Multi-Staged JSOutProx RAT Targets Indian Co-operative Banks and Finance Companies

blogs.quickheal.com/multi-staged-jsoutprox-rat-targets-indian-cooperative-banks-and-finance-companies/

October 21, 2021

Quick Heal Security Labs has been monitoring various attack campaigns using JSOutProx RAT against different SMBs in the BFSI sector since January 2021. We have found multiple payloads being dropped at different stages of its operations. Although the RAT campaigns have also been previously reported on other countries, those targeting Indian companies are operated through separate C2 domains. Let's dig deeper into the working of this targeted attack.

JSOutProx is a modular JScript-based RAT delivered to the user as a .hta file and first executed by the mshta.exe process. The initial attack vector is a spear-phishing email with a compressed attachment having a ".hta" file with a file name related to a financial transaction. The attachments have a double-extension-like format, for example "_pdf.zip", "_xlsx.7z", "_xls.zip", "_docx.zip", "_eml.zip", "_jpeg.zip", "_txt.zip" etc.

Stages

The RAT is delivered in 2 stages. In the first stage, a minimal version is provided with some functionalities stripped. In the second stage, a bigger version of the sample is delivered, which, apart from the existing functionalities of the first stage rat, has support for additional functions and plugins as well.

Initial Infection Vector

Spear Phishing emails are sent to targeted individuals who are employees of small finance banks from India. We believe the threat actor adds more targets to his list by stealing the email contacts of its victims. We have observed multiple campaigns from Jan 2021 to June 2021 where emails were sent to hundreds of targets in a single day. Sometimes, various emails with different attachment names are sent to a single target to increase the chances of the user downloading and opening the attachment file.

Obfuscation

The RAT was first observed two years ago, in 2019. Since then, the RAT has upgraded with new commands, more functionality, and increased obfuscation. The recent JScript files consist of more than one MB of obfuscated code, a vast array of base64-like strings, malware's configuration data, and an rc4 string decryption function. The obfuscation pattern remains the same as the older samples and is the same for both stages of RAT samples.

RAT Configuration Data

Once the configuration data is decrypted, we get a glimpse of the malware's capabilities. The "BaseUrl" field points to the C2 domain and port number it communicates using the HTTP protocol. "Password" field is used while downloading plugins and assemblies from C2. "Tag" field contains campaign ID. The first samples, which were reported two years back, had the tag name "JSOutProx," and hence it was named as such. Below is a list of initial fields present in the decrypted configuration data of one RAT sample.

Fig 1: RAT configuration fields

Few new fields like “ViewOnly” were seen in the recent samples, which allows the controller to monitor the victim to gather victim info and not write or execute anything on the machine. This ensures the malware is not creating any noisy events until the attacker decides to initiate the attack. Most of the initial fields are common in both stages.

First Stage RAT

The first stage RAT is a .hta file and executed by the mshta.exe process. It can create entries in registry and startup, create or terminate a process, perform file operations, download plugins, etc. It can also generate some mouse and keyboard operations using PowerShell scripts in the target machine through “ScreenPShell” commands, as mentioned in the below screenshot.

Fig 2: Few RAT functions for screen operations and shellcode execution

Following are the essential plugins supported and their functionalities:

- **InfoPlugin ->** Collects and sends victim machine info to C2.
- **File plugin ->** Perform all file system operations.
- **ProcessPlugin ->** Collects process information, creates or terminates a process.
- **ScreenPShellPlugin ->** Perform mouse and keyboard operations using PowerShell scripts.

- **ShellPlugin ->** In this, the “ShellExecute” option uses the ShellExecute method present in the object of Shell. Application. If the user has admin privileges, do call to ShellExecute method. If the command fails, then it tries to disable AntiSPyware of Windows Defender from Registry. If the user is non-Admin, it tries ShellExecute with elevated permissions using the ‘runas’ flag. The “get output” option uses the Run method present in the object of WScript.Shell. It saves the output in a local file. It also fetches the keyboard language/codepage of the user to format the output correctly.

Once the malware is executed, it communicates with C2, which first responds with a PowerShell script to capture the screenshot and save it in the temp directory. There are previous reports of the same PowerShell script being used in attacks against banks in the UK. Following is the PowerShell script:

```
c:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -exec bypass -c "
    [Reflection.Assembly]::LoadWithPartialName('System.Drawing.Imaging');
    [Reflection.Assembly]::LoadWithPartialName('System.Windows.Forms');
    [Reflection.Assembly]::LoadWithPartialName('System.Drawing');
        $s = [System.Windows.Forms.SystemInformation]::VirtualScreen;
        $bounds = [Drawing.Rectangle]::FromLTRB(0,0,$s.Width,$s.Height);
        $bmp = New-Object Drawing.Bitmap $bounds.width,$bounds.height;
        $g = [Drawing.Graphics]::FromImage($bmp);
        $g.copyFromScreen($bounds.Location,[Drawing.Point]::Empty,$bounds.size);
        $codec = [Drawing.Imaging.ImageCodecInfo]::GetImageEncoders()
            |Where-Object{$_.FormatDescription -eq "JPEG"};
        $ep = New-Object Drawing.Imaging.EncoderParameters;
        $ep.Param[0] = New-Object Drawing.Imaging.EncoderParameter
            ([System.Drawing.Imaging.Encoder]::Quality,[long] 10);
        $ratio = [100 / 100];
        $newWidth = [int] ($bmp.Width*$ratio);
        $newHeight = [int] ($bmp.Height*$ratio);
        $temp = New-Object System.Drawing.Bitmap($newWidth, $newHeight);
        $g2 = [System.Drawing.Graphics]::FromImage($temp);
        $g2.DrawImage($bmp,0,0 , $newWidth, $newHeight);
                $bmp.Save(\"$temp%\\<rand_name>.tmp\",$codec,$ep);
        $g.Dispose();
        $g2.Dispose();
        $bmp.Dispose();
        $temp.Dispose();"
```

Fig 3: PowerShell Script fetched from C2

Second Stage RAT

The second stage RAT is dropped as a “.js” file in a startup or as a “.tmp” file in the %temp% folder and is executed using wscript.exe. It also has a different C2 than the first stage sample. The size of these samples is around three MB and has additional plugins support. The inclusion of DotUtil functions enables it to download and execute .NET assemblies in memory. Following are some of the DotUtil functions:

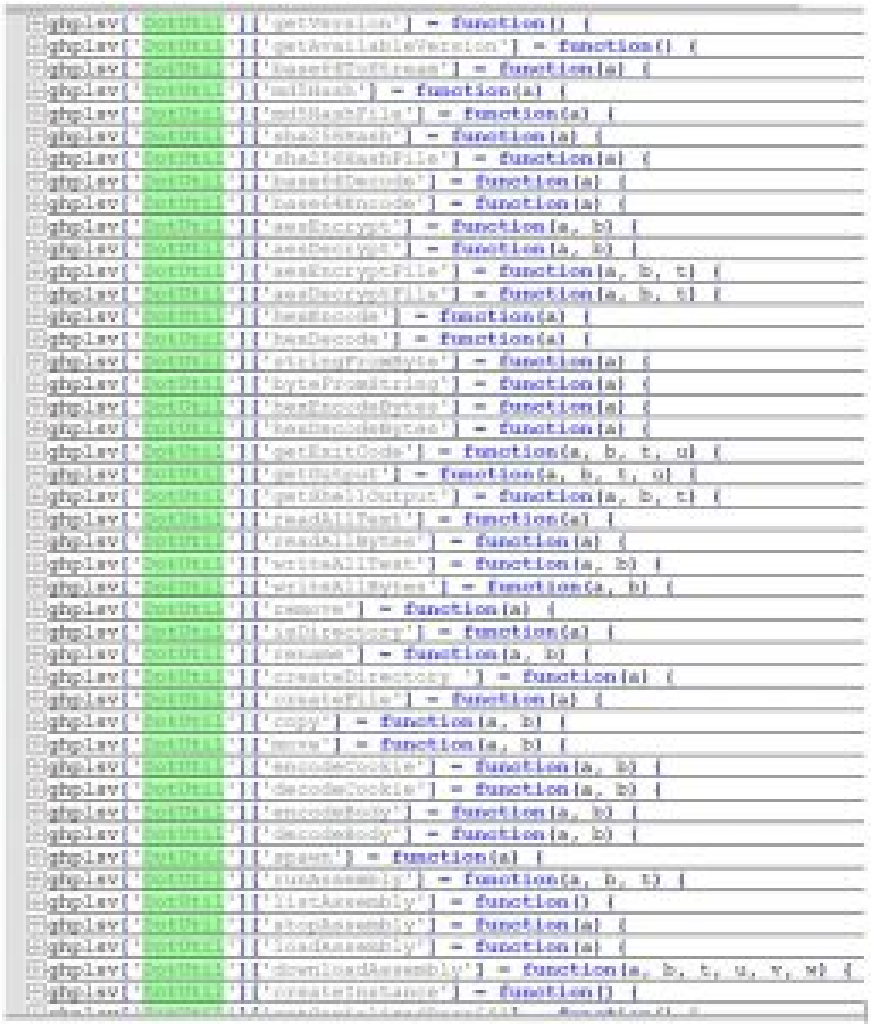

Fig 4: DotUtil functions to perform various .NET based tasks

Following are the additional plugins supported in the second stage:

- **ActivityPlugin** -> Enables the RAT to be in an Online or Offline state. When the state is online, it creates a adodb.stream object to save downloaded/collected data on disk.
- **CensorMiniPlugin** -> Enables/disables proxy settings on user machine by modifying registry key “Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable”
- **AdminConsolePlugin**
- **CensorPlugin**
- **ClipboardPlugin** -> It is used to copy the clipboard data and send it to C2. It can also modify clipboard data.
- **DnsPlugin** -> Used to set DNS path. Add or modify new path in C:\Windows\System32\drivers\etc\hosts.
- **LibraryPlugin** -> Sends list of dotnet versions installed on the machine to C2.
- **OutlookPlugin** -> It accesses the outlook account details and contacts list.

- **PriviledgePlugin** -> In this, the option “UAC” allows to write in registry location “SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\” by setting value 0 for keys EnableLUA and ConsentPromptBehaviorAdmin. The option “elevateScript” executes the script using wscript.exe with the batch mode option. The option “elevateCommand” executes the command using Wsh with ‘runas’ flag. It also has options for using UAC bypass techniques like fodhelper.exe, Slui File Handler Hijacking, CompMgmtLauncher, EventViewer.exe etc.
- **PromptPlugin**
- **ProxyPlugin** -> Sets DNS path. Add or modify new path in C:\Windows\System32\drivers\etc\hosts.
- **ShortcutPlugin** -> Create a shortcut file for a given executable. Execute the shortcut file. Get the target of a shortcut file or dump the content of the file.
- **RecoveryPlugin**
- **TokensPlugin** -> Steal OTP received from SymantecVIP application.

In the second stage, RAT finally drops a C++-based Netwire RAT with again a different C2 address. Last year we published our research about Java-based Adwind RAT (https://www.seqrite.com/blog/java-rat-campaign-targets-co-operative-banks-in-india/) in which jar file was the main component. It also targeted co-operative banks of India with Covid themed attachment names having a similar double-extension-like format. The various commands, configuration fields, and user-agent strings are identical in JSOutProx and Adwind RATs. We believe the same threat actor might be linked with JSOutProx RAT, where now they look to have changed their tactic to drop similar jar files as end payload, rather than as initial infection vector, to evade detections.

With multiple stages of payloads dropped by the threat actor, he can execute remote commands through any of the available stages, whichever can be seen as an attempt to evade antivirus detections.

We tracked the connections to the C2 domains to confirm if the exact fields are used in JSOutProx campaigns in other countries. But it turned out that only Indian IPs had connected to the C2 locations mentioned in the collected samples, confirming our assumption that it’s a targeted attack on Indian BFSI companies only.

With JavaScript, Jscript, or java-based malware, attackers keep inventing new ways to bypass static detections using different obfuscation techniques. But the behaviour-based detections are a suitable defence mechanism against such attacks. We continue to monitor such threats to protect our customers and mitigate the attacks at different levels. At the same time, people working in the finance sector are advised to stay alert from such attack campaigns as we expect more such attacks in the future as well.

IOCs

JSOutProx Stage 1

3c9f664193958e16c9c89423aefcb6c8
48adcbbc3ec003101b4a2bb0aa5a7e01
5D16911FE4BCC7D6A82C79B88E049AF2
0B9B2BF97CE805CA5930966FB4DA967A
5B2B4F989F684E265B03F8334576A20C
BEC6094A74E102A8D18630EE0EB053E3
988D384C68C95D28E67D6B8EDAF2EBE5
5111740D2EB8A8201231CB0E312DB88A

JSOutProx Stage 2

06396c2f1ac27f7a453d9461ad1af8a6
4876d3cc7b3b5990331a018c0b83ed03

Netwire

98fdee365893782b0639878c502fcfef

C2 Locations:

marcelbosgath.zapto.org:9790
ruppamoda.zapto.org:9099
apatee40rm.gotdns.ch:9897
mathepqo.serveftp.com:9059
protogoo.ddnsking.com:9081
riyaipopa.ddns.net:9098
dirrcharlirastrup.gotdns.ch:8037
uloibdrupain.hopto.org:8909
gensamogh.myq-see.com:9059
cccicpatooluma.hopto.org:5090
feednet.myftp.biz:6093

List of Filenames used in email attachments:

CBS_applcation_details_xlsx.hta
ANNEXURE_III_Exceeding_MDP_xlsx.hta
Nodal_Police_Stations_furnished_MHA_GOI_New_Delhi_xlsx.hta
Letter_dated_28_01_2021_jpg.hta
rtgs-credited-wrong_account_pdf__ 4.hta
Transaction report for_0127012021_docx.hta
Slip_RTGS_IDBI_To_HDFC_pdf.hta
Firewall_cRF_Login_access_details_pdf.hta
Comm_Bank_CLWS_Issues_&_Solutions_PDF.hta
Inspection_Compliance_pdf.hta

format-dist-wise-Cd_Ratio-pdf.hta
format_signatory_updation_PDO_138_docx__.hta
Information_regarding_CBS_details_update_xlsx.hta
Late_Return_docx.hta
Integrated_approach_brochure_pdf.hta
2685-Vishwambharlal_Kanahiyalal_Bhoot_Attachment_Order_pdf.hta
Pmay_infoletter_copy_of_houses-xlsx.hta
Annexure_Telangana_xlsx.hta
Compliances_Inspections_2020-pdf.hta
Circular-044_Introduction_Penalty_Charges_pdf.hta
NPCI_Compliance_Form_pdf.hta
Raise_chargeback_POS_txn-Reg_docx.hta
Karnataka_Vikas_Grameena_Bank_xlsx.hta
NFS_OC_No_354_RRN_format_pdf.hta
Exchange_information_details_pdf.hta
Neft_amount_credited_twice_dtd_09_03_2021_pdf.hta
KYC_Circular_from_AO__03_March_2021_pdf.hta
State_wise_ATM_Count_xls.hta
Payment_confirmation_details_acc_00190_pdf.hta
SR698684494_Transaction_Status_PDF.hta
SCAN1000000049A_JPEG.hta
Bridger_Sheet_OCSI_2_pdf.hta
Rewarding_SLBCs_for_APY_Performance_Pdf.hta
1_Format_EDU_LOAN_Annex_SLBC_April_March_2021_xlsx.hta
Importance_RBI_advisory_pdf.hta
Transaction_Amount_215000_pdf.hta
Submission_Returns_Ext_time_pdf.hta
PMJJBY_and_PMSBY_pdf.hta
3162_200727190525_001_pdf.hta
ISSUER_TRANSACTION_DT_17062021_docx.hta
Wrong_creditation_details_202101706_pdf.hta
MIS_080914_27804790_txt.hta
ICICBANK_Transaction_06172021_009122021_pdf.hta
NEFT_FORMAT_docx.hta
ISSUER_TRANSACTION_DT_17062021.XML.hta
Transaction_0578976746474754656866_pdf.hta
RTGS_FORM_AUTHORITY_LETTER_PDF.hta
CRF_NEFT_pdf.hta
STATUS_ENQUIRY_M0813100421890_docx.hta
Double_Neft_transactionS_Part_1_2_3_eml.hta

REF_NO_N0092010323095704_PDF.hta
SCAN_202024110816_122827484_pdf.hta
Annex_pdf.hta

Sameer Patil

Follow @