Analysis and Prevention of Memcache UDP Reflection Amplification Attack

Kai Bai [a]

[1] School of Computer Science, Yangtze University, Jingzhou 434023, Hubei, China;

[a] baikai@yangtzeu.edu.cn

Abstract

The Memcached application has been designed to speed up dynamic web applications by reducing stress on the database that helps administrators to increase performance and scale web applications. It's widely used by thousands of websites. Massive memcached-based reflection DDoS attacks with an unprecedented amplification factor have been ongoing for the last days, the attack apparently abuses unprotected Memcached servers. This paper describes the Principle of Memcache UDP Reflection Amplification Attack and the General Trend and the Solution to memcached attacks.

Keywords

Reflection Amplification Attack, Memcache DDoS, server, vulnerability scanning.

1. Introduction

Memcache is a distributed memory caching system and is used to speed up dynamic database-driven websites and Internet-facing services by caching data and objects in RAM.It is often deployed in data center, cloud, and IaaS networks.According to both Rapid7 and SANS ISC, there are currently over 100,000 exposed memcached servers on the Internet.

DDoS attacks remain one of the biggest internet security threat globally, the DDoSMon system detected roughly 20,000 attacks per day over the past period. This page contains the observations and insights derived from the various DDoS attacks that detected by the DDoSMon and our Botnet tracking system. It represents a unique view into the attack trends unfolding online, including attack statistics and behavioral trends for latest DDoS attacks.

Now we are aware of a new DDoS reflection attack vector: UDP-based memcached traffic. Memcached is a tool meant to cache data and reduce strain on heavier data stores, like disk or databases. The protocol allows the server to be queried for information about key value stores and is only intended to be used on systems that are not exposed to the Internet. There is no authentication required with memcached. When this is added to the ability to spoof IP addresses of UDP traffic, the protocol can be easily abused as a reflector when it is exposed to the Internet. Akamai has seen multiple attacks, some in excess of 190 Gbps, with the potential for much larger attacks.

How Memcrashed DDoS Amplification Attack Works?Like other amplification methods where hackers send a small request from a spoofed IP address to get a much larger response in return, Memcrashed amplification attack also works by sending a forged request to the targeted server (vulnerable UDP server) on port 11211 using a spoofed IP address that matches the victim's IP. According to the researchers, just a few bytes of the request sent to the vulnerable server can trigger the response of tens of thousands of times bigger. According to the researchers, most of the Memcached servers being abused for amplification DDoS attacks are hosted at VPS, and other small hosting providers. At peak we've seen 260Gbps of inbound UDP memcached traffic. This is massive for a new amplification vector. But the numbers don't lie. It's possible because all the reflected packets are very large.

This paper includes four main contributions: (1)We describe the evolution of Memcache UDP Reflection Amplification Attack. (2) We analysis the Principle of Memcache UDP Reflection

Amplification Attack. (3) We highlight mechanisms that improve our ability to operate our system at scale. (4)We characterize the production workloads defensed Attack on our system.

2. The Targets, the Sources and Breakdowns

Memcache DDoS attack has come out of nowhere and really captured lots of attentions. When we look at the news, we see all sort of reports but hardly can get a good idea what the real situation is, for example the most important question, how many victims are out there? And how big the attack army is?

We has been running the free ddosmon platform for quite some time and with all the massive amount of network data we have good visibility into the ddos world, so, we will provide our insights.we mentioned that there had been hardly any Memcache DDoS attacks in the last 9 months since our 360 0kee team publicly disclosed this vulnerability. However, since 2018-02-24, the frequency of attacks has increased dramatically. As shown in the following figures:

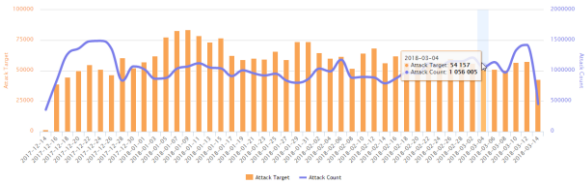

Fig. 1. Breakdowns

3. Analysis of Memcache Vulnerabilities

Reflection attacks happen when an attacker forges its victim's IP addresses in order to establish the victim's systems as the source of requests sent to a massive number of machines. The recipients of those requests then issue an overwhelming flood of responses back to the victim's network, ultimately crashing that network. These types of DDoS attacks differ from amplification attacks, where publicly accessible open DNS servers are used to flood victims with DNS responses.

In the case of the amplification attacks , attackers were able to send a small byte-sized UDP-based packet request to a memcached server (on port 11211). The packets would be spoofed to appear as if they were sent from the intended target of the DDoS attack. In response, the memcached server responds by sending the spoofed target a massively disproportionate response.

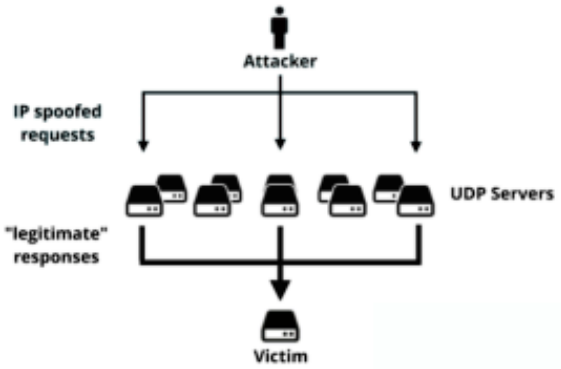

Fig. 2 The occurrence of reflection attack

3.1 What is UDP in Comparison to TCP

The memcache protocol was never meant to be exposed to the Internet, but there are currently more than 50,000 known vulnerable systems exposed at the time of this writing. By default, memcached listens on localhost on TCP and UDP port 11211 on most versions of Linux, but in some distributions it is configured to listen to this port on all interfaces by default.

When a system receives a memcached get request, it forms a response by collecting the requested values from memory, sending them over the wire in an uninterrupted stream. This response is sent to

the target in multiple UDP packets, each with a length of up to 1400 bytes. It is difficult to determine the exact amplification factor of memcached, but the attacks Akamai saw generated nearly 1 Gbps per reflector. Other organizations have reported attacks in excess of 500 Gbps using memcached reflection.

To make the situation worse, attackers can influence the amplification factor for a given node by inserting records into the open server, thus having a large object to use during reflection. By default memcached uses a limit of 1MB per stored value, however this constraint is user configurable. Even more worrisome is that multiple keys or duplicate keys can be requested multiple times in a single request. This allows attackers to load up a number of large values into the data store and then use them in attacks. It is possible that an attacker could purposely place a 1MB value in the data store, and using a spoofed UDP packet request that single 1MB value hundreds of times per request. This would result in a massive amplification factor where a 203 byte request results in 100MB response of reflected traffic, per request. It doesn't take much imagination to see how this could be and is being abused, resulting in considerable DDoS attacks.

Attacks of the size potentially created by memcached reflection cannot be easily defended against by data center solutions, requiring the cooperation of upstream ISPs and/or cloud based DDoS protection services. Blocking port 11211 is a starting point for defenses and will prevent systems on your network from being used as reflectors. Configuring mitigation controls, like port blocking, can allow for this traffic to be handled quickly and efficiently.

3.2 So What is the Catch

Well first of all UDP is really, really much less reliable. That means that it cant guarantee anything. You have to implement "TCP" yourself if you want any safety or error correction. This seems to be reason why its not fully supported by PHP memcache extension. It also seems that memcached itself limits the functionality in UDP mode.

Command to get stats executed over UDP fails:

```
echo $'stats\r\nquit\r\n' | nc -u 10.60.33.50 11211
Double-click on the code above and press CRTL+C to copy it.
```

Fig. 3. The stats of UDP

Produces:

```
SERVER_ERROR multi-packet request not supported
Double-click on the code above and press CRTL+C to copy it.
```

Fig. 4. Produces

3.3 How the Protocol Works

The protocol is documented here. It’s pretty straightforward.The easiest amplification attacks is to send the “stats” command. This is 15 byte UDP packet that causes the server to send back either a large response full of useful statistics about the server. You often see around 10 kilobytes of response across several packets.A harder, but more effect attack uses a two step process. You first use the “add” or “set” commands to put chunks of data into the server, then send a “get” command to retrieve it. You can easily put 100-megabytes of data into the server this way, and causes a retrieval with a single “get” command.That’s why this has been the largest amplification ever, because a single 100-byte packet can in theory cause a 100-megabytes response.

Doing the math, the 1.3 terabit/second DDoS divided across the 15,000 servers we can found vulnerable on the Internet leads to an average of 100-megabits/second per server. This is fairly minor, and is indeed something even small servers (like Raspberry Pis) can generate [6] .

We begin the analysis by presenting a detailed description ofmemcached’s request pipeline. There is substantially more to executing a memcached query than just looking up a value in a hash-table. In order to gain a more detailed understanding,we traced the life-cycle of a memcached request in Linux.

Fig. 1 depicts the basic steps involved.A client initiates a request by constructing a query andcalling the write system call (1). The request undergoes TCP/IP processing (2), is transmitted by the client's NIC (3),and is then sent to the server's NIC via cables and switches,where upon the processor core running memcached receives an interrupt.Linux quickly acknowledges the interrupt, constructs a struct skbuff, and calls netif receive skb in softIRQ context (4). After determining it is an IP packet, ip rcv is called (5), and after TCP/IP processing is complete.

3.4 Neutering the Attack

If they are using the more powerful attack against the server, we can neuter it: we can send a "flush_all" command back at the servers who are flooding you, causing them to drop all those large chunks of data from the cache.I'm going to describe how I would do this.

First, get a list of attackers, meaning, the amplifiers that are flooding you. The way to do this is grab a packet sniffer and capture all packets with a source port of 11211. Here is an example using tcpdump.

If they are using the more powerful attack against the server, we can neuter it: we can send a "flush_all" command back at the servers who are flooding you, causing them to drop all those large chunks of data from the cache.I'm going to describe how I would do this.

First, get a list of attackers, meaning, the amplifiers that are flooding you. The way to do this is grab a packet sniffer and capture all packets with a source port of 11211. Here is an example using tcpdump.

```
tcpdump -i -w attackers.pcap src port 11221
```

Fig. 5. tcpdump

Let that run for a while, then hit [ctrl-c] to stop, then extract the list of IP addresses in the capture file. The way I do this is with tshark:

```
tshark -r attackers.pcap -Tfields -eip.src | sort | uniq > amplifiers.txt
```

Fig. 6. Wireshark

Now, craft a flush_all payload. There are many ways of doing this. For example, if you are using nmap or masscan, you can add the bytes to the nmap-payloads.txt file. Also, masscan can read this directly from a packet capture file. To do this, first craft a packet, such as with the following command line foo:

```
echo -en "\x00\x00\x00\x00\x00\x01\x00\x00flush_all\r\n" | nc -q1
-u 11211
```

Fig. 7. craft a packet

Now that we have our list of attackers (amplifiers.txt) and a payload to blast at them (flush_all.pcap), use masscan to send it:

```
masscan -iL amplifiers.txt -pU:112211 –pcap-payload
flush_all.pcap
```

Fig. 8. list attackers

4. Recommended Actions

Memcached lacks access controls by design, and that's why it shouldn't be exposed to the Internet. Attacks of the size potentially created by memcached reflection cannot be easily defended against by data center solutions, requiring the cooperation of upstream ISPs and/or cloud-based DDoS protection services.

4.1 Neutering the Attack

1.is the Memcached 11211 UDP port open to the outside Internet test? You can use the NC tool to test the port and see whether the memcached process is running on the server.

Test port: NC -vuz IP address 11211

Test whether the memcached service is open to the outside world: telnet IP address 11211, if the 11211 port is open, it may be affected

The inspection process: PS -aux grep memcached | state

2. use the "echo -en" \x00\x00\x00\x00\x00\x01\x00\x00stats\r\n "NC -u IP | address 11211 command to view the content returned, if the return content is non empty, is that your server may be affected.

4.2 Solution

All relevant network infrastructure, host/application/service, and operational Best Current Practices (BCPs) should be implemented by network operators. In particular, state minimization is highly encouraged as a general operational principle to increase resilience in the face of attack. Situationally-appropriate network access policies should be implemented via transit ACLs (tACLs) on Internet Data Center (IDC) upstream transit links to block unauthorized network traffic destined for UDP/11211 and TCP/11211 from ingressing the IDC.

Network operators should export flow telemetry (e.g., NetFlow, IPFIX, s/Flow, cflowd/jflow, Netstream, et. al.) from their peering/transit/customer aggregation edges and Internet data center (IDC) distribution edges to Arbor SP, which provides the ability to detect, classify, and traceback DDoS attack traffic.

Given that intentional production use of memcached across the public Internet is vanishingly rare, traffic sourced from UDP/11211 may be safely rate-limited at peering/transit/customer aggregation edges by the application of situationally-appropriate policies deployed on edge routers. Alternately, transit ACLs (tACLs) may be deployed at peering edges, customer aggregation edges, and Internet data center (IDC) distribution gateway edges to block network traffic sourced from UDP/11211. In either case, care should be exercised to avoid unnecessary overblocking, and Arbor SP should be utilized in order to determine the efficacy of policies or tACLs implemented at network edges.

Arbor SP/TMS and APS IDMSes may be deployed in a situationally-appropriate manner to mitigate these attacks via multiple DDoS countermeasures, as well as flowspec for both attack mitigation and selective traffic diversion (SP/TMS).TMSes and/or APSes may be used both to mitigate reflected/amplified DDoS attack traffic as well as to prevent memecached priming queries and attack-stimulation queries from reaching misconfigured – and thus exploitable – memcached servers located in IDC networks and on end-customer premise networks.

As always, network operators are strongly encouraged to implement source address validation/BCP38/BCP84 in order to prevent their networks and the networks of their end-customers from being leveraged in reflection/amplification DDoS attacks. It is also recommended that network operators scan their IDC networks, as well as those of their end-customers, in order to identify abusable memcached installations so that remediation can take place on a timely basis.

4.3 Verification Method

After the repair, you can use the following methods to test whether the server repair measures are effective.

1., if you block the 11211 port of the external TCP protocol, you can use the command "telnet IP 11211" on the external network office computer. If the connection fails, it means that the 11211 port of the external TCP protocol has been closed.

2. if you disable the Memcached service on the server of the UDP protocol, you can run the following "echo -en" \x00\x00\x00\x00\x00\x01\x00\x00stats\r\n "NC -u IP | address 11211" command to detect whether to close the memcached UDP service agreement, see the return, if the return is empty,

that your server has successfully repaired vulnerabilities, can also use the "netstat -an grep UDP UDP 11211 | view port is in the listening state, if there is no monitoring, said it has successfully shut down memcached UDP protocol.

5. Conclusion

In this paper, we have analyzed the Principle of Memcache UDP Reflection Amplification Attack. We have profiled the Sources and Breakdowns of Memcache UDP Reflection Amplification Attack.. We have enumerated Recommended Actions and Solution to memcached attacks.From the analysis, The Memcached amplification attack can have a serious impact on network health and the stability of services. However, the attack can be mitigated effectively by following best practices for running networked services. After applying the changes in this paper, it is a good idea to continue to monitor services to ensure proper functionality and connectivity is maintained.

Acknowledgements

This study has been supported by the Scientific Research Projects of Hubei Provincial Department of Education (B2017032).

References

[1] Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2014). Node. js in Action. Manning
[2] Klein, A. (2005). DOM based cross site scripting or XSS of the third kind.Web Application Security Consortium, Articles, 4, 365-372
[3] Weinberger, J., Saxena, P., Akhawe, D., Finifter, M., Shin, R., & Song, D. (2011, September). A systematic analysis of XSS sanitization in web application frameworks. In European Symposium on Research in Computer Security (pp. 150-171). Springer Berlin Heidelberg
[4] Bogdanov, S., Patruno, A., Archibald, A. M., Bassa, C., Hessels, J. W., Janssen, G. H., & Stappers, B. W. (2014). X-ray observations of XSS J12270-4859 in a new low state: A transformation to a disk-free rotation-powered pulsar binary. The Astrophysical Journal, 789(1), 40.
[5] Papitto, A., Torres, D. F., & Li, J. (2014). A propeller scenario for the gamma-ray emission of low-mass X-ray binaries: the case of XSS J12270− 4859. Monthly Notices of the Royal Astronomical Society, 438(3), 2105-2116.
[6] Roy, J., Bhattacharyya, B., & Ray, P. S. (2014). GMRT discovery of a 1.69 ms radio pulsar associated with XSS J12270-4859. The Astronomer's Telegram, 5890.
[7] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, “Workload analysis of a large-scale key-value store,” in ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 1. ACM, 2012, pp. 53–64
[8] E. Cecchet, V. Udayabhanu, T. Wood, and P. Shenoy, “Benchlab: an open testbed for realistic benchmarking of web applications,” in Proceedings of the 2nd USENIX conference on Web application development. USENIX Association, 2011.
[9] “Mutilate: high-performance memcached load generator.” [Online]. Available: https:// github. com/leverich/mutilate.