Understanding the Evolution and Threat of Botnets in Cybersecurity

Introduction:

The cybersecurity landscape is continually evolving, with botnets representing one of the most formidable threats in this domain. A botnet, a network of compromised computers controlled by a malicious actor, can execute a range of cybercrimes, from DDoS attacks to data theft. This whitepaper delves into the nature of botnets, their evolution, significant examples, and the ongoing challenges they pose to cybersecurity.

Understanding Botnets:

A botnet is essentially a collection of internet-connected devices, including PCs, servers, and IoT devices, infected with malware that allows a remote attacker to control them. These “zombie” devices can then be used to launch coordinated cyber-attacks, send spam, or steal data, often without the knowledge of the device owner.

Methods of Botnet Expansion: Hackers employ various strategies to expand their botnet networks:

- IP Address Scanning: Automated tools scan for vulnerable devices across IP ranges.
- Use of Proxies and VPNs: To mask their real IP addresses.
- Botnets Control: Compromised devices (botnets) are spread across different IPs.
- Dynamic IPs: Utilizing networks that assign changing IP addresses.
- Compromised Devices: Using infected devices in diverse networks.
- IP Spoofing: Falsifying the source IP address (less common).

Evidence in the Log Files:

In recent years, a comprehensive analysis of log files has revealed critical insights. A server has automatically blocked over 513 networks operating under a 16-bit mask, potentially implicating over 33 million IP addresses. This extensive blocking highlights a concerning trend: the vast number of networks involved in these incidents. Notably, when a specific IP address or network is blocked, attackers swiftly migrate to a new IP in an alternate network, usually within less than an hour, to continue their persistent assaults.

Despite the vast number of networks that have been blocked, the bots are still capable of quickly finding an IP address that has not been added to a blacklist.

The blacklist has blocked as of today (1st February 2024) over 46 networks with a /8 subnet mask, and 513 networks with a /16 subnet mask: we speak about 800 million IP addresses blocked! And the Bot can still find in a short time an available IP which is not in the Blacklist.

This pattern underscores the alarming capability of hackers, who control a substantial number of devices. Equipped with these resources, they are capable of executing brute-force attacks from a wide array of networks. This situation demands heightened vigilance and advanced security measures to counteract these evolving cyber threats.

What does it mean:

- The botnets have an intricate understanding of network permissions and protocols for every internet server.
- Leveraging this knowledge, the botnet is able to rapidly identify a specific IP address within an hour for conducting persistent cyber-attacks.
- This capability demonstrates the botnet's advanced efficiency in targeting and exploiting network vulnerabilities for sustained operations.

This indicates that the attacks are not only systematic but also widespread in scale. Rather than being opportunistic, they should be defined as persistent, chronic, paranoid, and alarmingly targeted against all infrastructures.

Persistent attacks:

Hackers attempt to send SSH login commands from a wide range of IP addresses and networks using several methods.

- **Botnets:** Hackers often use networks of compromised computers, laptops, mobile phones, known as botnets, to conduct attacks. These botnets can consist of thousands of infected computers (including PCs, servers, and sometimes mobile devices) across the world. Each infected device can be used to send SSH login attempts to a target, making the attack come from a wide range of IP addresses.
- **Cloud Computing and Virtual Private Servers (VPS):** Hackers may rent cloud computing resources or VPS from various providers. These services can provide them with access to a large number of IP addresses from different networks and regions. They can launch attacks from these virtual machines, making it seem like the attacks are coming from legitimate and diverse sources.
- **Compromised IoT Devices**: Similar to botnets, hackers can exploit insecure Internet of Things (IoT) devices (like security cameras, smart appliances, etc.) and use them as a part of their attack infrastructure.

Some malicious apps can turn mobile devices into part of a botnet. These apps might be disguised as legitimate software but, once installed, can perform a range of unauthorized activities, including participating in distributed attacks like SSH login attempts.

To protect against such large attacks, it's important to implement strong security measures like using secure passwords, enabling two-factor authentication, using SSH key-based logins instead of password-based logins, regularly updating and patching systems, and monitoring network traffic for unusual activities. Additionally, employing intrusion detection and prevention systems can help identify and block such malicious traffic.

Notable Botnets:

- Mirai: Infected 600,000+ IoT devices, notorious for the 2016 Dyn DDoS attack.
- Conficker: Targeted Windows OS, infected millions since 2008.
- Zeus: Specialized in stealing banking information via keystroke logging.
- Storm Worm: Distributed via email spam, creating a massive botnet.
- WannaCry: 2017 ransomware, exploiting EternalBlue, affected 200,000+ computers.
- Satori: An evolution of Mirai, exploiting IoT device vulnerabilities.

- Avalanche Network: A multifaceted botnet for malware and phishing, active for seven years.
- Largest Known Botnet: "BredoLab" botnet, controlling over 30 million computers, primarily used for spamming and malware distribution.

Mitigation Strategies:

To combat botnets, cybersecurity specialists recommend:

- Intrusion Detection/Prevention Systems: For detecting suspicious activities.
- Rate-Limiting and Advanced Firewalls: To block repeated login attempts and advanced threats.
- SSH Keys and Fail2Ban: For secure authentication and auto-updating firewall rules.
- Geographical Blocking: If global access isn't required.
- Regular Monitoring: Keeping an eye on server logs and network traffic.

Practical Suggestion for Linux-Server:

Block all Networks and all protocols. Work with a whitelist and allow only specific IP (if possible, without ranges).

The files /etc/hosts.deny and /etc/hosts.allow are part of the TCP Wrapper system on Unix-like operating systems. They are used to control access to network services that are run from inetd and sometimes xinetd, including the Secure Shell Daemon (SSHD). These files allow system administrators to permit or deny access to these services based on IP addresses or hostnames.

/etc/hosts.deny: This file is used to specify which hosts are denied access to services on your system. Each line in the file specifies a service and a pattern that matches client hosts. If a client host matches a pattern in this file, it is denied access to the service.

Proposal for /etc/hosts.deny:

SSHD: ALL

In this example, SSHD: ALL means that all hosts are denied access to SSHD by default. This is a broad and restrictive setting. It means that unless a host is explicitly allowed in /etc/hosts.allow, it won't be able to access the SSHD service on your machine.

/etc/hosts.allow: This file specifies exceptions to the rules defined in /etc/hosts.deny. If a host matches a pattern in this file, it is allowed access to the specified service, even if it's denied in /etc/hosts.deny.

Proposal for /etc/hosts.allow:

SSHD: <IP>

The system checks /etc/hosts.allow first; if no match is found, it then checks /etc/hosts.deny. If no match is found in either file, access is granted. Therefore, by setting a default deny in /etc/hosts.deny and then selectively allowing access in /etc/hosts.allow, you can create a more secure and controlled environment for your network services.

Keep in mind that these settings only affect services managed by TCP Wrappers and don't apply to services that don't use it or to services managed by other means, like firewall rules. Also, the exact behavior can depend on how your system and its network services are configured.

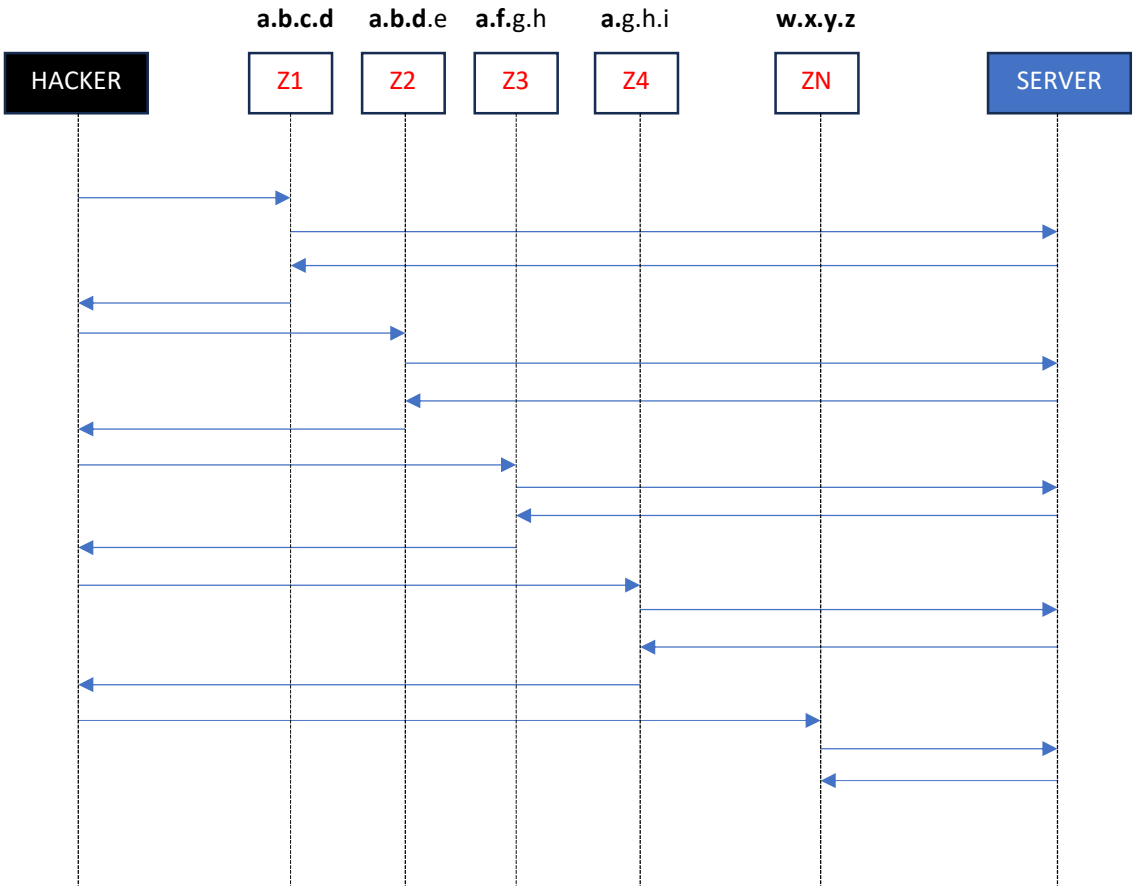
Reverse Engineering of the BotNet performing a persistent Attack

In the diagram provided, the reverse engineering of a botnet attack over the past five years is depicted. The hacker initiates a login request (SSH) from Zombie 1 (Z1). Zombie 1 then forwards the command to the target server. If the server denies the login, the hacker proceeds to send another request from Zombie 2 (Z2), located in a different network. Zombie 2 sends the command to the target server. Should this attempt fail, the hacker directs Zombie 3 (Z3), situated in another distinct network, to send the command. If the server's response is still negative, the hacker then uses Zombie 4 (Z4), which is in yet another separate network, to send the command. If this attempt is also unsuccessful, the hacker continues the attack using a new IP from a completely different network.

Statistics suggest that hackers have control over a zombie in all possible networks.

Legend:

a.b.c.d : IP1 a.b.d.e : IP2 a.f.g.h: IP3 a.g.h.i: IP4 w.x.y.z: IP5



Conclusion:

The fight against botnets represents a constantly evolving and persistent challenge in the field of cybersecurity. As these networks advance, the tactics used to counter them must also develop. The instances showcased in this article illustrate the vastness and intricacy of botnets. The most effective strategy is to deny all connections by default and permit only those from thoroughly vetted and highlighted IP addresses.

Cybersecurity professionals must remain vigilant and equipped with advanced tools and knowledge to stay ahead in this perpetual battle against cyber threats.