Ανάλυση βασικών DDoS επιθέσεων

CloudFare Inc

SYN flood attack

SYN flood attack

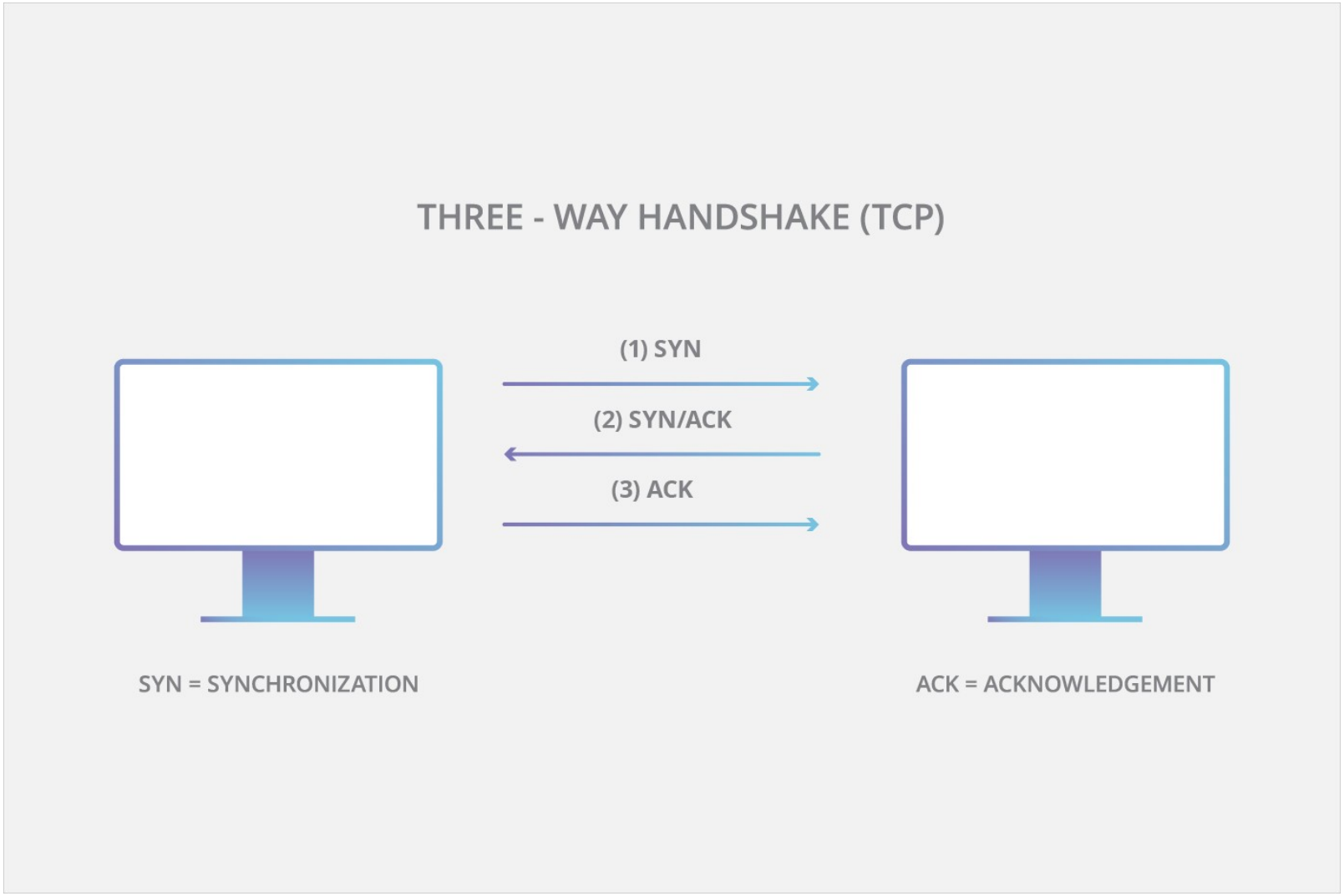
- A SYN flood (half-open attack) is a type of denial-of-service (DDoS) attack which aims to make a server unavailable to legitimate traffic by consuming all available server resources.
- Repeatedly sending initial connection request (SYN) packets, the attacker is able to overwhelm all available ports on a targeted server machine, causing the targeted device to respond to legitimate traffic sluggishly or not at all.

SYN flood attacks work by exploiting the handshake process of a TCP connection.

Under normal conditions, TCP connection exhibits three distinct processes in order to make a connection.

- First, the client sends a SYN packet to the server in order to initiate the connection.
- The server then responds to that initial packet with a SYN/ACK packet, in order to acknowledge the communication.
- Finally, the client returns an ACK packet to acknowledge the receipt of the packet from the server. After completing this sequence of packet sending and receiving, the TCP connection is open and able to send and receive data.

SYN flood attack



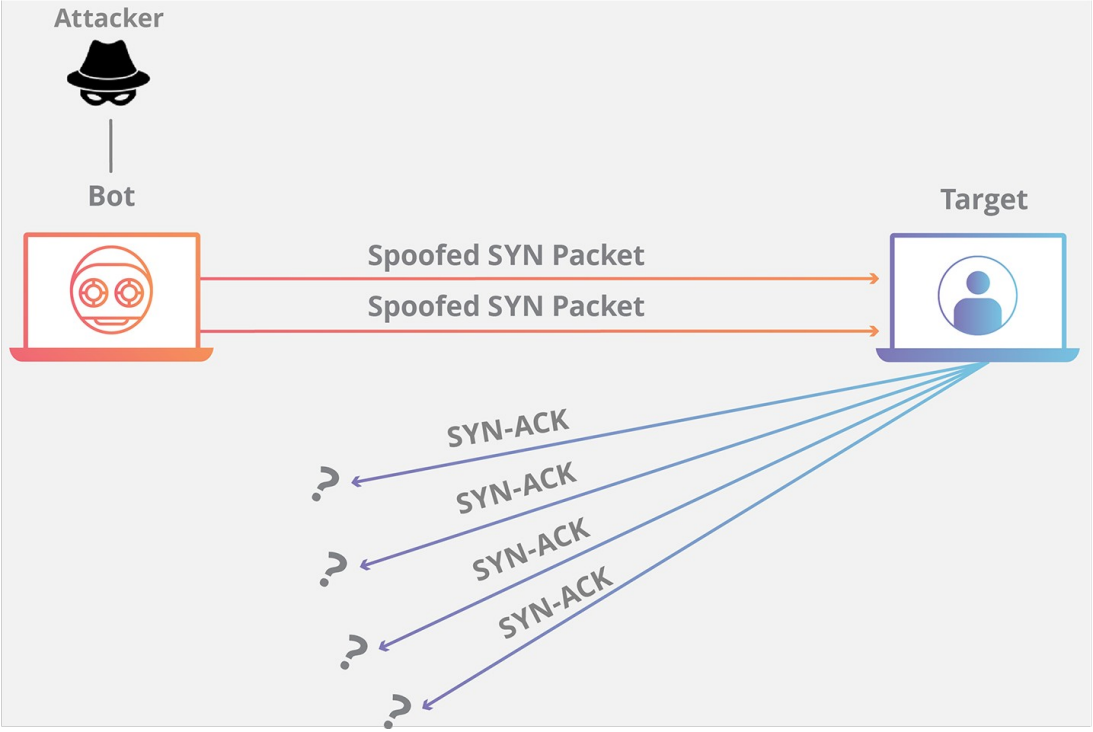
SYN flood attack

To create denial-of-service, an attacker exploits the fact that after an initial SYN packet has been received, the server will respond back with one or more SYN/ACK packets and wait for the final step in the handshake. Steps:

- The attacker sends a high volume of SYN packets to the targeted server, often with spoofed IP addresses.
- The server then responds to each one of the connection requests and leaves an open port ready to receive the response.
- While the server waits for the final ACK packet, which never arrives, the attacker continues to send more SYN packets. The arrival of each new SYN packet causes the server to temporarily maintain a new open port connection for a certain length of time, and once all the available ports have been utilized the server is unable to function normally.

SYN flood attack

In networking, when a server is leaving a connection open but the machine on the other side of the connection is not, the connection is considered half-open. In this type of DDoS attack, the targeted server is continuously leaving open connections and waiting for each connection to timeout before the ports become available again. The result is that this type of attack can be considered a “half-open attack”.



A SYN flood can occur in three different ways:

- **Direct attack**: A SYN flood where the IP address is not spoofed is known as a direct attack. In this attack, the attacker does not mask their IP address at all. As a result of the attacker using a single source device with a real IP address to create the attack, the attacker is highly vulnerable to discovery and mitigation. In order to create the half-open state on the targeted machine, the hacker prevents their machine from responding to the server's SYN-ACK packets. This is often achieved by firewall rules that stop outgoing packets other than SYN packets or by filtering out any incoming SYN-ACK packets before they reach the malicious user's machine. In practice this method is used rarely (if ever), as mitigation is fairly straightforward – just block the IP address of each malicious system. If the attacker is using a botnet such as the Mirai botnet they won't care about masking the IP of the infected device.
- **Spoofed Attack**: A malicious user can also spoof the IP address on each SYN packet they send in order to inhibit mitigation efforts and make their identity more difficult to discover. While the packets may be spoofed, those packets can potentially be traced back to their source. It's difficult to do this sort of detective work but it's not impossible, especially if Internet service providers (ISPs) are willing to help.
- **Distributed attack (DDoS)**: If an attack is created using a botnet the likelihood of tracking the attack back to its source is low. For an added level of obfuscation, an attacker may have each distributed device also spoof the IP addresses from which it sends packets. If the attacker is using a botnet such as the Mirai botnet, they generally won't care about masking the IP of the infected device.

Denial-of-service with SYN flooding attacks

By using a SYN flood attack, a bad actor can attempt to

- create denial-of-service in a target device or service with substantially less traffic than other DDoS attacks.
- Instead of volumetric attacks, which aim to saturate the network infrastructure surrounding the target, SYN attacks only need to be larger than the available backlog in the target's operating system.
- If the attacker is able to determine the size of the backlog and how long each connection will be left open before timing out, the attacker can target the exact parameters needed to disable the system, thereby reducing the total traffic to the minimum necessary amount to create denial-of-service.

How is a SYN flood attack mitigated?

Increasing Backlog queue

Each operating system on a targeted device has a certain number of half-open connections that it will allow. One response to high volumes of SYN packets is to increase the maximum number of possible half-open connections the operating system will allow. In order to successfully increase the maximum backlog, the system must reserve additional memory resources to deal with all the new requests. If the system does not have enough memory to be able to handle the increased backlog queue size, system performance will be negatively impacted, but that still may be better than denial-of-service.

Recycling the Oldest Half-Open TCP connection

Another mitigation strategy involves overwriting the oldest half-open connection once the backlog has been filled. This strategy requires that the legitimate connections can be fully established in less time than the backlog can be filled with malicious SYN packets. This particular defense fails when the attack volume is increased, or if the backlog size is too small to be practical.

SYN cookies

This strategy involves the creation of a cookie by the server. In order to avoid the risk of dropping connections when the backlog has been filled, the server responds to each connection request with a SYN-ACK packet but then drops the SYN request from the backlog, removing the request from memory and leaving the port open and ready to make a new connection. If the connection is a legitimate request, and a final ACK packet is sent from the client machine back to the server, the server will then reconstruct (with some limitations) the SYN backlog queue entry. While this mitigation effort does lose some information about the TCP connection, it is better than allowing denial-of-service to occur to legitimate users as a result of an attack.

Ping (ICMP) Flood DDoS Attack

A ping flood is a denial-of-service attack in which the attacker attempts to overwhelm a targeted device with ICMP echo-request packets, causing the target to become inaccessible to normal traffic. When the attack traffic comes from multiple devices, the attack becomes a DDoS or distributed denial-of-service attack.

How does a Ping flood attack work?

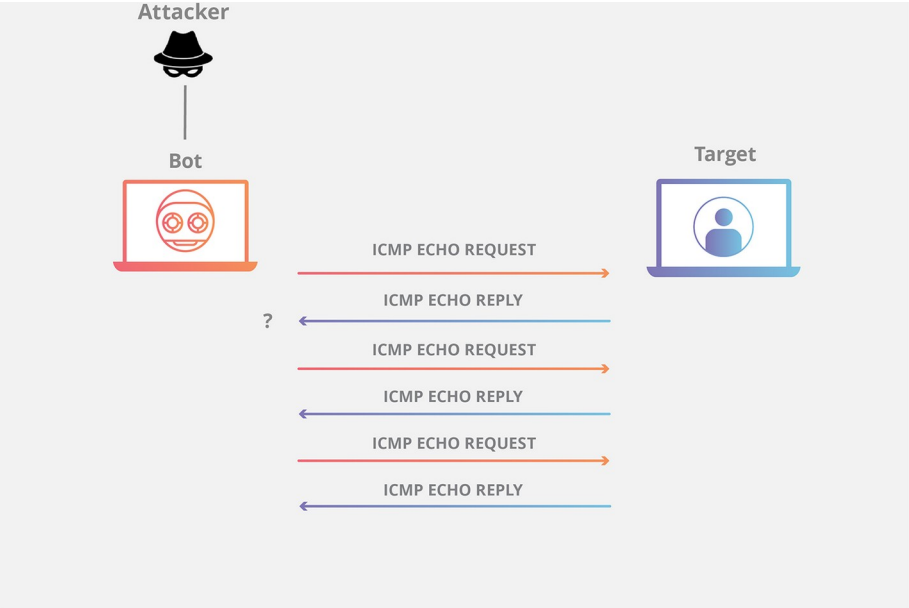
The Internet Control Message Protocol (ICMP), which is utilized in a Ping Flood attack, is an internet layer protocol used by network devices to communicate. The network diagnostic tools traceroute and ping both operate using ICMP. Commonly, ICMP echo-request and echo-reply messages are used to ping a network device for the purpose of diagnosing the health and connectivity of the device and the connection between the sender and the device.

An ICMP request requires some server resources to process each request and to send a response. The request also requires bandwidth on both the incoming message (echo-request) and outgoing response (echo-reply). The Ping Flood attack aims to overwhelm the targeted device's ability to respond to the high number of requests and/or overload the network connection with bogus traffic. By having many devices in a botnet target the same internet property or infrastructure component with ICMP requests, the attack traffic is increased substantially, potentially resulting in a disruption of normal network activity. Historically, attackers would often spoof in a bogus IP address in order to mask the sending device. With modern botnet attacks, the malicious actors rarely see the need to mask the bot's IP, and instead rely on a large network of un-spoofed bots to saturate a target's capacity.

Ping (ICMP) Flood DDoS Attack

The DDoS form of a Ping (ICMP) Flood can be broken down into 2 repeating steps:

- The attacker sends many ICMP echo request packets to the targeted server using multiple devices.
- The targeted server then sends an ICMP echo reply packet to each requesting device's IP address as a response.



- The damaging effect of a Ping Flood is directly proportional to the number of requests made to the targeted server. Unlike reflection-based DDoS attacks like NTP amplification and DNS amplification, Ping Flood attack traffic is symmetrical; the amount of bandwidth the targeted device receives is simply the sum of the total traffic sent from each bot.

Ping (ICMP) Flood DDoS Attack

How is a Ping flood attack mitigated?

Disabling a ping flood is most easily accomplished by disabling the ICMP functionality of the targeted router, computer or other device. A network administrator can access the administrative interface of the device and disable its ability to send and receive any requests using the ICMP, effectively eliminating both the processing of the request and the Echo Reply. The consequence of this is that all network activities that involve ICMP are disabled, making the device unresponsive to ping requests, traceroute requests, and other network activities.

How does Cloudflare mitigate Ping Flood attacks?

Cloudflare mitigates this type of attack in part by standing between the targeted origin server and the Ping flood. When each ping request is made, Cloudflare handles the processing and response process of the ICMP echo request and reply on our network edge. This strategy takes the resource cost of both bandwidth and processing power off the targeted server and places it on Cloudflare’s Anycast network.

Smurf DDoS Attack

What is a Smurf attack?

A Smurf attack is a distributed denial-of-service (DDoS) attack in which an attacker attempts to flood a targeted server with Internet Control Message Protocol (ICMP) packets. By making requests with the spoofed IP address of the targeted device to one or more computer networks, the computer networks then respond to the targeted server, amplifying the initial attack traffic and potentially overwhelming the target, rendering it inaccessible. This attack vector is generally considered a solved vulnerability and is no longer prevalent.

How does a Smurf attack work?

While ICMP packets can be utilized in a DDoS attack, normally they serve valuable functions in network administration. The ping application, which utilizes ICMP packets, is used by network administrators to test networked hardware devices such as computers, printers or routers. A ping is commonly used to see if a device is operational, and to track the amount of time it takes for the message to go round trip from the source device to the target and back to the source. Unfortunately, because the ICMP protocol does not include a handshake, hardware devices receiving requests are unable to verify if the request is legitimate.

This type of DDoS attack can be thought of metaphorically as a prankster calling an office manager and pretending to be the company's CEO. The prankster asks the manager to tell each employee to call the executive back on his private number and give him an update on how they're doing. The prankster gives the callback number of a targeted victim, who then receives as many unwanted phone calls as there are people in the office.

Smurf DDoS Attack

Here's How a Smurf attack works:

- First the Smurf malware builds a spoofed packet that has its source address set to the real IP address of the targeted victim.
- The packet is then sent to an IP broadcast address of a router or firewall, which in turn sends requests to every host device address inside the broadcasting network, increasing the number of requests by the number of networked devices on the network.
- Each device inside the network receives the request from the broadcaster and then responds to the spoofed address of the target with an ICMP Echo Reply packet.
- The target victim then receives a deluge of ICMP Echo Reply packets, potentially becoming overwhelmed and resulting in denial-of-service to legitimate traffic.

Smurf DDoS Attack

How can a Smurf attack be mitigated?

Several mitigation strategies for this attack vector have been developed and implemented over the years, and the exploit is largely considered solved. On a limited number of legacy systems, mitigation techniques may still need to be applied. A simple solution is to disable IP broadcasting addresses at each network router and firewall. Older routers are likely to enable broadcasting by default, while newer routers will likely already have it disabled. In the event that a Smurf attack occurs, Cloudflare eliminates the attack traffic by preventing the ICMP packets from reaching the targeted origin server. Learn more about how Cloudflare's DDoS Protection works.

Ping of Death DDoS attack

What is a Ping of Death attack?

A Ping of Death attack is a denial-of-service (DoS) attack, in which the attacker aims to disrupt a targeted machine by sending a packet larger than the maximum allowable size, causing the target machine to freeze or crash. The original Ping of Death attack is less common today. A related attack known as an ICMP flood attack is more prevalent.

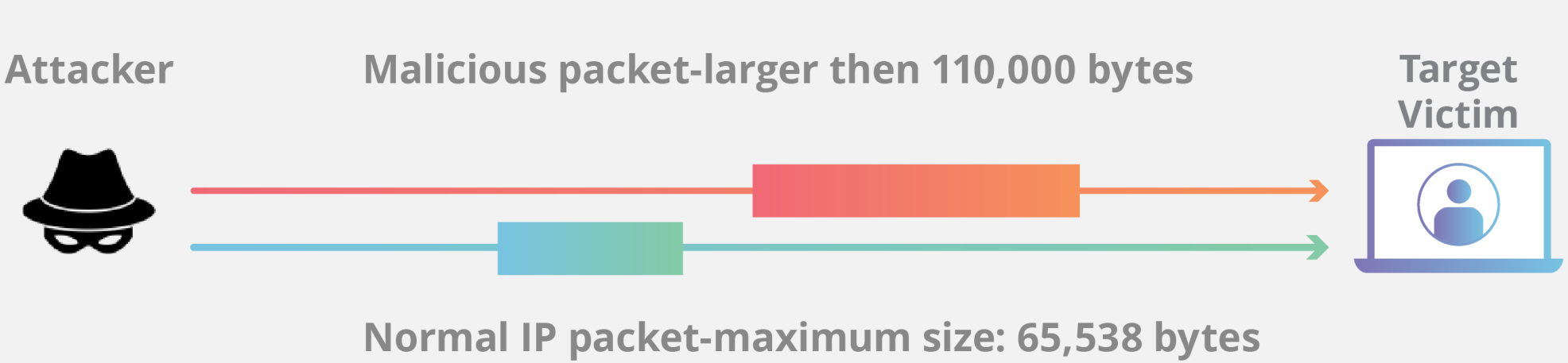
How does a Ping of Death work?

An Internet Control Message Protocol (ICMP) echo-reply message or “ping”, is a network utility used to test a network connection, and it works much like sonar – a “pulse” is sent out and the “echo” from that pulse tells the operator information about the environment. If the connection is working, the source machine receives a reply from the targeted machine.

While some ping packets are very small, IP4 ping packets are much larger, and can be as large as the maximum allowable packet size of 65,535 bytes. Some TCP/IP systems were never designed to handle packets larger than the maximum, making them vulnerable to packets above that size.

Ping of Death DDoS attack

- When a maliciously large packet is transmitted from the attacker to the target, the packet becomes fragmented into segments, each of which is below the maximum size limit. When the target machine attempts to put the pieces back together, the total exceeds the size limit and a buffer overflow can occur, causing the target machine to freeze, crash or reboot.
- While ICMP echo can be used for this attack, anything that sends an IP datagram can be used for this exploit. That includes TCP, UDP and IPX transmissions.



Ping of Death DDoS attack

How is a Ping of Death DDoS attack mitigated?

One solution to stop an attack is to add checks to the reassembly process to make sure the maximum packet size constraint will not be exceeded after packet recombination. Another solution is to create a memory buffer with enough space to handle packets which exceed the guideline maximum.

The original Ping of Death attack has mostly gone the way of the dinosaurs; devices created after 1998 are generally protected against this type of attack. Some legacy equipment may still be vulnerable. A new Ping of Death attack for IPv6 packets for Microsoft Windows was discovered more recently, and it was patched in mid 2013. Cloudflare DDoS Protection mitigates Ping of Death attacks by dropping malformed packets before they reach the targeted host computer.

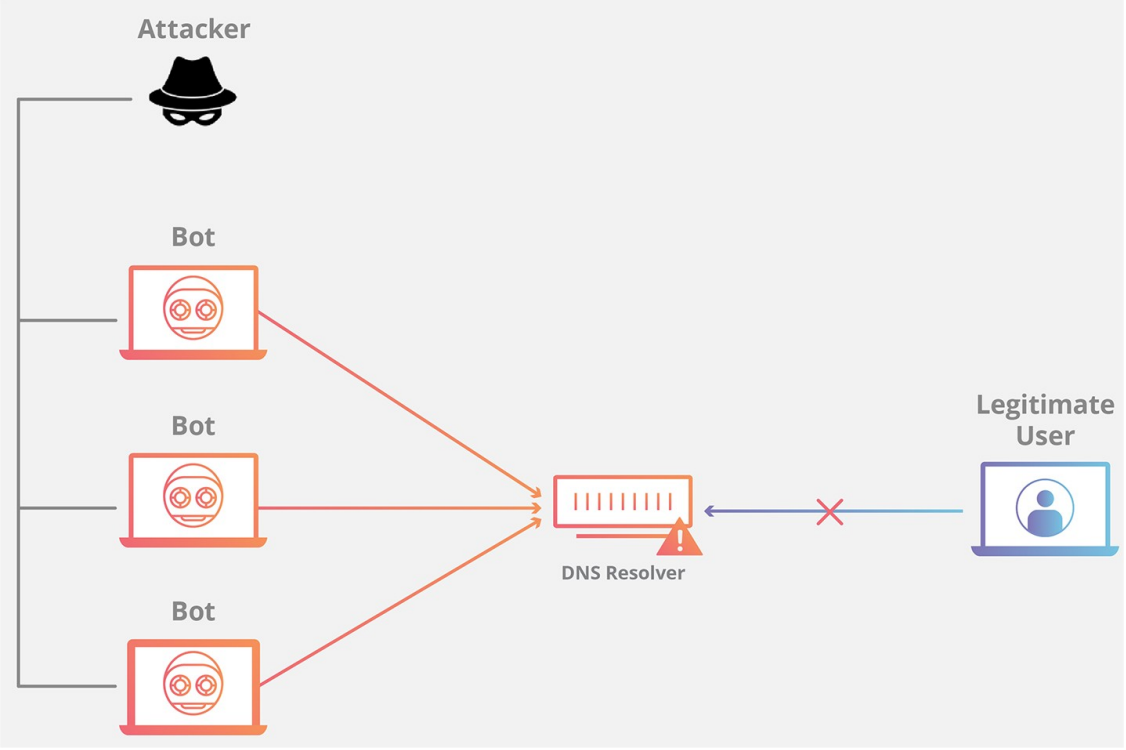
DNS Flood DDoS Attack

Domain Name System (DNS) servers are the “phonebooks” of the Internet; they are the path through which Internet devices are able to lookup specific web servers in order to access Internet content. A DNS flood is a type of distributed denial-of-service attack (DDoS) where an attacker floods a particular domain’s DNS servers in an attempt to disrupt DNS resolution for that domain. If a user is unable to find the phonebook, it cannot lookup the address in order to make the call for a particular resource. By disrupting DNS resolution, a DNS flood attack will compromise a website, API, or web application's ability respond to legitimate traffic. DNS flood attacks can be difficult to distinguish from normal heavy traffic because the large volume of traffic often comes from a multitude of unique locations, querying for real records on the domain, mimicking legitimate traffic.

How does a DNS flood attack work?

The function of the Domain Name System is to translate between easy to remember names (e.g. example.com) and hard to remember addresses of website servers (e.g. 192.168.0.1), so successfully attacking DNS infrastructure makes the Internet unusable for most people. DNS flood attacks constitute a relatively new type of DNS-based attack that has proliferated with the rise of high bandwidth Internet of Things (IoT) botnets like Mirai.

DNS flood attacks use the high bandwidth connections of IP cameras, DVR boxes and other IoT devices to directly overwhelm the DNS servers of major providers. The volume of requests from IoT devices overwhelms the DNS provider's services and prevents legitimate users from accessing the provider's DNS servers.



How does a DNS flood attack work?

DNS flood attacks differ from DNS amplification attacks. Unlike DNS floods, DNS amplification attacks reflect and amplify traffic off unsecured DNS servers in order to hide the origin of the attack and increase its effectiveness. DNS amplification attacks use devices with smaller bandwidth connections to make numerous requests to unsecured DNS servers. The devices make many small requests for very large DNS records, but when making the requests, the attacker forges the return address to be that of the intended victim. The amplification allows the attacker to take out larger targets with only limited attack resources.

How can a DNS Flood attack be mitigated?

DNS floods represent a change from traditional amplification-based attack methods. With easily accessible high bandwidth botnets, attackers can now target large organizations. Until compromised IoT devices can be updated or replaced, the only way to withstand these types of attacks is to use a very large and highly distributed DNS system that can monitor, absorb, and block the attack traffic in realtime. Learn about how Cloudflare's DDoS Protection protects against DNS flood attacks.

UDP Flood Attack

What is a UDP flood attack?

A UDP flood is a type of denial-of-service attack in which a large number of User Datagram Protocol (UDP) packets are sent to a targeted server with the aim of overwhelming that device's ability to process and respond. The firewall protecting the targeted server can also become exhausted as a result of UDP flooding, resulting in a denial-of-service to legitimate traffic.

How does a UDP flood attack work?

A UDP flood works primarily by exploiting the steps that a server takes when it responds to a UDP packet sent to one of it's ports. Under normal conditions, when a server receives a UDP packet at a particular port, it goes through two steps in response:

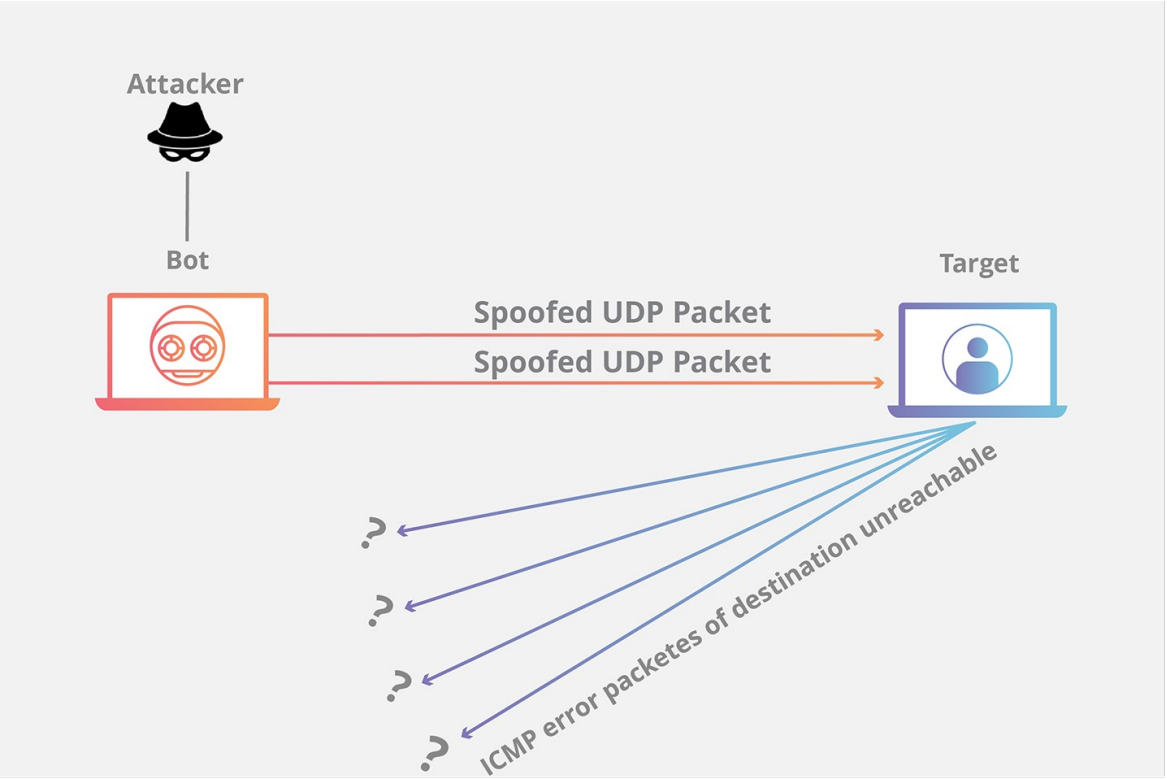
- The server first checks to see if any programs are running which are presently listening for requests at the specified port.
- If no programs are receiving packets at that port, the server responds with a ICMP (ping) packet to inform the sender that the destination was unreachable.

A UDP flood can be thought of in the context of a hotel receptionist routing calls. First, the receptionist receives a phone call where the caller asks to be connected to a specific room. The receptionist then needs to look through the list of all rooms to make sure that the guest is available in the room and willing to take the call. Once the receptionist realizes that the guest is not taking any calls, they have to pick the phone back up and tell the caller that the guest will not be taking the call. If suddenly all the phone lines light up simultaneously with similar requests then they will quickly become overwhelmed.

UDP Flood Attack

As each new UDP packet is received by the server, it goes through steps in order to process the request, utilizing server resources in the process. When UDP packets are transmitted, each packet will include the IP address of the source device. During this type of DDoS attack, an attacker will generally not use their own real IP address, but will instead spoof the source IP address of the UDP packets, impeding the attacker's true location from being exposed and potentially saturated with the response packets from the targeted server.

As a result of the targeted server utilizing resources to check and then respond to each received UDP packet, the target's resources can become quickly exhausted when a large flood of UDP packets are received, resulting in denial-of-service to normal traffic.



UDP Flood Attack

How is a UDP flood attack mitigated?

Most operating systems limit the response rate of ICMP packets in part to disrupt DDoS attacks that require ICMP response. One drawback of this type of mitigation is that during an attack legitimate packets may also be filtered in the process. If the UDP flood has a volume high enough to saturate the state table of the targeted server's firewall, any mitigation that occurs at the server level will be insufficient as the bottleneck will occur upstream from the targeted device.

How does Cloudflare mitigate UDP Flood attacks?

In order to mitigate UDP attack traffic before it reaches its target, Cloudflare drops all UDP traffic not related to DNS at the network edge. Because Cloudflare's Anycast network scatters web traffic across many Data Centers, we have sufficient capacity to handle UDP flood attacks of any size. Learn more about Cloudflare DDoS Protection.

DNS amplification attack

What is a DNS amplification attack?

This DDoS attack is a reflection-based volumetric distributed denial-of-service (DDoS) attack in which an attacker leverages the functionality of open DNS resolvers in order to overwhelm a target server or network with an amplified amount of traffic, rendering the server and its surrounding infrastructure inaccessible.

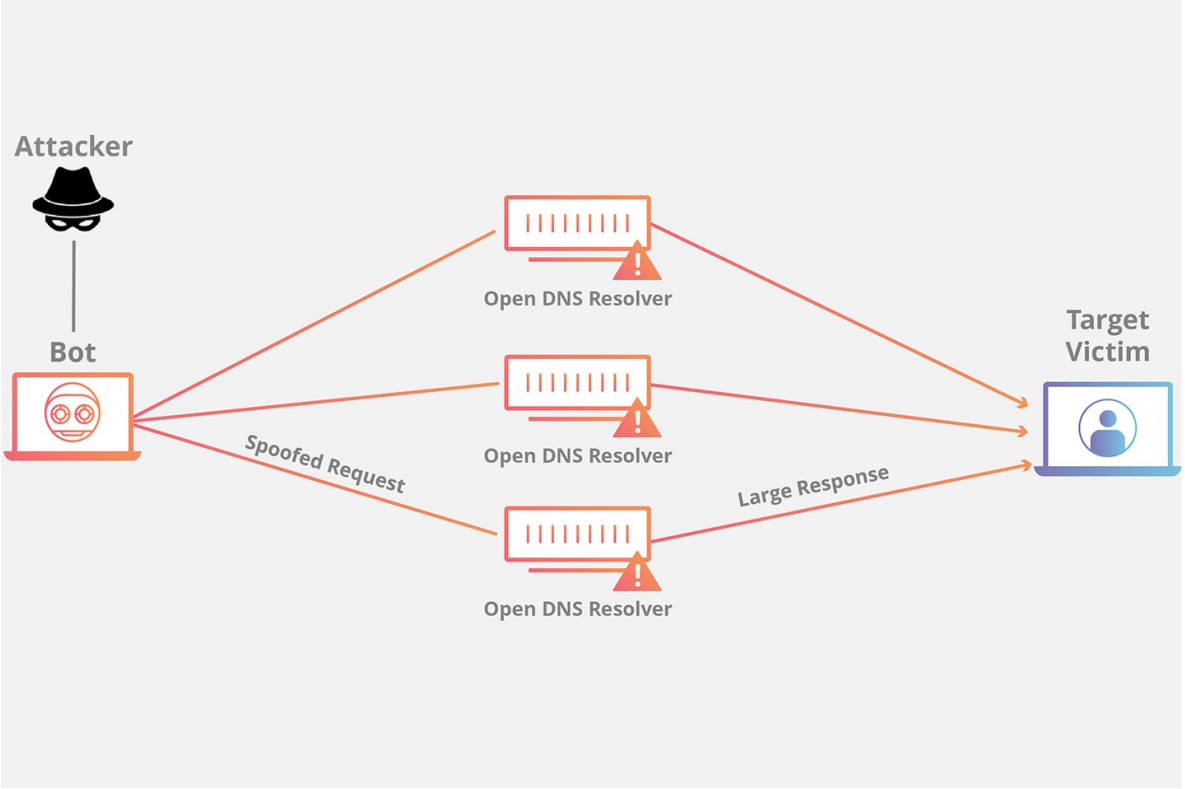
How does a DNS amplification attack work?

- All amplification attacks exploit a disparity in bandwidth consumption between an attacker and the targeted web resource. When the disparity in cost is magnified across many requests, the resulting volume of traffic can disrupt network infrastructure. By sending small queries that result in large responses, the malicious user is able to get more from less. By multiplying this magnification by having each bot in a botnet make similar requests, the attacker is both obfuscated from detection and reaping the benefits of greatly increased attack traffic.
- A single bot in a DNS amplification attack can be thought of in the context of a malicious teenager calling a restaurant and saying “I’ll have one of everything, please call me back and tell me my whole order.” When the restaurant asks for a callback number, the number given is the targeted victim’s phone number. The target then receives a call from the restaurant with a lot of information that they didn’t request.
- As a result of each bot making requests to open DNS resolvers with a spoofed IP address, which has been changed to the real source IP address of the targeted victim, the target then receives a response from the DNS resolvers. In order to create a large amount of traffic, the attacker structures the request in a way that generates as large a response from the DNS resolvers as possible. As a result, the target receives an amplification of the attacker’s initial traffic, and their network becomes clogged with the spurious traffic, causing a denial-of-service.

DNS amplification attack

DNS amplification can be broken down into four steps:

- The attacker uses a compromised endpoint to send UDP packets with spoofed IP addresses to a DNS recursor. The spoofed address on the packets points to the real IP address of the victim.
- Each one of the UDP packets makes a request to a DNS resolver, often passing an argument such as “ANY” in order to receive the largest response possible.
- After receiving the requests, the DNS resolver, which is trying to be helpful by responding, sends a large response to the spoofed IP address.
- The IP address of the target receives the response and the surrounding network infrastructure becomes overwhelmed with the deluge of traffic, resulting in a denial-of-service.



While a few requests is not enough to take down network infrastructure, when this sequence is multiplied across multiple requests and DNS resolvers, the amplification of data the target receives can be substantial. Explore more technical details on reflection attacks.

How is a DNS amplification attack mitigated?

How is a DNS amplification attack mitigated?

For an individual or company running a website or service, mitigation options are limited. This comes from the fact that the individual's server, while it might be the target, is not where the main effect of a volumetric attack is felt. Due to the high amount of traffic generated, the infrastructure surrounding the server feels the impact. The Internet Service Provider (ISP) or other upstream infrastructure providers may not be able to handle the incoming traffic without becoming overwhelmed. As a result, the ISP may blackhole all traffic to the targeted victim's IP address, protecting itself and taking the target's site off-line. Mitigation strategies, aside from offsite protective services like Cloudflare DDoS protection, are mostly preventative Internet infrastructure solutions.

Reduce the total number of open DNS resolvers

An essential component of DNS amplification attacks is access to open DNS resolvers. By having poorly configured DNS resolvers exposed to the Internet, all an attacker needs to do to utilize a DNS resolver is to discover it. Ideally, DNS resolvers should only provide their services to devices that originate within a trusted domain. In the case of reflection based attacks, the open DNS resolvers will respond to queries from anywhere on the Internet, allowing the potential for exploitation. Restricting a DNS resolver so that it will only respond to queries from trusted sources makes the server a poor vehicle for any type of amplification attack.

DNS amplification attack

Source IP verification – stop spoofed packets leaving network

Because the UDP requests being sent by the attacker's botnet must have a source IP address spoofed to the victim's IP address, a key component in reducing the effectiveness of UDP-based amplification attacks is for Internet service providers (ISPs) to reject any internal traffic with spoofed IP addresses. If a packet is being sent from inside the network with a source address that makes it appear like it originated outside the network, it's likely a spoofed packet and can be dropped. Cloudflare highly recommends that all providers implement ingress filtering, and at times will reach out to ISPs who are unknowingly taking part in DDoS attacks and help them realize their vulnerability.

How does Cloudflare mitigate DNS amplification attacks?

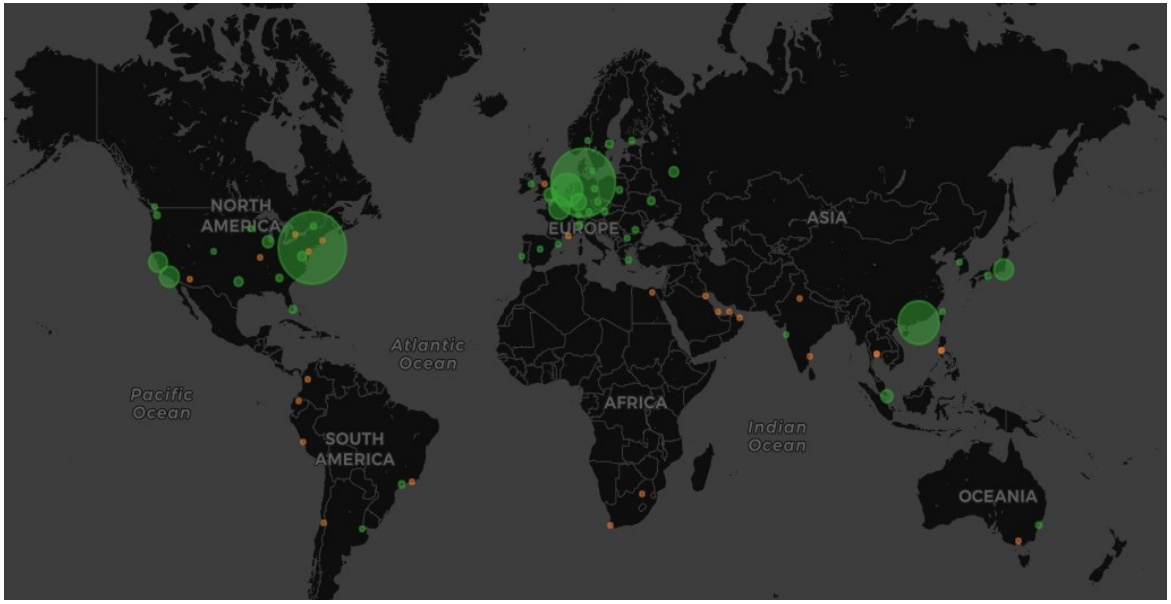
With a properly configured firewall and sufficient network capacity (which isn't always easy to come by unless you are the size of Cloudflare), it's trivial to block reflection attacks such as DNS amplification attacks. Although the attack will target a single IP address, our Anycast network will scatter all attack traffic to the point where it is no longer disruptive. Cloudflare is able to use our advantage of scale to distribute the weight of the attack across many Data Centers, balancing the load so that service is never interrupted and the attack never overwhelms the targeted server's infrastructure. During a recent six month window our DDoS mitigation system "Gatebot" detected 6,329 simple reflection attacks (that's one every 40 minutes), and the network successfully mitigated all of them.

memcached DDoS attack

What is a memcached DDoS attack?

A memcached distributed denial-of-service (DDoS) attack is a type of cyber attack in which an attacker attempts to overload a targeted victim with internet traffic. The attacker spoofs requests to a vulnerable UDP memcached* server, which then floods a targeted victim with internet traffic, potentially overwhelming the victim's resources. While the target's internet infrastructure is overloaded, new requests cannot be processed and regular traffic is unable to access the internet resource, resulting in denial-of-service.

Memcached is a database caching system for speeding up websites and networks.
Here are data centers in Cloudflare's global network and the relative amount of memcached attack traffic they received during a recent attack.



memcached DDoS attack

How does a memcached attack work?

A Memcached attacks operates similarly to all DDoS amplification attacks such as NTP amplification and DNS amplification. The attack works by sending spoofed requests to a vulnerable server, which then responds with a larger amount of data than the initial request, magnifying the volume of traffic.

Memcached amplification can be thought of in the context of a malicious teenager calling a restaurant and saying "I'll have one of everything, please call me back and tell me my whole order." When the restaurant asks for a callback number, the number given is the targeted victim's phone number. The target then receives a call from the restaurant with a lot of information that they didn't request.

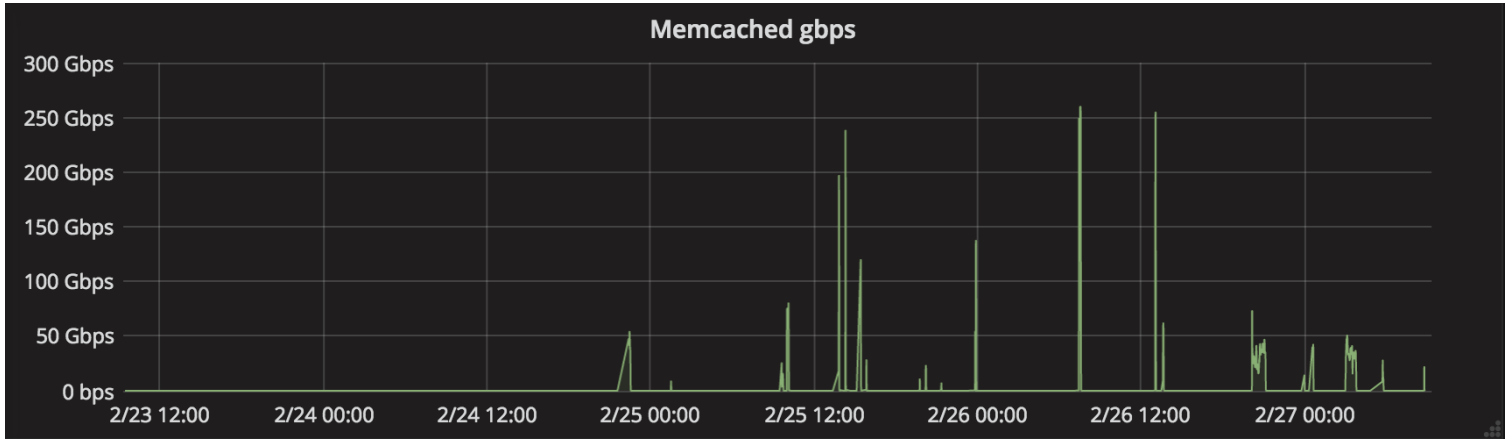
This method of amplification attack is possible because memcached servers have the option to operate using the UDP protocol. UDP is a network protocol that allows for the sending of data without first getting what's known as a handshake, which is a network process where both sides agree to the communication. UDP is utilized because the targeted host is never consulted on whether or not they're willing to receive the data, allowing for a massive amount of data to be sent to the target without their prior consent.

memcached DDoS attack

A memcached attack occurs in 4 steps:

- An attacker implants a large payload* of data on an exposed memcached server.
- Next the attacker spoofs an HTTP GET request with the IP address of the targeted victim.
- The vulnerable memcached server that receives the request, which is trying to be helpful by responding, sends a large response to the target.
- The targeted server or its surrounding infrastructure is unable to process the large amount of data sent from the memcached server, resulting in overload and denial-of-service to legitimate requests.

This is a 260 GB per second memcached attack against Cloudflare's network being mitigated



This is a 260 GB per second memcached attack against Cloudflare's network being mitigated

memcached DDoS attack

How can a memcached attack be mitigated?

- Disable UDP - For memcached servers, make sure to disable UDP support if you do not need it. By default, memcached has UDP support enabled, potentially leaving a server vulnerable.
- Firewall memcached servers - by firewalling memcached servers from the Internet, system administrators are able to use UDP for memcached if necessary without exposure.
- Prevent IP spoofing - as long as IP addresses can be spoofed, DDoS attacks can make use of the vulnerability to direct traffic to a victim's network. Preventing IP spoofing is a larger solution that cannot be implemented by any particular system administrator, and it requires transit providers to not allow any packets to leave their network that have a source IP address originating outside the network. In other words, companies such as internet service providers (ISPs) must filter traffic such that the packets that leave their network are not allowed to pretend to be from a different network somewhere else. If all major transit providers implemented this type of filtration, spoofing-based attacks would disappear overnight.
- Develop software with reduced UDP responses - another way to eliminate amplification attacks is to remove the amplification factor to any incoming request; If the response data sent as a result of a UDP request is smaller than or equal to the initial request, amplification is no longer possible.

memcached DDoS attack

How big can a memcached amplification attack be?

The magnification factor of this type of attack is truly staggering; in practice we have witnessed amplification factors of up to a whopping 51,200x! That means that for a 15 byte request, a 750 kB response can be sent. This represents a massive amplification factor and security risk to web properties that are unable to shoulder the weight of this volume of attack traffic. Having such a large amplification factor coupled with vulnerable servers makes memcached a prime use case for attackers looking to launch DDoS against various targets.

Cloudflare filters UDP traffic at our network edge, eliminating the risk posed by amplification attacks such as this one. Explore Cloudflare's advanced DDoS protection.

For a more in-depth look at Cloudflare encountering memcached attacks and specific commands and processes for mitigation, explore the blog post Memcrashed - Major amplification attacks from UDP port 11211.

TCP reset attack

- TCP reset attack, also known as "forged TCP resets", "spoofed TCP reset packets" or "TCP reset attacks", is a way to tamper and terminate the Internet connection by sending a forged TCP reset packet. This tampering technique can be used by a firewall in goodwill, or abused by a malicious attacker to interrupt Internet connections.
- The Great Firewall of China and Iranian Internet censors are known to use TCP reset attacks to interfere with and block connections, as a major method to carry out Internet censorship.
- In a stream of packets of a TCP connection, each packet contains a TCP header. Each of these headers contains a bit known as the "reset" (RST) flag. In most packets this bit is set to 0 and has no effect; however, if this bit is set to 1, it indicates to the receiving computer that the computer should immediately stop using the TCP connection; it should not send any more packets using the connection's identifying numbers, called ports, and discard any further packets it receives with headers indicating they belong to that connection. A TCP reset basically kills a TCP connection instantly.

TCP reset attacks

- When used as designed, this can be a useful tool. One common application is the scenario where a computer (computer A) crashes while a TCP connection is in progress. The computer on the other end (computer B) will continue to send TCP packets since it does not know that computer A has crashed. When computer A reboots, it will then receive packets from the old pre-crash connection.
- Computer A has no context for these packets and no way of knowing what to do with them, so it might send a TCP reset to computer B. This reset lets computer B know that the connection is no longer working. The user on computer B can now try another connection or take other action.
- It is possible for a 3rd computer to monitor the TCP packets on the connection and then send a "forged" packet containing a TCP reset to one or both endpoints. The headers in the forged packet must indicate, falsely, that it came from an endpoint, not the forger. This information includes the endpoint IP addresses and port numbers. Every field in the IP and TCP headers must be set to a convincing forged value for the fake reset to trick the endpoint into closing the TCP connection. Properly formatted forged TCP resets can be a very effective way to disrupt any TCP connection that the forger can monitor.
- TCP connections is disrupted without the consent of both the endpoints
- open source Snort software used TCP resets to disrupt suspicious connections

ACK flooding attacks

What is an ACK flood DDoS attack?

- An ACK flood attack is when an attacker attempts to overload a server with TCP ACK packets. Like other DDoS attacks, the goal of an ACK flood is to deny service to other users by slowing down or crashing the target using junk data. The targeted server has to process each ACK packet received, which uses so much computing power that it is unable to serve legitimate users.
- Imagine a caller filling up someone's voicemail box with fake messages so that voicemails from real callers cannot get through.

ACK packets in web traffic

- The TCP protocol requires that connected devices acknowledge they have received all packets in order. Suppose a user visits a webpage that hosts an image. The image is broken up into data packets and sent to the user's browser. Once the entire image arrives, the user's device sends an ACK packet to the host server to confirm that not one pixel is missing. Without this ACK packet, the host server has to send the image again.
- Since an ACK packet is any TCP packet with the ACK flag set in the header, the ACK can be part of a different message the laptop sends to the server. If the user fills out a form and submits data to the server, the laptop can make one of those packets the ACK packet for the image. It doesn't need to be a separate packet.

ACK flooding attacks

How does an ACK flood attack work ? (εκτός του TCP handshake μηχανισμό)

- ACK flood attacks target devices that need to process every packet that they receive. **Firewalls and servers** are the most likely targets for an ACK flood. Load balancers, routers, and switches are not susceptible to these attacks.
- Legitimate and illegitimate ACK packets look essentially the same, making ACK floods difficult to stop without using a content delivery network (CDN) to filter out unnecessary ACK packets. Although they look similar, packets used in an ACK DDoS attack do not contain the main part of a data packet, also known as a payload. In order to appear legitimate, they only have to include the ACK flag in the TCP header.
- ACK floods are layer 4 (transport layer) DDoS attacks

How does a SYN ACK flood attack work?

- A SYN ACK flood DDoS attack is slightly different from an ACK attack, although the basic idea is still the same: to overwhelm the target with too many packets.
- Remember how a TCP three-way handshake works: The second step in the handshake is the SYN ACK packet. Usually a server sends this SYN ACK packet in response to a SYN packet from a client device. In a SYN ACK DDoS attack, the attacker floods the target with SYN ACK packets. These packets are not part of a three-way handshake at all; their only purpose is to disrupt the target's normal operations.

ACK flooding attacks

How does Cloudflare stop ACK flood DDoS attacks?

- The Cloudflare CDN proxies all traffic to and from a Cloudflare customer's origin server. The CDN does not pass along any ACK packets that are not associated with an open TCP connection. This ensures that the malicious ACK traffic does not reach the origin server. The Cloudflare network of data centers is large enough to absorb DDoS attacks of almost any size, so ACK floods have little to no effect on Cloudflare as well.
- Cloudflare Magic Transit and Cloudflare Spectrum also stop these kinds of DDoS attacks. Magic Transit proxies layer 3 traffic and Spectrum proxies layer 4 traffic, instead of layer 7 traffic like the CDN. Both products block ACK floods by automatically detecting attack patterns and blocking attack traffic.

Major attacks took place

What was the largest DDoS attack of all time?

The biggest DDoS attack to date took place in February of 2020. At its peak, this attack saw incoming traffic at a rate of 2.3 terabits per second (Tbps). AWS reported that it mitigated this massive attack but did not disclose which customer was targeted by the attack.

The attackers responsible used hijacked Connection-less Lightweight Directory Access Protocol (CLDAP) web servers. CLDAP is a protocol for user directories. It is an alternative to LDAP, an older version of the protocol. CLDAP has been used in multiple DDoS attacks in recent years.

The February 2018 GitHub DDoS attack

Prior to this 2.3 Tbps attack, the largest verifiable DDoS attack on record targeted GitHub, a popular online code management service used by millions of developers. This attack reached 1.3 Tbps, sending packets at a rate of 126.9 million per second.

The GitHub attack was a memcached DDoS attack, so there were no botnets involved. Instead the attackers leveraged the amplification effect of a popular database caching system known as memcached. By flooding memcached servers with spoofed requests, the attackers were able to amplify their attack by a magnitude of about 50,000x.

Luckily, GitHub was using a DDoS protection service, which was automatically alerted within 10 minutes of the start of the attack. This alert triggered the process of mitigation and GitHub was able to stop the attack quickly. The massive DDoS attack only ended up lasting about 20 minutes.

It should also be noted that there was an alleged 1.7 Tbps DDoS attack 5 days after the attack on GitHub. However the victim of this attack was never publicly disclosed and there was not very much information released about it, making it difficult to verify.

Major attacks took place

The 2016 Dyn attack

Another massive DDoS attack was directed at Dyn, a major DNS provider, in October of 2016. This attack was devastating and created disruption for many major sites, including Airbnb, Netflix, PayPal, Visa, Amazon, The New York Times, Reddit, and GitHub. This was done using malware called Mirai. Mirai creates a botnet out of compromised Internet of Things (IoT) devices such as cameras, smart TVs, radios, printers, and even baby monitors. To create the attack traffic, these compromised devices are all programmed to send requests to a single victim.

Fortunately Dyn was able to resolve the attack within one day, but the motive for the attack was never discovered. Hacktivist groups claimed responsibility for the attack as a response to WikiLeaks founder Julian Assange being denied Internet access in Ecuador, but there was no proof to back up this claim. There are also suspicions that the attack was carried out by a disgruntled gamer.

The 2015 GitHub attack

The largest DDoS attack ever at the time, this one also happened to target GitHub. This politically motivated attack lasted several days and adapted itself around implemented DDoS mitigation strategies. The DDoS traffic originated in China and specifically targeted the URLs of two GitHub projects aimed at circumventing Chinese state censorship. It is speculated that the intent of the attack was to try and pressure GitHub into eliminating those projects.

The attack traffic was created by injecting JavaScript code into the browsers of everyone who visited Baidu, China's most popular search engine. Other sites who were using Baidu's analytics services were also injecting the malicious code; this code was causing the infected browsers to send HTTP requests to the targeted GitHub pages. In the aftermath of the attack it was determined that the malicious code was not originating from Baidu, but rather being added by an intermediary service.

Major attacks took place

The 2013 Spamhaus attack

- Another largest-ever-at-the-time attack was the 2013 attack directed at Spamhaus, an organization that helps combat spam emails and spam-related activity. Spamhaus is responsible for filtering as much as 80% of all spam, which makes them a popular target for people who would like to see spam emails reach their intended recipients.
- The attack drove traffic to Spamhaus at a rate of 300 Gbps. Once the attack began, Spamhaus signed up for Cloudflare. Cloudflare's DDoS protection mitigated the attack. The attackers responded to this by going after certain Internet exchanges and bandwidth providers in an attempt to bring down Cloudflare. This attack did not achieve its goal, but it did cause major issues for LINX, the London Internet exchange. The main culprit of the attack turned out to be a teenage hacker-for-hire in Britain who was paid to launch this DDoS attack.

The 2000 Mafiaboy attack

- In 2000, a 15-year-old hacker known as 'Mafiaboy' took down several major websites including CNN, Dell, E-Trade, eBay, and Yahoo!, the last of which at the time was the most popular search engine in the world. This attack had devastating consequences, including creating chaos in the stock market.
- Mafiaboy, who was later revealed to be a high schooler named Michael Calce, coordinated the attack by compromising the networks of several universities and using their servers to conduct the DDoS attack. The aftermath of this attack directly led to the creation of many of today's cybercrime laws.

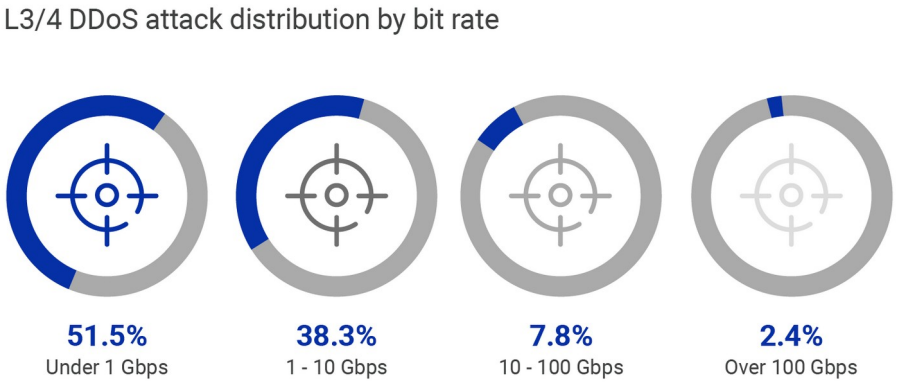
The 2007 Estonia attack

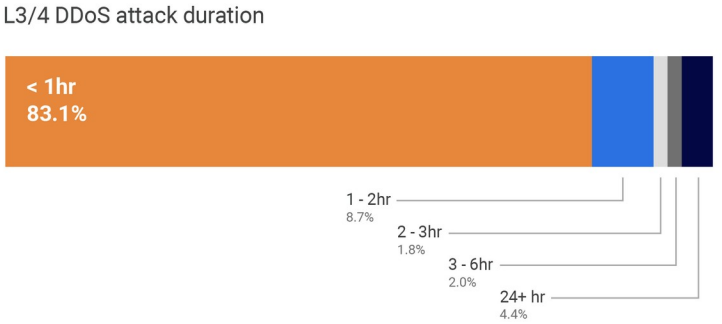
In April 2007 the nation of Estonia was hit with a massive DDoS attack targeting government services, financial institutions, and media outlets. This had a crushing effect since Estonia’s government was an early adopter of online government and was practically paperless at the time; even national elections were conducted online.

The attack, considered by many to be the first act of cyber warfare, came in response to a political conflict with Russia

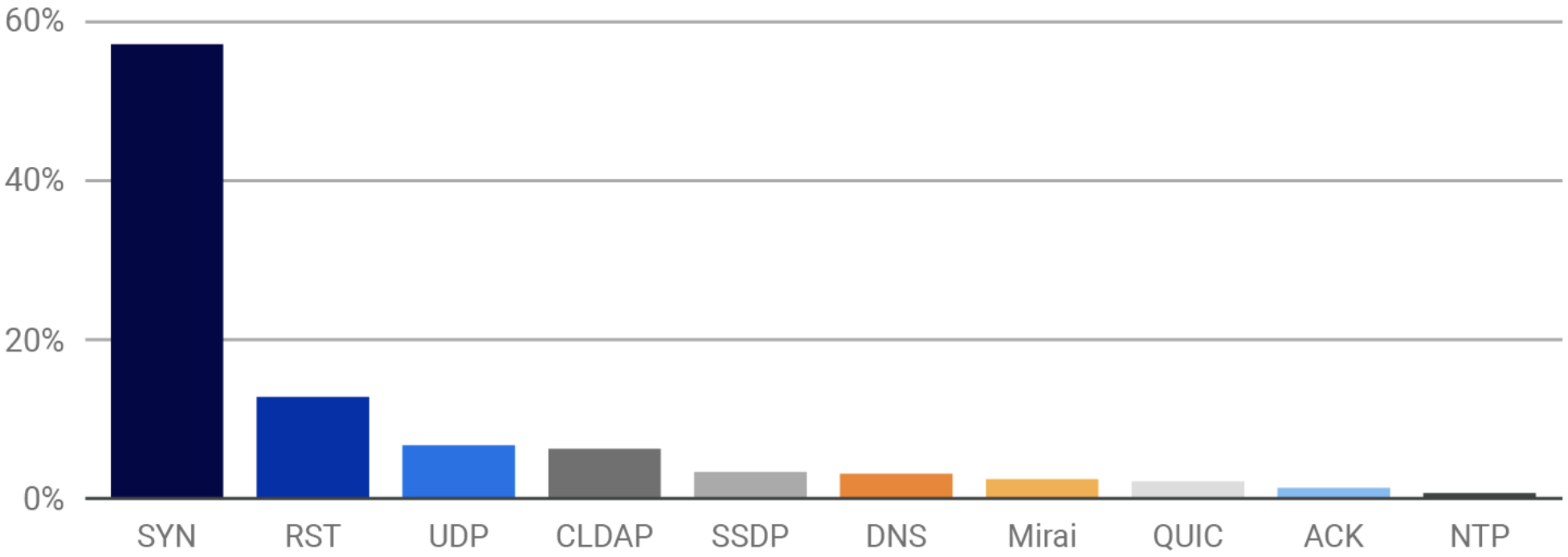
Cloudflare has mitigated in terms of volume 942 Gbps traffic, and features 42 Tbps of network capacity,

Cloudflare's research shows that most DDoS attacks do not exceed 10 Gbps.





L3/4 DDoS attack vector distribution

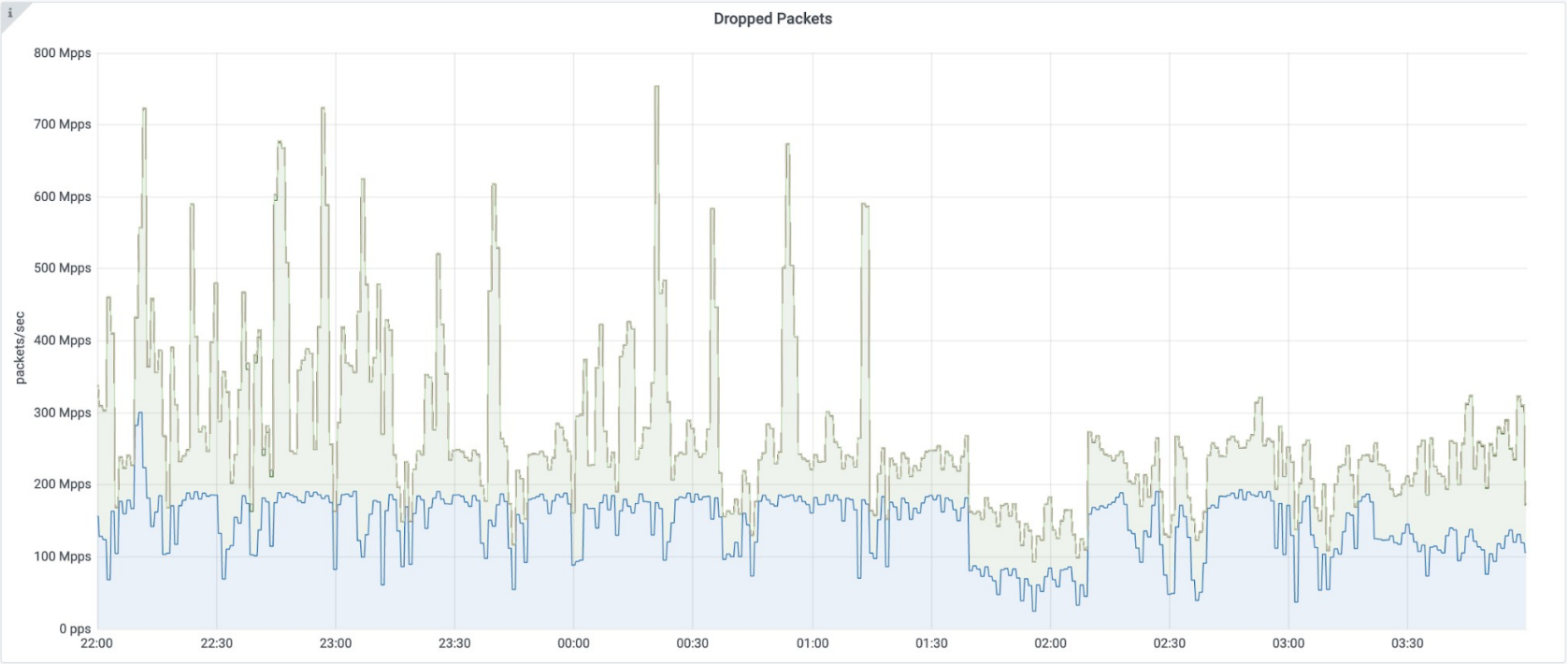


CLOUDFLARE

Cloudflare attacks

Mitigating a 754 Million PPS DDoS Attack Automatically

- attack traffic was sent from over 316,000 IP addresses towards a single Cloudflare IP address that was mostly used for websites on our Free plan



Cloudfare Capacity

- Attack traffic was sent from over 316,000 IP addresses towards a single Cloudflare IP address
- The attack utilized a combination of three attack vectors over the TCP protocol: SYN floods, ACK floods and SYN-ACK floods.
- The attack campaign sustained for multiple hours at rates exceeding 400-600 million packets per second and peaked multiple times above 700 million packets per second, with a top peak of 754 million packets per second
- The attack peaked at a mere 250 Gbps but howerver Cloudfata global capacity exceeds 37 Tbps.
- Cloudflare’s network utilizes BGP Anycast to spread attack traffic globally across our fleet of data centers.
- DDoS protection systems, **Gatebot** and **dosd**, which drop packets inside the Linux kernel for maximum efficiency in order to handle massive floods of packets.
- Custom built L4 load-balancer, Unimog, which uses our appliances' health and other various metrics to load-balance traffic intelligently within a data center.
- **Gatebot** asynchronously samples traffic from every one of our data centers in over 200 locations around the world. It also monitors our customers’ origin server health.
- It then analyzes the samples to identify patterns and traffic anomalies that can indicate attacks. Once an attack is detected, Gatebot sends mitigation instructions to the edge data centers.