Objective--Recover Alabaster's Password (Part1)

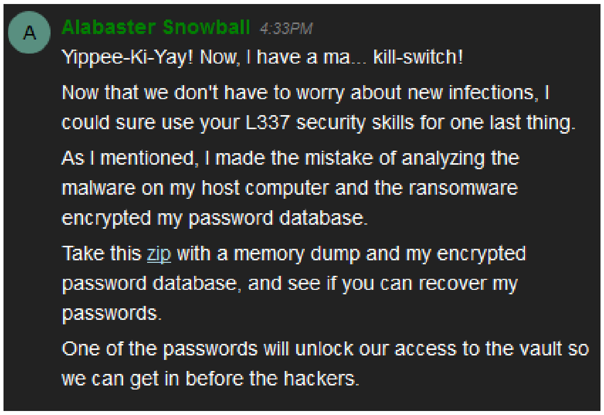
What you can learn from this

This objective takes a dive into encryption and decryption. Both symmetric encryption (AES) and asymmetric or public key encryption (RSA) are in play. There are many tools we can use: PowerShell, openssl, and Python were helpful for me. You will probably learn that encryption routines are very fussy about data format, block size, and other details that can be most frustrating.

There will be more reverse-engineering of malware written in PowerShell, probably as much as you will ever want! You will also learn to extract part of the information you need from a memory dump of the PowerShell malware as it was running on Alabaster's computer.

The Objective

Alabaster ignored the OPSEC rules we have been talking about and tried to analyzer the WannaCookie malware on his workstation instead of on an encrypted VM. Now his personal password database has been encrypted and he needs our help to decrypt it. The zip file that Alabaster links to is available here.



A

Alabaster Snowball 4:33PM

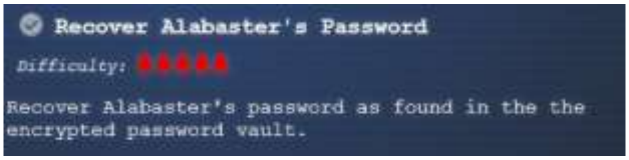
Yippee-Ki-Yay! Now, I have a ma... kill-switch!

Now that we don't have to worry about new infections, I could sure use your L337 security skills for one last thing.

As I mentioned, I made the mistake of analyzing the malware on my host computer and the ransomware encrypted my password database.

Take this zip with a memory dump and my encrypted password database, and see if you can recover my passwords.

One of the passwords will unlock our access to the vault so we can get in before the hackers.



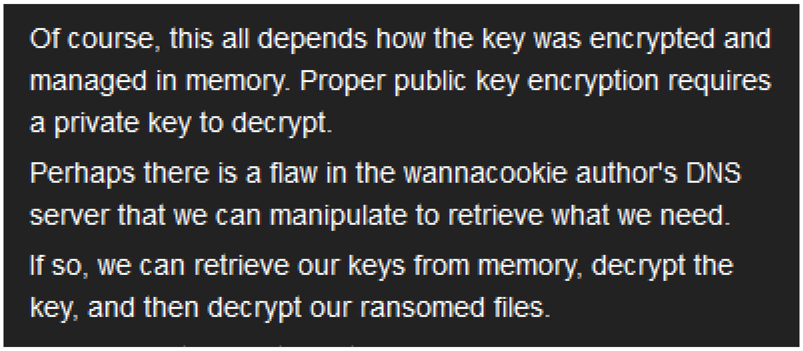
Recover Alabaster's Password

Difficulty: ●●●●●

Recover Alabaster's password as found in the the encrypted password vault.

Hints

This is the end of a conversation with Shinny Upatree after we helped him win the Sleigh Bell Lottery. It describes the job before us very well. We will do well to remember Shinny's advice.

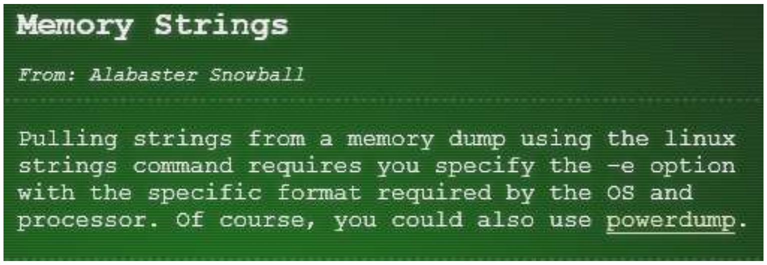


Of course, this all depends how the key was encrypted and managed in memory. Proper public key encryption requires a private key to decrypt.

Perhaps there is a flaw in the wannacookie author's DNS server that we can manipulate to retrieve what we need.

If so, we can retrieve our keys from memory, decrypt the key, and then decrypt our ransomed files.

Also, Alabaster reminds us about powerdump. Chris Davis demonstrated its use in his Analyzing PowerShell Malware talk.



Memory Strings

From: Alabaster Snowball

Pulling strings from a memory dump using the linux strings command requires you specify the -e option with the specific format required by the OS and processor. Of course, you could also use powerdump.

Get Started

So far, we have analyzed the functions in wannacookie.ps1 that convert data, and the first lines of the wannacookie function that terminate execution. The core of evil in the wannacookie function is in lines 193 through 203. Now is the time to analyze them in detail.

Hand in

1) Create a flowchart, a discussion, comment the code, or whatever helps you understand the process that wannacookie follows in the 20 lines of evil (193-203). Turn in your flowchart, discussion, commented code, or screenshots of whatever you did.

2) As you document the malware, create a list of interesting variables, their types and their lengths. We will use this later.

3) As you document the malware, keep a list of the command codes and their meanings.