DDOS

DOS (Denial of Service) is an attack that prevents legitimate users from accessing a resource, like a **website, email, network**, etc. **Distributed Denial of Service (DDoS)** is a type of **DoS attack** which is carried out by a group of compromised machines that all target the same victim. It floods the computer network with data packets.

There are three types of DDoS Attacks:

1. Volume-based attacks
2. Application layer attacks
3. Protocol attacks

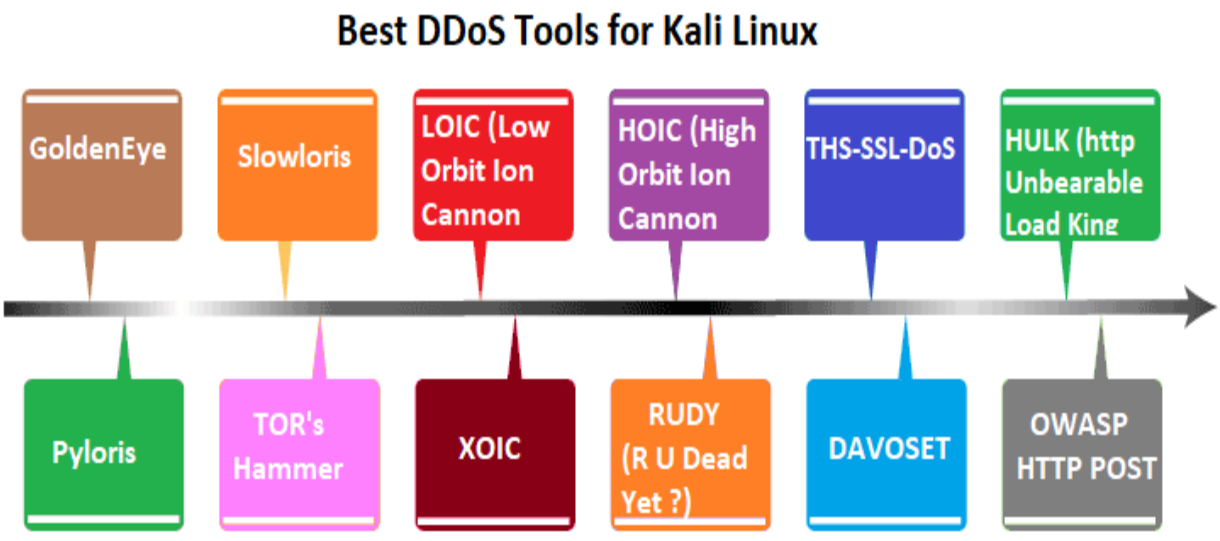
A **distributed denial-of-service** attack against a target server can be created using a variety of **DDoS** attack tools. In this tutorial, we will discuss various types of DDoS tools for Kali Linux.

Purpose of DDoS Attack

Usually, the purpose of a DDoS attack is to crash the website. The duration of a DDoS attack is determined by whether the attack is on the **network layer** or the **application layer. Network layer** attacks last for **48 to 49** hours at the most. **Application layer** attack last for **60 to 70** days at most.

The DDoS or any other attack like this attack is illegal as per the **Computer Misuse act 1990**. Since it is illegal, an attacker could face a punishment of imprisonment.

Best DDoS Tools for Kali Linux



The following is the list of Best DDoS Tools for Kali Linux:

1. GoldenEye
2. Slowloris
3. LOIC (Low Orbit Ion Cannon)
4. HOIC (High Orbit Ion Cannon)
5. THC-SSL-DoS
6. HULK (http Unbearable Load King)
7. Pyloris
8. TOR's Hammer
9. XOIC
10. RUDY (R U Dead Yet ?)
11. DAVOSET
12. OWASP HTTP POST

1. GoldenEye

In Kali Linux, **GoldenEye** is a free and open-source tool that is available on **GitHub**. With the help of this tool, we can perform a **denial-of-service** attack. The framework of this tool is written in **.NET Core**. This tool comes with a lot of **base classes** and **extensions** that we can use in our regular work. This tool allows a single machine to take down another web server of the machine by using totally legal **HTTP** traffic. It establishes a full **TCP** connection and then needs only a few hundred requests at long-term and consistent intervals. As a result, the tool does not require a large amount of traffic to exhaust the server's available connections.

Features of GoldenEye

The following are the features of the GoldenEye:

- GoldenEye is an **open-source** tool; as a result, we can download it from **GitHub** at no cost.
- GoldenEye can be used to carry out a denial-of-service attack by creating a large amount of botnet traffic.
- GoldenEye uses fully legitimate **HTTP**
- With the help of this tool, we can perform DDoS attacks on any webserver.
- GoldenEye sends numerous requests to the target, resulting in generating heavy traffic botnets.

2. Slowloris

The most effective tool for initiating a dos attack is **slowloris**. It operates by establishing numerous connections to the targeted web server and maintaining them open as long as possible. It accomplishes this

by repeatedly sending incomplete **HTTP** requests that are never completed. The attacked server continues to open connections and open more as they wait for each of the attack requests to be completed.

Because of the attack's simple yet elegant form, it uses very little bandwidth and exclusively impacts the target server's web server, with nearly no side effects on other services or ports.

Features of Slowloris

The following are the features of Slowloris:

- In slowloris, a perfectly legitimate **HTTP** traffic is used.
- With the help of this tool, we can perform **ddos attacks** on any **webserver**.
- As this tool is an open-source tool so, we can download it from **github** free of cost.
- Slowloris can be used to carry out a **denial-of-service** attack by creating a large amount of botnet traffic.
- Slowloris sends many requests to the target resulting in a heavy traffic botnet.

3. LOIC (Low Orbit Ion Cannon)

It is the most well-known **DoS** tool, and it has become a legend among hackers. **LOIC** was initially developed by **Praetox Technologies** in **C#**, however, it was later released into the public domain.

LOIC essentially converts a computer's network connection into a firehouse of garbage request, directed towards a target web server. One computer hardly creates **TCP, UDP**, or **HTTP** requests to overwhelm a web server on its own- garbage requests are readily disregarded, while legitimate requests for web pages are handled normally.

Features of LOIC (Low Orbit Ion Cannon)

The following are the features of LOIC (Low Orbit Ion Cannon:

- **LOIC** is a free **DDoS** attack tool that allows us to test our **network's performance.**
- It enables us to perform **stress testing** in order to ensure its stability.
- With the help of this tool, we can create a **DDoS** attack online against any website that they control.
- We can use this **DDoS** software to identify **DDoS** programs that hackers can use to attack a computer network.
- LOIC does not hide an **IP address** even if the proxy server is down.

4. HOIC (High Orbit Ion Cannon

The **High Orbit Ion Cannon (HOIC**) is a tool that can be used by an unauthenticated, remote attacker to launch **distributed denial of service (DDoS**) attacks. The High Orbit Ion Canon or HOIC is developed by the well-known group Anonymous, a hacktivist collective, in order to replace the **Low Orbit Ion Cannon (LOIC)** tool.

It works by flooding target systems with junk **HTTP GET** and **POST** requests.

The tool can open up to **256** concurrent attack sessions, bringing down the target system by sending a steady stream of junk traffic until it can process legitimate requests.

Traditional security technologies and firewalls find it more difficult to locate and block **DDoS** attacks because of **HOIC's** misleading and varied strategies.

The HOIC is a well know **DDoS** attack tool available for **Linux, Windows,** and **Linux platforms** and is free to use.

Features of HOIC (High Orbit Ion Cannon)

The following are the features of HOIC (High Orbit Ion Cannon):

- With the help of this tool, we can attack up to **256 DDoS websites** at once,
- With the help of this tool, we can control attacks with low, medium, and high settings.
- It contains a counter that we can use to measure the output.
- This DDoS machine-free tool can be run on Linux and **Mac OS.**
- We can choose the number of threads in the current attack.

5. THC-SSL-DoS

This DDoS tool (included in Kali) differs from typical **DoS** tools in that it doesn't require a lot of bandwidth and can be carried out with just one computer. It tries to take down the server by exploiting **SS**L flaws. It attacks vulnerabilities in **SSL** to bring down servers. We can easily download it from THC, but if we are using Kali, and we already have it.

6. HULK (Http Unbearable Load King)

HULK is another useful **DOS** attack tool that generates a unique request for each generated request to conceal traffic on a web server. **HULK** uses a variety of additional techniques in order to prevent attack detection via recognized patterns.

The HULK is a **Denial of Service (DoS)** tool that we can use to perform stress testing of the web servers. The **HULK DoS** tool is extremely effective since it can create a large amount of obscured and unique traffic.

The HULK tool is written in **Python** and can be run on any operating system that has Python installed, including **Linux, Windows**, and **Mac.** We can use the HULK tool to test network devices such as **switches, routers,** and **firewalls.** HULK traffic can also avoid cache engines and go straight to the server's direct resource pool. As a result, it can be quite hazardous.

Features of HULK (Http Unbearable Load King)

The following are the features of HULK are:

- With the help of this tool, we can generate unique network traffic.
- HULk can bypass the cache server.

- We can use this tool for research purposes.

7. PyLoris

PyLoirs is a network vulnerability testing software that uses a **distributed denial-of-Service (DDoS**) attack to evaluate network vulnerabilities. It allows us to control badly managed concurrent connections and handle DDoS online. This tool can perform a DoS attack on a server by using **SOCKS proxies** and **SSL connections**. **HTTP, IMAP, FTP, SMTP,** and **Telnet** are among the protocols it can target.

Features of PyLoris

The following are the features of PyLoris:

- We can use this tool to attack utilizing **HTTP request headers.**
- This tool supports **Windows, Linux**, and Mac OS.
- It offers easy to use **GUI (Graphical User Interface**).
- It has a more advanced option with a limit of **50 threads** and a total of **10 connections.**
- **PyLoris** can be run using a **Python**
- It uses the most recent codebase (collection of source code used to build a particular software system).

8. Tor's Hammer

The Tor's Hammer was designed to run across the Tor network in order to anonymize the attack and limit mitigation options. This **DDoS** online tool can be used to target web applications and a web server. It performs browser-based internet requests, which we use to load web pages.

Features of Tor's Hammer

The following are the features of **Tor's Hammer**:

- The Tor's Hammer automatically converts the **URL** into **links**.
- With the help of this tool, we can quickly link other artifacts in our project.
- It enables us to create rich text markup using Markdown (a plain text formatting syntax tool).
- It holds **HTTP POST** requests and connections for **1000** to **30000**
- Tor's Hammer uses web server resources by creating several network connections.

9. XOIC

XOIC is another DOS attack tool that has an **IP address**, a user-selected port, and a user-selected protocol. It is a **graphical user interface (GUI)** based tool that is simple to use for beginners. **XOIC**, according to its developers, is more powerful than **LOIC**. There are three attack modes available. The first mode is basis. The second mode is the normal DOS attack mode. The third one is a DoS attack mode that comprises **TCP/HTTP/UDP/ICMP** Message.

Features of XOIC

The following are the features of XOIC:

- XOIC is easy to use.
- There are three modes of XOIC:

1. Testing mode.
2. Normal DoS attack mode.
3. DoS attack with **TCP/HTTP/UDP/ICMP**

10. RUDY

RUDY stands for **R-U-Dead-Yet**. It is a free DDoS attack tool that allows us to easily carry out an online **DDoS attack**. It targets cloud applications by starvation the number of sessions on the web server.

Features of RUDY

The following are the features of RUDY:

- RUDY is a simple and easy tool.
- This tool offers an interactive console menu.
- This **DDoS** Free attack tool automatically identifies form fields for data submission.
- It automatically browsers the target DDoS website and detects embedded web forms.
- **R-U-Dead-Yet** allows us to launch an **HTTP DDoS** attack using long-form field submission.

11. DAVOSET

DAVOSET is software that allows us to launch **DDoS** attacks by abusing any website's functionality. This command-line tool makes it simple to carry out widespread **denial-of-service** attacks.

Features of DAVOSET

The following are the features of DAVOSET:

- This **DDoS** attack for free software offers a command-line interface to perform an attack.
- DAVOSET can also help us to hit attacks using XML external entities (attack against an app that pares **XML** input).
- It is one of the **DDoS tools** which offers support for cookies.

12. OWASP HTTP POST

The OWASP **(Open Web Application Security Project)** HTTP Post software allows us to test our web applications for network performance. It enables us to carry out denial service from a single DDoS machine online.

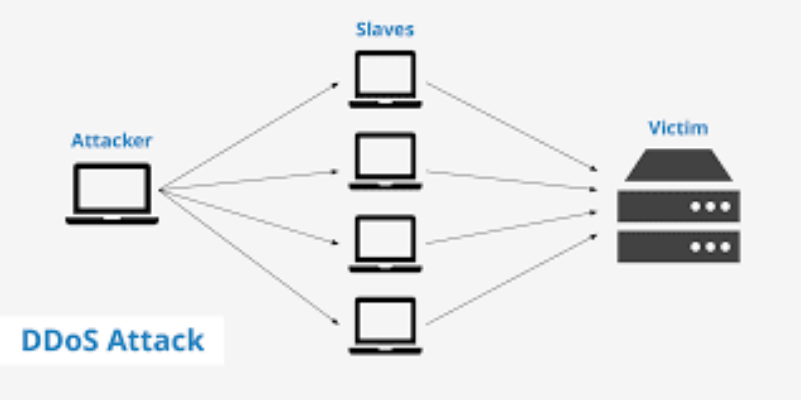
Features of OWASP HTTP POST

The following are the features of OWASP HTTP POST:

- **OWASP HTTP POST** allows us to share the result under the terms of the license.

- It enables us to share and distribute the tool with others.
- It helps us in determining the server's capacity.
- We can use this tool to test against the application layer attacks.
- This tool can be used commercially without restriction.

DENIAL OF SERVICE (DoS)



Denial-of-service (**DoS**) **attack** is an attempt to make a machine or network resource unavailable to its legitimate users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

A **distributed denial-of-service (DDoS)** is where the attack source is more than one, often thousands of, unique IP addresses. It is same to a group of people crowding the entry door or gate to a shop or business, and not letting legitimate parties enter into the shop or business, disrupting normal operations.

Outcome

- Unusually slow network performance (opening files or accessing web sites)
- Unavailability of a particular web site
- Inability to access any web site
- Dramatic increase in the number of spam emails received—this type of DoS attack is considered an e-mail bomb
- Disconnection of a wireless or wired internet connection
- Long term denial of access to the web or any internet services

If the attack is conducted on a sufficiently large scale, entire geographical regions of Internet connectivity can be compromised without the attacker's knowledge or intent by incorrectly configured or flimsy network infrastructure equipment.

Attack techniques

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. There are two general forms of DoS attacks: those that crash services and those that flood services.

The most serious attacks are distributed and in many or most cases involve forging of IP sender addresses so that the location of the attacking machines cannot easily be identified, nor can filtering be done based on the source address.

Attack tools

A wide array of programs are used to launch DoS-attacks.

In cases such as My Doom the tools are embedded in malware, and launch their attacks without the knowledge of the system owner. Stacheldraht is a classic example of a DDoS tool. It utilizes a layered structure where the attacker uses a client program to connect to handlers, which are compromised systems that issue commands to the zombie agents, which in turn facilitate the DDoS attack. Agents are compromised via the handlers by the attacker, using automated routines to exploit vulnerabilities in programs that accept remote connections running on the targeted remote hosts. Each handler can control up to a thousand agents

In other cases a machine may become part of a DDoS attack with the owner's consent, for example, in Operation Paybac,k organized by the group Anonymous. The LOIC has typically been used in this way. Along with HOIC a wide variety of DDoS tools are available today, including paid and free versions, with different features available. There is an underground market for these in hacker related forums and IRC channels.The attack using slowloris is shown below

(S)SYN flood

A SYN flood occurs when a host sends a flood of TCP/SYN packets, often with a forged sender address. Each of these packets are handled like a connection request, causing the server to spawn a half-open connection, by sending back a TCP/SYN-ACK packet (Acknowledge), and waiting for a packet in response from the sender address (response to the ACK Packet). However, because the sender address is forged, the response never comes. These half-open connections saturate the number of available connections the server can make, keeping it from responding to legitimate requests until after the attack ends.

What are DoS and DDoS attacks?

Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks are malicious attempts to disrupt the normal operations of a targeted server, service, or network by overwhelming it with a flood of Internet traffic.

DoS attacks accomplish this disruption by sending malicious traffic from a single machine — typically a computer. They can be very simple; a basic ping flood attack can be accomplished by sending more ICMP (ping) requests to a targeted server than it is able to process and respond to efficiently.

DDoS attacks, meanwhile, use more than one machine to send malicious traffic to their target. Often, these machines are part of a botnet — a collection of computers or other devices that have been infected with malware and can thus be controlled remotely by an individual attacker. In other circumstances, multiple individual attackers launch DDoS attacks by working together to send traffic from their individual computers.

DDoS attacks are more prevalent and damaging in the modern Internet for two reasons. First, modern security tools have evolved to stop some ordinary DoS attacks. Second, DDoS attack tools have become relatively cheap and easy to operate.

What is a denial-of-service attack?

A denial-of-service (DoS) attack is a type of cyber attack in which a malicious actor aims to render a computer or other device unavailable to its intended users by interrupting the device's normal functioning. DoS attacks typically function by overwhelming or flooding a targeted machine with requests until normal traffic is unable to be processed, resulting in denial-of-service to addition users. A DoS attack is characterized by using a single computer to launch the attack.

How does a DoS attack work?

The primary focus of a DoS attack is to oversaturate the capacity of a targeted machine, resulting in denial-of-service to additional requests. The multiple attack vectors of DoS attacks can be grouped by their similarities.

DoS attacks typically fall in 2 categories:

Buffer overflow attacks

An attack type in which a memory buffer overflow can cause a machine to consume all available hard disk space, memory, or CPU time. This form of exploit often results in sluggish behavior, system crashes, or other deleterious server behaviors, resulting in denial-of-service.

Flood attacks

By saturating a targeted server with an overwhelming amount of packets, a malicious actor is able to oversaturate server capacity, resulting in denial-of-service. In order for most DoS flood attacks to be successful, the malicious actor must have more available bandwidth than the target.

What are some historically significant DoS attacks?

Historically, DoS attacks typically exploited security vulnerabilities present in network, software and hardware design. These attacks have become less prevalent as DDoS attacks have a greater disruptive capability and are relatively easy to create given the available tools. In reality, most DoS attacks can also be turned into DDoS attacks.

A few common historic DoS attacks include:

Smurf attack - a previously exploited DoS attack in which a malicious actor utilizes the broadcast address of vulnerable network by sending spoofed packets, resulting in the flooding of a targeted IP address.

Ping flood - this simple denial-of-service attack is based on overwhelming a target with ICMP (ping) packets. By inundating a target with more pings than it is able to respond to efficiently, denial-of-service can occur. This attack can also be used as a DDoS attack.

Ping of Death - often conflated with a ping flood attack, a ping of death attack involves sending a malformed packet to a targeted machine, resulting in deleterious behavior such as system crashes.

How can you tell if a computer is experiencing a DoS attack?

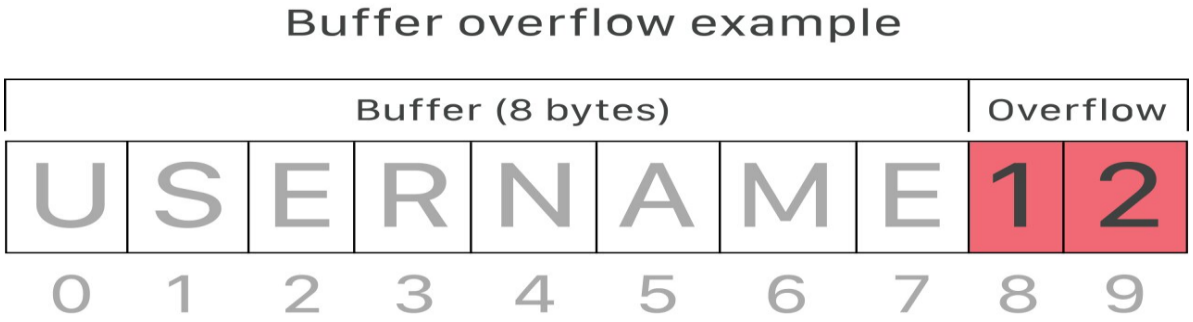
While it can be difficult to separate an attack from other network connectivity errors or heavy bandwidth consumption, some characteristics may indicate an attack is underway.

Indicators of a DoS attack include:

- Atypically slow network performance such as long load times for files or websites
- The inability to load a particular website such as your web property
- A sudden loss of connectivity across devices on the same network

What is buffer overflow?

Buffer overflow is an anomaly that occurs when software writing data to a buffer overflows the buffer's capacity, resulting in adjacent memory locations being overwritten. In other words, too much information is being passed into a container that does not have enough space, and that information ends up replacing data in adjacent containers.



Buffer overflows can be exploited by attackers with a goal of modifying a computer's memory in order to undermine or take control of program execution.

What's a buffer?

A buffer, or data buffer, is an area of physical memory storage used to temporarily store data while it is being moved from one place to another. These buffers typically live in RAM memory. Computers frequently use buffers to help improve performance; most modern hard drives take advantage of buffering to efficiently access data, and many online services also use buffers. For example, buffers are frequently used in online video streaming to prevent interruption. When a video is streamed, the video player downloads and stores perhaps 20% of the video at a time in a buffer and then streams from that buffer. This way, minor drops in connection speed or quick service disruptions won't affect the video stream performance.

Buffers are designed to contain specific amounts of data. Unless the program utilizing the buffer has built-in instructions to discard data when too much is sent to the buffer, the program will overwrite data in memory adjacent to the buffer.

Buffer overflows can be exploited by attackers to corrupt software. Despite being well-understood, buffer overflow attacks are still a major security problem that torment cyber-security teams. In 2014 a threat known as 'heartbleed' exposed hundreds of millions of users to attack because of a buffer overflow vulnerability in SSL software.

How do attackers exploit buffer overflows?

An attacker can deliberately feed a carefully crafted input into a program that will cause the program to try and store that input in a buffer that isn't large enough, overwriting portions of memory connected to the buffer space. If the memory layout of the program is well-defined, the attacker can deliberately overwrite areas known to contain executable code. The attacker can then replace this code with his own executable code, which can drastically change how the program is intended to work.

For example if the overwritten part in memory contains a pointer (an object that points to another place in memory) the attacker's code could replace that code with another pointer that points to an exploit payload. This can transfer control of the whole program over to the attacker's code.

Who is vulnerable to buffer overflow attacks?

Certain coding languages are more susceptible to buffer overflow than others. C and C++ are two popular languages with high vulnerability, since they contain no built-in protections against accessing or overwriting data in their memory. Windows, Mac OSX, and Linux all contain code written in one or both of these languages.

More modern languages like Java, PERL, and C# have built-in features that help reduce the chances of buffer overflow, but cannot prevent it altogether.

How to protect against buffer overflow attacks

Luckily, modern operating systems have runtime protections which help mitigate buffer overflow attacks. Let's explore 2 common protections that help mitigate the risk of exploitation:

- **Address space randomization** - Randomly rearranges the address space locations of key data areas of a process. Buffer overflow attacks generally rely on knowing the exact location of important executable code, randomization of address spaces makes that nearly impossible.
- **Data execution prevention** - Marks certain areas of memory either executable or non-executable, preventing an exploit from running code found in a non-executable area.

Software developers can also take precautions against buffer overflow vulnerabilities by writing in languages that have built-in protections or using special security procedures in their code.

Despite precautions, new buffer overflow vulnerabilities continue to be discovered by developers, sometimes in the wake of a successful exploitation. When new vulnerabilities are discovered, engineers need to patch the affected software and ensure that users of the software get access to the patch.

What are the different types of buffer overflow attacks?

There are a number of different buffer overflow attacks which employ different strategies and target different pieces of code. Below are a few of the most well-known.

- **Stack overflow attack** - This is the most common type of buffer overflow attack and involves overflowing a buffer on the call stack*.
- **Heap overflow attack** - This type of attack targets data in the open memory pool known as the heap*.
- **Integer overflow attack** - In an integer overflow, an arithmetic operation results in an integer (whole number) that is too large for the integer type meant to store it; this can result in a buffer overflow.
- **Unicode overflow -** A unicode overflow creates a buffer overflow by inserting unicode characters into an input that expect ASCII characters. (ASCII and unicode are encoding standards that let computers represent text. For example the letter 'a' is represented by the number 97 in ASCII. While ASCII codes only cover characters from Western languages, unicode can create characters for almost every written language on earth. Because there are so many more characters available in unicode, many unicode characters are larger than the largest ASCII character.)

What is a Botnet?

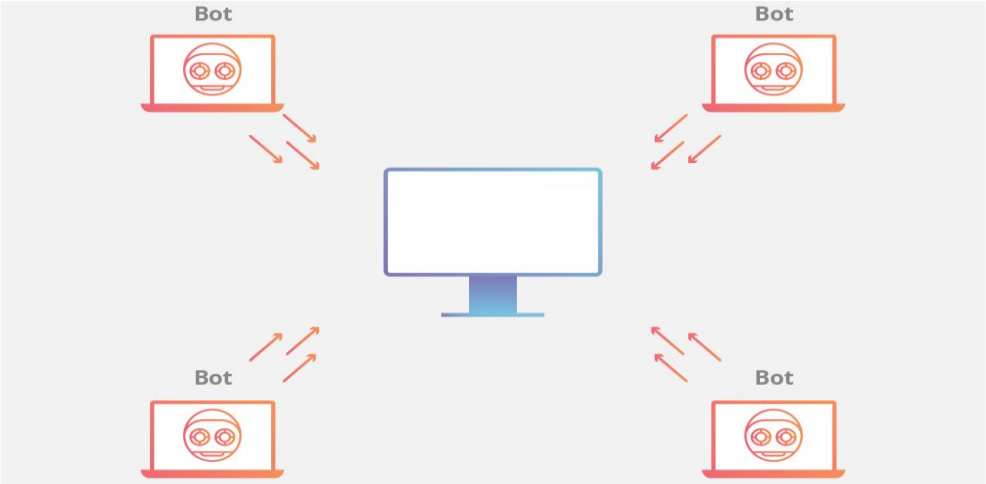
A botnet refers to a group of computers which have been infected by malware and have come under the control of a malicious actor. The term botnet is a portmanteau from the words robot and network and each infected device is called a bot. Botnets can be designed to accomplish illegal or malicious tasks including sending spam, stealing data, ransomware, fraudulently clicking on ads or distributed denial-of-service (DDoS) attacks.

While some malware, such as ransomware, will have a direct impact on the owner of the device, DDoS botnet malware can have different levels of visibility; some malware is designed to take total control of a device, while other malware runs silently as a background process while waiting silently for instructions from the attacker or "bot herder."

Self-propagating botnets recruit additional bots through a variety of different channels. Pathways for infection include the exploitation of website vulnerabilities, Trojan horse malware, and cracking weak authentication to gain remote access. Once access has been obtained, all of these methods for infection result

in the installation of malware on the target device, allowing remote control by the operator of the botnet. Once a device is infected, it may attempt to self-propagate the botnet malware by recruiting other hardware devices in the surrounding network.

While it's infeasible to pinpoint the exact numbers of bots in a particular botnet, estimations for total number of bots in a sophisticated botnet have ranged in size from a few thousand to greater than a million.



Why are botnets created?

Reasons for using a botnet ranges from activism to state-sponsored disruption, with many attacks being carried out simply for profit. Hiring botnet services online is relatively inexpensive, especially in relationship to the amount of damage they can cause. The barrier to creating a botnet is also low enough to make it a lucrative business for some software developers, especially in geographic locations where regulation and law enforcement are limited. This combination has led to a proliferation of online services offering attack-for-hire.

How is a botnet controlled?

A core characteristic of a botnet is the ability to receive updated instructions from the bot herder. The ability to communicate with each bot in the network allows the attacker to alternate attack vectors, change the targeted IP address, terminate an attack, and other customized actions. Botnet designs vary, but the control structures can be broken down into two general categories:

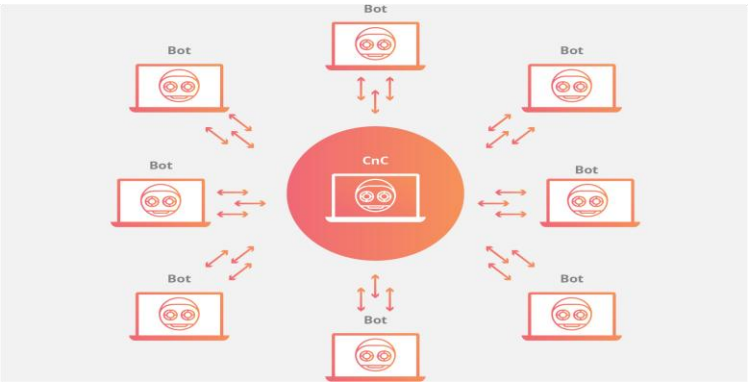
1). The client/server botnet model

The client/server model mimics the traditional remote workstation workflow where each individual machine connects to a centralized server (or a small number of centralized servers) in order to access information. In

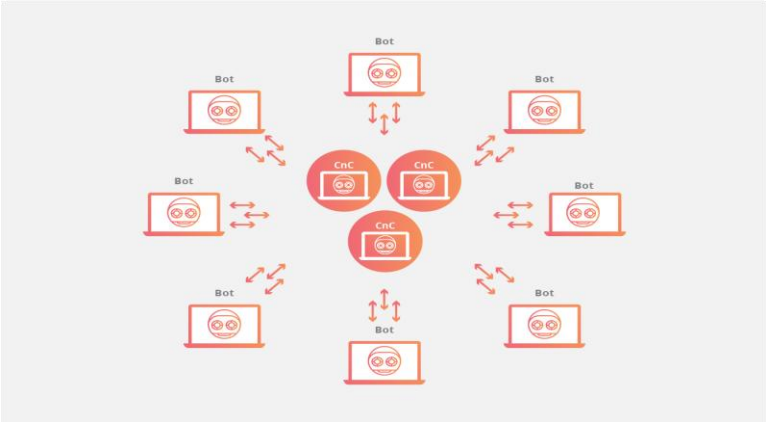
this model each bot will connect to a command-and-control center (CnC) resource like a web domain or an IRC channel in order to receive instructions. By using these centralized repositories to serve up new commands for the botnet, an attacker simply needs to modify the source material that each botnet consumes from a command center in order to update instructions to the infected machines. The centralized server in control of the botnet may be a device owned and operated by the attacker, or it may be an infected device.

A number of popular centralized botnet topologies have been observed, including:

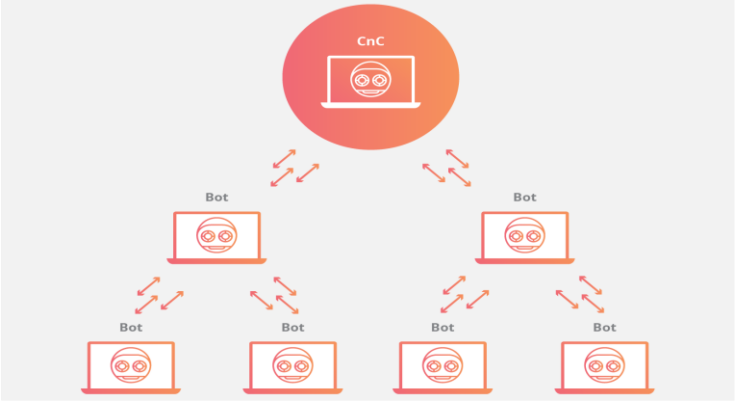
Star Network Topology



Multi Server Network Topology



Hierarchical Network Topology

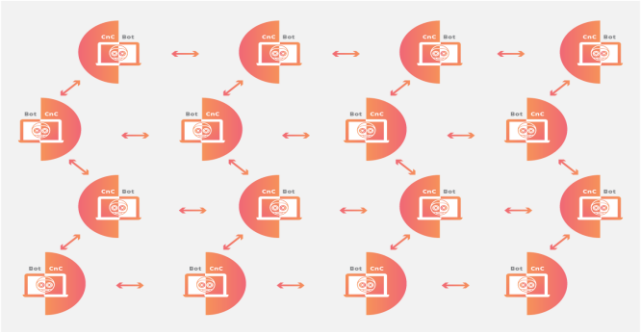


In any of these client/server models, each bot will connect to a command center resource like a web domain or an IRC channel in order to receive instructions. By using these centralized repositories to serve up new commands for the botnet, an attacker simply needs to modify the source material that each botnet consumes from a command center in order to update instructions to the infected machines.

Hand-in-hand with the simplicity of updating instructions to the botnet from a limited number of centralized sources is the vulnerability of those machines; in order to remove a botnet with a centralized server, only the server needs to be disrupted. As a result of this vulnerability, the creators of botnet malware have evolved and moved towards a new model that is less susceptible to disruption via a single or a few points of failure.

2). The peer-to-peer botnet model

To circumvent the vulnerabilities of the client/server model, botnets have more recently been designed using components of decentralized peer-to-peer filesharing. Embedding the control structure inside the botnet eliminates the single point-of-failure present in a botnet with a centralized server, making mitigation efforts more difficult. P2P bots can be both clients and command centers, working hand-in-hand with their neighboring nodes to propagate data.



Peer to peer botnets maintain a list of trusted computers with which they can give and receive communications and update their malware. By limiting the number of other machines the bot connects to, each bot is only exposed to adjacent devices, making it harder to track and more difficult to mitigate. Lacking a centralized command server makes a peer-to-peer botnet more vulnerable to control by someone other than the botnet's creator. To protect against loss of control, decentralized botnets are typically encrypted so that access is limited.

How is an existing botnet disabled?

Disable a botnet's control centers:

Botnets designed using a command-and-control schema can be more easily disabled once the control centers can be identified. Cutting off the head at the points of failure can take the whole botnet offline. As a result, system administrators and law enforcement officials focus on closing down the control centers of these botnets. This process is more difficult if the command center operates in a country where law enforcement is less capable or willing to intervene.

Eliminate infection on individual devices:

For individual computers, strategies to regain control over the machine include running antivirus software, reinstalling software from a safe backup, or starting over from a clean machine after reformatting the system. For IoT devices, strategies may include flashing the firmware, running a factory reset or otherwise formatting the device. If these option are infeasible, other strategies may be available from the device's manufacturer or a system administrator.

How can you protect devices from becoming part of a botnet?

Create secure passwords:

For many vulnerable devices, reducing exposure to botnet vulnerability can be as simple as changing the administrative credentials to something other than the default username and password. Creating a secure password makes brute force cracking difficult, creating a very secure password makes brute force cracking virtually impossible. For example, a device infected with the Mirai malware will scan IP addresses looking for responding devices. Once a device responds to a ping request, the bot will attempt to login to that found device with a preset list of default credentials. If the default password has been changed and a secure password has been implemented, the bot will give up and move on, looking for more vulnerable devices.

Allow only trusted execution of third-party code:

If you adopt the mobile phone model of software execution, only allowed applications may run, granting more control to terminate software deemed as malicious, botnets included. Only an exploitation of the supervisor software (i.e. kernel) may result in exploitation of the device. This hinges on having a secure kernel in the first place, which most IoT devices do not have, and is more applicable to machines that are running third party software.

Periodic system wipe/restores:

Restoring to a known good state after a set time will remove any gunk a system has collected, botnet software included. This strategy, when used as a preventative measure, ensures even silently running malware gets thrown out with trash.

Implement good ingress and egress filtering practices:

Other more advanced strategies include filtering practices at network routers and firewalls. A principle of secure network design is layering: you have the least restriction around publicly accessible resources, while continually beefing up security for things you deem sensitive. Additionally, anything that crosses these boundaries has to be scrutinized: network traffic, usb drives, etc. Quality filtering practices increase the likelihood that DDoS malware and their methods of propagation and communication will be caught before entering or leaving the network.

What is a DDoS attack?

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices. From a high level, a DDoS attack is like an unexpected traffic jam clogging up the highway, preventing regular traffic from arriving at its destination.

How does a DDoS attack work?

DDoS attacks are carried out with networks of Internet-connected machines.

These networks consist of computers and other devices (such as IoT devices)which have been infected with malware, allowing them to be controlled remotely by an attacker. These individual devices are referred to as bots (or zombies), and a group of bots is called a botnet.

Once a botnet has been established, the attacker is able to direct an attack by sending remote instructions to each bot.

When a victim's server or network is targeted by the botnet, each bot sends requests to the target's IP address, potentially causing the server or network to become overwhelmed, resulting in a denial-of-service to normal traffic.

Because each bot is a legitimate Internet device, separating the attack traffic from normal traffic can be difficult.

How to identify a DDoS attack

The most obvious symptom of a DDoS attack is a site or service suddenly becoming slow or unavailable. But since a number of causes — such a legitimate spike in traffic — can create similar performance issues, further investigation is usually required. Traffic analytics tools can help you spot some of these telltale signs of a DDoS attack:

- Suspicious amounts of traffic originating from a single IP address or IP range
- A flood of traffic from users who share a single behavioral profile, such as device type, geolocation, or web browser version
- An unexplained surge in requests to a single page or endpoint
- Odd traffic patterns such as spikes at odd hours of the day or patterns that appear to be unnatural (e.g. a spike every 10 minutes)

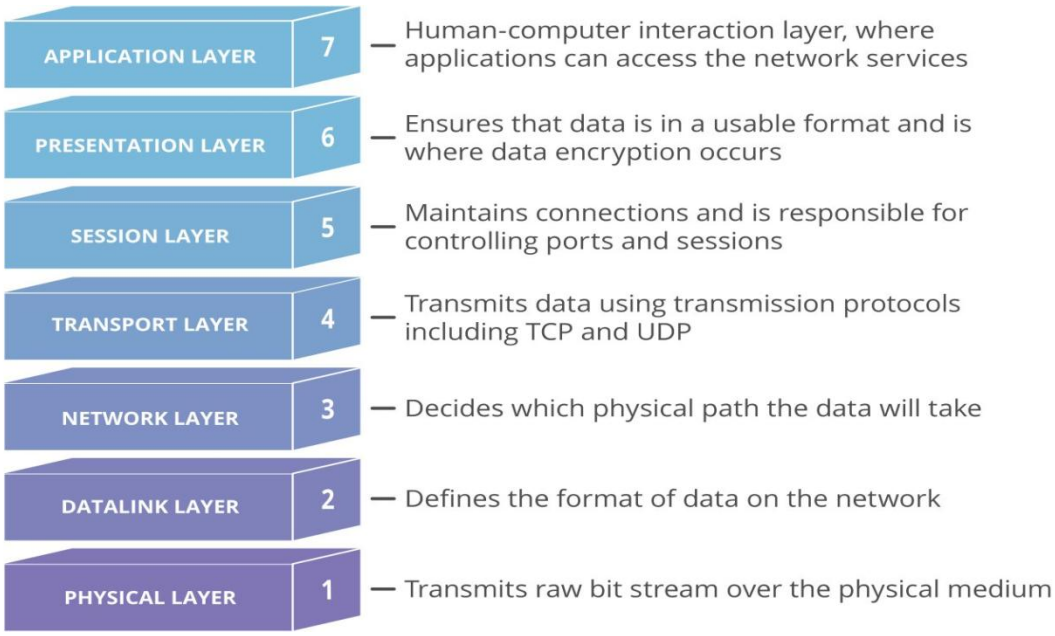
There are other, more specific signs of DDoS attack that can vary depending on the type of attack.

What are some common types of DDoS attacks?

Different types of DDoS attacks target varying components of a network connection. In order to understand how different DDoS attacks work, it is necessary to know how a network connection is made.

A network connection on the Internet is composed of many different components or “layers”. Like building a house from the ground up, each layer in the model has a different purpose.

The OSI model, shown below, is a conceptual framework used to describe network connectivity in 7 distinct layers.



While nearly all DDoS attacks involve overwhelming a target device or network with traffic, attacks can be divided into three categories. An attacker may use one or more different attack vectors, or cycle attack vectors in response to counter measures taken by the target.

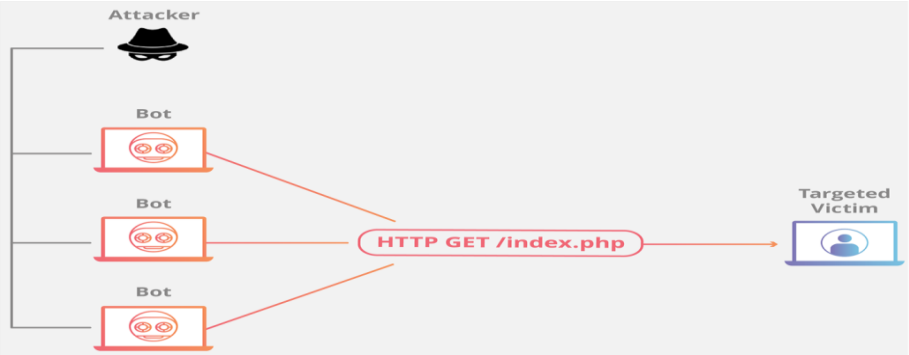
Application layer attacks

The goal of the attack:

Sometimes referred to as a layer 7 DDoS attack (in reference to the 7th layer of the OSI model), the goal of these attacks is to exhaust the target’s resources to create a denial-of-service. The attacks target the layer where web pages are generated on the server and delivered in response to HTTP requests. A single HTTP request is computationally cheap to execute on the client side, but it can be expensive for the target server to respond to, as the server often loads multiple files and runs database queries in order to create a web page.

Layer 7 attacks are difficult to defend against, since it can be hard to differentiate malicious traffic from legitimate traffic.

Application layer attack example:



HTTP flood

This attack is similar to pressing refresh in a web browser over and over on many different computers at once – large numbers of HTTP requests flood the server, resulting in denial-of-service.

This type of attack ranges from simple to complex.

Simpler implementations may access one URL with the same range of attacking IP addresses, referrers and user agents. Complex versions may use a large number of attacking IP addresses, and target random urls using random referrers and user agents.

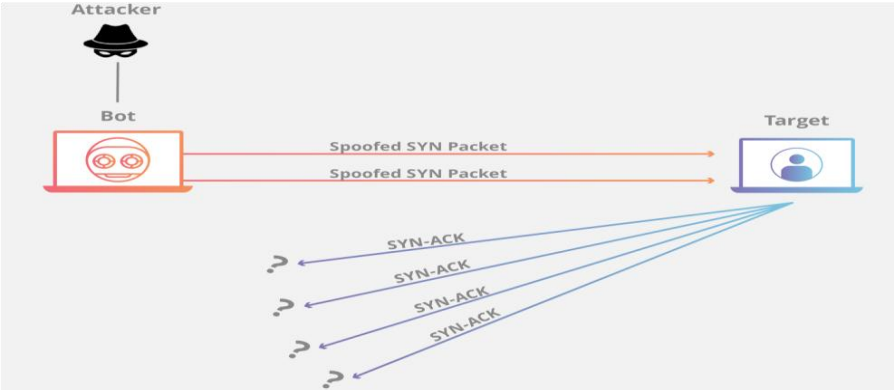
Protocol attacks

The goal of the attack:

Protocol attacks, also known as state-exhaustion attacks, cause a service disruption by over-consuming server resources and/or the resources of network equipment like firewalls and load balancers.

Protocol attacks utilize weaknesses in layer 3 and layer 4 of the protocol stack to render the target inaccessible.

Protocol attack example:



SYN flood

A SYN Flood is analogous to a worker in a supply room receiving requests from the front of the store. The worker receives a request, goes and gets the package, and waits for confirmation before bringing the package out front. The worker then gets many more package requests without confirmation until they can't carry any more packages, become overwhelmed, and requests start going unanswered. This attack exploits

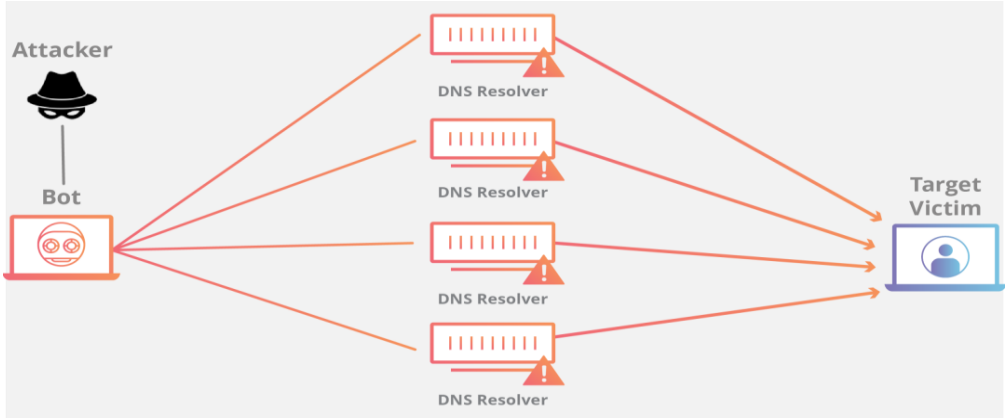
the TCP handshake — the sequence of communications by which two computers initiate a network connection — by sending a target a large number of TCP "Initial Connection Request" SYN packets with spoofed source IP addresses. The target machine responds to each connection request and then waits for the final step in the handshake, which never occurs, exhausting the target's resources in the process.

Volumetric attacks

The goal of the attack:

This category of attacks attempts to create congestion by consuming all available bandwidth between the target and the larger Internet. Large amounts of data are sent to a target by using a form of amplification or another means of creating massive traffic, such as requests from a botnet.

Amplification example:



DNS Amplification

A DNS amplification is like if someone were to call a restaurant and say "I'll have one of everything, please call me back and repeat my whole order," where the callback number actually belongs to the victim. With very little effort, a long response is generated and sent to the victim.

By making a request to an open DNS server with a spoofed IP address (the IP address of the victim), the target IP address then receives a response from the server.

What is the process for mitigating a DDoS attack?

The key concern in mitigating a DDoS attack is differentiating between attack traffic and normal traffic.

- For example, if a product release has a company's website swamped with eager customers, cutting off all traffic is a mistake. If that company suddenly has a surge in traffic from known attackers, efforts to alleviate an attack are probably necessary.
- The difficulty lies in telling the real customers apart from the attack traffic.

- In the modern Internet, DDoS traffic comes in many forms. The traffic can vary in design from un-spoofed single source attacks to complex and adaptive multi-vector attacks.
- A multi-vector DDoS attack uses multiple attack pathways in order to overwhelm a target in different ways, potentially distracting mitigation efforts on any one trajectory.
- An attack that targets multiple layers of the protocol stack at the same time, such as a DNS amplification (targeting layers 3/4) coupled with an HTTP flood (targeting layer 7) is an example of multi-vector DDoS.
- Mitigating a multi-vector DDoS attack requires a variety of strategies in order to counter different trajectories.
- Generally speaking, the more complex the attack, the more likely it is that the attack traffic will be difficult to separate from normal traffic - the goal of the attacker is to blend in as much as possible, making mitigation efforts as inefficient as possible.
- Mitigation attempts that involve dropping or limiting traffic indiscriminately may throw good traffic out with the bad, and the attack may also modify and adapt to circumvent countermeasures. In order to overcome a complex attempt at disruption, a layered solution will give the greatest benefit.

Blackhole routing

One solution available to virtually all network admins is to create a blackhole route and funnel traffic into that route. In its simplest form, when blackhole filtering is implemented without specific restriction criteria, both legitimate and malicious network traffic is routed to a null route, or blackhole, and dropped from the network.

If an Internet property is experiencing a DDoS attack, the property's Internet service provider (ISP) may send all the site's traffic into a blackhole as a defense. This is not an ideal solution, as it effectively gives the attacker their desired goal: it makes the network inaccessible.

Rate limiting

Limiting the number of requests a server will accept over a certain time window is also a way of mitigating denial-of-service attacks.

While rate limiting is useful in slowing web scrapers from stealing content and for mitigating brute force login attempts, it alone will likely be insufficient to handle a complex DDoS attack effectively.

Nevertheless, rate limiting is a useful component in an effective DDoS mitigation strategy.

Web application firewall

A Web Application Firewall (WAF) is a tool that can assist in mitigating a layer 7 DDoS attack. By putting a WAF between the Internet and an origin server, the WAF may act as a reverse proxy, protecting the targeted server from certain types of malicious traffic.

By filtering requests based on a series of rules used to identify DDoS tools, layer 7 attacks can be impeded. One key value of an effective WAF is the ability to quickly implement custom rules in response to an attack.

Anycast network diffusion

This mitigation approach uses an Anycast network to scatter the attack traffic across a network of distributed servers to the point where the traffic is absorbed by the network.

Like channeling a rushing river down separate smaller channels, this approach spreads the impact of the distributed attack traffic to the point where it becomes manageable, diffusing any disruptive capability.

The reliability of an Anycast network to mitigate a DDoS attack is dependent on the size of the attack and the size and efficiency of the network. An important part of the DDoS mitigation implemented by Cloudflare is the use of an Anycast distributed network.

Cloudflare has a 155 Tbps network, which is an order of magnitude greater than the largest DDoS attack recorded.

If you are currently under attack, there are steps you can take to get out from under the pressure.

What is a Web Application Firewall (WAF)?

WAF or web application firewall helps protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet. It typically protects web applications from attacks such as cross-site forgery, cross-site-scripting (XSS), file inclusion, and SQL injection, among others. A WAF is a protocol layer 7 defense (in the OSI model), and is not designed to defend against all types of attacks. This method of attack mitigation is usually part of a suite of tools which together create a holistic defense against a range of attack vectors.

By deploying a WAF in front of a web application, a shield is placed between the web application and the Internet. While a proxy server protects a client machine’s identity by using an intermediary, a WAF is a type of reverse-proxy, protecting the server from exposure by having clients pass through the WAF before reaching the server.

A WAF operates through a set of rules often called policies. These policies aim to protect against vulnerabilities in the application by filtering out malicious traffic. The value of a WAF comes in part from

the speed and ease with which policy modification can be implemented, allowing for faster response to varying attack vectors; during a DDoS attack, rate limiting can be quickly implemented by modifying WAF policies.

What is the difference between blocklist and allowlist WAFs?

A WAF that operates based on a blocklist (negative security model) protects against known attacks. Think of a blocklist WAF as a club bouncer instructed to deny admittance to guests who don't meet the dress code. Conversely, a WAF based on an allowlist (positive security model) only admits traffic that has been pre-approved. This is like the bouncer at an exclusive party, he or she only admits people who are on the list. Both blocklists and allowlists have their advantages and drawbacks, which is why many WAFs offer a hybrid security model, which implements both.

What are network-based, host-based, and cloud-based WAFs?

A WAF can be implemented one of three different ways, each with its own benefits and shortcomings:
A network-based WAF is generally hardware-based. Since they are installed locally they minimize latency, but network-based WAFs are the most expensive option and also require the storage and maintenance of physical equipment.
A host-based WAF may be fully integrated into an application's software. This solution is less expensive than a network-based WAF and offers more customizability. The downside of a host-based WAF is the consumption of local server resources, implementation complexity, and maintenance costs. These components typically require engineering time, and may be costly.
Cloud-based WAFs offer an affordable option that is very easy to implement; they usually offer a turnkey installation that is as simple as a change in DNS to redirect traffic. Cloud-based WAFs also have a minimal upfront cost, as users pay monthly or annually for security as a service. Cloud-based WAFs can also offer a solution that is consistently updated to protect against the newest threats without any additional work or cost on the user's end. The drawback of a cloud-based WAF is that users hand over the responsibility to a third party, therefore some features of the WAF may be a black box to them. (A cloud-based WAF is one type of cloud firewall; learn more about cloud firewalls.)

How are DoS/DDoS attack tools categorized?

A number of tools exist that can be adapted to launch DoS/DDoS attacks, or are explicitly designed for that purpose. The former category are often ""stressors"" — tools with the stated purpose of helping security

researchers and network engineers perform stress tests against their own networks, but which can also be used to perform genuine attacks.

Some are specialized and only focus on a particular layer of the OSI model, while others are designed to allow for multiple attack vectors. Categories of attack tools include:

Low and slow attack tools

As the name implies, these types of attack tools use a low volume of data and operate very slowly. Designed to send small amounts of data across multiple connections in order to keep ports on a targeted server open as long as possible, these tools continue to take up the server's resources until it is unable to maintain additional connections. Uniquely, low and slow attacks may at times be effective even when not using a distributed system such as a botnet and are commonly used by a single machine.

Application layer (L7) attack tools

These tools target layer 7 of the OSI model, where Internet-based requests such as HTTP occur. Using an HTTP flood attack to overwhelm a target with HTTP GET and POST requests, a malicious actor can launch attack traffic that is difficult to distinguish from normal requests made by actual visitors.

Protocol and transport layer (L3/L4) attack tools

Going further down the protocol stack, these tools utilize protocols like UDP to send large volumes of traffic to a targeted server, such as during a UDP flood. While often ineffective individually, these attacks are typically found in the form of DDoS attacks where the benefit of additional attacking machines increases the effect.

What are commonly used DoS/DDoS attack tools?

Some commonly used tools include:

Low Orbit Ion Cannon (LOIC)

The LOIC is an open-source stress testing application. It allows for both TCP and UDP protocol layer attacks to be carried out using a user-friendly WYSIWYG interface. Due to the popularity of the original tool, derivatives have been created that allow attacks to be launched using a web browser.

High Orbit Ion Cannon (HOIC)

This attack tool was created to replace the LOIC by expanding its capabilities and adding customizations. Using the HTTP protocol, the HOIC is able to launch targeted attacks that are difficult to mitigate. The software is designed to have a minimum of 50 people working together in a coordinated attack effort.

Slowloris

Slowloris is an application designed to instigate a low and slow attack on a targeted server. It needs a relatively limited amount of resources in order to create a damaging effect.

R.U.D.Y (R-U-Dead-Yet)

R.U.D.Y. is another low and slow attack tool designed to allow the user to easily launch attacks using a simple point-and-click interface. By opening multiple HTTP POST requests and then keeping those connections open as long as possible, the attack aims to slowly overwhelm the targeted server.

How can I defend against DoS/DDoS tools?

Since DoS and DDoS attacks take a variety of forms, mitigating them requires a variety of tactics. Common tactics for stopping DDoS attacks include:

- Rate limiting: Limiting the number of requests a server will accept over a certain time window
- Web application firewalls: Tools that filter web traffic based on a series of rules
- Anycast network diffusion: Placing a large, distributed cloud network between a server and incoming traffic, providing additional computing resources with which to respond to requests.