



○ SOTI SUMMER 2018

[state of the internet] / security

ATTACK SPOTLIGHT

ATTACK SPOTLIGHT

Memcached

1.0 OVERVIEW

Earlier this year, Akamai mitigated the largest DDoS attack in its history. Targeting a software company, the attack broke through the 1 Tbps threshold for the first time. A closer look at this new vector — a memcached reflection attack — offers significant warnings and lessons learned for anyone in charge of an organization’s online security and business continuity planning, as well as for the security community at large.

The initial wave of attacks began to trickle in at the end of February 2018. Initial suspicions indicated this was just the beginning, and attacks would get worse before they got better. Some of the first attacks exceeded 100 Gbps — already a substantial size for a reflection vector. Only a few days later, Akamai found itself at the center of the largest attack ever mitigated, topping 1.3 Tbps, more than twice the traffic generated for the 623 Gbps attack in September 2016.

This record-setting attack is the largest attack Akamai has seen to date. Organizations need to take this new high-water mark for DDoS attacks into account in their threat evaluation and mitigation plans. The median of attacks Akamai defends customers against is currently 1.3 Gbps, but organizations that can’t tolerate any downtime must be prepared to combat this new vector and its potential for attacks of massive scale.

1 Tbps

Threshold broken by the memcached reflector

Memcached How?

Mirai held the previous record for volumetric DDoS attacks. At the time, the record-setting Mirai attack garnered much attention into the potential of repeat attacks using Mirai or other IoT-based botnets. In general, a heightened level of awareness was warranted. The largest DDoS attacks have been generated by DDoS malware like Mirai. These tools attempt to infect as many devices as possible and use reflectors like memcached to achieve the greatest potential attack volume. Memcached went largely unnoticed as a reflection vector until February 2018 because it had not been integrated into attack tools until then.



fig 1.1 **The Memcached attack vector was well known for months before initial observed attacks.**

6/17	Memcached DRDoS disclosed by Okee team
11/17	Research released publicly at PoC 2017 conference
1/18	Articles begin to surface with detailed examples on how to exploit servers and increase attack amplification

2.0 ATTACK AND PROTOCOL OVERVIEW


50,000x

The potential amplification factor for a single UDP packet request when being exploited by memcached

Memcached was developed to act as a distributed memory caching system. Since the protocol uses UDP, an insecure protocol, and carries the potential for tremendous amplification, it has the key traits of a successful reflection-based attack vector. By default, the memcached protocol allows a specific key value to store 1 MB of data. A single UDP packet can request that the data be delivered to the DDoS target multiple times, creating a potential amplification factor in excess of 50,000 times the traffic sent. For perspective, a DNS reflection attack typically uses domains that contain 3,000 to 5,000 bytes of data with an amplification factor below 500 times the traffic generated by offending tools.

Since the attack is based on whatever is stored for a specific memcached key, or output from the stats command, there is some variation in the potential amplification factor. We explore what is possible with the default attack payload found in available attack

scripts as well as what was observed during the 1.3 Tbps attack. The following payload is from an attack script available to weaponize this vector, in a similar way to other reflection methods.

 **fig 1.2 Default attack payload sent by attack script**

```
\x00\x01\x00\x00\x00\x01\x00\x00stats\r\n
```

By default, this script will send a “stats” query over UDP on port 11211 to any number of listed memcached servers. Using a default memcached server instance in a lab environment, the above query produced the following.

 **fig 2.2 Default memcached server instance**

```
23:37:56.587705 IP 192.168.20.62.80 >
192.168.20.20.11211: UDP, length 15
E..+....?.....>.....P+....      .....stats

23:37:56.587994 IP 192.168.20.20.11211 >
192.168.20.62.80: UDP, length 1349
E..a..@.@..I.....>+..P.M.....STAT pid 2096
STAT uptime 9090
STAT time 1526008544
STAT version 1.4.33
STAT libevent 2.0.21-stable
STAT pointer_size 64
STAT rusage_user 1.220000
STAT rusage_system 296.324000
STAT curr_connections 9
STAT total_connections 11
STAT connection_structures 11
<snip>
```

The standard [BAF](#) (base amplification factor) calculation takes into account the payload portion of the query and response and results in a 90-fold amplification —

impressive, but not enough to achieve the massive attack that occurred. For the largest peak attack of 1.3 Tbps, the following payloads were observed.



fig 2.3 Payloads during the peak attack of 1.3 Tbps

```
13:27:06.587956 IP X.X.X.X.11211 > X.X.X.X.12251: UDP,
length 1400
E...XP@.8....).....q+./.....?...<snip>
VALUE a 1 1000000 58
abcdefghijklmnopabcdefghijklmnopabcdefghijklmnopabcdefghijklmnop
hijabcdefghijklmnopabcdefghijklmnopabcdefghijklmnopabcdefghijklmnop
efghijabcdefghijklmnopabcdefghijklmnopabcdefghijklmnopabcdefghijklmnopghija
bcdefghijabcdefghijklmnopabcdefghijklmnopabcdefghijklmnopghij
ij<snip>
```

The [protocol documentation](#) for memcached outlines the server response data as follows:



fig 2.4 Memcached response data outline

```
VALUE <key> <flags> <bytes> [<cas unique>]\r\n
<data block>\r\n
```

- <key> is the key for the item being sent
- <flags> is the flags value set by the storage command
- <bytes> is the length of the data block to follow,
 - *not* including its delimiting \r\n
- <cas unique> is a unique 64-bit integer that uniquely identifies this specific item
- <data block> is the data for this item

Using this breakdown, it can be determined that the value key “a” in the observed attack payload contains 1,000,000 bytes of data.

The following re-creation of that single key value query, along with data response from the server, was done in a lab environment.

fig 2.5 Query and response for attack payload in lab environment

```
1 query of 16 bytes (Note: using gets or get would work here)
```

```
22:58:37.482340 IP X.X.X.X.80 > X.X.X.X.11211: UDP, length 16
```

```
E..,....?.....>.....P+.....w.....gets a
```

```
Response from server = 719 total packets.
```

```
718 payloads of 1,400 bytes (1,428 with headers) + 1 response of 580 bytes (608 with headers) in the final data packet
```

```
Response packets with full header length and payload lengths of 1,400 bytes per packet and 580 respectively.
22:58:37.482513 IP (tos 0x0, ttl 64, id 23540, offset 0, flags [DF], proto UDP (17), length 1428)
```

```
    X.X.X.X.11211 > X.X.X.X.80: UDP, length 1400
```

```
E...[.@@./.....>+..P...4w.....VALUE a 0 1000000 1 abcdefghijabcdefghij<snip>
```

```
22:58:37.517452 IP (tos 0x0, ttl 64, id 24258, offset 0, flags [DF], proto UDP (17), length 608)
```

```
    X.X.X.X.11211 > X.X.X.X.80: UDP, length 580
```

```
E..`^.@.@.0(.....>+..P.L.w.....fghijabcdefghij<snip>
END
```

Since the query produces multiple response packets, the BAF should only be applied to the query and one response packet. The memcached server will continue sending full packet headers, along with 1,400 byte payloads for the remaining replies. In this example, a 16-byte payload query resulted in 1,025,884 bytes of response data, or 64,118x amplification.

Although a high amplification factor can be achieved by the single key value query, attackers took it a step further and combined multiple key requests into a single query. Queries like this were observed in the days following the 1.3 Tbps attack.

fig 2.6 Memcached query and response suspected in 1.3 Tbps attack

```
1 query of 1,154 bytes with multiple requests for key "a"
23:28:39.492218 IP X.X.X.X.48779 > X.X.X.X.11211: UDP,
length 1154
E....~@.@.V.....+. ....w.....gets a a a <snip>
```

```
Response from server = 409,493 total packets.
409,492 payloads of 1,400 bytes (1,428 with headers) + 1
response of 259 bytes (287 with headers) in the final
packet
```

```
Response packets with full header length and payload
lengths of 1,400 bytes per packet and 259 respectively.
16:24:08.739151 IP (tos 0x0, ttl 64, id 552, offset 0,
flags [DF], proto UDP (17), length 1428)
X.X.X.X.11211 > X.X.X.X.80: UDP, length 1400
E....(@.@.....>+..P...4w...?...VALUE a 0 1000000 1
abcdefghijklmnopqrs<snip>
```

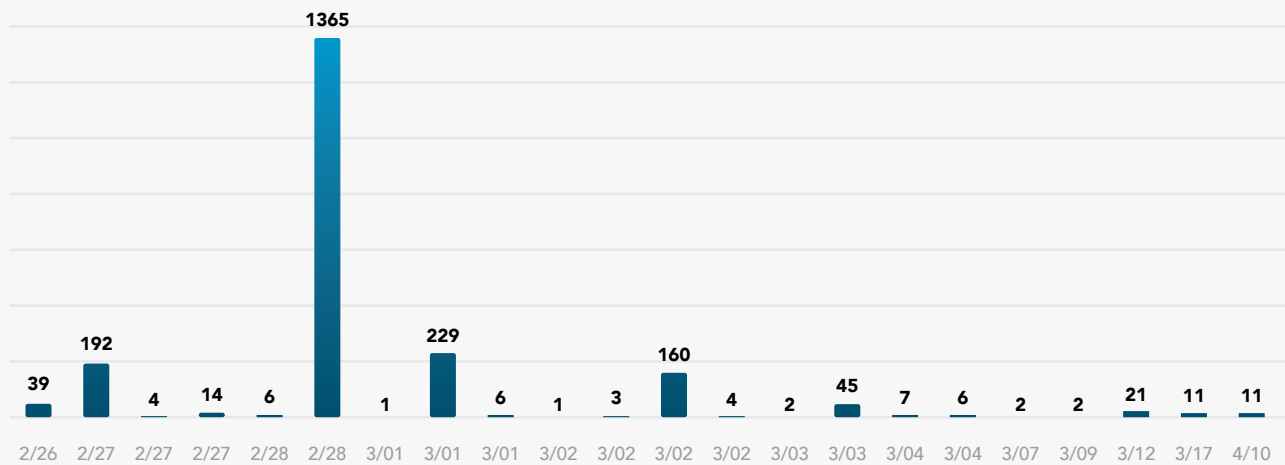
```
16:24:29.008163 IP (tos 0x0, ttl 64, id 17041, offset 0,
flags [DF], proto UDP (17), length 287)
X.X.X.X.11211 > X.X.X.X.80: UDP, length 259
E...B.@.@.M.....>+..P...w.?.?...ghijklmnopqrs<snip>
END
```

The query for this example is 1,154 bytes in length. It carried a bigger outbound data requirement from the attacker-controlled host. From this single query, the attack target would receive 584,754,835 bytes of data. In other words, the single packet query resulted in traffic being received by the target that was 506,720 times the amount of traffic sent! Memcached, unfortunately, appears to be tailor-made to be a vector for reflection and amplification attacks.

3.0 OBSERVED ATTACK TIMELINE

Most reflection vectors typically start off with lower volume attacks before they reach their full potential. There is indication that memcached reflection attacks were ongoing in Asia in advance of more widespread attacks. Akamai mitigated five of these attacks prior to the record-setting 1.3 Tbps attack.

fig 3.1 February 28 stands out as the peak of memcached attacks through April.



Attacks of this size are bad news, but they do generate a lot of awareness in the community and help create urgency for those working to mitigate the threat. For memcached, the [developer](#) acted quickly to disable the UDP port by default. ISPs also provided filtering and/or rate limiting of the UDP port used by default for this protocol. The timeline provides some insights into the effects of this effort, as attacks over 100 Gbps were nonexistent by the end of March.

4.0 MITIGATION

Mitigating memcached reflection attacks begins at the source. If the number of reflectors is limited, then the attack becomes much less effective. As of memcached version 1.5.6, UDP, which is required for this attack to work, is disabled by default. Where disabling the UDP protocol is not possible, consider completely filtering external access or limiting access to a fixed set of required source IPs.

5.0 CONCLUSION

5

The number of times Akamai mitigated these attacks before the 1.3 Tbps record-setting attack

The timeline on the previous page indicates that the memcached vector has largely been tamed by mitigation efforts. Most organizations are unlikely to be able to withstand an attack on the order of 1 Tbps, as it would exhaust bandwidth and severely impact performance. For smaller attacks, a filter for source port 11211 can help, although there should be consideration of how this could impact any regular traffic.

Since the possibility exists for any UDP service to become the next big reflection vector, it is very difficult to know in advance which services could create the most devastating attacks in the future. Early in 2018, Akamai SOC mitigated a new wave of previously unused reflection methods, such as [IKE](#) and IPMI reflection. Neither of these protocols come close to the amplification factor of other reflection vectors, but they do have a vast number of available reflector sources.

It's important for those who deploy and use these services to take time to become familiar with the traffic patterns associated with the services. Does the service work off of one specific port? Does response traffic from this service sometimes exceed 500 bytes or more than the 1,500-byte Ethernet default MTU (maximum transmission unit)? Can the data provided in the response be manipulated by outside sources as it was with memcached, and NTP prior to that?

It is important that notification of these protocol vulnerabilities to vendors for patching and mitigation happens before it's too late. For an attack discovered in June 2017, memcached should have been under control before 2018, before a 1.3 Tbps attack woke the world to its potential. Yet even following the public disclosure of this vector in November 2017, detailed proof-of-concept and instructional articles were released in China; preceding the large-scale attacks themselves, little attention was paid to memcached.

Once a service is determined to have the potential for large volumetric DDoS attacks, developers, vendors, and other parties responsible for the service should evaluate the impact and remedy the situation. Could this vector have been mitigated sooner? Are there any more hidden surprises in the vast array of UDP services available, and can they be detected before they become a problem? We have already surpassed the 1 Tbps attack threshold; how much more can the Internet take? Records are made to be broken, and the next new reflector or family of botnets may be what it takes to break memcached's spot. And what breaks the Internet.

ABOUT AKAMAI

As the world's largest and most trusted cloud delivery platform, Akamai makes it easier for its customers to provide the best and most secure digital experiences on any device, anytime, anywhere. Akamai's massively distributed platform is unparalleled in scale with over 200,000 servers across 130 countries, giving customers superior performance and threat protection. Akamai's portfolio of web and mobile performance, cloud security, enterprise access, and video delivery solutions are supported by exceptional customer service and 24/7 monitoring. To learn why the top financial institutions, online retail leaders, media & entertainment providers, and government organizations trust Akamai please visit www.akamai.com, blogs.akamai.com, and @Akamai on Twitter. You can find our global contact information at www.akamai.com/locations or call 877-425-2624. Published 06/18.

For more information on current attacks and Internet trends, read the [State of the Internet Security report](#) and Akamai's [Security Intelligence and Threat Research blog](#).

Questions? Email us at SOTI@akamai.com

