTTP-Method`
- `X-HTTP-Method-Override`
- `X-Method-Override`

Bunu test etmek için, PUT veya DELETE gibi kısıtlı fiillerin "405 Yönteme izin verilmemesi" gibi bir "405 Yönteme izin verilmediği" senaryolarda, HTTP yönteminin geçersiz kılması için alternatif
başlıkların eklenmesiyle aynı isteği tekrar oynatın ve sistemin nasıl tepki verdiğini gözlemleyin. Uygulama, yöntem geçersiz kılmanın desteklendiği durumlarda farklı bir durum koduyla (
örneğin 200) yanıt vermelidir.

Aşağıdaki örnekteki web sunucusu izin vermez `DELETE` Yöntem ve engelleme:

```
$ ncat www.example.com 80
DELETE /resource.html HTTP/1.1
Host: www.example.com

HTTP/1.1 405 Method Not Allowed
Date: Sat, 04 Apr 2020 18:26:53 GMT
Server: Apache
Allow: GET,HEAD,POST,OPTIONS
Content-Length: 320
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
```

Bunu ekledikten sonra `X-HTTP-Header` Sunucu talebe 200 ile yanıt verir:

```
$ ncat www.example.com 80
DELETE /resource.html HTTP/1.1
Host: www.example.com
X-HTTP-Method: DELETE
```

```
HTTP/1.1 200 OK
Date: Sat, 04 Apr 2020 19:26:01 GMT
Server: Apache
```

Remediation (Düzeltme)

- Sadece gerekli başlıklara izin verildiğinden ve izin verilen başlıkların uygun şekilde yapılandırıldığından emin olun.
- Kullanıcı ajanları, çerçeveler veya web sunucuları tarafından uygulanan güvenlik önlemlerini atlamak için hiçbir geçici çözüm uygulanmamasını sağlayın.

Tools (Araçlar)

- Ncat
- cURL
- nmap http-methods NSE script
- w3af plugin htaccess_methods

References (Referanslar)

- RFC 2109 and RFC 2965: “HTTP State Management Mechanism”
- HTACCESS: BILBAO Method Exposed
- Amit Klein: “XS(T) attack variants which can, in some cases, eliminate the need for TRACE”
- Fortify - Misused HTTP Method Override
- CAPEC-107: Cross Site Tracing