What is a bot?

A bot is a computer program that is designed to perform automated tasks over the internet. These tasks are typically relatively simple and repeatable, and a bot can execute them much faster and much more accurately than a human user. Googlebot, for example, is a bot owned by Google that is utilized to crawl and index websites, including this website. It is technically performing a job that can be done by any human: copy and pasting data, but it can do it at a much faster rate with higher accuracy.

What is a botnet?

The term botnet is actually short for “ro **bot net** work”, which refers to a group of robot devices (computers, mobile phones, IoT devices) that are now under the control of an attacking party. Typically the devices in a botnet have been infected by malware, and the attacker controlling the botnet is called a “bot herder”.

The devices under the control of a bot herder are called “ zombie ” devices, although sometimes they are also called “bot devices” or just “bots”—hence the confusion mentioned above.

How does a botnet attack work?

Once a cybercriminal (or bot herder) has control of a group of infected devices (composing a botnet) they can remotely command every device to simultaneously carry out coordinated activities, including malicious and criminal activities like:

- **DDoS Attacks :** Distributed denial of service attacks leverage the number of devices in the botnet to send a massive amount of requests or payloads to overload a target server or website, rendering the service inaccessible to legitimate users. Even large companies like Sony and Electronic Arts have been victims of large-scale DDoS attacks caused by botnets, and could benefit from better DDoS protection services.

- **Spam Attacks:** The majority of online spam attacks (email spam, comment section spam, form spam, etc.) are performed by botnets. Spam attacks are often used for spreading malware and phishing, and there are botnets that can launch tens of *billions* of spam messages per day.

- **Data Breach:** Some botnets are specially designed to steal confidential and valuable information like banking details, credit card numbers, and so on. The ZeuS botnet , for example, is primarily designed to steal account information from various e-commerce, banking, and social media sites. Botnets may be

designed to target very specific high-value services and digital assets in breaches and data leaks .

- **Monitoring:** Compromised zombie devices will monitor the user's activities and can scan passwords and financial information to report back to the bot herder.
- **Spreading the Botnet:** A botnet will search for vulnerabilities in other devices, websites, and networks to spread the malware to other machines and grow the botnet, making the overall botnet attack worse.

How is a botnet created?

Botnets always consist of a command and control (C&C or C2) server and a group of zombie devices.

1. C&C Server

The bot herder runs a device (or a group of devices) that acts as a C&C server, which sends command programming to the zombie devices to drive their next activities. The bot herder may also rent the C&C server (or the commands) out to other parties. For example, a business can rent a C&C server and command a botnet to launch a DDoS attack on its competitors.

There are two different types of C&C servers:

- **Centralized:** A centralized C&C server follows a client-server bot herding model, in which a single C&C server commands *all* the zombie devices in a centralized network. In some cases, additional C&C servers might be tasked as sub-herders between the main C&C server and the zombie devices, creating a proxy-based hierarchy, but all commands come from the main C&C server. The centralized model, allows the bot herder to be more easily discovered, and is thus considered obsolete/no longer ideal.

- **Decentralized:** The more modern and preferred approach is the decentralized model, where *all* zombie devices can send and redirect commands to one another. The main C&C server only needs to contact one of the zombie devices to ensure the command gets relayed to the other zombies, which is why this model is also called a “peer-to-peer” model. The decentralized model protects the identity of the bot herder by making the C&C server more difficult to trace.

2. Zombie Devices

As discussed, zombie devices—often called bots—are devices that have been taken over by the bot herder as a part of the botnet, mainly due to malware infection. These devices, once compromised, will blindly follow the C&C server’s command.

While different bot herders might use varying techniques for “recruiting” new devices into the botnet, recruitment generally involves three core phases:

Phase 1: Finding and Exploiting Vulnerabilities

The first, and arguably the most crucial, step involves the attacker searching and finding vulnerabilities in a human user's behavior—an application/software, a website, or any other potential inroad to a device. The objective of the attacker in phase 1 is to find ways to take control of a device, such as any vulnerability that can be exposed to a malware infection, for example.

For instance, the attacker might find vulnerabilities in a device user's behavior that indicate malware could be successfully delivered to the user via a spam/phishing email. Or the attacker might notice an unpatched vulnerability on an application or an OS to exploit.

Phase 2: Actual Malware Infection

In this phase, the device is infected by malware and is now under the control of the bot herder.

Infection can happen in two different ways:

- **Passive:** The infection happens without any help from human users. For example, when the infection happens due to an exploited software vulnerability.
- **Active:** The infection happens after the human user takes a certain action, like downloading an email attachment or clicking a link on an already infected website.

Once infected, no matter what method was used, the device falls under the control of the attacker.

Phase 3: Taking Control of the Compromised Device

The third and final phase involves the bot herder organizing the infected devices into a network, the *botnet,* and performing the necessary configurations to ensure each of the infected devices can be remotely managed via the C&C server.

The bot herder will repeat these three phases with many devices to grow the botnet's size. In some cases, botnets are able to grow to include millions of computers, smartphones, and IoT devices.

3 Most Concerning Botnet Attacks

While using a botnet to spread malware infections is a botnet attack by itself, botnets are more notably used to execute secondary cybercrime attacks, often on a very large scale due to the size of botnets.

While large-scale botnet attacks can come in various different forms, there are three vital types to protect your system/website against::

1. Phishing

Most botnets rely on spam and phishing tactics to infect more devices and grow the botnet in size.

In phishing and other forms of social engineering attacks, the botnet will send emails, post comments, and create messages on social media platforms imitating people and/or organizations that are known and trusted by the target victim.

As a botnet attack, phishing can be particularly difficult to defend against because no matter how advanced your security infrastructure is, it can totally crumble when just one employee or user in your network falls victim to a phishing attack. That's why human error remains the *top cause* for successful data breach attacks and cybersecurity attacks.

2. Distributed Denial of Service (DDoS)

The idea behind using botnets for DDoS attacks is to overwhelm a target server with a massive number of requests (from the zombie devices) to crash, or at least slow down, the server significantly.

DDoS is one of the most common ways botnets are utilized in criminal attacks, and often the most dangerous. Damages resulting from DDoS attacks can be severe and long-lasting, not only in terms of financial damages, but also reputational damages.

It's very important for any business that serves its customers online (which includes having a functional website) to have proper anti-DDoS measures and a functional anti-DDoS response plan. When a DDoS attack occurs, it *is* already too late to plan your response, so an actionable response plan must be prepared beforehand. Many businesses use a DDoS protection software to do this.

3. Account Takeover Attack

Bot herders can use botnets to perform various forms of Account Takeover (ATO) attacks, especially brute force (credential cracking) and credential stuffing attacks.

In a brute force attack, the zombie devices are commanded to try the different possibilities of a user password to "crack" the password. For example, if it's a 4-digit pin, zombie device 1 will try "0000", the second zombie device will try "0001", and so on up to "9999" or until the right PIN has been guessed.

Defending against botnet ATO attacks can be quite challenging, and fairly effective in exploiting weak user credentials to take over accounts. Some businesses use an account takeover prevention software to minimize this risk.

How to Protect Yourself and Your Business Against Botnets

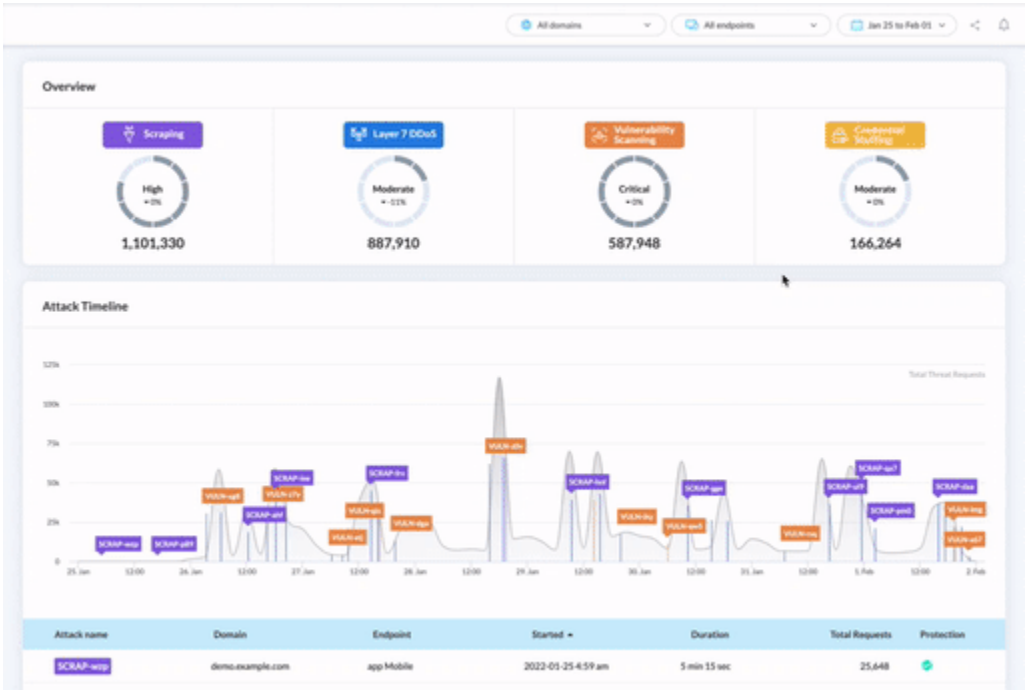
The significant threats presented by botnets and various forms of botnet attacks make it critical to protect your business, customers, and systems against botnet attacks. Here are some important best practices to consider when defending against botnet attacks:

1. Update Everything ASAP

Your team should regularly update your OS, applications, and software solutions—ideally as soon as updates are made available. Security patches exist for a reason, and most software vendors will publish known vulnerabilities as soon as they are patched. If you haven’t updated your OS when vulnerabilities are made known to hackers, it may now be a potential gateway for bot herders to infect your system and recruit it into a botnet.

If you can’t set up automatic software and firmware updates on all your systems and devices for any reason, schedule regular manual updates for at least once a week.

2. Choose the Right Botnet Detection Solution



To effectively protect your full online ecosystem from botnet attacks, you need a professional botnet detection software solution that is designed to quickly and accurately detect malicious botnet activity in real time.

As botnets continue to grow more sophisticated in masking their identities, classic (signature-based) fingerprinting and IP reputation detection are no longer sufficient protection. Instead, choose a solution with supervised AI and machine learning capabilities that

harness behavior-based analyses and considers 5 trillion signals per day for the most accurate botnet detection.

3. Educate Your Users and Employees to Avoid Phishing Attacks

Ensure you have adequate security training and educational materials about how to avoid phishing attacks for your organization, your users, and especially employees that may be at a higher risk of being targeted due to their role in the organization. Some general guidelines include:

- Avoid downloading email attachments unless you are 100% sure of the identity of the sender. Carefully investigate and verify the sender's email address before clicking on any unexpected attachment or link.
- Invest in a proper antivirus/anti-malware software that can automatically and accurately scan attachments for malware.
- Avoid clicking on any links in any messages (email, text messages, social media direct messages, etc.) unless you are 100% sure about the sender. When you *must* visit the link, it's better to manually enter the URL into your browser's address bar to avoid DNS cache poisoning, rather than clicking.

4. Use Strong and Unique Passwords

Always use strong/complex passwords for all your accounts, including admin accounts across all devices that connect to other devices or directly to the internet. Also, make sure each password is unique for only one account.

5. Anti-Malware Software

Since most botnet conversions happen due to malware infections, it's crucial to invest in a strong enough antivirus/anti-malware software that can protect your device and system against botnet malware and other digital threats. Also, make sure to update your antivirus solution **regularly**.

We will be using Selenium (python library) for making the auto-login bot. Python Selenium library helps us to access all functionalities of Selenium WebDriver like Firefox, Chrome, Remote etc.

Installation

First of all, we have to install selenium using the below command:

```
pip install selenium
```

After successful installation of selenium, we also have to install chromedriver for accessing the chrome webdriver of selenium. You can download the same from here (Download version according to your system chrome version and according to your OS).

Make sure that you have noted the location where the chromedriver has been downloaded (as it is used in our python script). Now After downloading extract the zip file and please note the file location of the extracted file as we have needed it later in python code. (You can find the location by clicking on properties and then details).

Stepwise Implementation:

- First of all import the webdrivers from the selenium library.
- Find the URL of the login page to which you want to logged in.
- Provide the location executable chrome driver to selenium webdriver to access the chrome browser.
- Finally, find the name or id or class or CSS selector of username and password by right-clicking inspect on username and password.

Below is the implementation:

- Python3

```
# Used to import the webdriver from selenium

from selenium import webdriver

import os



# Get the path of chromedriver which you have install



def startBot(username, password, url):

    path = "C:\\Users\\hp\\Downloads\\chromedriver"



    # giving the path of chromedriver to selenium webdriver
```

```
    driver = webdriver.Chrome(path)



    # opening the website  in chrome.

    driver.get(url)



    # find the id or name or class of

    # username by inspecting on username input

    driver.find_element_by_name(

        "id/class/name of username").send_keys(username)



    # find the password by inspecting on password input

    driver.find_element_by_name(

        "id/class/name of password").send_keys(password)



    # click on submit

    driver.find_element_by_css_selector(

        "id/class/name/css selector of login button").click()




# Driver Code
```

```
# Enter below your login credentials

username = "Enter your username"

password = "Enter your password"



# URL of the login page of site

# which you want to automate login.

url = "Enter the URL of login page of website"



# Call the function

startBot(username, password, url)
```