Classification: **TLP-GREEN**

RISK LEVEL: **MEDIUM**

Release Date: 6.1.16

Threat Advisory: *Trivial File Transfer Protocol (TFTP) Reflection DDoS*

1.0 / OVERVIEW / Akamai SIRT is investigating a new DDoS reflection and amplification method that abuses TFTP. This is yet another UDP-based protocol that has been added to the list of DDoS amplification scripts available for malicious use.

A weaponized version of the TFTP attack script began circulating around the same time as publications regarding research on the possibility of this attack method were posted. The research was conducted by Edinburgh Napier University.

As of April 20, 2016, Akamai has mitigated 10 attacks using this method against our customer base. Most of the attack campaigns consisted of multi-vector attacks which included TFTP reflection. An indication that this method has possibly been integrated into at least one site offering DDoS as a service.

Details of these attacks follow along with a revealing lack of distribution based on IP sources observed during early attacks.

2.0 / HIGHLIGHTED CAMPAIGN ATTRIBUTES / Here are the basic details of what is involved in these attacks:

- Peak bandwidth: 1.2 Gigabits per second
- Peak packets per second: 176.4 Thousand Packets per second
- Attack Vector: TFTP Reflection
- Source port: 69(TFTP)
- Destination port: Random

Attack payload response with default size 512 + 4 byte data block
16:00:11.497689 IP x.x.x.x.69 > x.x.x.x.10009: 516 DATA block 1
16:00:11.497833 IP x.x.x.x.69 > x.x.x.x.10009: 516 DATA block 1

Attack payload response with server default response 1456 + 4 byte data block
13:12:34.511256 IP x.x.x.x.69 > x.x.x.x.19636: 1460 DATA block 1
<snip>...PXE->EB:... PXENV+ at . !PXE at . No PXE stack found!
. entry point at .
UNDI code segment ., data segment . (.kB)
. UNDI device is PCI . Unable to determine UNDI physical device., type . (workaround enabled)..... .kB free base memory after PXE unload
. UNDI API call . failed: status code .
<snip>

Attack payload 3
08:53:03.540217 IP x.x.x.x.69 > x.x.x.x.51716: 1460 DATA block 1
08:53:03.541582 IP x.x.x.x.214.69 > x.x.x.x.46625: 1460 DATA block 1

Attack payload 4
18:38:18.086417 IP x.x.x.x.69 > x.x.x.x.41886: 516 DATA block 1
<snip>.L.!This program cannot be run in DOS mode.
<snip>
18:38:18.090832 IP 209.242.10.150.69 > 185.34.104.45.62798: 516 ERROR ENOTFOUND "Can't open file for read/write"

Figure 1: Payload samples from all 4 attacks. Only the first block of DATA (block 1) is sent to the target.

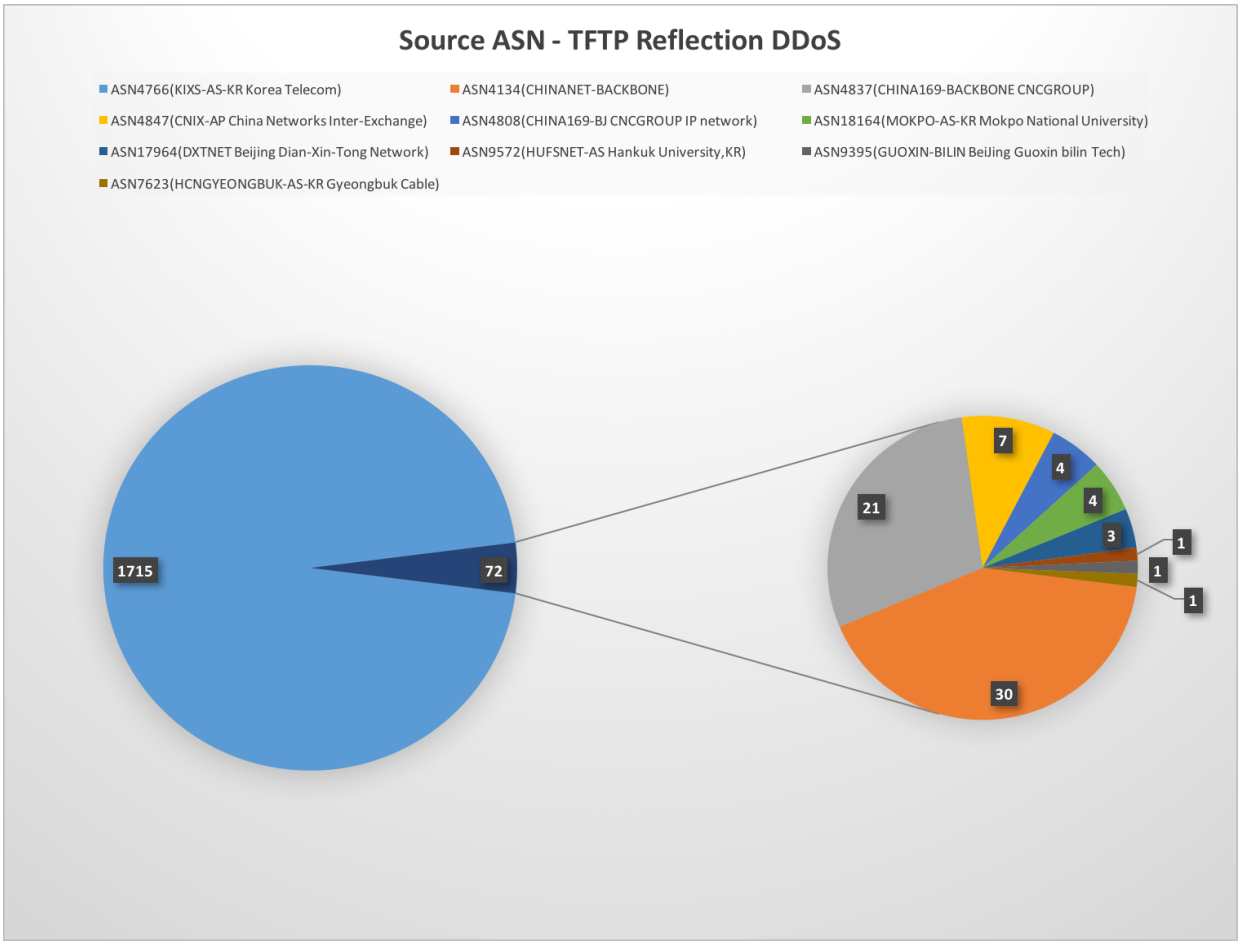

Figure 2: Respresents source ASN information of reflectors used in DDoS attacks against our customers.

3.0/ ATTACK CHARACTERISTICS / Trivial File Transfer Protocol has been around for years. It can be used for file transfers of firmware and configuration files, typically for networking devices, but it's not limited to just those devices.

Its simple design leaves out many features like authentication and directory listing capabilities. This simplicity also makes it ideal for use in PXE (Preboot eXecution Environment) deployments which are normally only LAN accessible and listen on UDP port 69 by default. Malicious actors have now added this protocol to the growing arsenal of reflection based amplification DDoS attack vectors using TFTP servers that are exposing this port to the internet.

Still, there are some limitations to the effectiveness of this attack using the currently observed methods.

Based on observed attack payloads, the behavior seems consistent with what is expected and described in RFC 1350. The targets of the TFTP reflection DDoS are flooded with RRQ(read request) DATA responses. The attack tool, described later, makes a default request for a file, "/x" in this case from the TFTP server. The victim TFTP server returns data to the requesting target host as a result of this request regardless of the filename mismatch.

A similar request can be made using a tftp client from the command line on Linux. Running a command such as "tftp localhost -c get /x" will result in the request payload below which would subsequently timeout unless tested against a real tftp server.

Command: tftp localhost -c get /x
15:21:43.291149 IP (tos 0x0, ttl 64, id 58345, offset 0, flags [none], proto UDP (17), length 42)
x.x.x.x.49915 > x.x.x.x.69: [udp sum ok] 14 RRQ "/x" netascii
E..*....@.....E...p../x.netascii.

Figure 3: Payload sample of basic tftp read request for file "/x" using regular TFTP client.

Based on lab testing, most TFTP servers won't respond to this request. The result would normally be a file not found or other error message. As with other popular methods of reflection like NTP, SSDP, and DNS, the requests are sent at alarming rates and simultaneously to multiple TFTP servers.
The request is forged in a way that forces the victim TFTP server to respond back to the malicious actors intended target IP.

Although the TFTP reflectors used thus far contain large files, sometimes over 20K bytes, only a limited portion is returned. TFTP sends back data in specific block sizes, by default this is 512 bytes of data + an additional 4 bytes of options (516 total bytes). The largest replies observed in attacks have contained 1,460 bytes all together as part of the payload.

This puts amplification at 36.86 and 104.29 for those two payloads respectively without taking IP and UDP headers into consideration. Luckily TFTP only sends out data in specific block sizes and requires acknowledgement of each block being received. Since the target of the attack will never acknowledge the data, only the first block is sent. This mitigates the potential of higher amplification based on single requests.

The next section will delve into a weaponized version of this attack tool already in the wild.

4.0/ AMPLIFICATION DDOS TOOL / Not much time was wasted it seems by malicious actors in creating a scripted attack tool for TFTP DDoS. A total of 4 attacks have been observed so far starting in March 14th. The largest attack using only TFTP reflection peaked at 1.2 Gbps. The release of the attack script also seems to coincide with media publications regarding the research into the possibility of this attack method.

The attack tool borrows much of the same code as other UDP based reflection tools. The command line is similar as well. The input required is a target IP (used as the source of the attack tool requests), the port(usually seen as the destination port at the target),file listing TFTP server addresses, number of threads, packet per second rate limit, and attack run time.

The attacks observed in most cases ignored the port parameter and resulted in random ports. Below is a sample of the requests going out as seen in tcpdump within a lab environment.

```
13:37:28.646587 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.647979 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.648357 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.648617 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.651597 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.652093 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.653410 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.655413 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.656291 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
13:37:28.657912 IP x.x.x.x.44235 > x.x.x.x.69:  14 RRQ "/x" netascii
```

Figure 4: Ten packet sample of the attack tool flood of requests.

The payload in the attack request is the same as the command line version performed previously. The code contains a section defining the parameters used in the attack request payload as shown below.

```
memcpy((void *)udph + sizeof(struct udphdr),
"\x00\x01\x2f\x78\x00\x6e\x65\x74\x61\x73\x63\x69\x69\x00", 14);
```

Figure 5: Attack script tool payload portion.

The values translate to the following TFTP options:

00 01 - opcode 1 = read request(RRQ)
2f 78 - /x = filename specified
00 = filename terminating byte
6e 65 74 61 73 63 69 69 - mode netascii = using mode netascii
00 = mode terminating byte

Figure 6: Represents the byte translation of TFTP options.

The same values can be seen in wireshark when examining either a regular TFTP request done from the command line with mode "netascii" or using the attack tool.

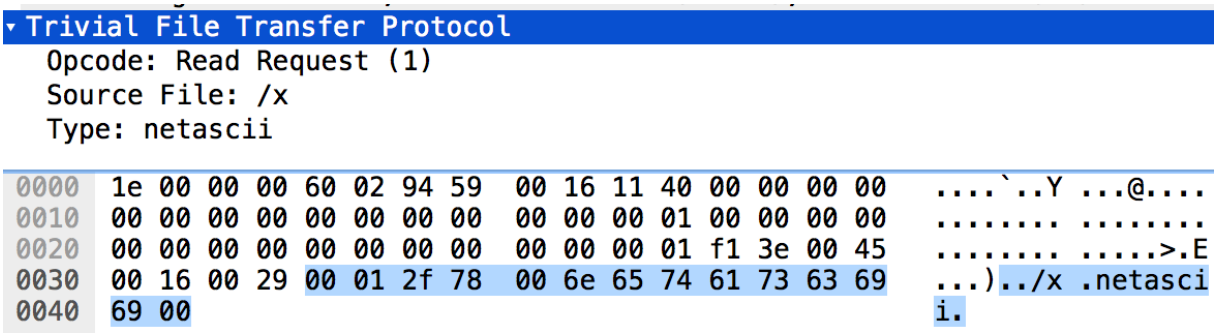

Figure 7: Wireshark view of tftp request payload.

The specific reasoning behind using"/x" as a filename is unknown at this point. This is likely the first thing that worked to initiate a file transfer on some TFTP servers. Inspection of attack payloads so far seems to indicate that the affected victims being leveraged for this reflection are part of PXE deployments. Testing with regular standalone TFTP servers reveals that these are not suitable reflectors. A common error from these servers is a simple file not found message.

5.0 / RECOMMENDED MITIGATION / This method of attack will not generate a high packet rate but the volume generated may be enough to consume bandwidth at the target site. So far the peak traffic for a single vector TFTP only attack has been measured at just over 1 Gbps.

TFTP is not recommended to be used over the internet. As such here are some precautions that may mitigate further use of this reflection method.

For those hosting TFTP servers:

- Assess the need to have UDP port 69 exposed to the internet. This should be firewalled and only allowed to trusted sources.
- Snort or a similar IDS can be used to detect for the abuse of TFTP servers in your network(rule provided below)

Customized Snort Detection:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 69 \
(msg: "TFTP DDoS Abuse request"; \
flow: to_server; \
content: "|00 01 2f 78 00 6e 65 74 61 73 63 69 69 00|"; dsize:14<>14; \
classtype:Reflection-Abuse; \
sid: 201600003; rev:1;)
```

For targets of TFTP amplification DDoS:

- Upstream filtering of UDP source port 69 can be applied if possible
- A DDoS mitigation provider can also be leveraged to absorb attack traffic generated

5.0 / CONCLUSION / This attack will likely see more use as part of multi-vector attack campaigns. The appearance of this vector in multi-vector campaigns is early evidence of possible inclusion into one or more sites offering DDoS as a service.

Alone, TFTP has produced a 1.2 Gbps attack but multi-vector campaigns, where TFTP is just one of many vectors, have peaked at just over 44 Gbps. So far, sources of TFTP reflection attacks collected during the early stages of attacks are poorly distributed. Mostly these are originating out of Asia with later attacks adding in sources from Europe.

This attack is also limited by the nature of TFTP as it's designed to deliver files, typically configuration files, but to a limited number of hosts at a time. In fact, messages like "Out of memory" in attack signatures allude to TFTP servers not being able to handle the rapid fire queries sent by the TFTP flood attack tool.

As stated above, we recommend the following steps to mitigate the threat:

For those hosting TFTP servers, assess the need to have UDP port 69 exposed to the Internet. This should be firewalled and only allowed to trusted sources. Snort or a similar IDS can be used to detect for the abuse of TFTP servers in your network.

Customers who believe they are at risk and need additional direction can contact Akamai directly through CCare at 1- 877-4-AKATEC (US And Canada) or 617-444-4699 (International), their Engagement Manager, or account team.

Non-customers can submit inquiries through Akamai's hotline at 1.877.425.2624, the contact form on our website at http://www.akamai.com/html/forms/sales_form.html, the chat function on our website at http://www.akamai.com/ or on twitter @akamai.

To access other white papers, threat bulletins and attack reports, please visit our Security Research and Intelligence section on Akamai Community.

About Akamai Security Intelligence Response Team (SIRT) Focuses on mitigating malicious global cyber threats and vulnerabilities, the Akamai Security Intelligence Response Team (SIRT) conducts and shares digital forensics and post-event analysis with the security community to proactively protect against threats and attacks. As part of its mission, the Akamai SIRT maintains close contact with peer organizations around the world and trains Akamai's Professional Services and Customer Care teams to both recognize and counter attacks from a wide range of adversaries. The research performed by the Akamai SIRT is intended to help ensure Akamai's cloud security products are best of breed and can protect against any of the latest threats impacting the industry.

About Akamai® As the global leader in Content Delivery Network (CDN) services, Akamai makes the Internet fast, reliable and secure for its customers. The company's advanced webperformance, mobile performance, cloud security and media delivery solutions are revolutionizing how businesses optimize consumer, enterprise and entertainment experiences for any device, anywhere. To learn how Akamai solutions and its team of Internet experts are helping businesses move faster forward, please visit www.akamai.comor blogs.akamai.com, and follow @Akamai on Twitter

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 40 offices around the world. Our services and renowned customer care enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers and contact information for all locations are listed on www.akamai.com/locations