Http Slow Saldırıları

Bu yazıda http slow saldırılarının arkaplanı ve uygulaması gösterilecektir:

a. Http Slow Saldırıları Arkaplanı
b. Http Slow Saldırıları Uygulama

slowhttptest kurulum işlemleri için bk. Yaz Tatili 2014 / slowhttptest Kurulumu.docx

a. Http Slow Saldırıları Arkaplanı

Gelen talebin tamamını beklemek web sunucularının doğası gereğidir. Bundan dolayı http talebini yavaş gönderme ya da yavaş alma sonucu web sunucuyu meşgul etme ve servis dışı bırakma saldırıları gerçekleşebilir.

Http Slow saldırıları üç türdür:

- Slow Headers Attack (Slowloris attack olarak da bilinir)
- Slow Message Body Attack (Slow Post attack olarak da bilinir)
- Slow Read (Slow Read olarak bilinir)

Read / Write (Okuma / Gönderme) arasındaki zaman aralığında veriyi küçük küçük parçalar halinde gönderme veya okumasını yapma bu saldırıların temelini oluşturur.

i) Slow Headers Attack

Slowloris olarak bilinen bu atak bilinen tüm Apache sürümlerinde uygulanabilmektedir. Bu saldırının temel çalışma prensibi http request'teki header'ların yavaş yavaş gönderilmesi sonucu sunucunun meşgulde bırakılmasıdır. Örneğin bu saldırı türünü gerçekleştirebilen bir araç (tool) aşağıdaki gibi bir http talebini hedef sunucuya yapar.

```
GET / HTTP/1.1CRLF
Host: localhost:80CRLF
User-Agent: Mozilla/4.0 (Windows NT 6.1; Trident/4.0; SLCC2)CRLF
```

Sunucu http talebi GET olması dolayısıyla paketin sonlandığını çift CRLF ile anlar. İstemci http talebini çift CR LF ile bitirmediğinde web sunucu paketin henüz tamamlanmadığını kabul eder ve paketin arta kalan parçalarını bekler. Tool sunucuya yaptığı bu yarım talebe eklemlenmek üzere sırayla ve yavaş yavaş header gönderiminde bulunur. .

```
.
. n seconds
.
X-HMzV2bwpzQw9jU9fGjZRknd7Sa54J: u6RrIoLRrte4QV92yojeewiuBL2N7CRLF
.
. n seconds
.
X-nq0HRGnv1W: T5dSLCRLF
.
. n seconds
```

.
X-iFrjuN: PdR7Jcj27PCRLF
.
.
.

Not: Tool http talebine koyulacak header isim ve değerlerini rasgele string'lerden oluşturur ve header isim ve değerlerinin max limit'lerini belirler.

Web sunucu paketin her gelen yeni satırını (parçasını) mevcut pakete alt alta ekler. Çift CR LF belirteçleri her defasında gönderilmediğinden sunucuda bağlantı açık tutulur ve sunucu meşgulde kalır. Tool tarafından bu bağlantı gibi aynı işlemin gerçekleştirileceği 100'lerce bağlantı açıldığında ise web sunucunun bağlantı havuzu tükenir ve sunucu başka istemcilere bağlantı sunamayacağı için servis dışı kalır.

ii) Slow Message Body Attack

Slow Header saldırında temel mantık http talebindeki header'ların yavaş yavaş ve belirli aralıklarla peyderpey gönderilmesinden ibaretti. Slow Message Body (diğer adıyla Slow Post) saldırı türünde ise temel mantık http talebindeki body kısmına koyulacak parametre ve değerlerinin yavaş yavaş ve belirli aralıklarla peyderpey gönderilmesinden oluşur. Örneğin Slow Message Body saldırısı yapabilen bir araç sunucuya şöyle bir http talebinde bulunabilir:

POST / HTTP/1.1CRLF
Host: 10.10.25.116:80CRLF
User-Agent: Mozilla/5.0 (Macintosh; Mac OS X 10.7;) Gecko/2101 Firefox/5.0.1CRLF
Content-Length: 8192CRLF
Connection: closeCRLF
Referer: http://code.google.com/p/slowhttptest/CRLF
Content-Type: application/x-www-form-urlencodedCRLF
Accept: text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5CRLF
CRLF
foo=bar
.
. n seconds
.
&rjP8=du7FKMe
.
. n seconds
.
&93zgIx=jgfpopJ
.
.
.

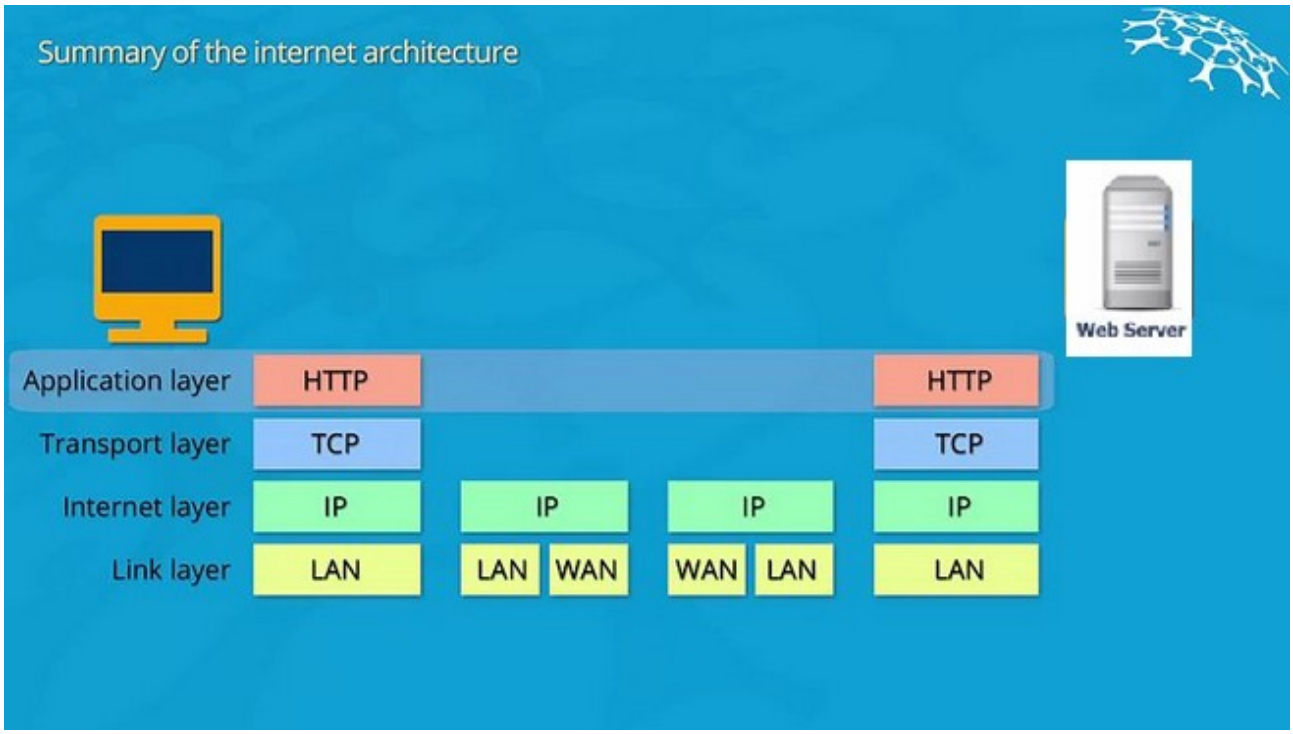
Not: Tool http talebinin body'sinde yer alacak parametre ve değerlerini rasgele string'lerden oluşturur ve parametre ismi ve değerinin max limit'lerini belirler.

Sunucu http talebi POST olması dolayısıyla paketin sonlandığını çift CRLF ile değil, http talebindeki Content-Length header'ının aldığı byte değeri ile anlar. Http talebinin body'sindeki boyut bu byte değerine ulaştığında sunucu paket tamamlandı der ve bağlantıyı kapatır. Dolayısıyla eğer http yanıtının body'sindeki parametre ve değerleri yavaş yavaş gönderilirse sunucuda bir bağlantı sürekli açık tutulmuş olur. Tool tarafından bu bağlantı gibi aynı işlemin gerçekleştirileceği 100'lerce bağlantı açılırsa web sunucunun bağlantı havuzu tükenir ve sunucu başka istemcilere bağlantı sunamayacağı için servis dışı kalır.

iii) Slow Read Attack

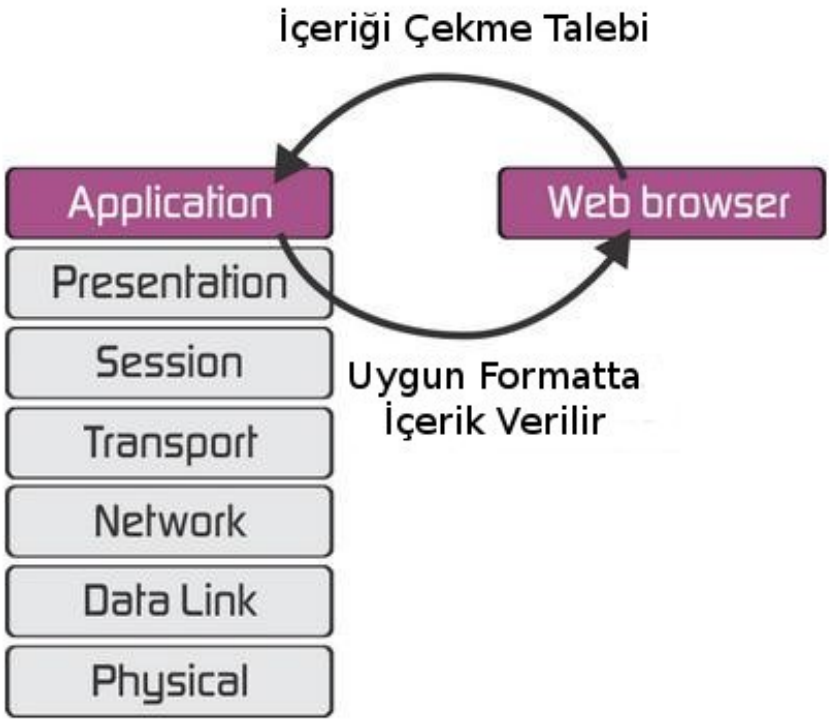
Slow http saldırıları arasında önlenmesi en güç saldırı türü olan Slow Read Attack gönderilen http talebi sonrası gelen yanıtın istemci tarafında yavaştan okunması suretiyle gerçekleşir. Bu şekilde hedef sunucuda olabildiğince çok bağlantı aktif bırakılmaya çalışılır. Bu saldırı türünde istemci http yanıt paketini normalde olması gerektiği gibi alır ve http yanıt paketi istemcinin kernel buffer'ına (çekirdek tamponuna) yerleşir. Ancak sunucudan http yanıt paketinin istemciye transferi ilk aşamada network layer'da gerçekleşir. Dolayısıyla kernel buffer'daki http yanıt paketinin uygulama katmanına (web tarayıcıya) iletilmesi sonraki aşamadır. İşte slow message body attack bu yapıdan faydalanır ve Application Layer'daki uygulama kernel buffer'daki http yanıt paketini yavaş yavaş okuyarak paketin tamamını okumayı geciktirir. Böylelikle uygulama katmanında paket tamamen okunana kadar hedef web sunucu beklemede bırakılmış olur. Bu işlem web sunucuda bir bağlantının açık bırakılmasını sağlar. Tool tarafından bu bağlantı gibi aynı işlemin (yavaş okuma işleminin) gerçekleştirileceği 100'lerce bağlantı açılırsa web sunucunun bağlantı havuzu tükenecektir. Bunu sonucunda sunucu başka istemcilere bağlantı sunamayacağı için yine servis dışı kalacaktır.

Aşağıda istemci ve Web sunucu arasındaki http talep ve yanıt paketlerinin transfer koridorunu görüntülemektesiniz:



Görüldüğü üzere Network Layer (Data-Link Layer) da veri transferi gerçekleşiyor. Gelen bu veri

uygulama katmanına ait olduğu için daha sonra Application Layer'a iletilecektir. Aşağıda ise Application Layer'daki uygulamanın alt katmandan gelen veriyi çektiğini görüntülemektesiniz:



b. Uygulama

(+) Bu başlık birebir denenmiştir ve başarıyla uygulanmıştır.

slowhttptest tool'u slow tekniğini kullanarak web sunucularına dos saldırıları yapmaya yaran bir araçtır.

Aşağıdaki dört slow saldırı türünü gerçekleştirir:

- Slow Headers (Slowloris olarak da bilinir)
- Slow Message Body (Slow Http Post olarak da bilinir)
- Slow Read (Slow Read olarak bilinir)

i) Slow Header Attack (Slowloris Attack) - [Başarılı Olundu]

Aşağıdaki komut ile hedef web sunucuya saniyede 200 tane olmak üzere toplam 1000 adet bağlantı isteği gönderilir. Açılan bu bağlantılardan gönderilen http taleplerine 10 saniye aralıklarla sürekli http header eklemlenmek üzere gönderilir. Hedef web sunucu 3 saniye boyunca http yanıt gönderemediğinde ise servis dışı kalmıştır denir.

Saldıran Sistem Ubuntu 14.04 LTS Ana Makinası
Hedef Sistem DVWA - WebGoat (ubuntu 14.04) Sanal Makinası (**Apache**)

> ./slowhttptest -H -c 1000 -r 200 -i 10 -x 24 -p 3 -t GET -u http://172.16.3.122 -g -o slow_header_stats

Çıktı:
…
service available: NO

-H : slowhttptest'in slow header attack parametresidir. Http header'larının yavaş yavaş ve sırayla yollandığı bitmemiş http talepleri yapılmasını sağlar.
-c : Test sırasında kullanılmak üzere bağlantı sayısını belirler. (connection)
-r : Birim saniyede kurulacak bağlantı sayısını belirler. (rate)
-i : Birbirini takip eden veriler arasındaki gönderim zaman aralığını belirler. (interval)

-x : Birbirini takip eden verilerden her biri için geçerli maksimum boyutu belirler.
-p : Http yanıtını bekleme süresini belirtir. Http yanıtın gelmesi beklenirken bu süre aşıldığında DoS gerçekleşti denir. (probe)
-t : Http talep ismi alır. Örn; GET,HEAD,POST, FAKEVERB,...
-u : Hedef web uygulamasının adresi girilir.
-g : Test bittiğinde csv ve html formlarında iki belge hazırlar.
-o : Test bittiğinde -g ile oluşan csv ve html formlarının ismini belirler.

ii) Slow Message Body Attack (Slow Http Post Attack) - [Başarılı Olundu]

Aşağıdaki komut ile hedef web sunucuya saniyede 200 tane olmak üzere 3000 adet bağlantı isteği gönderilir. Açılan bu bağlantılardan gönderilen http taleplerinin body'sine 110 saniye aralıklarla sürekli POST parametre ve değeri eklenmek üzere gönderilir. Content-Length değeri olarak 8192 byte belirtildiği için body'deki POST parametre ve değerleri toplam boyutu 8192 byte olana kadar bağlantı açık kalır (not: parametre ve değerlerinin her birinin maksimum karakter sayısı 10 byte olarak belirlenir). Hedef web sunucu tüm bu işlemler olurken 3 saniye boyunca http yanıt gönderemez ise servis dışı kalmıştır denir.

Saldıran Sistem Ubuntu 14.04 LTS Ana Makinası
Hedef Sistem Windows Server 2008 Standard (R2 Öncesi) (**IIS 7.0**)

> ./slowhttptest -B -c 1000 -r 200 -i 110 -s 8192 -x 10 -p 3 -t POST -u http://172.16.3.128/deneme.php -g -o slow_body_stats

Çıktı:
…
service available: NO

-B : slowhttptest'in slow message body attack parametresidir. Http taleplerinin body'sinde yer alan parametre ve değerlerinin sırayla ve yavaş yavaş yollandığı bitmemiş http talepleri yapılmasını sağlar.
-c : Test sırasında kullanılmak üzere bağlantı sayısını belirler. (connection)
-r : Birim saniyede kurulacak bağlantı sayısını belirler. (rate)

-i : Birbirini takip eden veriler arasındaki gönderim zaman aralığını belirler. (interval)
-s : Http taleplerindeki Content-Type header'ının değerini belirler. Böylece http talebinin body'sinde gidecek verinin toplam büyüklüğü belirlenir.
-x : Birbirini takip eden verilerden her biri için geçerli maksimum boyutu belirtir. Böylece http talebinin body'sine koyulacak parametre ve değerlerinin her birinin maksimum uzunluk limiti belirlenir.
-p : Http yanıtını bekleme süresini belirtir. Http yanıtın gelmesi beklenirken bu süre aşıldığında DoS gerçekleşti denir. (probe)
-t : Http talep ismi alır. Örn; GET,HEAD,POST, FAKEVERB,...
-u : Hedef web uygulamasının adresi girilir.
-g : Test bittiğinde csv ve html formlarında iki belge hazırlar.
-o : Test bittiğinde -g ile oluşan csv ve html formlarının ismini belirtir.

Not:

Slow Message Body Attack post işlemini handle edebilen bir sayfaya ancak yapılabilir. Dolayısıyla bu saldırıda vulnerable url'yi bulmak şarttır (Yani post işlemini handle eden bir sayfayı bulmak şarttır). Windows Server 2008 Standard makinasında deneme.php dosyası bu şekilde oluşturulmuştur. Örn;

deneme.php

```
<?php
	header("Content-Type: text/html; charset=UTF-8");

	if (isset($_POST["name"])) {
		echo "username：" . $_POST["name"] . "";
	}

	if (isset($_POST["psw"])) {
		echo "password：" . $_POST["psw"] . "";
	}
?>
```

Bu şekilde http://172.16.3.128/deneme.php url'sine yapılacak Slow Message Body Attack ile hedef IIS 7.0 sunucusu servis dışı kalacaktır.

Burada bir nokta vardır: Post handle eden sayfadaki $_POST ile çekilen değişkenlerin isimlerinin ne olduğunun bir önemi yoktur. Sadece POST taleplerine açık bir sayfa bulmamız yeterlidir. Böyle bir sayfaya yapılacak Slow Message Body saldırısı ile IIS 7.0 servis dışı kalacaktır.

Eğer POST taleplerine açık bir sayfa değil de örneğin GET taleplerine açık bir sayfaya Slow Message Body saldırısı yapılırsa sunucu zafiyete sahip olsa bile DoS gerçekleşmeyecektir. Bu durum Windows Server 2008 Standard sanal makinasındaki IIS 7.0 sunucuya yaptığım ilk Slow Message Body DoS saldırılarında (default IIS 7 welcome sayfasına yaptığım saldırılarda) başıma gelmiştir. Ardından https://github.com/shekyan/slowhttptest/issues/49 linkinden edindiğim bilgi ile POST handle eden bir sayfa olması gerektiğini öğrenip IIS 7.0 sunucusuna PHP kurdum ve POST handle eden deneme.php isimli bir sayfa oluşturdum. Son olarak bu sayfaya yaptığım Slow Message Body saldırısı sonucunda ise başarıya ulaştım.

Not 2:

Yukarıdaki uygulama sonrası tecrübe edildiği üzere saldırısı sırasında Windows Server 2008 (R2 Öncesi) makinesinde servis dışı kalma etkisinin gerçekleştiği görülmüştür. Ancak bu etki sadece betik dili (php, asp,...) uzantılı dosyalara (web sayfalarına) erişimde görülmüştür. Diğer uzantılı (txt, html,…) dosyalara (web sayfalarına) erişimin o sıralarda halen devam edebildiği görülmüştür. Anlaşıldığı üzere yaptığımız bu saldırı IIS 7 sunucusunun sadece PHP modülünü etkilemiştir. IIS 7 'ye sonradan eklenen bu PHP modülü servis dışı kalma saldırısı ile düşse bile IIS web sunucu yazılımı kendine default olarak yer alan txt, html,… gibi uzantılara sahip dosyaları istemciye döndürebilme servisine sahip olduğundan ayakta kalabilmiştir. Burada etkilenen sadece betik dili uzantılı web sayfalarına erişim olmuştur. Diğer uzantılı dosyalar saldırıdan etkilenmemiştir. Örneğin saldırı sırasında deneme.php'ye ya da ekstradan test amaçlı oluşturulmuş başka bir php dosyasına (abc.php'ye) erişimin engelli olduğu görülmüşken o sıralarda default.html sayfasına ya da yine test amaçlı oluşturulmuş başka bir dosyaya (xyz.txt dosyasına) erişimin sürekli olarak devam ettiği görülmüştür.

iii) Slow Read Attack - **[Hedef Sistem Bu Zafiyete Sahip Olmadığından Başarılamadı]**

Aşağıdaki komut ile hedef web sunucuya saniyede 200 tane olmak üzere 8000 bağlantı isteği gönderilir. TCP window boyutu olarak 512 ile 1024 arası rasgele değerler seçilir. Açılan bağlantıların her biri için 5 saniyede bir 32 byte veri okuması yapılır. Tüm bu işlemler olurken eğer web sunucudan gelen http yanıtları 3 saniye boyunca gelmezse hedef web sunucu servis dışı kalmıştır denir.

./slowhttptest -X -c 8000 -r 200 -n 5 -w 512 -y 1024 -z 32 -p 3 -l 350 -u http://172.16.3.122 -g -o slow_read_stats

-X	: slowhttptest'in slow read attack parametresidir. Http yanıtının yavaş yavaş okunmasını sağlar.
-c	: Test sırasında kullanılmak üzere bağlantı sayısını belirler. (connection)
-n	: Http yanıt paketinin yavaş yavaş ve parça parça okunma zaman aralığını saniye cinsinden belirler.
-w	: TCP window aralığının başlangıcını belirler.
-y	: TCP window aralığının bitişini belirler.
-z	: Her bir read() işleminde kernel buffer'dan alınacak byte'ların sayısını belirler.
-p	: Http yanıtını bekleme süresini belirtir. Http yanıtın gelmesi beklenirken bu süre aşıldığında DoS gerçekleşti denir. (probe)
-l	: Test süresini belirtir.
-u	: Hedef web uygulamasının adresi girilir.
-g	: Test bittiğinde csv ve html formlarında iki belge hazırlar.
-o	: Test bittiğinde -g ile oluşan csv ve html formlarının ismini belirtir.

Kaynaklar

https://blog.qualys.com/tag/slow-http-attack

https://www.systutorials.com/docs/linux/man/1-slowhttptest/

https://github.com/shekyan/slowhttptest/issues/49

https://www.slideshare.net/jseidl/latinoware-2013-supereffectivedosattacks

https://github.com/valyala/goloris

https://github.com/shekyan/slowhttptest/wiki/InstallationAndUsage