radware

ERT Threat Alert

Inside the Hackers Arsenal: HTTP Flood Tools

December 7, 2016

Abstract

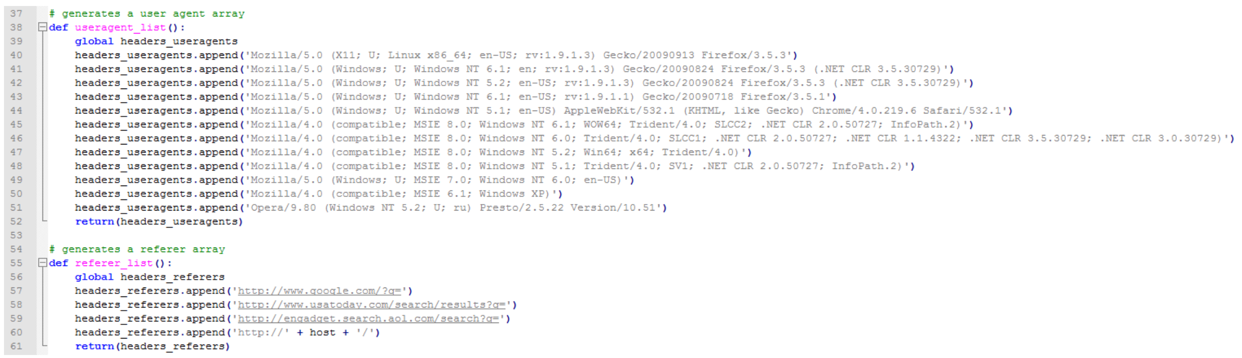
While record-breaking volumetric DDoS attacks flash in the headlines, low profile Denial-of-Service attacks continue to hit business worldwide. Radware's Emergency Response Team has battled over 385,000 HTTP floods launched across the world. HTTP flood is one of the hackers' favorites when it comes to causing a business disruption, as they are much harder to block (compared to a typical Layer 3 and Layer 4 network attack) and are prevalent - dozens of HTTP flood tools are available to the community and are constantly being improved. The majority of those tools leverage botnets for rent (DDoSaaS or Stresser services) that include HTTP flood attacks as part of their offering.

Attack Methods

An HTTP flood is an attack method used by hackers to attack web servers and applications. These floods consist of seemingly legitimate session-based sets of HTTP GET or POST requests sent to a targeted web server. HTTP floods do not use spoofing, reflective techniques or malformed packets. These requests are specifically designed to consume a significant amount of the server's resources, and therefore can result in a denial-of-service. Such requests are often sent en masse by means of a botnet, increasing the attack's overall power. HTTP and HTTPS flood attacks are one of the most advanced threats facing web servers today since it's very hard for network security devices to distinguish between legitimate HTTP traffic and malicious HTTP traffic.

HULK

HULK stands for HTTP-Unbearable-Load-King. It's an HTTP denial-of-service tool that has been around for several years. This tool, written in Python, is able to generate unique HTTP requests that are designed to stress test web servers against resource exhaustion (see Figure 1). HULK generates unique requests based on the user agent/referrer strings. By requesting the HTTP server for no cache, the server presents a unique page for each request. The tool bypasses caching engines and hits the server directly. Since its release, several actors have optimized and republished this code to include more user agents and referrers in an attempt to generate more randomized requests while launching an attack.



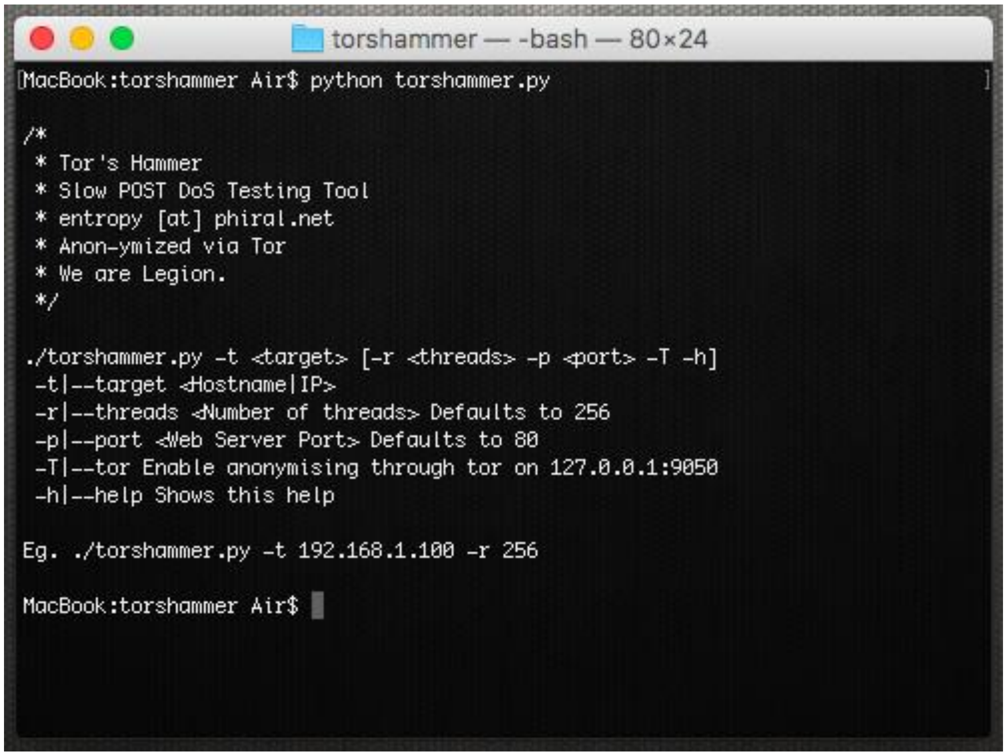
```python
# generates a user agent array
def useragent_list():
    global headers_useragents
    headers_useragents.append('Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.3) Gecko/20090913 Firefox/3.5.3')
    headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 6.1; en; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)')
    headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)')
    headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.1) Gecko/20090718 Firefox/3.5.1')
    headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/532.1 (KHTML, like Gecko) Chrome/4.0.219.6 Safari/532.1')
    headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; InfoPath.2)')
    headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR 3.5.30729; .NET CLR 3.0.30729)')
    headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.2; Win64; x64; Trident/4.0)')
    headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; SV1; .NET CLR 2.0.50727; InfoPath.2)')
    headers_useragents.append('Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 6.0; en-US)')
    headers_useragents.append('Mozilla/4.0 (compatible; MSIE 6.1; Windows XP)')
    headers_useragents.append('Opera/9.80 (Windows NT 5.2; U; ru) Presto/2.5.22 Version/10.51')
    return(headers_useragents)

# generates a referer array
def referer_list():
    global headers_referers
    headers_referers.append('http://www.google.com/?q=')
    headers_referers.append('http://www.usatoday.com/search/results?q=')
    headers_referers.append('http://engadget.search.aol.com/search?q=')
    headers_referers.append('http://' + host + '/')
    return(headers_referers)
```

Figure 1: Original HULK code

TorHammer

Tor's Hammer is a slow-rate HTTP POST (Layer 7) DoS tool. This tool executes a DoS attack by using a classic slow POST attack where HTML POST fields are transmitted in slow rates under the same session. Similar to the former R.U.D.Y. (R-U-Dead-Yet) tool, the slow POST attack causes the web server application threads to await the end of boundless posts in order to process them. This causes the exhaustion of the web server resources and causes it to enter a denial-of-service state for any legitimate traffic.

DoS attacks via TorsHammer can be carried out via the Tor Network by using a native socks proxy integrated in Tor clients. This enables launching the attack from random source IP addresses, which makes tracking the attacker almost impossible (see Figure 2).

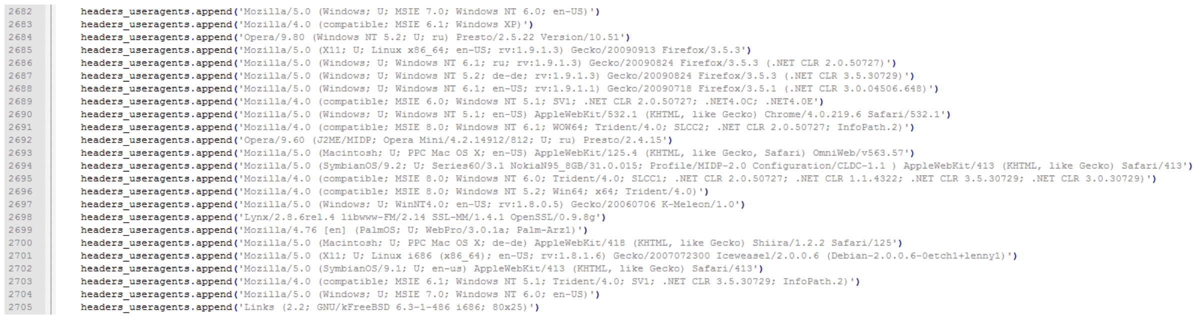


```
MacBook:torshammer Air$ python torshammer.py

/*
 * Tor's Hammer
 * Slow POST DoS Testing Tool
 * entropy [at] phiral.net
 * Anon-ymized via Tor
 * We are Legion.
 */

./torshammer.py -t <target> [-r <threads> -p <port> -T -h]
 -t|--target <Hostname|IP>
 -r|--threads <Number of threads> Defaults to 256
 -p|--port <Web Server Port> Defaults to 80
 -T|--tor Enable anonymising through tor on 127.0.0.1:9050
 -h|--help Shows this help

Eg. ./torshammer.py -t 192.168.1.100 -r 256

MacBook:torshammer Air$
```

Figure 2: TorsHammer interface

Saphyra

Saphyra is a HULK variant that has been used in several Anonymous operations throughout 2016. It includes over 3200 unique user agent strings and over 300 unique referrer field strings resulting in the possibility of over one million unique requests generated during an attack (see Figure 3). This combination of user agents and referrer instances makes it very difficult for most DDoS protection solutions (in particular rate-limiting technologies) to identify patterns in the attack traffic.

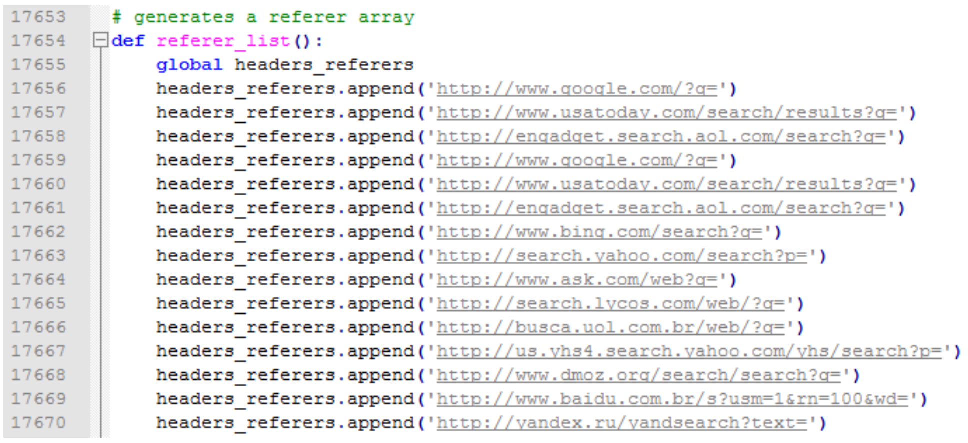


```
headers_useragents.append('Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 6.0; en-US)')
headers_useragents.append('Mozilla/4.0 (compatible; MSIE 6.1; Windows XP)')
headers_useragents.append('Opera/9.80 (Windows NT 5.2; U; ru) Presto/2.5.22 Version/10.51')
headers_useragents.append('Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.3) Gecko/20090913 Firefox/3.5.3')
headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 6.1; ru; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 2.0.50727)')
headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 5.2; de-de; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)')
headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.1) Gecko/20090718 Firefox/3.5.1 (.NET CLR 3.0.04506.648)')
headers_useragents.append('Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E')
headers_useragents.append('Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/532.1 (KHTML, like Gecko) Chrome/4.0.219.6 Safari/532.1')
headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; InfoPath.2)')
headers_useragents.append('Opera/9.60 (J2ME/MIDP; Opera Mini/4.2.14912/812; U; ru) Presto/2.4.15')
headers_useragents.append('Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-US) AppleWebKit/125.4 (KHTML, like Gecko, Safari) OmniWeb/v563.57')
headers_useragents.append('Mozilla/5.0 (SymbianOS/9.2; U; Series60/3.1 NokiaN95_8GB/31.0.015; Profile/MIDP-2.0 Configuration/CLDC-1.1 ) AppleWebKit/413 (KHTML, like Gecko) Safari/413')
headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR 3.5.30729; .NET CLR 3.0.30729)')
headers_useragents.append('Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.2; Win64; x64; Trident/4.0)')
headers_useragents.append('Mozilla/5.0 (Windows; U; WinNT4.0; en-US; rv:1.8.0.5) Gecko/20060706 K-Meleon/1.0')
headers_useragents.append('Lynx/2.8.6rel.4 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/0.9.8g')
headers_useragents.append('Mozilla/4.76 [en] (PalmOS; U; WebPro/3.0.1a; Palm-Arz1)')
headers_useragents.append('Mozilla/5.0 (Macintosh; U; PPC Mac OS X; de-de) AppleWebKit/418 (KHTML, like Gecko) Shiira/1.2.2 Safari/125')
headers_useragents.append('Mozilla/5.0 (X11; U; Linux i686 (x86_64); en-US; rv:1.8.1.6) Gecko/2007072300 Iceweasel/2.0.0.6 (Debian-2.0.0.6-0etch1+lenny1)')
headers_useragents.append('Mozilla/5.0 (SymbianOS/9.1; U; en-us) AppleWebKit/413 (KHTML, like Gecko) Safari/413')
headers_useragents.append('Mozilla/4.0 (compatible; MSIE 6.1; Windows NT 5.1; Trident/4.0; SV1; .NET CLR 3.5.30729; InfoPath.2)')
headers_useragents.append('Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 6.0; en-US)')
headers_useragents.append('Links (2.2; GNU/kFreeBSD 6.3-1-486 i686; 80x25)')
```

Figure 3: Saphyra – User agents strings in script

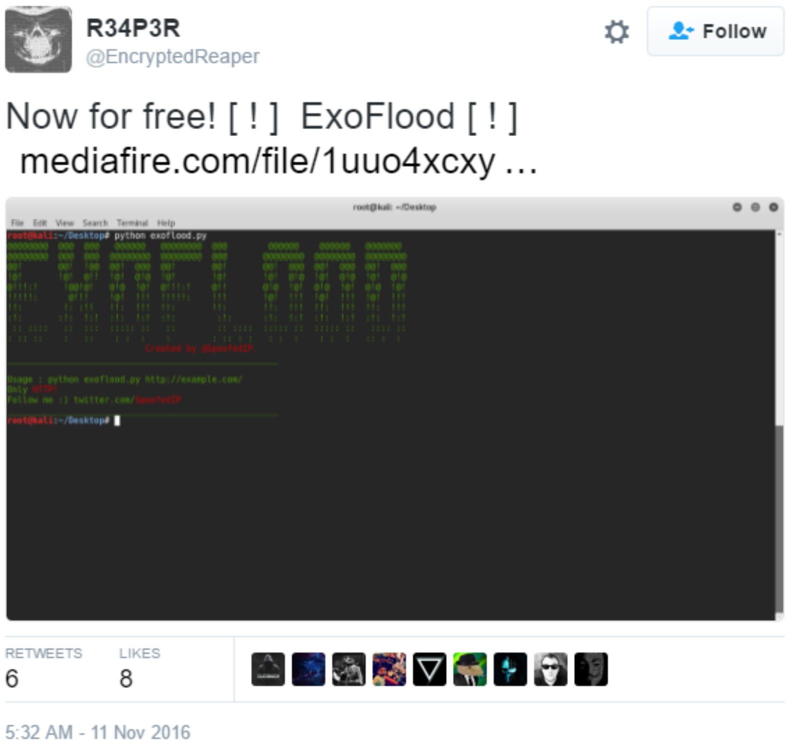
ExoFlood

ExoFlood is another HULK variant that has recently been published via Twitter. ExoFlood is very similar to Saphyra in the terms of user agents and referrer strings, and features over 4000 unique user agents and over 500 referrer stings resulting in event more unique request being generated by this tool (see Figures 4 and 5).



```
# generates a referer array
def referer_list():
    global headers_referers
    headers_referers.append('http://www.google.com/?q=')
    headers_referers.append('http://www.usatoday.com/search/results?q=')
    headers_referers.append('http://engadget.search.aol.com/search?q=')
    headers_referers.append('http://www.google.com/?q=')
    headers_referers.append('http://www.usatoday.com/search/results?q=')
    headers_referers.append('http://engadget.search.aol.com/search?q=')
    headers_referers.append('http://www.bing.com/search?q=')
    headers_referers.append('http://search.yahoo.com/search?p=')
    headers_referers.append('http://www.ask.com/web?q=')
    headers_referers.append('http://search.lycos.com/web/?q=')
    headers_referers.append('http://busca.uol.com.br/web/?q=')
    headers_referers.append('http://us.yhs4.search.yahoo.com/yhs/search?p=')
    headers_referers.append('http://www.dmoz.org/search/search?q=')
    headers_referers.append('http://www.baidu.com.br/s?usm=1&rn=100&wd=')
    headers_referers.append('http://yandex.ru/yandsearch?text=')
```

Figure 4: ExoFlood – Referer strings in the script



R34P3R
@EncryptedReaper

Now for free! [!] ExoFlood [!]
mediafire.com/file/1uuo4xcxy …

```
root@kali:~/Desktop# python exoflood.py
                    Created by @SpoofedIP.
--------------------------------------------------
Usage : python exoflood.py http://example.com/
Only HTTP!
Follow me :) twitter.com/SpoofedIP
--------------------------------------------------
root@kali:~/Desktop# 
```

RETWEETS 6 LIKES 8

5:32 AM - 11 Nov 2016

Figure 5: Attacker releasing ExoFlood

Reasons for Concern

Attackers are slowly moving away from easy-to-mitigate reflective attacks and starting to leverage simpler and defensively complex tools such as Layer 7 HTTP and HTTPS flood attacks. HTTP flood attacks are some of the most advanced cyber security threats to web servers. These attacks are hard to distinguish between legitimate and malicious traffic, creating a challenge to rate-based detection solutions.

Low and slow attacks use slow traffic that appears to be legitimate HTTP requests to pass traditional detection and mitigation systems undetected. In addition, these tools also keep a persistent foothold in the system by sending a standard HTTP command keep-alive to force the server to maintain the open connections.

All of this prevents servers' protection controls from recognizing a pattern and filtering out the attack traffic. The real concern is that these tools can be used on almost any device from laptops and tablets to desktops and cloud instances controlled via a cellphone.

How to Prepare

HTTP floods will continue to grow in popularity as more tools become available and those using the tools modify them in more complicated ways to defeat mitigation techniques. We are also seeing more attackers attempting to leverage HTTPS floods as the tools become more popular and widely available. There are several ways Radware can help mitigate Layer 7 attacks. By responding to the request and verifying the correct network behavior from the source, identifying repeating patterns in the traffic headers, and rate limit based on traffic parameters, we can help keep your network and application online.

DDoS Protection Considerations for Organizations Under Threat

- Hybrid DDoS Protection – on-premise and cloud-based solutions for real-time protection that also addresses high volume attacks and protects from pipe saturation.
- Behavioral-Based Detection - to quickly and accurately identify and block anomalies while allowing legitimate traffic through.
- Real-Time Signature Creation - to promptly protect from unknown threats and zero-day attacks.
- Cyber-Security Emergency Response Plan - that includes a dedicated emergency team of experts.

Radware urges companies to inspect and patch their network in order to defend against risks and threats.

Under Attack and in Need of Expert Emergency Assistance? Radware Can Help.

Radware offers a service to help respond to security emergencies, neutralize the risk and better safeguard operations before irreparable damages occur. If you're under DDoS attack or malware outbreak and in need of emergency assistance, Contact us with the code "Red Button".

Learn More at DDoS Warriors

To know more about today's attack vector landscape, understand the business impact of cyber-attacks or learn more about emerging attack types and tools visit DDoSWarriors.com. Created by Radware's Emergency Response Team (ERT), it is the ultimate resource for everything security professionals need to know about DDoS attacks and cyber security.