What is Arid Gopher? An Analysis of a New, Never-Before-Seen Malware Variant

deepinstinct.com/blog/arid-gopher-the-newest-micropsia-malware-variant

March 21, 2022

Learn more

March 21, 2022 | Simon Kenin, Asaf Gilboa

Executive Summary

- Deep Instinct's Threat Research team has found a new, undocumented malware developed in Golang
- The malware is attributed to APT-C-23 (Arid Viper)
- Further research revealed additional, previously unseen second-stage payloads

New Malware Variant Discovery: Arid Gopher

Our Threat Research team maintains a vigilant watch over the cyber threat landscape, hunting for malware as a normal course of operations. The team recently encountered an executable file written in the Go programming language. The identified file was initially submitted to VirusTotal on December 29, 2021 and was

detected by only six security vendors.

After initial inspection, two additional similar files written in Go have been found. During the analysis of these files, the team identified a previously unseen variant of Arid Gopher malware; the new unknown malware is a variant of the Micropsia malware, written and used exclusively by APT-C-23 (Arid Viper).

Micropsia and Arid Viper

This strain of malware was first identified in 2017 by "360 Security," but later re-named to Micropsia. This malware targets computers running Windows OS.

The threat actor behind the Micropsia malware is known by the name APT-C-23 or Arid Viper. This malware has primarily been used to target the Middle East region, with specific interest against Palestinian targets.

Arid Viper also has a unique Android malware that has been used against Israeli targets. Arid Viper has been previously linked to the Hamas organization.

Both the Windows and Android malware versions are constantly evolving. In April 2021, Facebook (now Meta) published a threat report about Arid Viper. In the report they identified a new iOS malware developed by APT-C-23. Facebook highlighted the specifics of how the threat actor had constantly changed the programming language used for developing the Micropsia malware which included Pascal, Delphi, C++, and even Python.

What is Arid Gopher?

During our investigation of the three files written in Go, we uncovered a novel variant of the Micropsia malware family written in Go, which we named Arid Gopher.

This new variant is still being developed; all the three files share a common baseline, but each file contains unique code which is not present in the other files.

Beside the main implant, our investigation revealed a "helper" malware, also written in Go, and a second-stage malware which was downloaded from the C2 server.

We provide a brief analysis of all the newly found samples and in-depth analysis of just one of them following.

What is Arid Gopher V1?

This variant is written in Go 1.16.5gs and contains public code from libraries found on GitHub:

```
strings 99544057a5215e756b67aa47815e27dc157eb850792e5eacda6796922bb9a50b | grep "go1\."
   ebp     ebx     ecx     edi     edx     eflags  eip     esi     esp     exporterfinishedfs      go1.16.5gs
dep     github.com/go-ole/go-ole        v1.2.5  h1:t4MGB5xEDZvXI+0rMjjsfBsD7yAgp/s9ZDkL1JndXwY=
dep     github.com/gonutz/w32/v2        v2.2.1  h1:OZj3YFJPlZ29VgVldt3IF3s1iHJiozmWu5CrCtvwypA=
dep     github.com/kbinani/screenshot   v0.0.0-20210326165202-b96eb3309bb0      h1:lICR2wyk9J6T709NawrhNTDi9DjMIbQqdlPT/EE0xBI=
dep     github.com/lxn/win      v0.0.0-20210218163916-a377121e959e      h1:H+t6A/QJMbhCSEH5rAuRxh+CtW96g0Or0Fxa9IKr4uc=
dep     github.com/zetamatta/go-windows-shortcut        v0.0.0-20200826154755-11707f2ef14c      h1:AF5xWROLzLP4Nco97nXs2SSU6fq3Y8SL5TXnXhkldKI=
```

Arid Gopher variant contains public code from libraries in GitHub

Developing the variant in this manner saves the author time by not needing to write some features from scratch. It also adds some degree of legitimacy because those libraries are not malicious, but the malware author abuses the libraries' capabilities for malicious purposes.

Library	Usage
https://github.com/zetamatta/go-windows-shortcut	Create Shortcut for persistence.
https://github.com/go-ole/go-ole	A dependency of the go-windows-shortcut library.

https://github.com/lxn/win	A Windows API wrapper package for the Go Programming Language.
https://github.com/kbinani/screenshot	Create screenshots of the infected computer
https://github.com/gonutz/w32	Windows API wrapper for the Go Programming Language.

All of the above libraries also exist in Arid Gopher V2 alongside additional libraries, except for the “go-windows-shortcut” library which has been replaced by another library with similar functionality.

The function names in V1 have unique and innocent names such as “infoSchoolManagerAboutRecievingHomeworksDone” and “wakeUpWhatIsInMyBag,” function names have been renamed in V2 to be more generic.

This variant is using the domain “grace-fraser[.]site” as a C2.

Grace Fraser is the name of a character from HBO TV show “The Undoing.” Arid Viper is known to use many references to TV shows; similar behavior was observed with Arid Gopher V2.

The C2 is using the “Laravel” framework which was used by Arid Viper in previous campaigns.

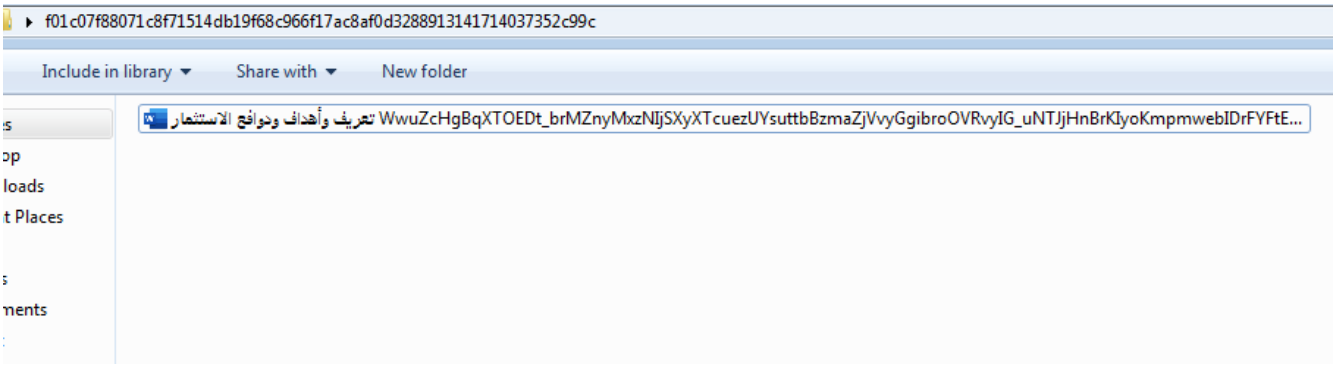
This variant was uploaded to VirusTotal at the end of August 2021 inside a RAR archive named “تعريف وأهداف ودوافع الاستثمار.xz” from the UAE, which might indicate the region in which the target is located.

The practice of sending variants of Micropsia inside archives with the extension “.xz” has been observed several times in Arid Viper campaigns.

The filename roughly translates to “definition, objectives, and motives for investment.”

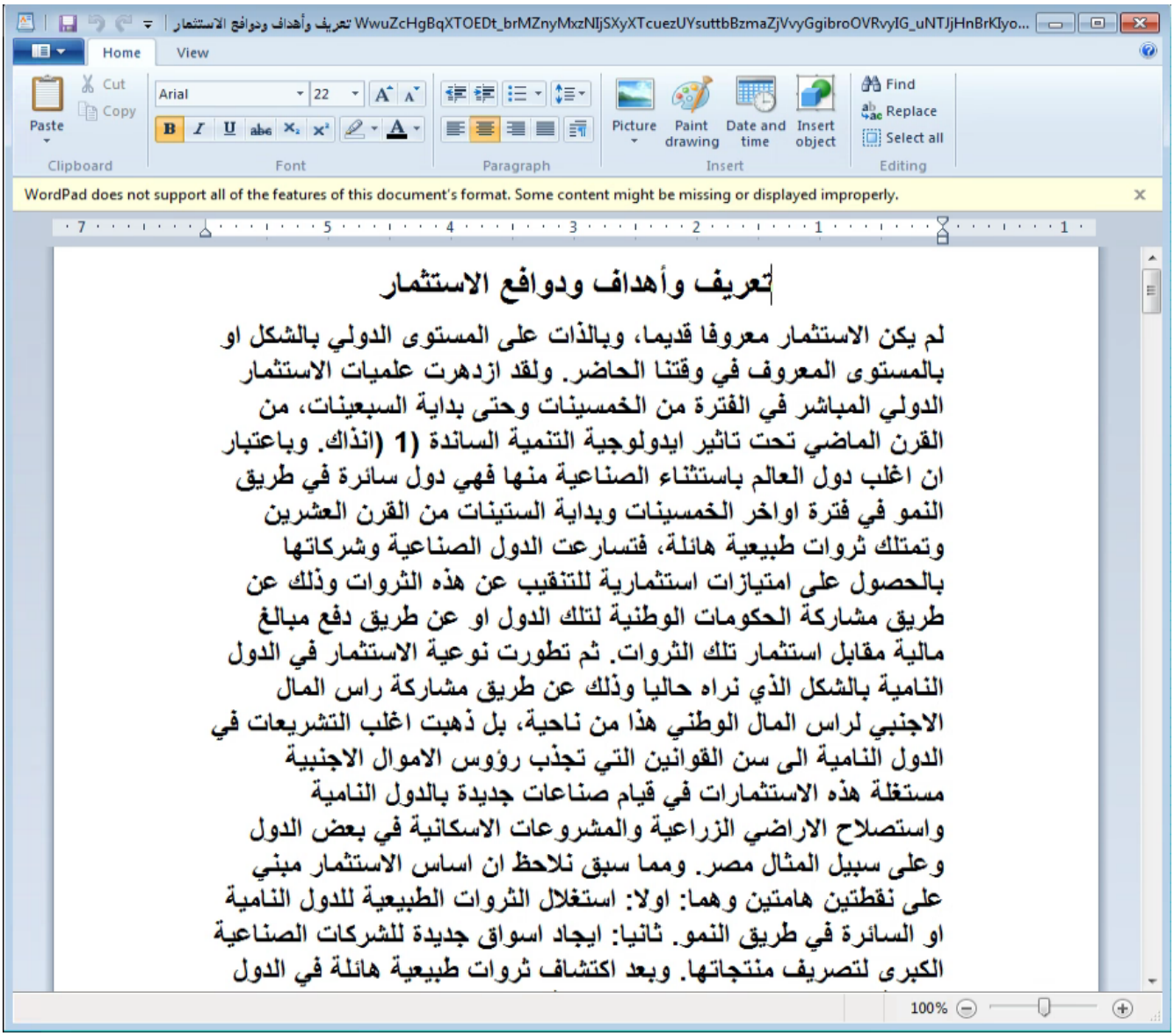
In order to trick the user into thinking they are opening an innocent Word document the threat actor uses two masquerading techniques:

- First, it uses the Microsoft Word Office document icon.
- Second, it uses a very long file name (see image below), preventing the user from seeing the ‘.exe’ file extension.



Arid Gopher File Name

Lastly, upon execution of the file, the malware will write a benign decoy office document to the folder “C:\ProgramData\NotificationControllerPS” and will present It to the victim:



Arid Gopher Benign Decoy Office Document

The combination of those three social engineering elements is intended to fool unsuspecting victims to run the malware and present them with decoy documents as they would expect from opening a Word document. This behavior is consistent in Arid Viper attacks utilizing Micropsia.

The decoy document contains sections from an academic publication regarding financial investments. The original article can be found in this link.

The malware creates a LNK file and copies it to the startup folder for persistence using the name of the malware executable.

System info collection:

1. The malware writes a base64 blob containing <current_user>_<random_ID> to "C:\ProgramData\NotificationControllerPS\MSAProfileNotificationHandler.txt"
2. The malware checks for installed Antivirus products by running the following command: cmd /c WMIC /Node:localhost /Namespace:\\root\SecurityCenter2 Path AntiVirusProduct Get displayName /Format:List
3. The function "app_myLib_student_GetFatherName" returns a string containing the OS version, which uses RtlGetVersion() and returns a string such as "Microsoft Windows [version 6.1.7601]"

The malware uses the custom User-Agent "aimxxhwpcc" while sending data to the C2 server.

The malware creates a mutex named “ABCMedia”.

What is Arid Gopher V2?

We found two versions of this variant that had been used in the beginning of 2022.

The main difference between the two samples is the decoy content. We will first show the similarities and later will show the difference between the two V2 samples.

Those samples are written in Go 1.17.4 and contain some of the public libraries found at V1 and the following libraries:

Arid Gopher V2 Libraries

Library	Usage
https://github.com/bi-zone/wmi	Windows Management Instrumentation (WMI) for Go.
https://github.com/bi-zone/go-ole	Dependency for the above WMI library.
https://github.com/hashicorp/go-multierror	Error handling.
https://github.com/hashicorp/errwrap	Error handling.
https://github.com/scjalliance/comshim	Ensures that at least one thread within a Go process maintains an initialized connection to the component object model runtime in Windows.
https://github.com/btcsuite/winsvc	Windows service library written in Go.
https://github.com/danieloliveira085/autostarter	Go library that creates a shortcut to run automatically at startup and supports cross-compilation between Windows and Linux.
https://github.com/GeertJohan/go.rice (only in PDF sample)	Go package that makes working with resources such as html, js, css, images, templates, etc. very easy.

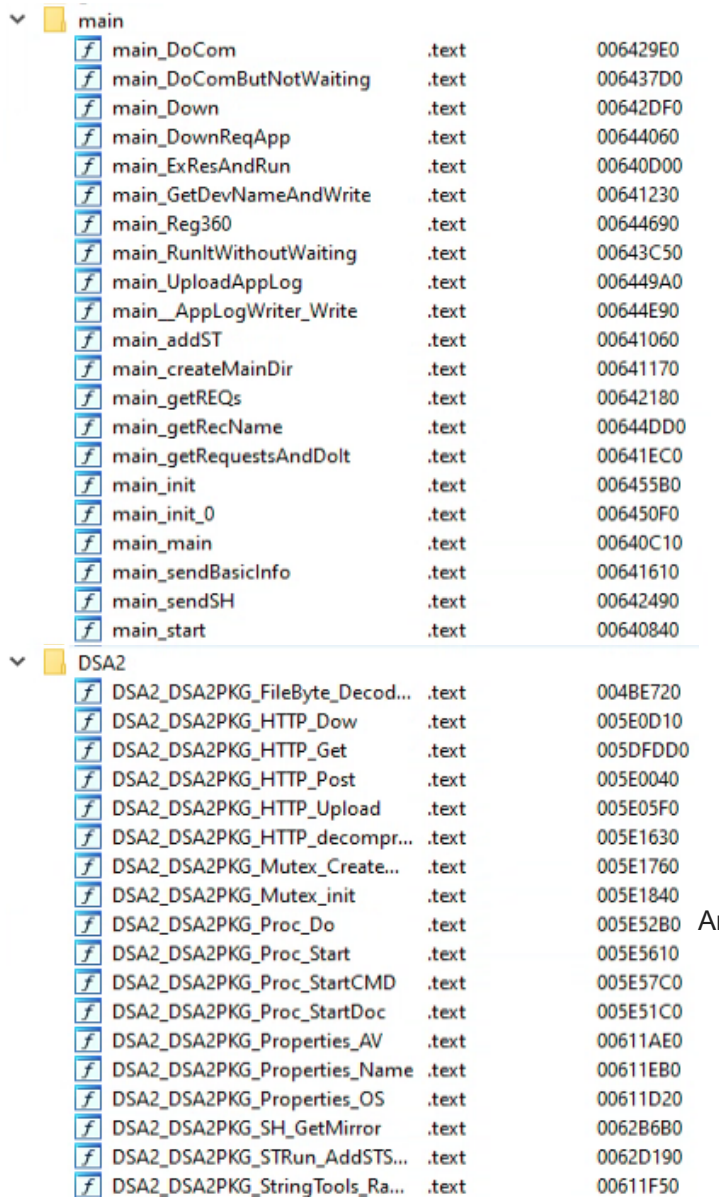
Those samples don’t have the Student/School functions name theme, however, using BinDiff we identified an exact match between functions in V1 & V2:

V1 function name	V2 function name	Functionality
app_myLib_student_GetFatherName	DSA2_DSA2PKG_Properties_OS	Retrieves OS version using RtlGetVersion()
app_myLib_student_GetStudentName	DSA2_DSA2PKG_Properties_Name	V1: Generates an identifier based on the %USERNAME%, current time, and random seed V2: The identifier is made from the hostname and %USERNAME%
app_myLib_driver_DoWhatIsTheDriverWants	DSA2_DSA2PKG_Proc_StartCMD	Runs “cmd /c <argument>” and retrieves the output
main_CreateMutex	DSA2_DSA2PKG_Mutex_CreateMutex	Calls kernel!CreateMutexW with the given string

Those samples are using the domain “pam-beesly[.]site” as a C2.

Pam Beesly is a yet another name of a character from TV show (The Office) and the same motive has been observed in V1 and older Micropsia variants.

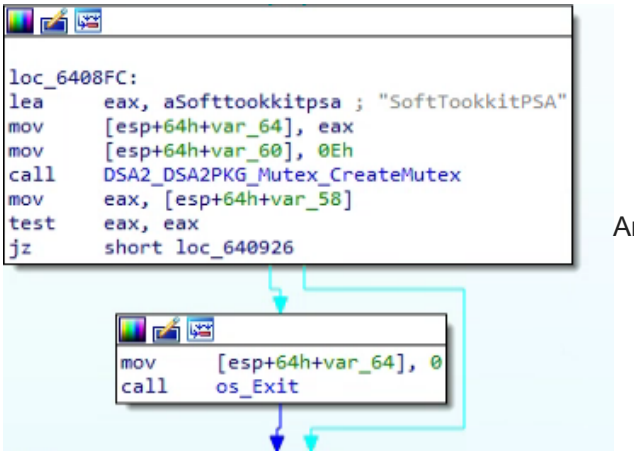
The following functions exist in V2 version:



Arid Gopher V2 Functions

If the process command line doesn't contain "-st", then the "main_ExResAndRun" function is called, which extracts the decoy document from the assets of the file using the library bindata.

The "SoftTookkitPSA" mutex is attempted to be created and if it fails then os.Exit() is called, terminating the process:



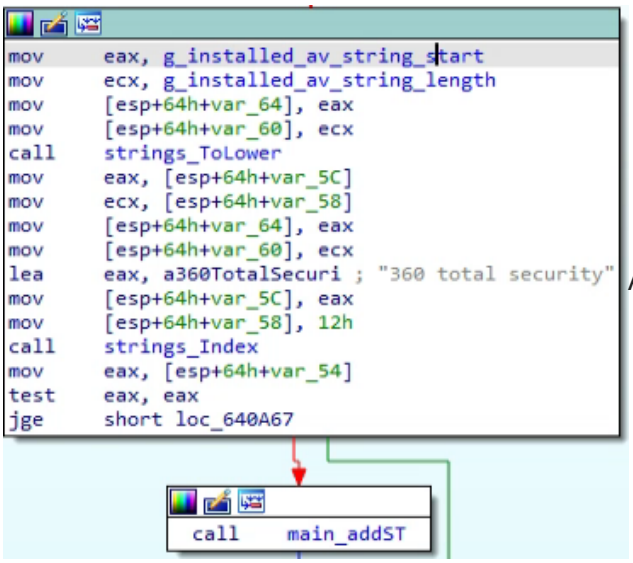
Arid Gopher V2 Mutex code

As can be seen in the image above the author most likely made a typo: "Tookkit" instead of Toolkit.

This mutex ensures there's only one instance of the malware running.

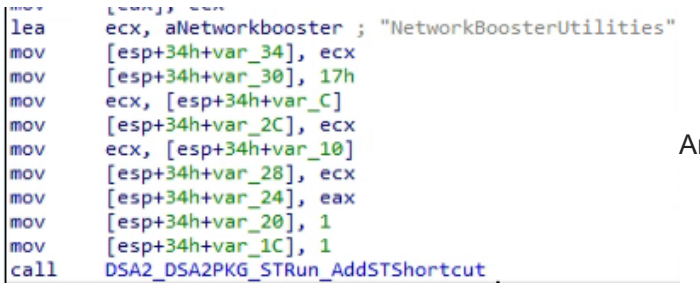
The malware queries WMI for installed AV products. If "360 Total Security" is present, the malware will call the function "main_Reg360" to download and execute a second-stage malware we called "Arid Helper." A description of "Arid Helper" is provided in a later section of this article.

If the computer does not have "360 Total Security" antivirus installed, "main_addST" is called:



AridHelper's AV check

The function "main_addST" creates a LNK shortcut named "NetworkBoosterUtilities.lnk" in the Startup folder, which links to the malware full-file path.



Arid Gopher V2 LNK persistence code

The LNK contains the argument "-st" which is used to start the malware without displaying the decoy document:

Names

C:\Users\<USER>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\NetworkBoosterUtilities.lnk

ExifTool File Metadata

MIMEType	application/octet-stream
TargetFileDOSName	THEMOD~1.EXE
LocalBasePath	C:\Users\
ModifyDate	2022:02:01 18:12:16+00:00
WorkingDirectory	C:\Users\admin\Downloads
RunWindow	Normal
CommandLineArguments	"-st"
AccessDate	2022:02:01 18:12:15+00:00
RelativePath	..\..\..\..\..\..\Downloads\The modified opening session program 29-1-2022_page-0001 98656456363546 4565546454645 98984938493854 pdf .exe
CreateDate	2022:02:01 18:12:15+00:00
TargetFileSize	5341184
MachineID	work
NetProviderType	Unknown (0x20000)
Flags	IDList, LinkInfo, RelativePath, WorkingDir, CommandArgs, Unicode
FileTypeExtension	lnk
NetName	8
HotKey	(none)
DriveType	Fixed Disk
IconIndex	(none)
FileType	LNK
FileAttributes	Archive

Arid Gopher: -st argument to start the malware
"-st" is most likely a prefix for the word start. In the Delphi variants of Micropsia, the LNK contained the argument "-start" for the same purpose.

The function "DSA2_DSA2PKG_STRun_AddSTShortcut" uses the autostarter library to set up the malware persistence in the Startup directory of the running user.

Afterwards, the function "main_createMainDir" is called which creates the C:\ProgramData\NotificationControllerPSK directory that will store screenshots and other information collected by the malware.

System info collection:

1. Similar to V1, the malware writes a base64 blob, this time also containing the computer name, in the format <computer_name>_<current_user>_<random_ID> to "C:\ProgramData\NotificationControllerPSK\MSAProfileNotificationHandler.txt"
2. The malware takes a screenshot and saves it as a PNG file to the same folder mentioned above "C:\ProgramData\NotificationControllerPSK"
3. The malware (only 3d7d75d66428c55dc81563c3bde5477977fadb3325d0224ef9313da133940077) is executing the following commands:
   cmd /c ipconfig /release
   cmd /c ipconfig /renew
   This is done by calling "cmd.exe," similar to the WMI query in V1.

4. The malware checks for installed Antivirus products by running a WMI query. Unlike V1 that used cmd.exe to query WMI, the malware uses the bi-zone WMI imported library by calling a function named “DSA2_DSA2PKG_Properties_AV":

```
lea     edx, aAntivirusprodu ; "AntivirusProduct"
mov     [esp+5Ch+var_54], edx
mov     [esp+5Ch+var_50], 10h
mov     [esp+5Ch+var_4C], 0
mov     [esp+5Ch+var_48], 0
call    github_com_bi_zone_wmi_CreateQueryFrom
mov     eax, [esp+5Ch+var_44]
mov     ecx, [esp+5Ch+var_40]
lea     edx, aRootSecurityce ; "root/SecurityCenter2"
mov     [esp+5Ch+var_18], edx
mov     [esp+5Ch+var_14], 14h
mov     [esp+5Ch+var_10], 0
mov     [esp+5Ch+var_C], 0
mov     [esp+5Ch+var_8], 0
mov     [esp+5Ch+var_4], 0
lea     edx, type__string ; *string
mov     [esp+5Ch+var_8], edx
lea     ebx, [esp+5Ch+var_18]
mov     [esp+5Ch+var_4], ebx
mov     ebx, off_8CB2B4
mov     [esp+5Ch+var_5C], ebx
mov     [esp+5Ch+var_58], eax
mov     [esp+5Ch+var_54], ecx
lea     eax, unk_6561C0
mov     [esp+5Ch+var_50], eax
mov     eax, [esp+5Ch+var_1C]
mov     [esp+5Ch+var_4C], eax
lea     ecx, [esp+5Ch+var_10]
mov     [esp+5Ch+var_48], ecx
mov     [esp+5Ch+var_44], 2
mov     [esp+5Ch+var_40], 2
call    github_com_bi_zone_wmi__Client_Query
```

Arid Gopher malware checks for installed antivirus products by running a WMI query

5. The function “app_myLib_student_GetFatherName” that was present in V1 was renamed to “DSA2_DSA2PKG_Properties_OS”

The function “main_sendBasicInfo” collects the following information about the computer and sends it to the C&C server:

Field Name	Field Value
ename	The asset name containing the name of the decoy .pdf to be opened
en	Set to 2
device_name	Base64 encoded unique string ID made up of the computer name, running username, and a random string
av	Result of the WMI query of the running antivirus product
os	OS version, e.g., “Microsoft Windows [Version 10.0.0.1836]

C2 Communication

The malware calls “main_sendSH” to save a screenshot using kbinani’s library and sends it to the server.

Then, “main_getRequestsAndDoIt” is responsible for a loop which sends GET requests to the C2.

The response is a JSON, unmarshalled into a struct named “main.REQ," which has the following definition:

```
type main.REQ {
  Id int
  Type string
  Value String
  Status int
  EncodedFileUrl string
}
```

Arid Gopher V2: main.REQ definition

The C&C server needs to send a response with the following JSON object:

```
[
  {“id“ : <int>,
  “type“ : <string>,
  “value“ : <string>,
  “status“ : <int>,
  “encoded_file_url“ : <string>,
  }
]
```

Arid Gopher V2 server response with JSON object

ID: A number that is used later for C&C commands to build a URL
Type: The name of the command to run
Value: The value of the command
Status: Unused
encoded_file_url: Used in the "d" command as the resource file to download

Most of the time an empty array will be sent by the server, which will make the sample wait a random number of seconds until it sends another GET request.

Here are the following supported commands (i.e., for the possible “type” field):

Type	Details
s	Takes a screenshot and sends it to the server
c	Runs the command in the “value” field in CMD
d	If the field “encoded_file_url” is present, download and execute a 2nd-stage payload from “<C2_address>/<id>/download_app_download-by-id/<encoded_file_url> to the “NotificationControllerPSK” directory
cwr	Same as type “c” command, but don’t wait for exit.
ra	Run a process with the path supplied in the “value” field

al	Uploads the SoftTookkitPSA.txt log file

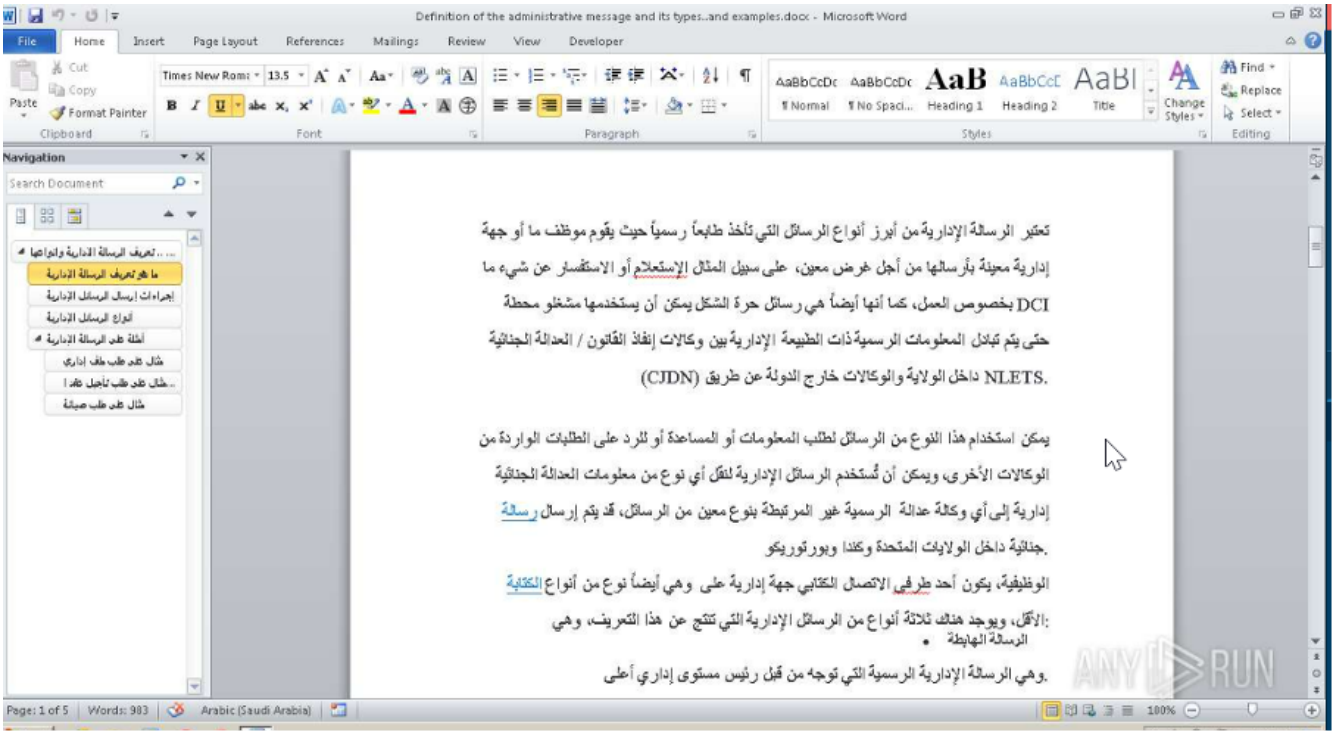
Differences Between V2 Samples:

3d7d75d66428c55dc81563c3bde5477977fadb3325d0224ef9313da133940077

This variant was uploaded on December 19, 2021 to VirusTotal from Palestine, which might indicate the country in which the target is located, since Arid Viper is known to focus on attacking Palestinian targets.

The icon of the malware is the exact same Word Icon that was used in V1.

If a victim opens the file, the malware will write a benign decoy office document to the folder "%AppData%\Local\Temp\<random_number>\" and will present It to the victim:



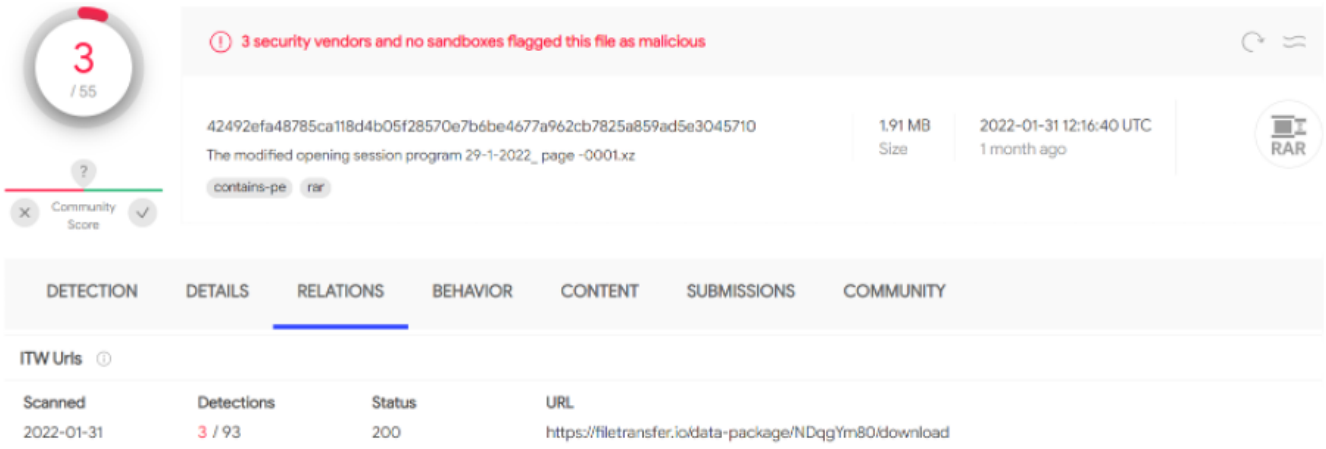
Arid Gopher V2: decoy office document opens in %AppData%\Local\Temp\\ folder
This document contains content from a Saudi blog named "Almrsal" with information on how to write a formal letter.

5588f6fab387133c21b06f6248259c64260435898edd61866fad50312c2d3b25

This variant was uploaded on January 31, 2022 to VirusTotal inside a RAR archive named "The modified opening session program 29-1-2022_ page -0001.xz" from Palestine, which likely indicates which country the target is located.

The practice of sending variants of Micropsia inside archives with the extension ".xz" has been observed with V1, as well as in several others Arid Viper campaigns.

The archive was downloaded from the URL "https://filetransfer[.]io/data-package/NDqgYm80/download," a free file upload service, however it is unknown how the victim(s) receive this URL.



”In the wild” URL containing archive with Arid Gopher V2 variant
This file has been observed by security expert “MalwareHunterTeam” as suspicious, but no formal attribution has been made.

Unlike V1 and the previous V2 sample, this variant has a PDF icon.

The malware executable inside the archive contains a double extension “The modified opening session program 29-1-2022_page-0001 98656456363546 4565546454645 98984938493854 pdf .exe”

The usage of double extensions, specifically “pdf.exe” with a combination of a long filename, has been observed in previous Arid Viper campaigns.

If the victim is opening the file, the malware will write a benign decoy PDF document named “The modified opening session program 29-1-2022_page-0001 98656456363546 4565546454645 98984938493854.pdf” to the folder “%TEMP%\<current_date_and_time>” and will present It to the victim:

Palestine National Council

Speaker Office

Ref. .....

Date .....

المجلس الوطني الفلسطيني

مكتب الرئيس

الرقم: م.و.ف ع/١/٣٠٢٩

التاريخ: ٢٠٢٢/١/٢٩

برنامج الجلسة الافتتاحية

للدورة الحادية والثلاثين للمجلس المركزي الفلسطيني

دورة " تطوير وتفعيل منظمة التحرير الفلسطينية، وحماية المشروع الوطني، والمقاومة الشعبية"

رام الله- فلسطين

قاعة أحمد الشقيري للمؤتمرات

2022/2/6

الساعة السادسة مساء

1. النشيد الوطني الفلسطيني.
2. الوقوف دقيقة وقراءة الفاتحة على أرواح الشهداء.
3. كلمة رئيس المجلس الوطني الفلسطيني الأخ سليم الزعنون يلقيها الأب قسطنطين قرمش نائب رئيس المجلس الوطني.
4. كلمة سيادة الأخ الرئيس محمود عباس أبو مازن رئيس دولة فلسطين.
5. كلمة الاخ محمد بركة رئيس لجنة المتابعة العربية العليا.
6. كلمة الاخ محمد اشتية رئيس الوزراء .
7. رفع الجلسة الافتتاحية من قبل نائب رئيس المجلس، وإعلان موعد الجلسات المغلقة بعد ربع ساعة.

* اجراء فحص PCR بمقر م.ت.ف بالقرب من مقر الرئاسة ما بين الساعة 10:00 -12:00

* الحضور الساعة الخامسة مساء لاجراء فحص كورونا السريع قبل الدخول الى القاعة

رئيس المجلس الوطني

٢٠٢٢/١/٢٩

Arid

Gopher V2: document meeting summary of Palestinian National Council
The document contains an official meeting summary of the Palestinian National Council.

Additional 2nd Stage Payloads

Arid Helper:
During the analysis of the V2 sample we noticed that if the WMI query for installed security products returns “360 total security” then the malware will call an additional function instead of “main_addST” as can be seen in Figure 10.

The function “main_addST” is responsible for creating the LNK shortcut as we described in V2 analysis.

However, if “360 total security” is found to be installed, the malware calls a function named “main_Reg360.” This function then calls another function named “main_DownReqApp” which downloads additional payload from the C2 server from the following URL: “http[:]//pam-beesly[.]site/J2FWAHfmgH573SUB/download_app/download-by-name/SystemNetworkEventsNotification”

The file is saved as “C:\ProgramData\NotificationControllerPSK\SystemNetworkEventsNotification.txt”.

Afterwards, a function named “DSA_DSA2PKG_FileByte_DecodeByte” is called to convert the downloaded file into an executable named “C:\ProgramData\NotificationControllerPSK\SystemNetworkEventsNotification.exe” and the text file is being deleted.

The final executable is also written in Go language and its’ sole purpose appears to be to create an alternative persistence mechanism in case “360 total security” is installed.

As a side note, 360 Total Security were the first to publish about the Windows malware which was later named Micropsia that Arid Viper continues to develop and improve.

This “helper” executable can receive the following parameters:
v=<PersistenceName> - Creates a registry run key with the given value name.
d=<PathToExecutable> - Sets the registry value to the path provided
-st - Unused
-old - Unused

The executable will use the Golang App Shutdown Hooks library, and set a shutdown hook that will add a run key with the parameters used to set persistence.

A shutdown hook function is called when the process receives an event for termination. Since the console for the process is hidden, it is in high likelihood that the intention of this executable is to add a registry run key when the computer shuts down.

Additional Second Stage Payload:

While analyzing the traffic from the V2 variant, we noticed that at some point a URL that was returning an empty JSON response started to respond with data.

The URL was called from the “main_getRequestsAndDoIt” function.

At one point, the URL returned a base64 blob, which when decoded, revealed another executable file:

property	value
md5	CA8C3C835877319FA927939DDAE31DE1
sha1	87BB1047298B51786F1D799FE8237FEA33D829D6
sha256	57674D0ED1E03807AD9D53A9087388B1B9BF6E9E5D120DBE834730AFFEBE2675
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	1057280 (bytes)
size-without-overlay	n/a
entropy	6.697
imphash	n/a
signature	Microsoft Visual C++ 8
entry-point	E8 A2 0B 00 00 E9 87 FE FF FF CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC 51 8D 4C 24 04 2B ...
file-version	n/a
description	n/a
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x620B591A (Mon Feb 14 23:41:14 2022)
debugger-stamp	0x620B591A (Mon Feb 14 23:41:14 2022)

PE information for the second stage malware

This sample is not written in Go, but in Visual C++ and was compiled on February 14, 2021. It uses mozelllittel[.]com as a C2.

The C2 uses the Laravel framework, which is also used in other Arid Viper C2 servers.

The sample also creates an LNK in startup for persistence, like the Gopher variants.

This sample doesn't display any decoy documents.

We will provide a further analysis of this sample in another blog post.

Conclusion

Most current cybersecurity products fail to detect new malware and APTs (Advanced Persistent Threats) because they rely on manually tuned heuristics. More advanced solutions use manually selected features, which are then fed into classical machine learning modules to classify the file as malicious or legitimate (and even then, the malware detection rates are quite poor). Several methods rely on running the malware in a sandbox environment to obtain more information. While this allows for more accurate detection, it comes at the cost of protection because it's a very time-intensive process and will not prevent threats from executing.

The Deep Instinct Prevention Platform stops known, unknown, and zero-day threats with the highest accuracy and lowest false-positive rate in the industry. We stop attacks before they happen, identifying malicious files in <20ms, before execution. Deep Instinct prevents >99% of unknown threats faster and with greater efficacy than existing EPP and EDR solutions, ensuring malware never enters your environment.

Deep Instinct is built on a first-of-its-kind deep learning cybersecurity framework inspired by the brain's ability to learn. Once a brain learns to identify an object, its identification becomes second nature. When applied to cybersecurity, deep learning facilitates new predictive capabilities of detecting, with unmatched accuracy, any type of cyber threat, including never-before-seen malware as described in this blog.

Deep Instinct customers are protected against the variants of Micropsia described in this blog, as well as other unknown, never-before-seen malicious threats. If you'd like to see the platform in action for yourself, we'd be honored to show you what *true* prevention looks like. Please request a demo.

Indicators of Compromise

SHA256	Description
f01c07f88071c8f71514db19f68c966f17ac8af0d3288913141714037352c99c	Archive containing AridGopher V1
99544057a5215e756b67aa47815e27dc157eb850792e5eacda6796922bb9a50b	AridGopher V1
42492efa48785ca118d4b05f28570e7b6be4677a962cb7825a859ad5e3045710	Archive containing AridGopher V2 (PDF)
5588f6fab387133c21b06f6248259c64260435898edd61866fad50312c2d3b25	AridGopher V2 (PDF)
3d7d75d66428c55dc81563c3bde5477977fadb3325d0224ef9313da133940077	AridGopher V2 (Word)
fa257cca88522e76a7dc4a10311f739d17587f25fe447ae2b4c84027f2246705	AridHelper
57674d0ed1e03807ad9d53a9087388b1b9bf6e9e5d120dbe834730affebe2675	2nd stage malware

Domain	Description
grace-fraser[.]site	AridGopher V1 C2
pam-beesly[.]site	AridGopher V2 C2
mozelllittel[.]com	2nd stage C2

Indicators of Attack

Folders:
C:\ProgramData\NotificationControllerPS
C:\ProgramData\NotificationControllerPSK

User-Agent:
aimxxhwpcc

Mutex:
ABCMedia
SoftTookkitPSA

Commands:
cmd /c WMIC /Node:localhost /Namespace:\\root\SecurityCenter2 Path AntiVirusProduct Get displayName /Format:List