Challenge: T1598.002 Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Easy

Tools Used: Oledump, Google Admin Toolbox Messageheader

Summary: This lab involved analysing a spearphishing email, using tools such as oledump and Google's Message Header Analyser to extract artifacts from a msg file. The investigation involved identifying msg stream metadata with oledump plugins, determining attachment details, extracting message-class information, and analysing header fields to uncover the senders IP address, total delivery delay, and the antispam software used by the recipient. Further analysis included dumping and examining a malicious HTML attachment that downloaded a password-protected ZIP archive, from which a malicious LNK file was extracted. Inspection of the LNK file uncovered a curl command used to retrieve a malicious payload from an external URL. Overall, this was an enjoyable lab, I highly recommend you giving it a go especially if you enjoy email analysis.

Scenario: Adversaries may send spearphishing messages with malicious attachments to elicit sensitive information that can be used during targeting. Spearphishing is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Spearphishing for information frequently involves social engineering techniques, such as posing as a source with a reason to collect information (e.g., Establish Accounts or Compromise Accounts) and/or sending multiple, seemingly urgent messages.

All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email, usually relying upon the recipient populating information and returning the file.[1][2] The text of the spearphishing email usually tries to give a plausible reason why the file should be filled in, such as a request for information from a business associate. Adversaries may also use information from previous reconnaissance efforts (ex: Search Open Websites/Domains or Search Victim-Owned Websites) to craft persuasive and believable lures.

What plugin is included in the oledump directory that scans streams in MSG files?

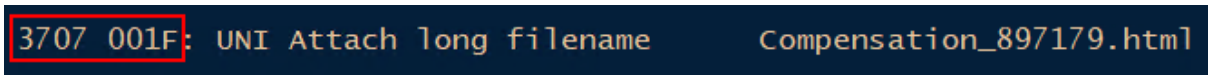
If you look at the oledump install directory, we can see a plugin called "plugin_msg". Upon searching for this plugin, we can see that it is used to identify streams in MSG files based on the 8-digit hexadecimal codes in the stream name.

Answer: plugin_msg

What are the 8-digit hexadecimal codes related to the "Attach long filename" stream?

Using the following command, we can identify the 8-digit hexadecimal codes related to the “Attach long filename” stream:

- `python .\oledump.py -p plugin_msg C:\Users\Administrator\Desktop\Challenge\T1598.msg | Select-String "Attach long filename"`



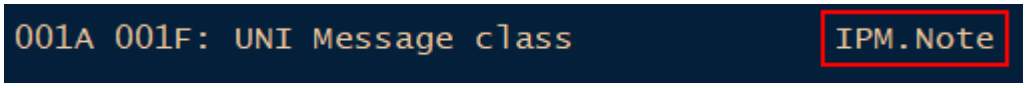
Answer: 0x3707 001F

During the analysis of the streams. What is the message class?

A message class in an email is an internal identifier that tells an email client which form to use for displaying and managing an item. For example. IPM.Note is the message class for a standard email message. Using this command:

- `python .\oledump.py -p plugin_msg C:\Users\Administrator\Desktop\Challenge\T1598.msg | Select-String "Message class"`

We can see that this msg is a standard email message:



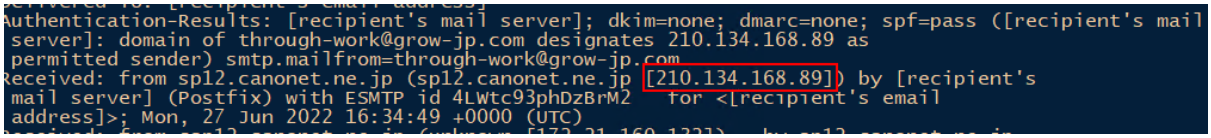
Answer: IPM.Note

To understand spearphishing, it's important to find the sender's source. What is the sender's IP address?

Another plugin we can use is plugin_msg_summary, which enables us to scan the msg file for email headers:

- `python .\oledump.py -p plugin_msg_summary --pluginoptions -H C:\Users\Administrator\Desktop\Challenge\T1598.msg`

If you look at the first received line, we can see where the email originated from:



Answer: 210.134.168.89

What was the total delay (in seconds) between the sender and the email receiver?

To determine the delay, we can compare the first and last received headers from the output of the command used previously. To make analysis easier, I copied the entire email header into Google's message header analyser:

1	1 sec	unknown
2		localhost
3	1 sec	echeck12.canonet.ne.jp
4		unknown
5	1 sec	sp12.canonet.ne.jp

We can see that the total delay is 3 seconds.

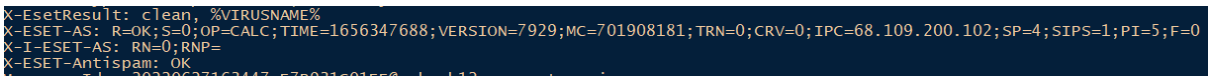
Answer: 3

What is the company that developed the antispam software used by the target?

In emails, X headers are non-standard custom fields added to the header of an email to provide extra information beyond the standard header fields. They are often used by email providers for internal processing like spam filtering and authentication results. In the output of the following command:

- `python .\oledump.py -p plugin_msg_summary --pluginoptions -H C:\Users\Administrator\Desktop\Challenge\T1598.msg`

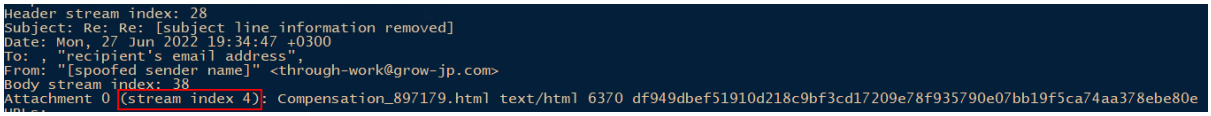
We can see that ESET is the company that developed the antispam software:



Answer: ESET

Dumping the malicious HTML file and opening it will download a zip file. What is the zip file's MD5 hash?

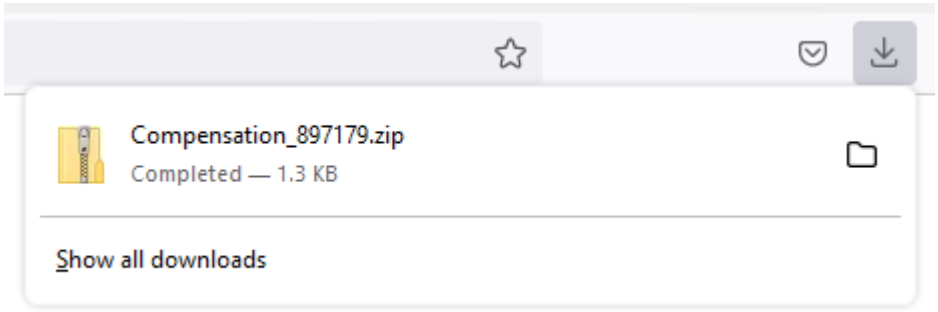
In the output of the previous command, we can see that the html attachment is located in stream number 4:



Using the following command, we can dump that stream:

- `python .\oledump.py -p plugin_msg_summary -s 4 -d C:\Users\Administrator\Desktop\Challenge\T1598.msg > .output.html`

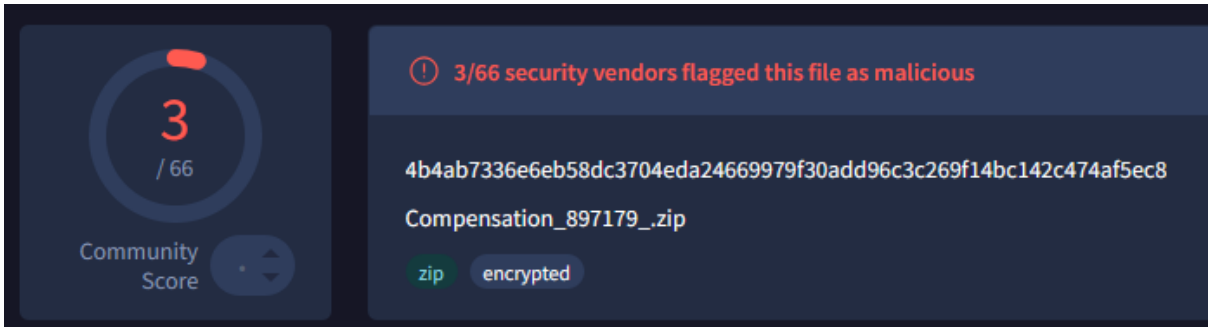
Upon opening this HTML file, it downloads a zip file called “Compoensation_897179.zip”:



We can use the Get-FileHash cmdlet to generate the MD5 hash of the zip file:

- `Get-FileHash -Algorithm MD5 -Path .\Compensation_897179.zip`

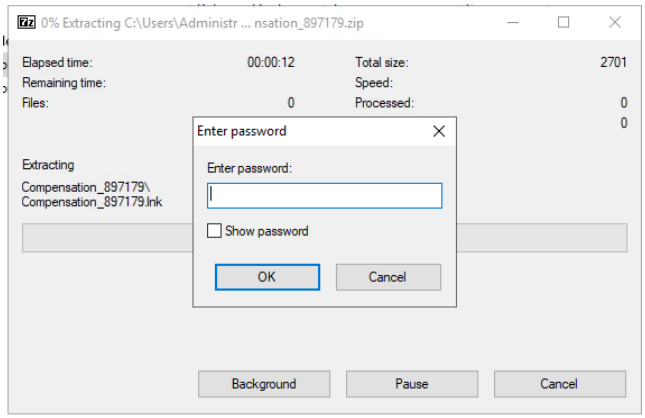
Submitting this to VirusTotal, it yields 3 detections:



Answer: 7E42F69B2ADCE7166408F635A093266E

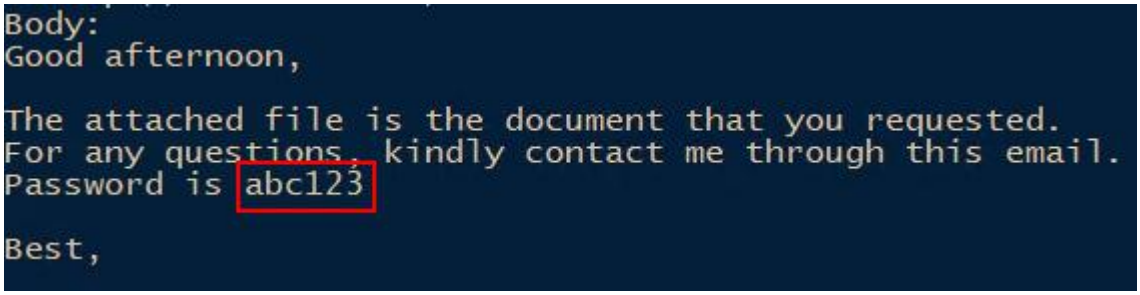
What full URL is used by the malicious shortcut embedded in the zip file?

If you try to extract the ZIP archive, we can see that its password protected:

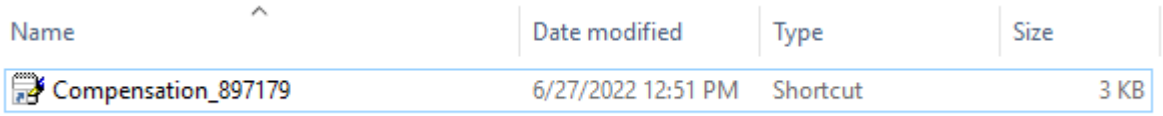


Threat actors often password protect archives to prevent it from being analysed by antivirus engines. The password is likely included in the email body:

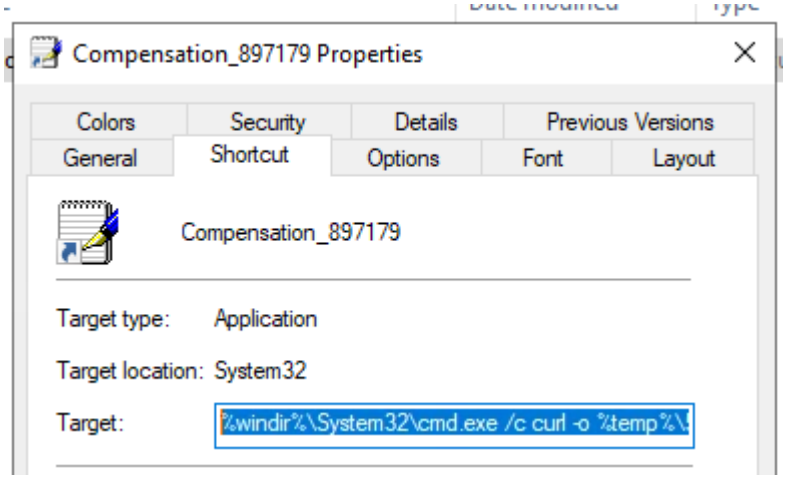
- `python .\oledump.py -p plugin_msg_summary --pluginoptions -b C:\Users\Administrator\Desktop\Challenge\T1598.msg`



After extracting this archive using the discovered password, we can see it extracts a LNK (shortcut) file:



If you right-click this file and select Properties, we can see the target of this shortcut is a curl command:



This command uses curl to download a file called herALook.dat from http://23.29.125.210 and saves it to the temp directory as 5552.jpg. Threat actors often abuse LNK files to deliver malware by including malicious commands in the target field. See here for more details on how threat actors abuse LNK files.

Answer: http://23.29.125.210/herALook.dat