Use Searching Engines to Hunt For Threat Actors

Background

Hi folks, today, I'd like to discuss how to leverage search engines to identify vulnerable servers used by threat actors. These actors often employ multiple servers for various purposes, such as phishing infrastructure, command and control (C2) infrastructure, and tool/payload servers. Due to poor operational security (OPSEC) or budget constraints, some may even use a single server for multiple functions.

People make mistakes, including threat actors. While they may employ advanced C2 frameworks, custom C2 profiles, redirectors, legitimate domains and certificates, and evasive tradecraft, a single mistake, such as an open directory misconfiguration, can undermine their entire effort.

As I am not a threat-hunting expert, my approach to identifying threat actors' infrastructure may not be exhaustive. However, in this article, I will demonstrate how to use the search engine Quake (https://quake.360.net/) to locate these misconfigured (Open Directory) infrastructures and provide an analysis of one specific case.

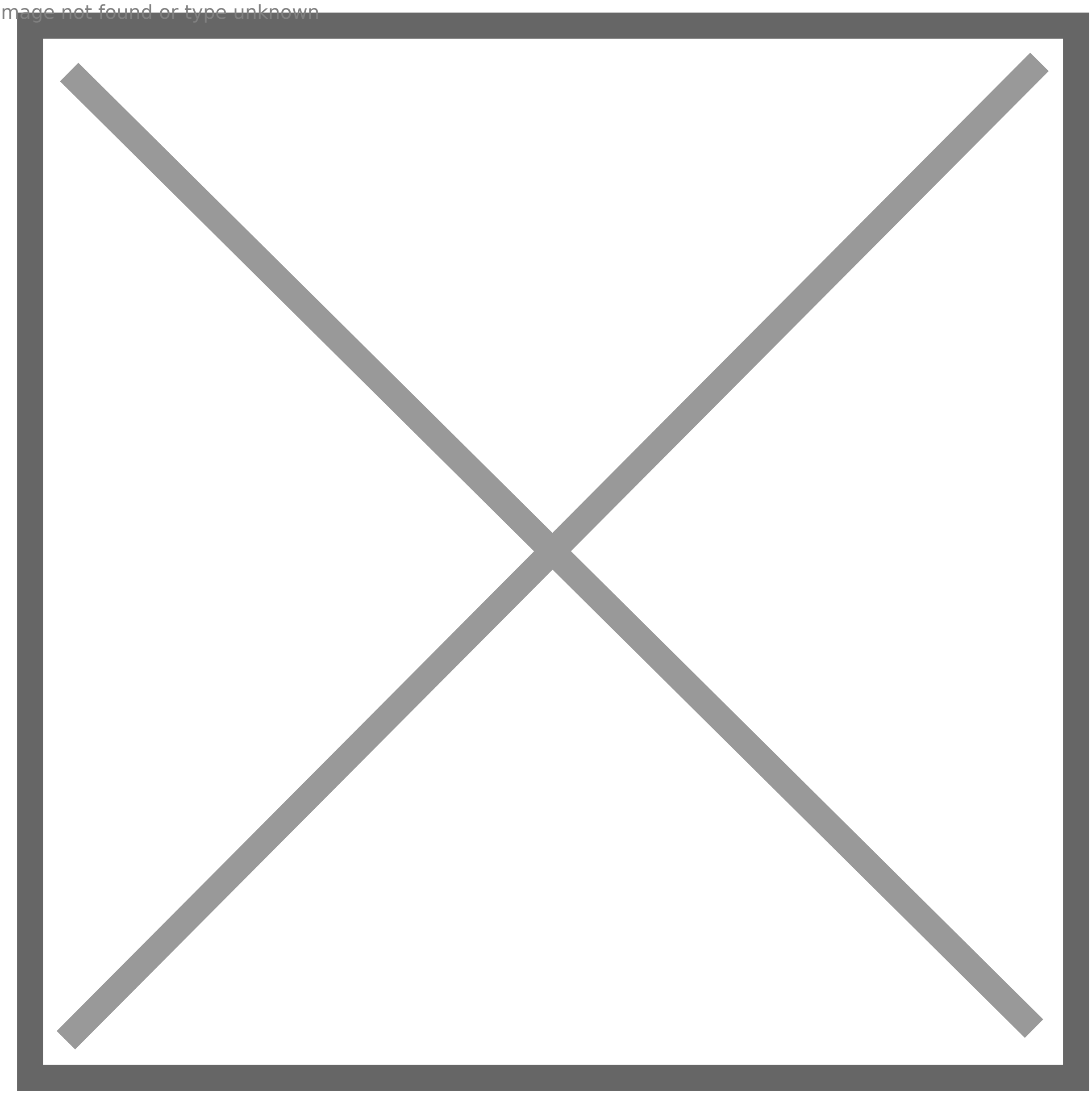
Motivation

As a red team operator, I fully understand the importance of OPSEC. Although I am not a threat-hunting expert, utilizing threat intelligence to track and locate other hackers and observing their mistakes can help enhance my own OPSEC awareness, allowing me to avoid low-level mistakes. Moreover, whether we are red team or blue team operators, our common goal is to make cyberspace safer.

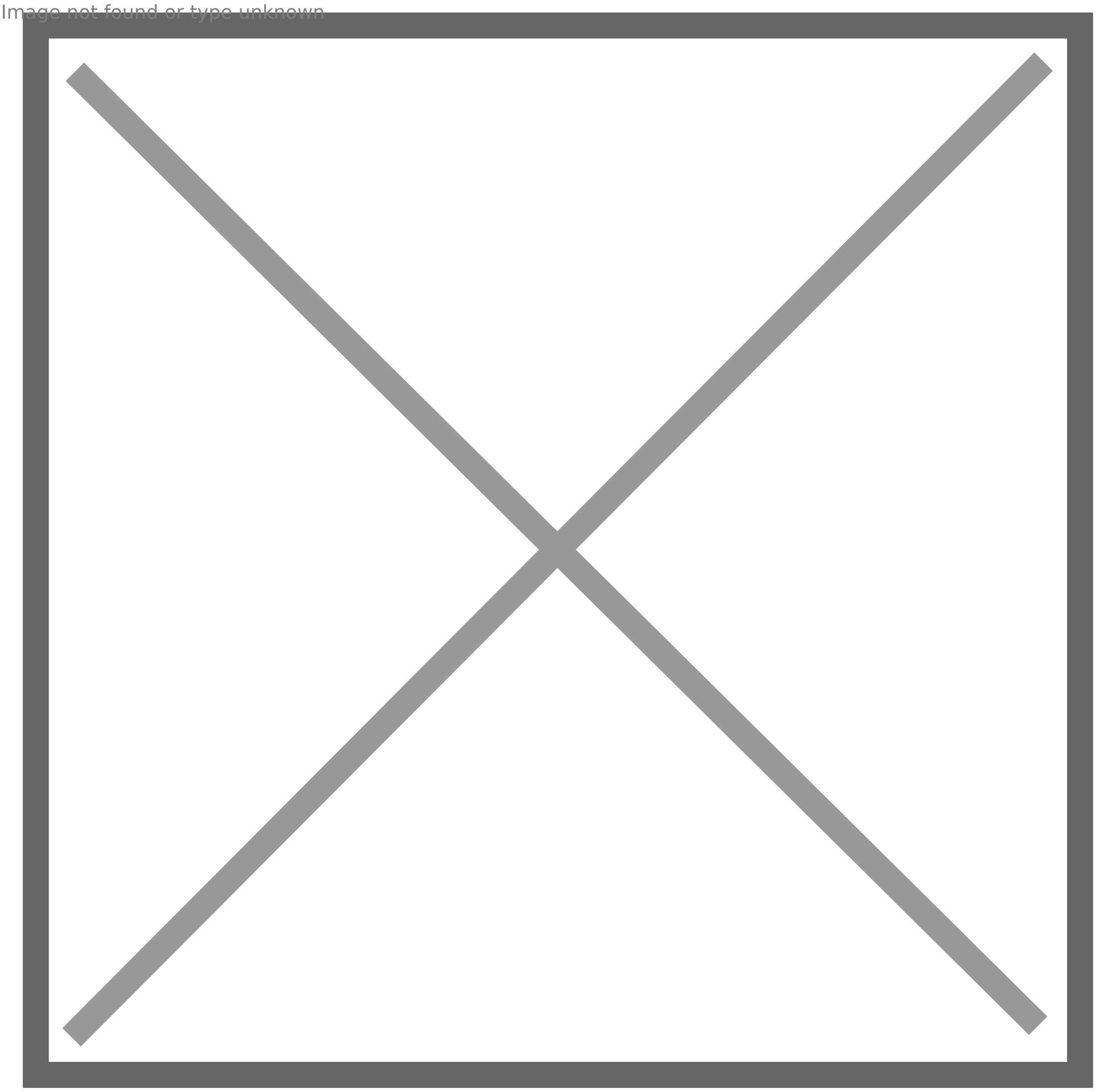
Threat actors who engage in malicious cyber activities should not have a foothold in cyberspace. While we may not possess law enforcement authority, we can at least expose their activities and warn others about their danger and existence.

The threat intelligence community boasts numerous outstanding threat hunters, such as Michael Koczwara, whose articles have provided me with significant insights. These threat hunters expose threat actors’ infrastructure IPs and domain names, assisting in enriching blacklists for both individuals and cybersecurity products. This article (https://bank-security.medium.com/hunting-cobalt-strike-servers-385c5bedda7b) explains how to use different methods, such as default Cobalt Strike certificates and default 404 responses, to search for Cobalt Strike servers on the internet using the Shodan search engine. Other articles, like https://michaelkoczwara.medium.com/hunting-c2-with-shodan-223ca250d06f, analyze the characteristics of C2 servers beyond Cobalt Strike and how to locate them using search engines.

Image not found or type unknown



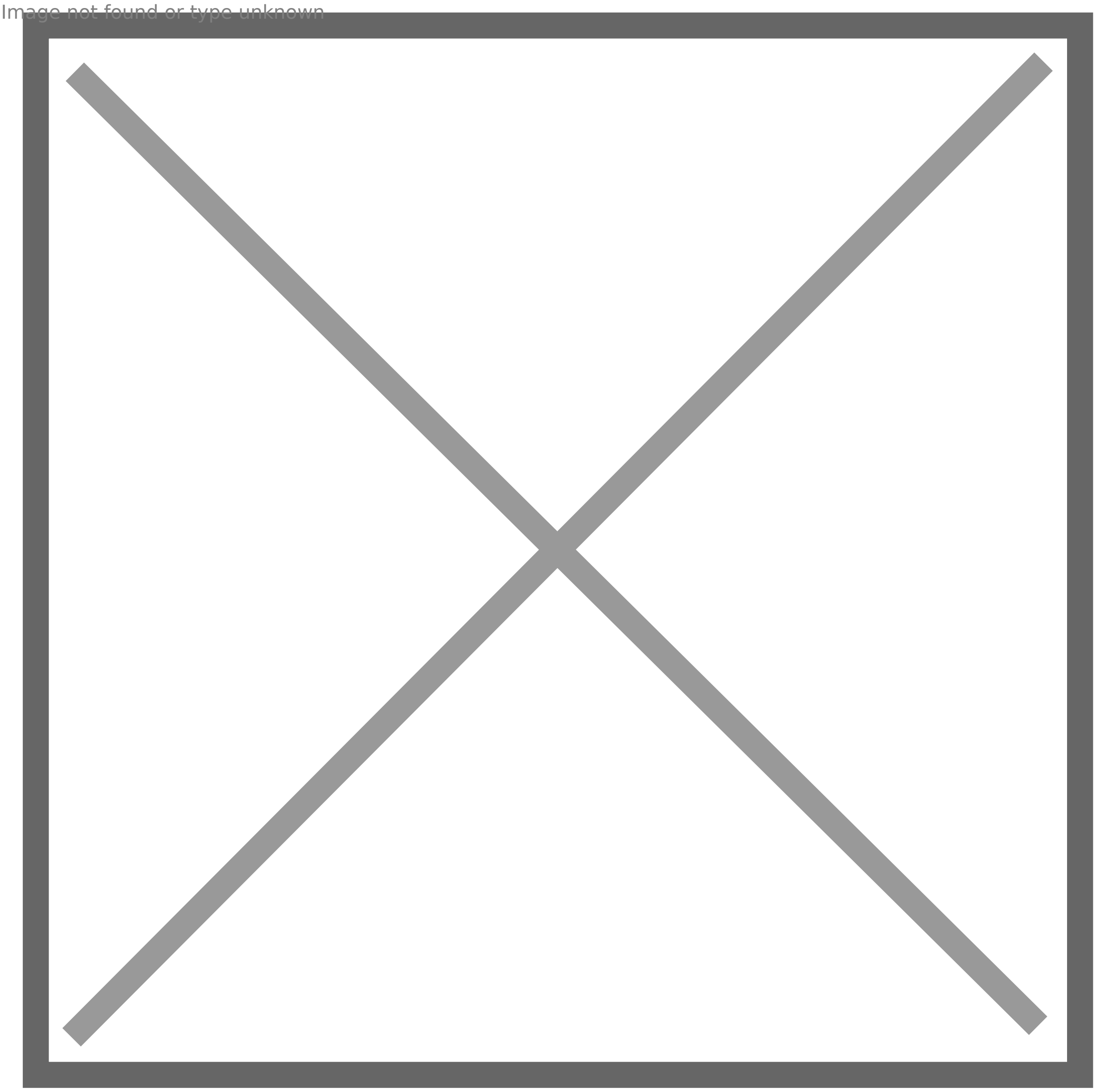
Building on this foundation, we can compile a list of threat actor servers and feed it to security products, automating the isolation of communication with these malicious servers and alerting people to their existence. However, since most of us do not have law enforcement powers, our ability to take further action is limited. That said, if threat actors make low-level mistakes like open directory, we can counter them more effectively, for example, by gathering more comprehensive evidence, analyzing malicious file samples, and potentially identifying the threat actors based on the downloadable files.



Searching Engine Stuff

There is a wide variety of internet asset search engines available, such as Shodan, Censys, Zoomeye, and Fofa. However, when it comes to locating threat actor servers with open directory configurations, my personal choice is the Quake search engine. Quake's syntax supports keyword searches based on HTTP titles and responses. For sites with open directory configurations, the webpage title is typically "**Directory listing for /**" or "**Index of /**". The open directory page title is primarily determined by the web server in use. For instance, if it's an Apache2 server, the title would be "Index of /", and if it's a Python HTTP server, the title would be "Directory listing for /".

Image not found or type unknown

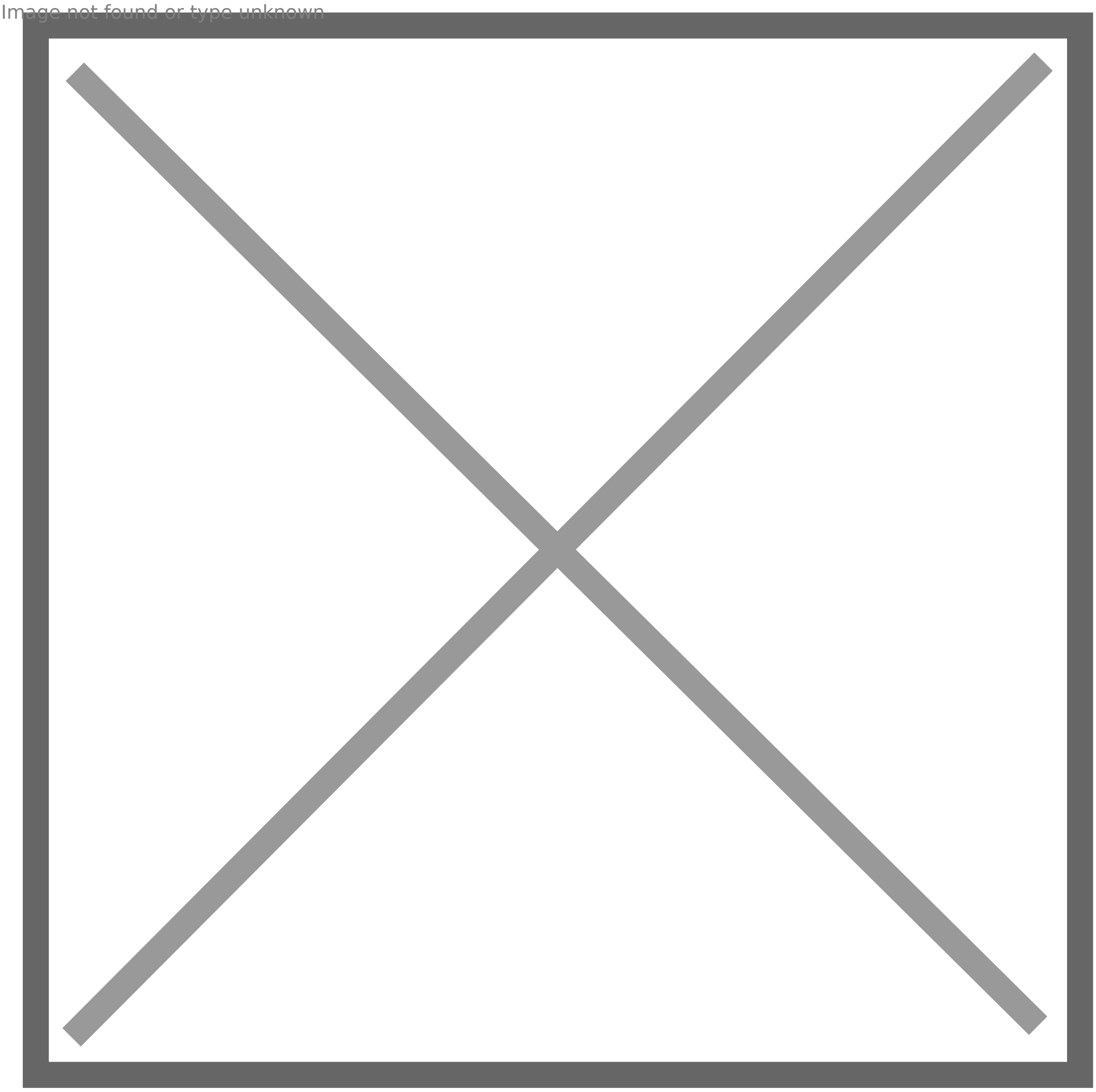


Threat actors tend to prefer setting up temporary HTTP file servers using **Python** due to its simplicity and convenience. However, sometimes they forget to shut down the Python HTTP server promptly, leaving traces we can track. Filtering by HTTP response is relatively straightforward; we can enter the name of any security tool or malware, such as Mimikatz, Cobalt Strike, or Rubeus.

Some query examples:

title: “Directory listing for /” and response:”cobaltstrike”

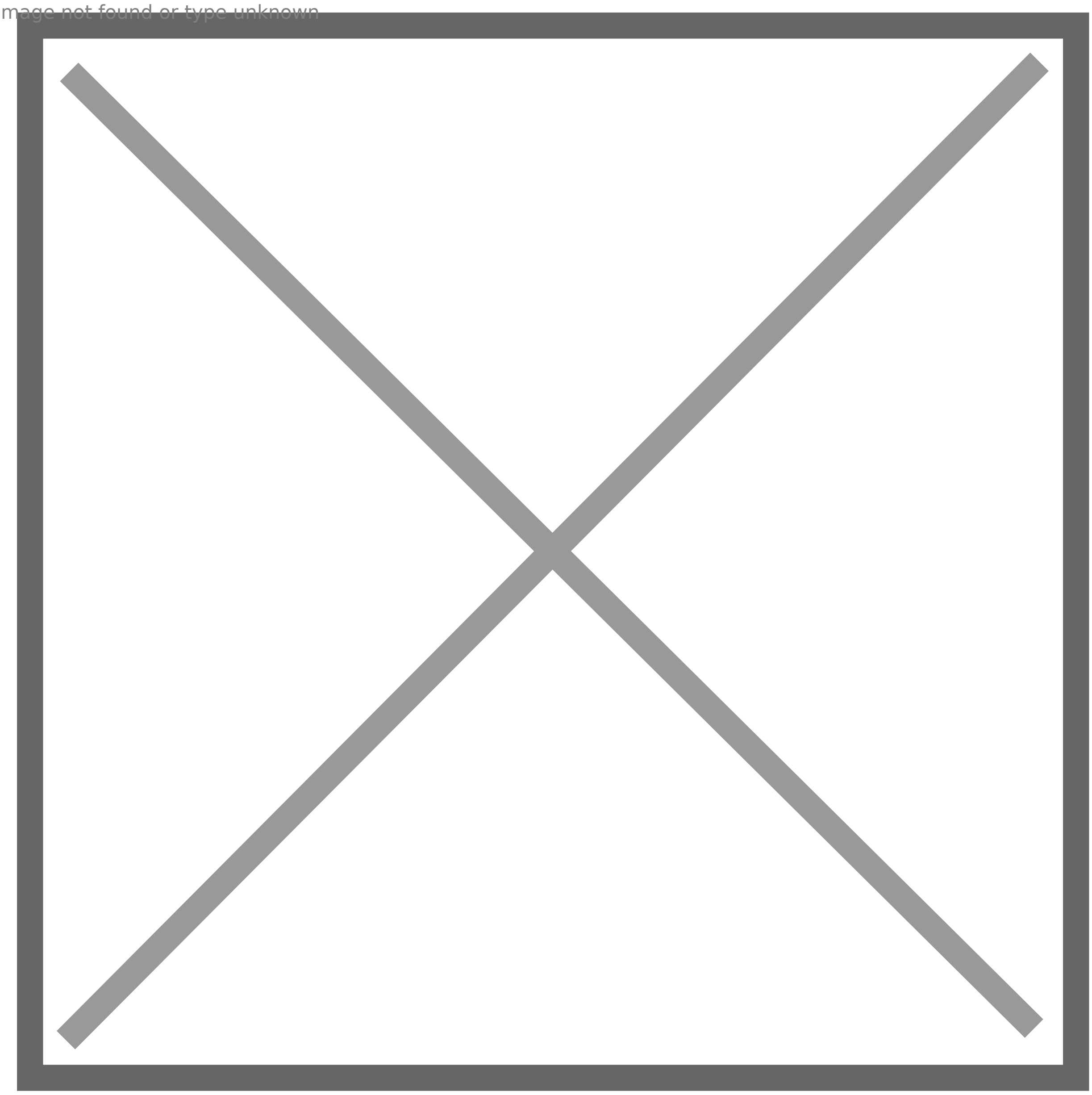
Based on the provided screenshot, we can see that 67 servers are currently or have previously host the Cobalt Strike C2 framework tool for threat actors to download. I speculate that the majority of these Cobalt Strike instances are likely to be unauthorized copies.



title: “Directory listing for /” and response:”mimikatz”

49 servers are currently or have previously host the hack tool mimikatz for threat actors to download.

Image not found or type unknown

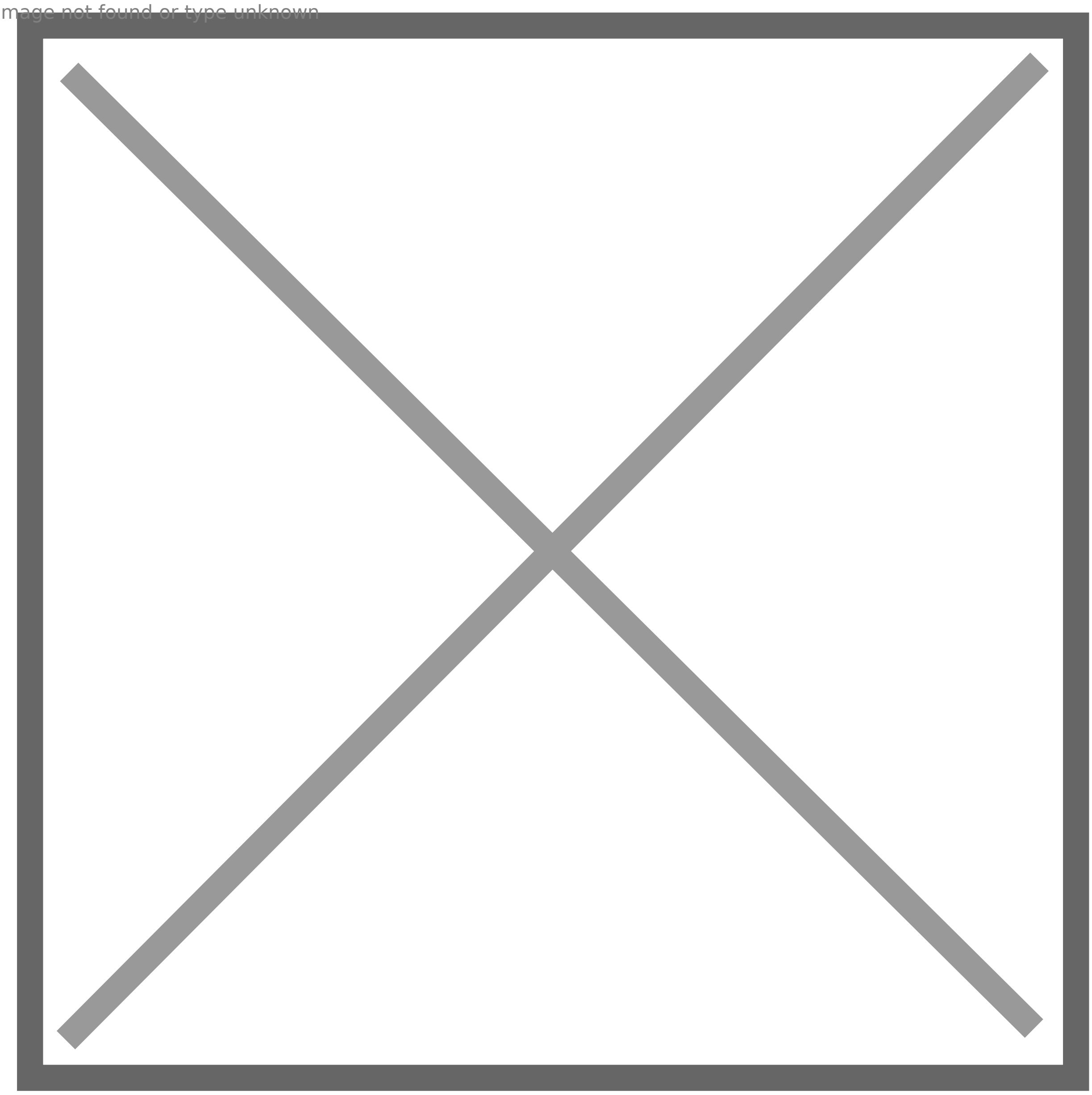


title: “Directory listing for /” and response:”exp”

We use this query to search servers that are currently or have previously host vulnerability exploits.

Take a close look at this server, from the preview, we can see a .ovpn file. It could be used to connect to the threat actor’s internal network (Or it is a victim’s .ovpn file). Unfortunately, the threat actor already shut down the Python HTTP server by the time I found it.

Image not found or type unknown



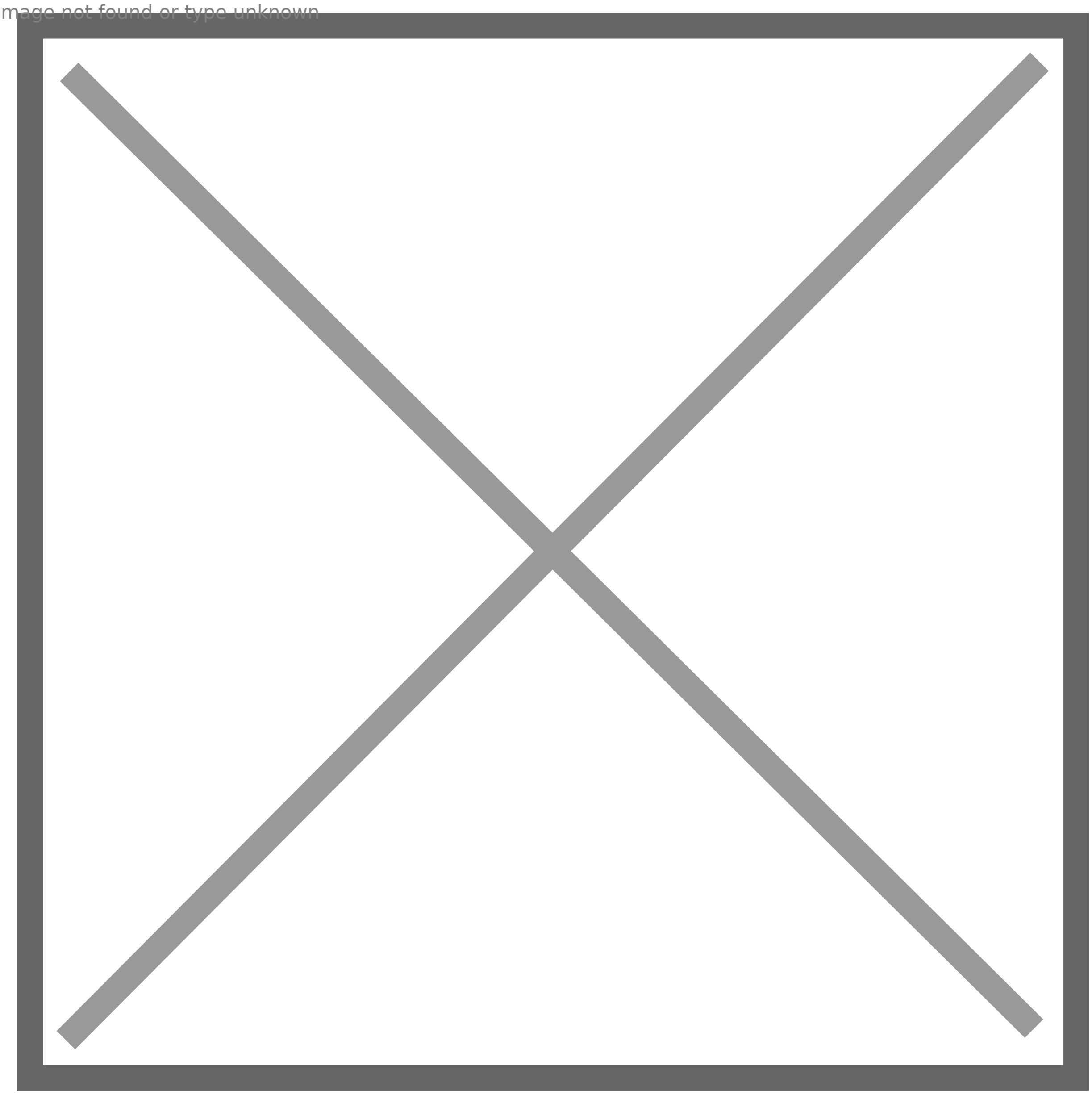
title: “Directory listing for /” and response:”lsass”

We can use this query to search servers that are currently or have previously host lsass dump. We can see server **80.85.156.184** was used by a threat actor to save lsass dump file.

Image not found or type unknown

Currently, this server is not accessible, some other threat hunters already noticed this server before.

Image not found or type unknown

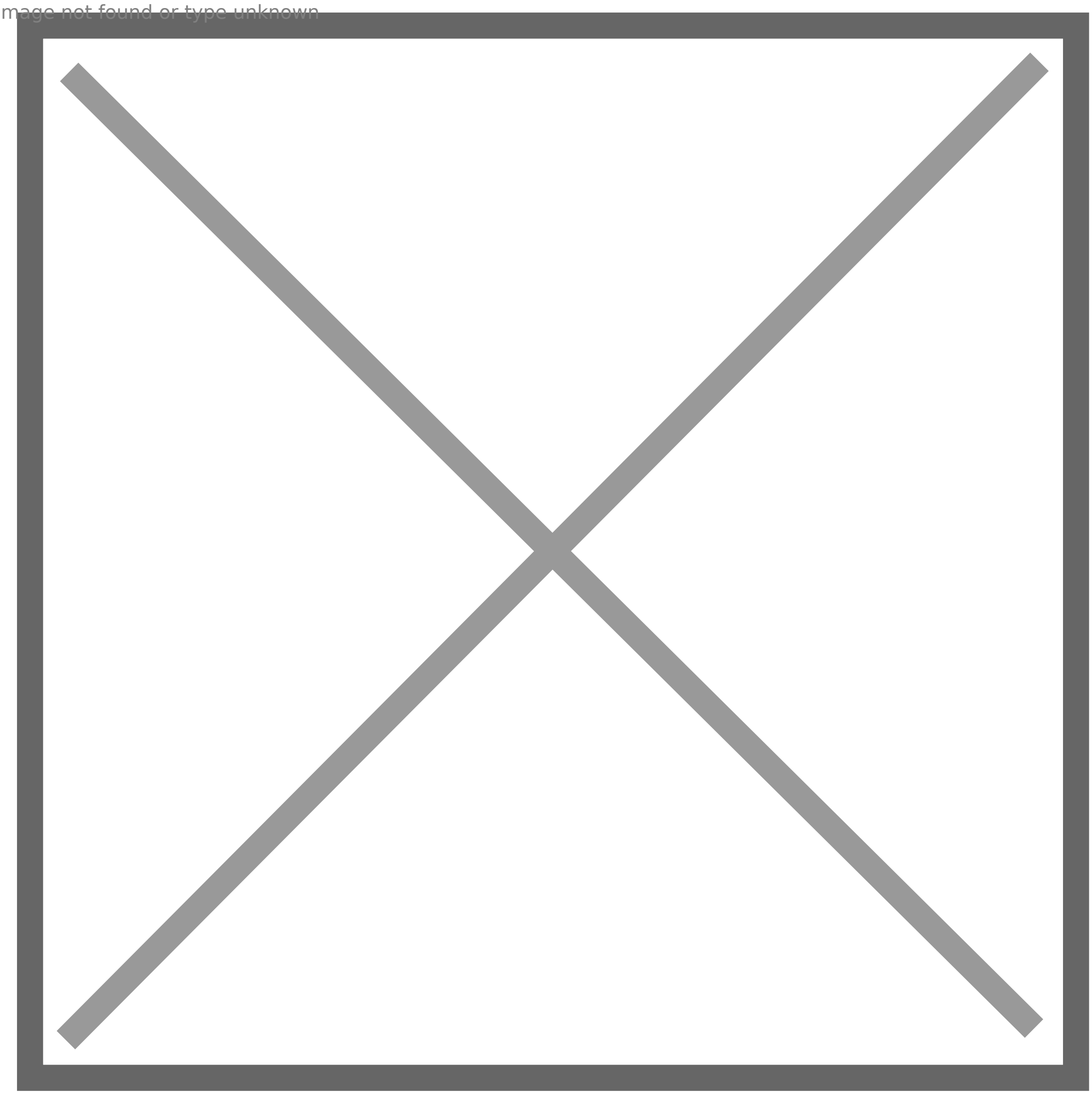


In conclusion, we can flexibly adjust the HTTP response keywords for endless search possibilities. However, the ultimate goal remains the same: to locate threat actors’ hacking tools, malware, vulnerability exploits, and “spoils” obtained from their victims. If we happen to find their personal files, like an .ssh private key, a .ovpn file, document-based files, their bad day go to worse day : D

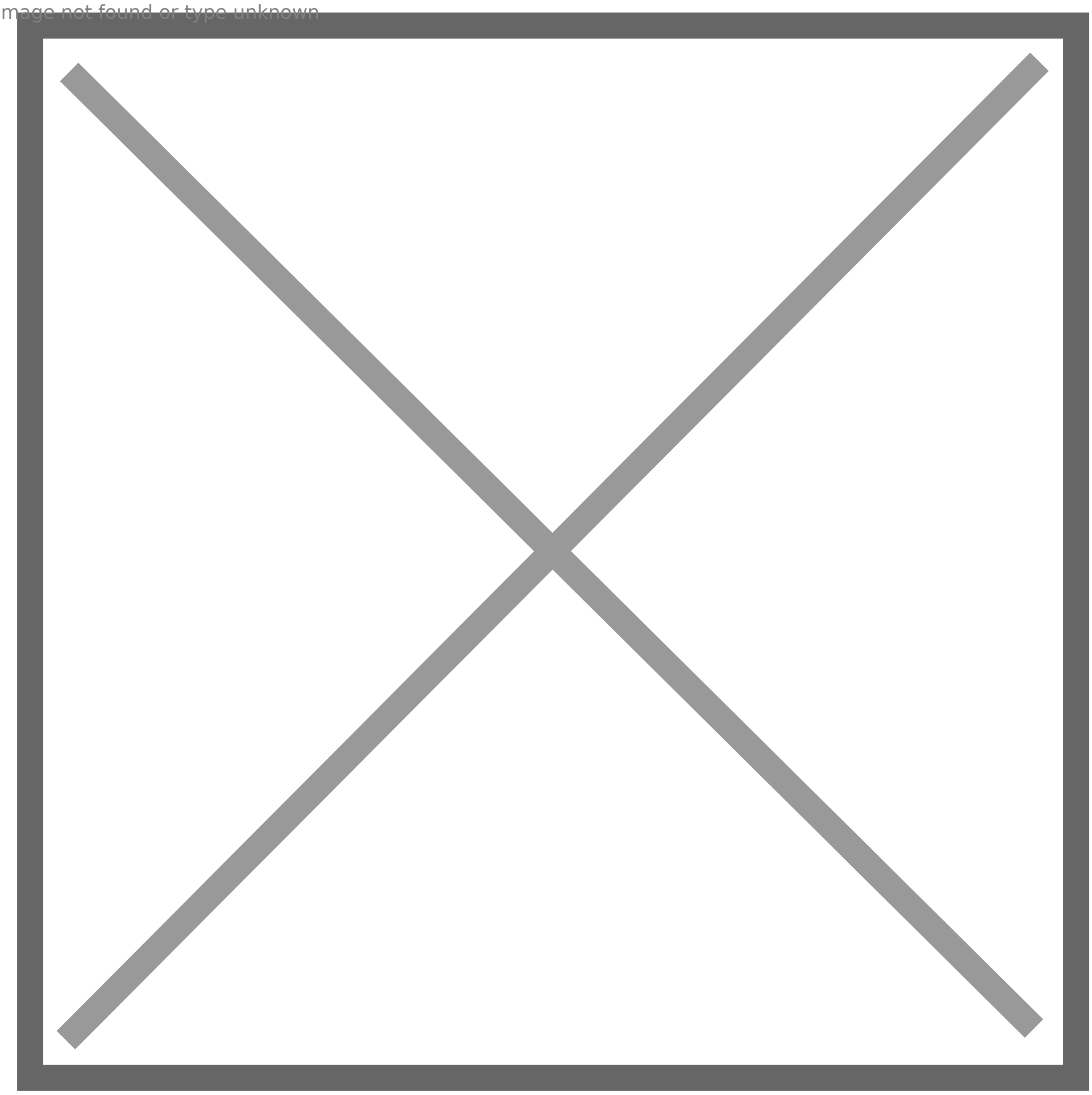
Case Analysis

Among the improperly configured servers discovered, I would like to share one of the most interesting examples (http://81.68.227.204:8000/). If this server is not a honeypot, then this hacker certainly has **zero OPSEC**. Upon visiting this URL, it becomes evident that the hacker is running a Python HTTP server at the root directory of their C drive, as we can see directories such as Windows, Users, and Program Files. We can easily find some hack tools, such as CS4.4 K8 (Downloaded from https://github.com/k8gege/Aggressor/releases/tag/cs), □□□(scanner).

Image not found or type unknown



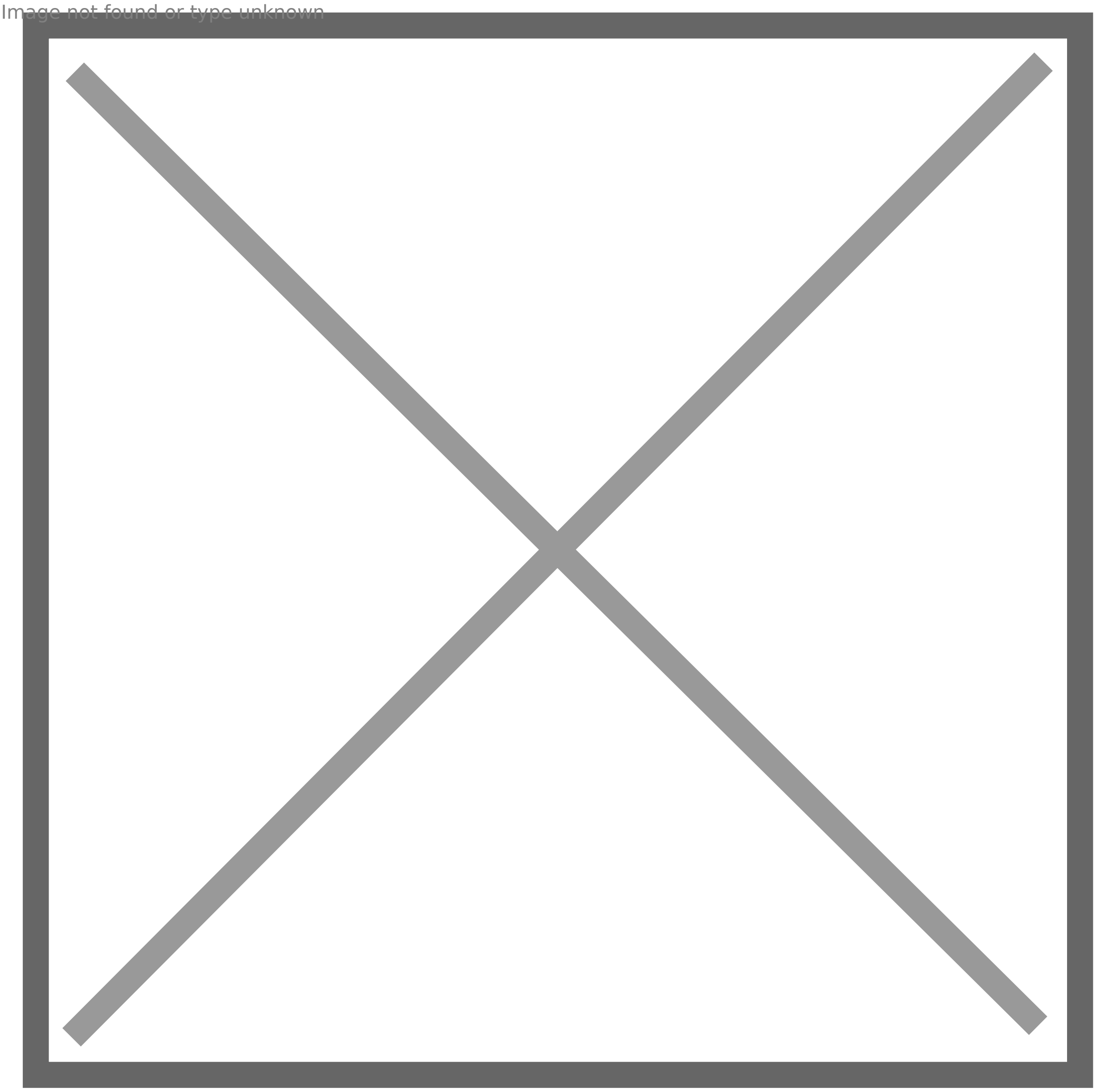
This hacker seems to have a romantic side, as there is a folder named ‘love’ in the C drive root directory, which contains a web animation as shown in the picture. It appears to be a small surprise prepared for his girlfriend, and from this, we can know her name. However, it’s worth noting that this information could also be fabricated.



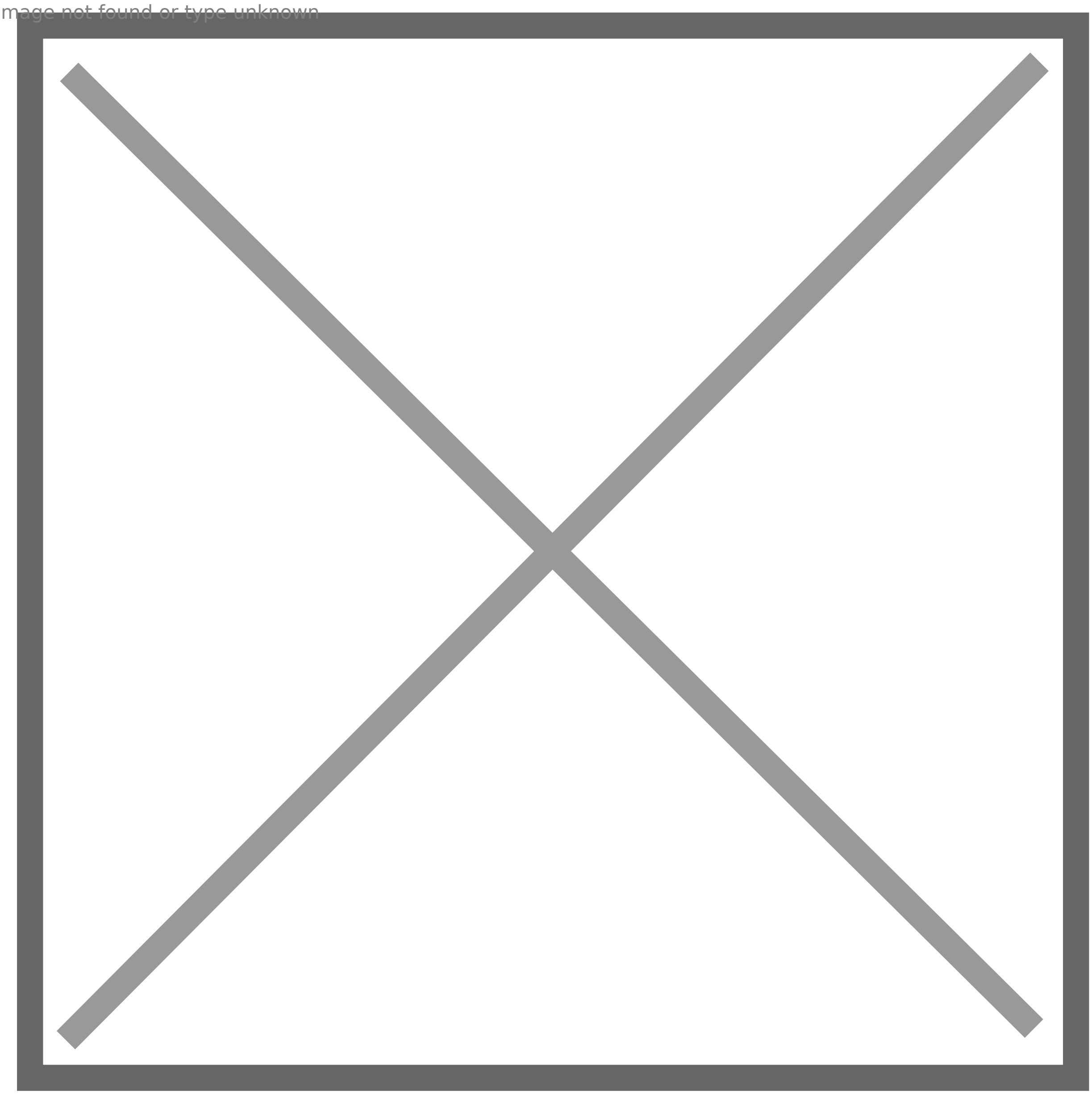
The hacker installed multiple programs, such as Redis, Python, GO language, Metasploit, QQ, etc. And this HTTP server was run as Administrator

Image not found or type unknown

We can see his Cobalt Strike team server’s log:



The hacker used some insecure commands like shell whoami, sleep 0.

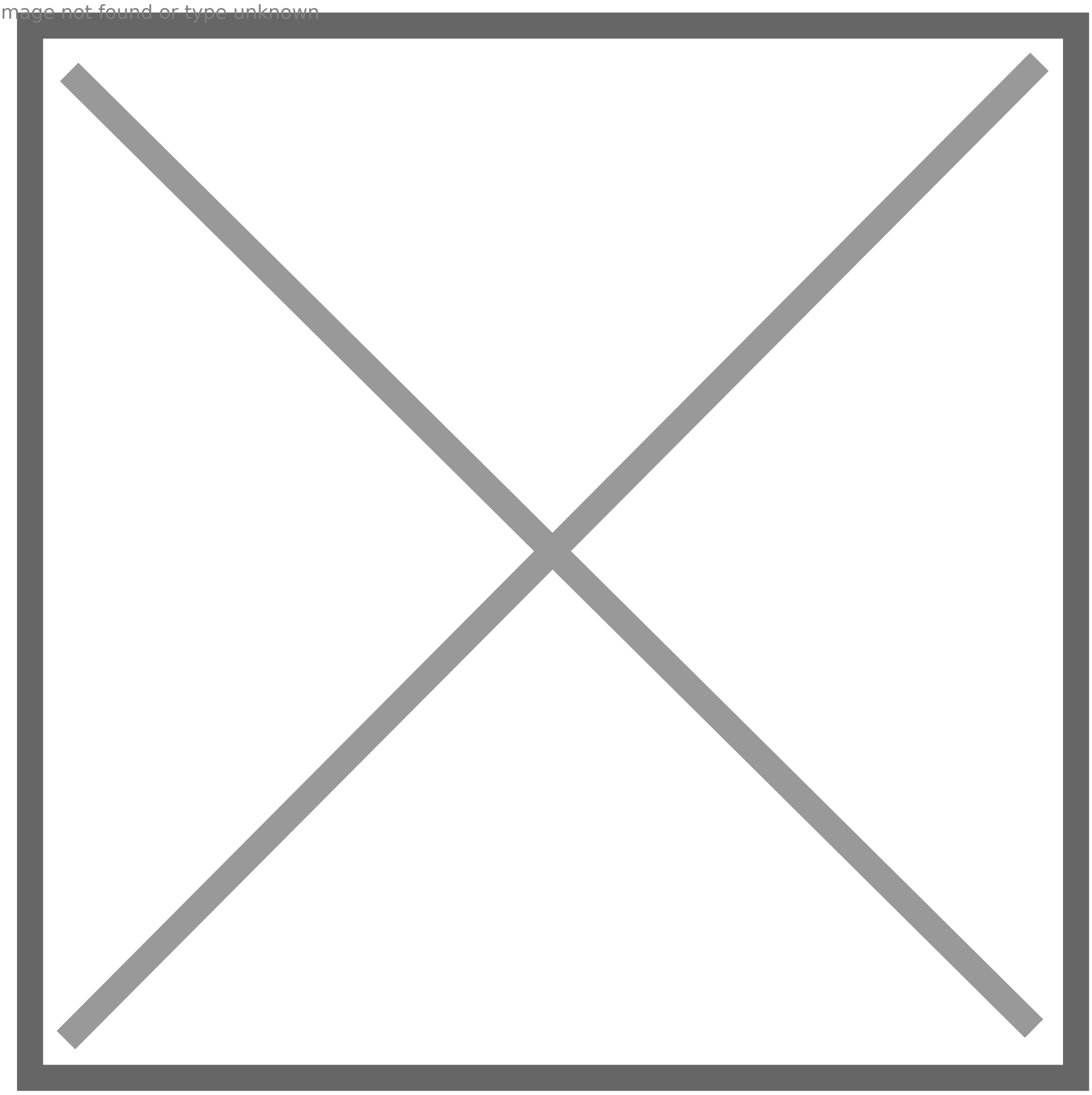


The hacker enabled **stager**, and we can see some victims he controlled.

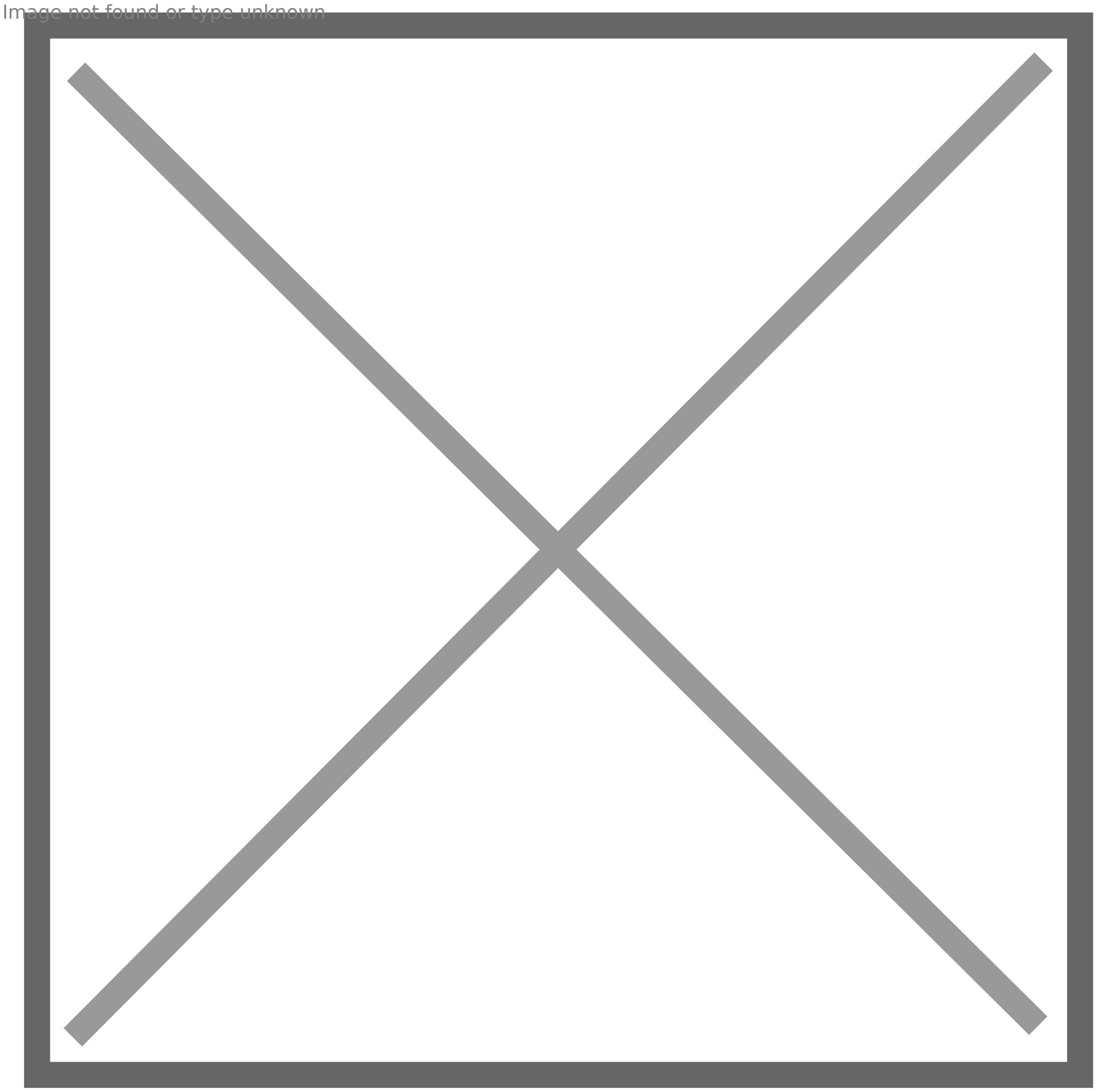
Image not found or type unknown

Image not found or type unknown

The hacker also uses this server as a challenge box by setting up a local lab.



The hacker deleted some other user accounts and their directories. The SID ends with 500, so it is a local administrator account. Maybe the account was created by other hackers?



From **ToDesk config file**, I found this hacker’s phone number, and I found his social media account, should I add him and say hello? : D

Image not found or type unknown

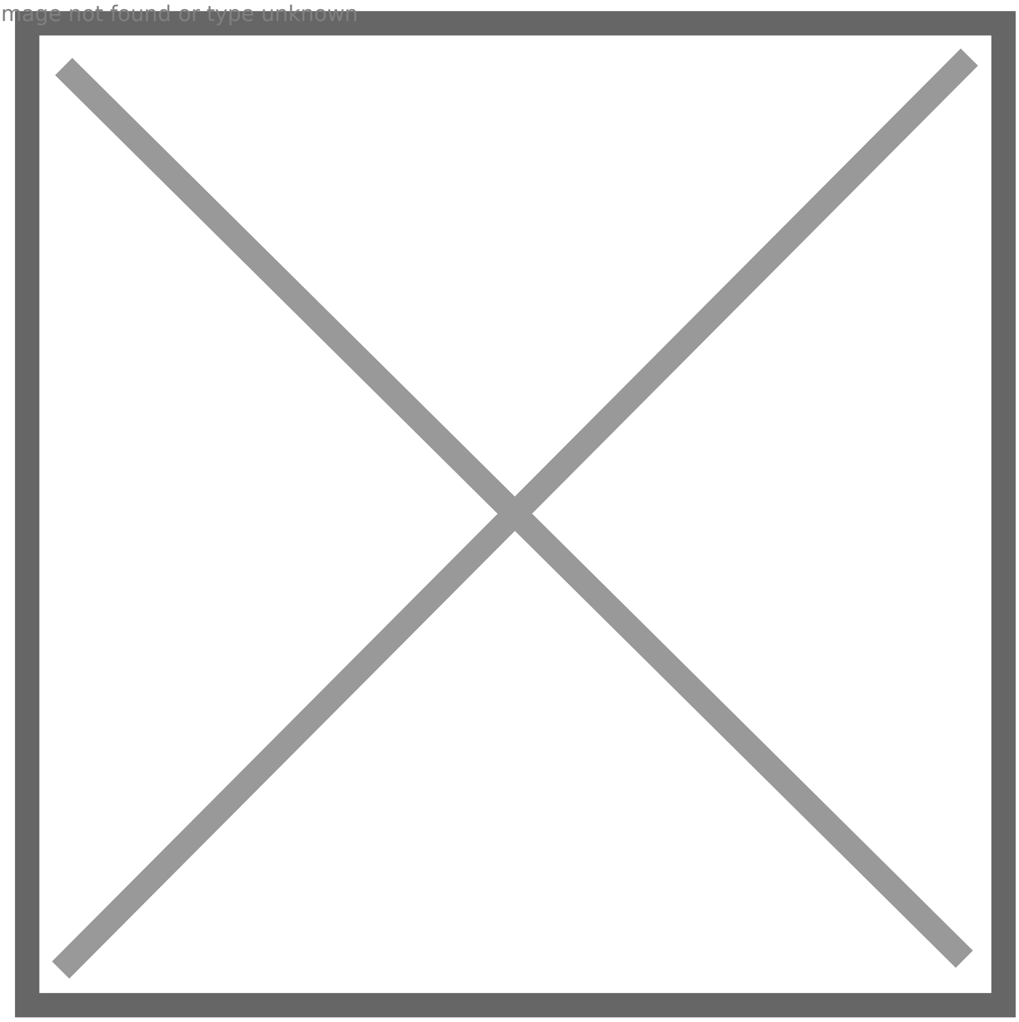
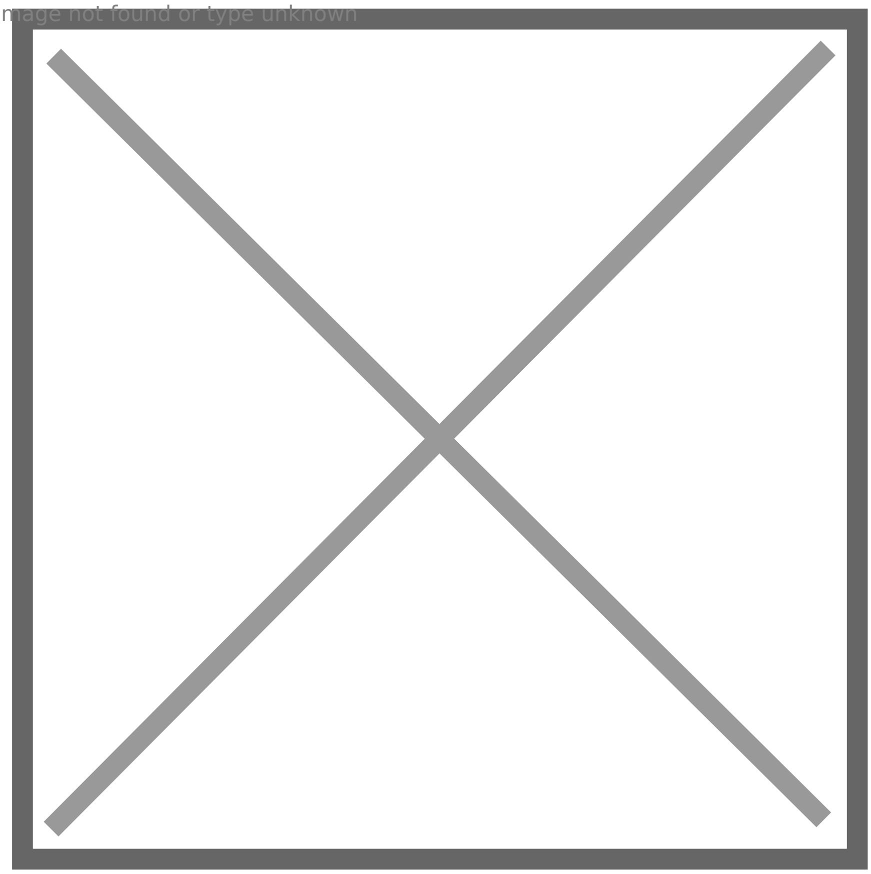


Image not found or type unknown



Alright, considering we have access to the entire C drive, on the surface, we are examining this open directory, but in reality, what we are doing is almost akin to local reconnaissance. Let’s directly jump to an issue that could lead to this hacker’s server being compromised.

The server is running IIS server, and we can access IIS directory. We find a file aspx.aspx, it is basically a one-liner webshell.

Image not found or type unknown

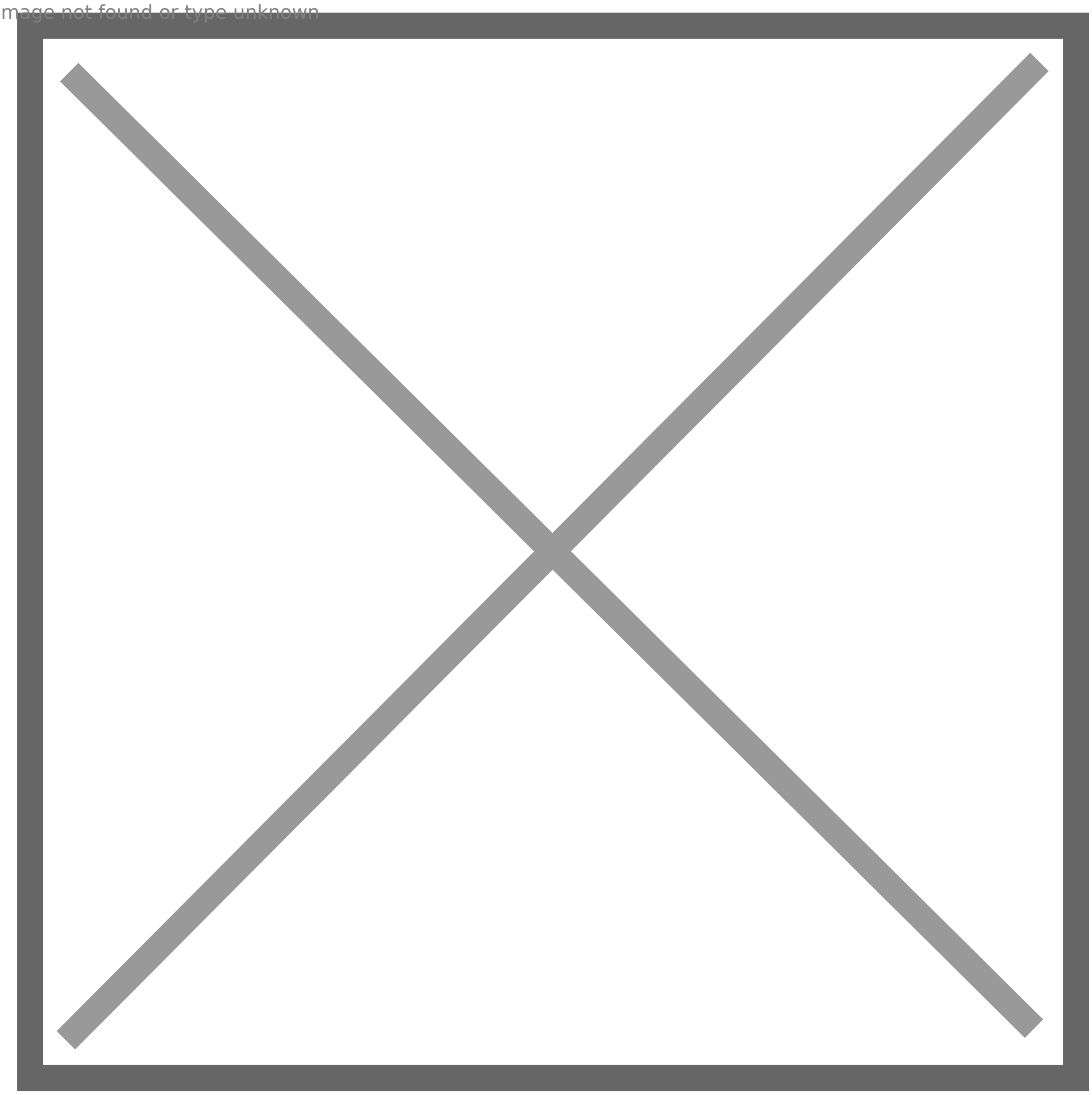
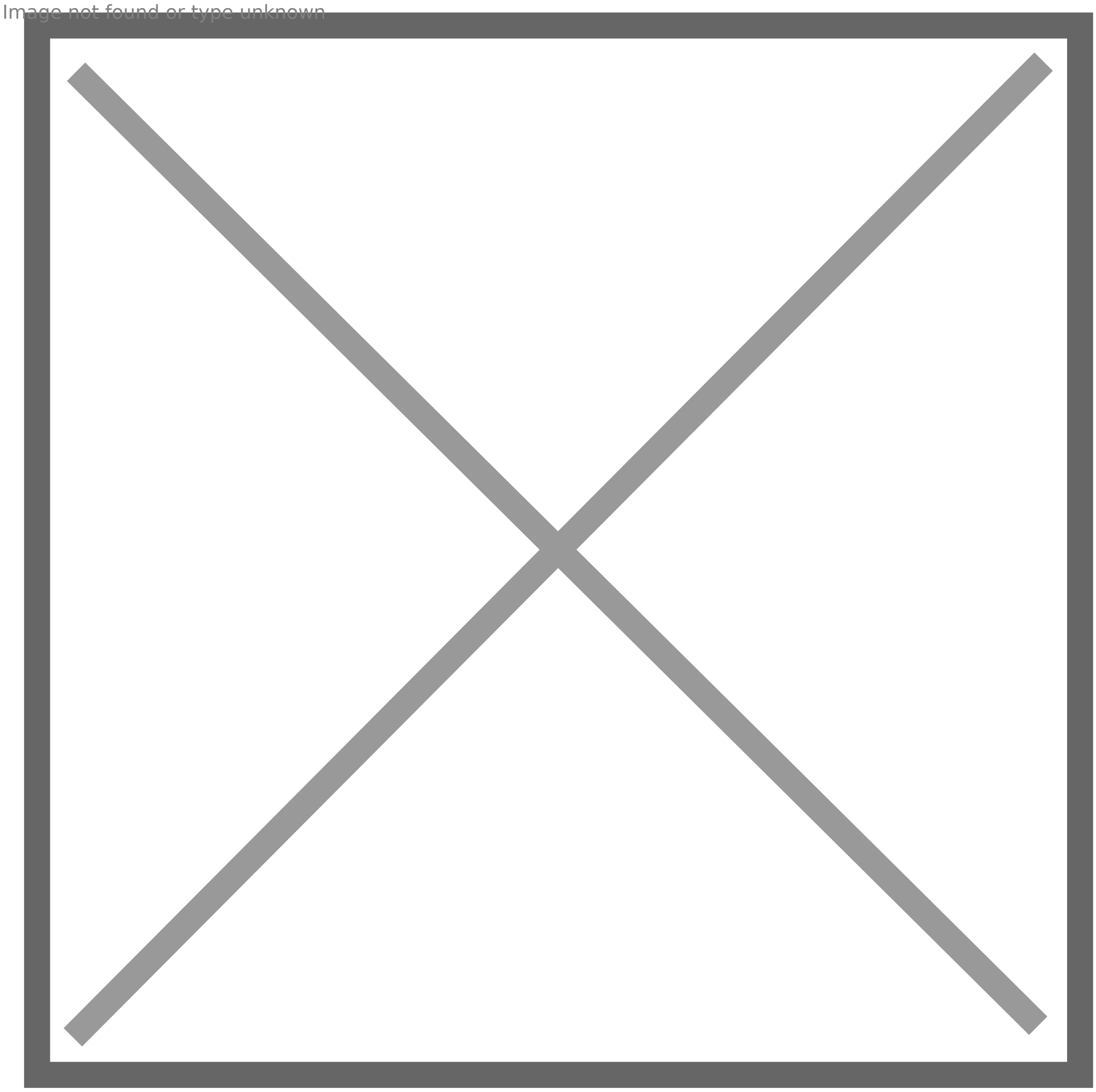


Image not found or type unknown

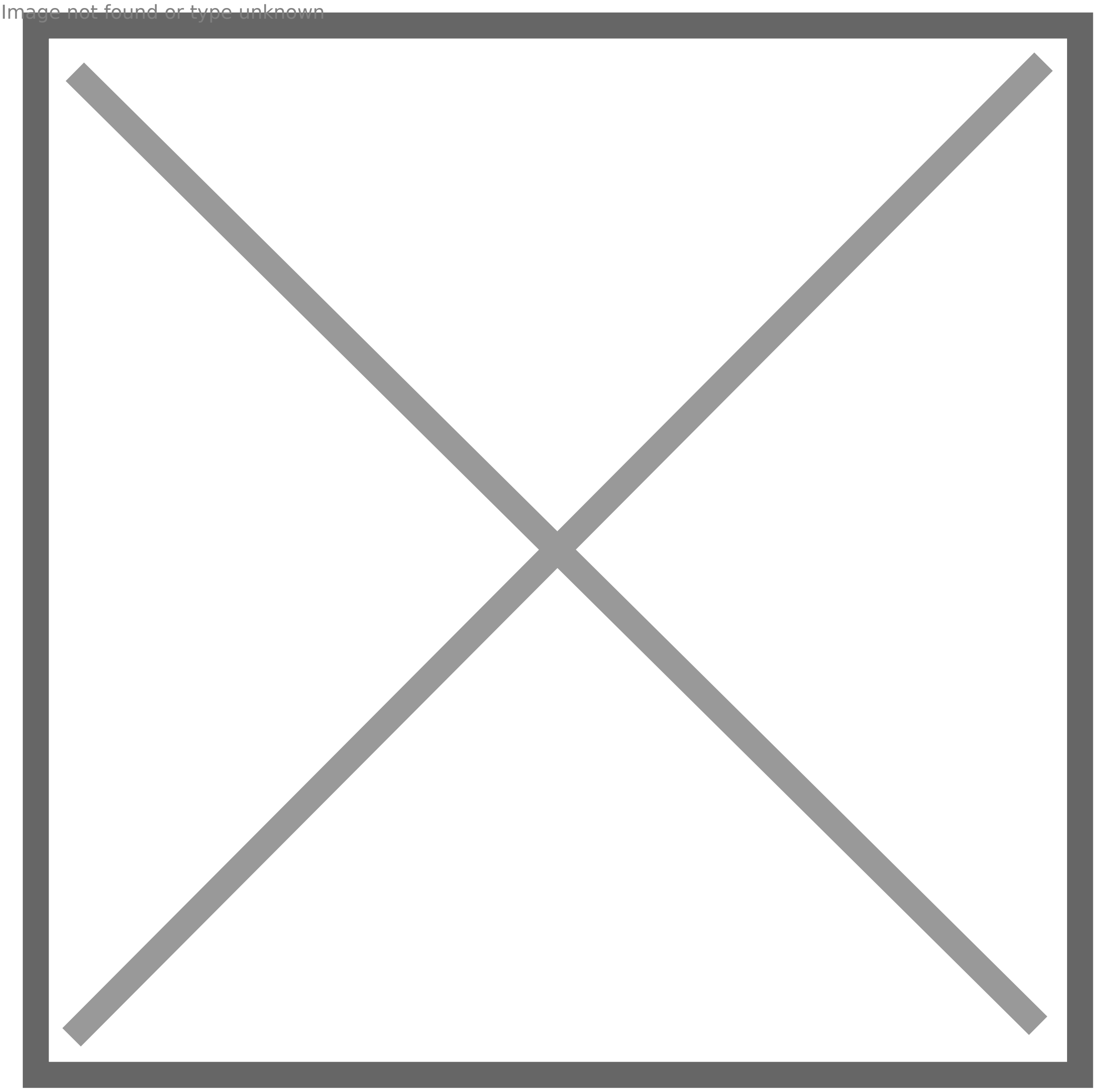


(Please stop trying “?chopper=whoami”, the payload is not this one, and you cannot see the output.)

I am not sure if the backdoor is used by the hacker for remote management, or other hackers already compromised this server and left this backdoor lol

Though he is a threat actor, we still should not attempt to exploit it.

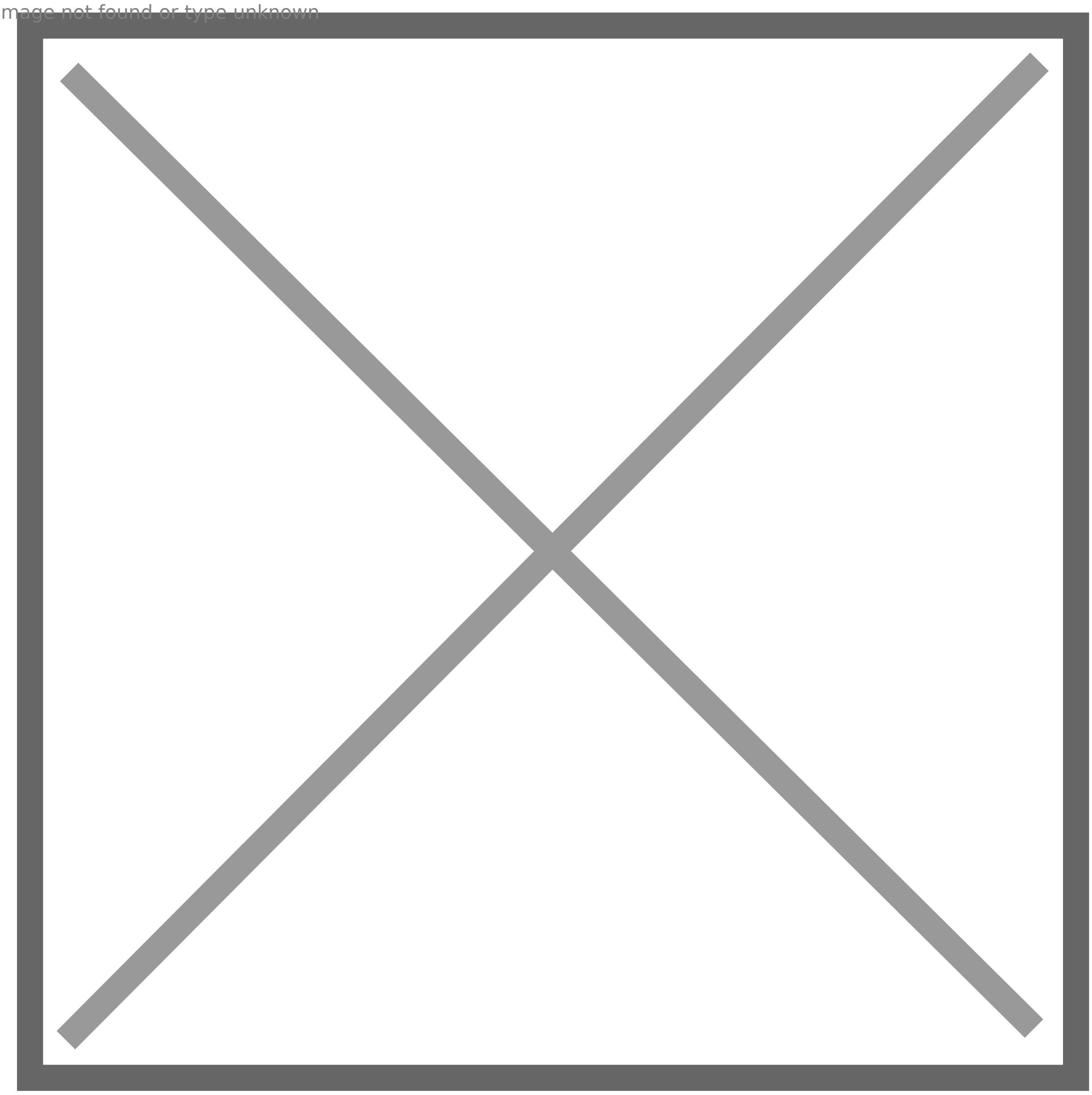
Finally, let’s use Shodan to analyze this server:



It opens multiple ports, and he did not enforce authentication for Redis. Some people already noticed this unauthenticated Redis server and connected to it.

Image not found or type unknown

Image not found or type unknown



Summary

Thank you for reading my article, and I’d like to express my gratitude to the threat hunters who have inspired me. Threat intelligence is a fascinating field that has greatly aided in enhancing my red team skills. Together, let’s work towards exposing threat actors and countering them within the bounds of authorization (Directly attacking these servers is not legitimate). Happy hunting!

References

https://michaelkoczwara.medium.com/adversaries-infrastructure-ransomware-groups-apts-and-red-teams-7a6dd761c50e

https://michaelkoczwara.medium.com/hunting-c2-with-shodan-223ca250d06f

https://bank-security.medium.com/hunting-cobalt-strike-servers-385c5bedda7b

https://quake.360.net/quake/

https://twitter.com/MichalKoczwara?source=post_page-----74be52976e9f--------------------------------

https://twitter.com/1ZRR4H/status/1631466978132074498?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1631466978132074498%7Ctwgr%5E7a3a86883a64953451d0aa545b0d6d0a18594740%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fcdn.embedly.com%2Fwidgets%2Fmedia.html%3Ftype%3Dtext2Fhtmlkey%3Da19fcc184b9711e1b4764040d3dc5c07schema%3Dtwitterurl%3Dhttps3A%2F%2Ftwitter.com%2F1ZRR4H%2Fstatus%2F1631466978132074498image%3Dhttps3A%2F%2Fi.embed.ly%2F1%2Fimage3Furl3Dhttps253A252F252Fabs.twimg.com252Ferrors252Flogo46x38.png26key3Da19fcc184b9711e1b4764040d3dc5c07

Revision #1
Created 28 February 2024 18:26:34 by winslow
Updated 28 February 2024 19:38:48 by winslow