Seth Ayers

IT 330 – 01

08/30/2017

Homework Assignment 2

SAyers\_HWA2.docx

Chapter two questions:

1. C. Virus

4. C. Intimidation

5. C. Trojan

6. D. Reformat the hard drive and reinstall the operating system

8. B. Typo Squatting

11. B. Botnets

15. B. Impersonation

16. B. A hoax could convince a user that a bad Trojan is circulating and that he should change his security setting.

18. C. Tailgating

Chapter three questions:

1. B. The processors on the clients are smaller than on web servers and thus they are easier to defend.
2. D. Content Length
3. A. Flash Cookie
4. B. Can be embedded inside a webpage but add-ons cannot

6 C. Transitive

8. C. Point to another area in data memory that contains the attacker’s malware code

9. B. XSS requires use of a browser.

11. B. To inject SQL statements through unfiltered user input

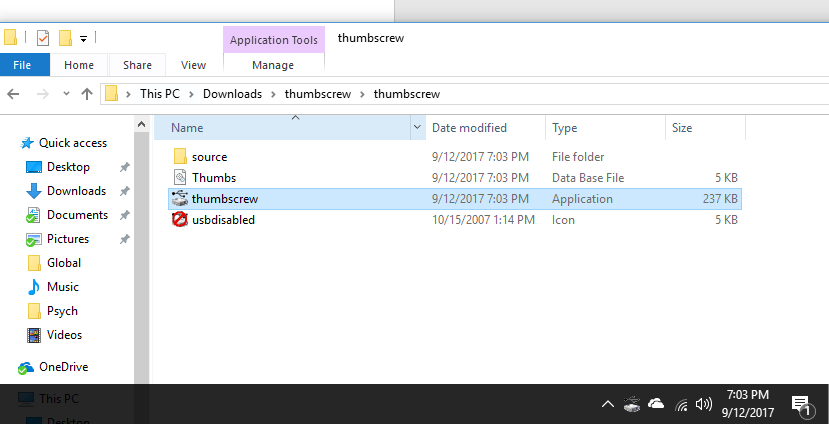
12. A. Reformat the web application server’s hard drive

14. D. Directory traversal

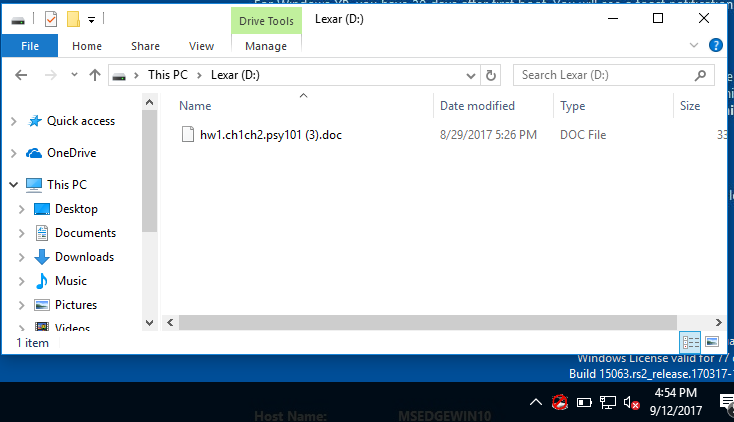
16. B. A random string assigned to the web server

**Project 2 – 1:**

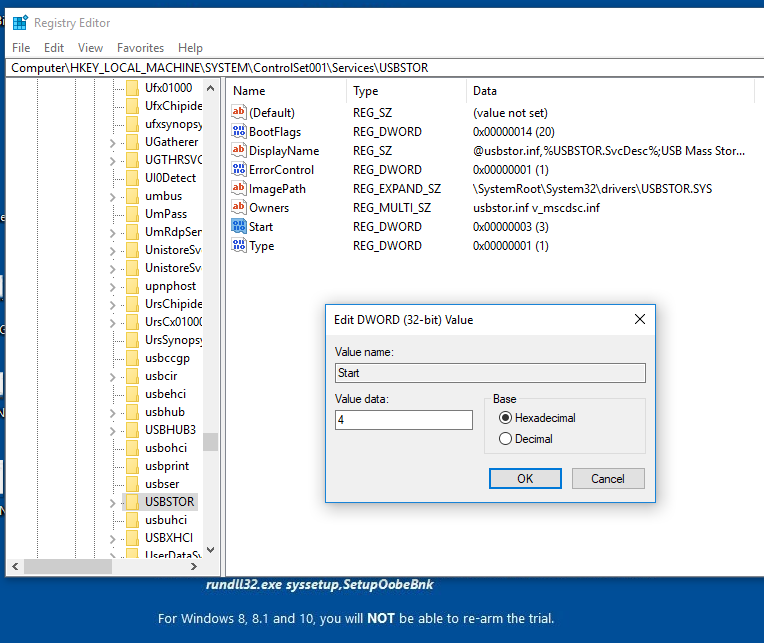
Parts 1-5



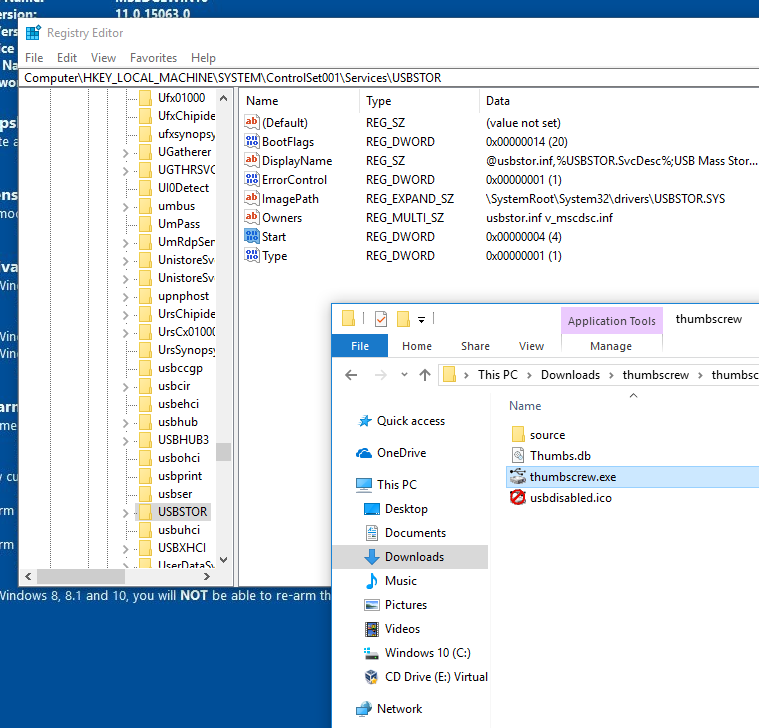
Parts 6 – 15



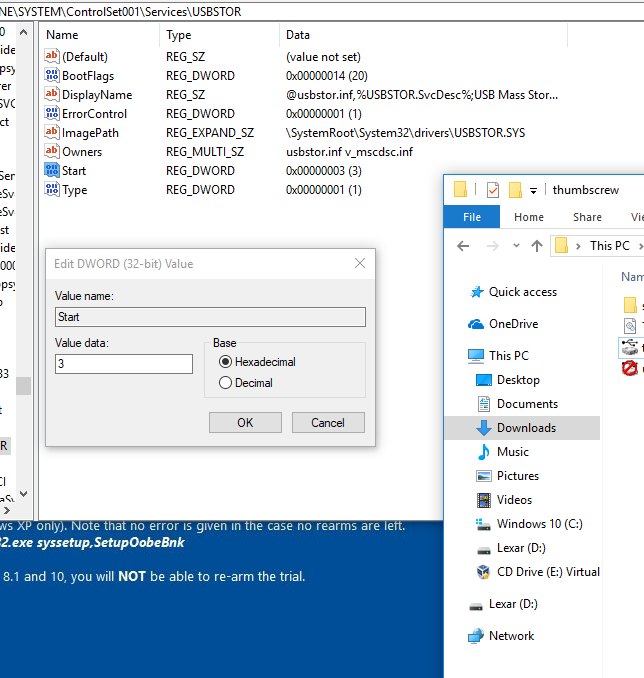
Parts 16-24



Part 25



Part 26

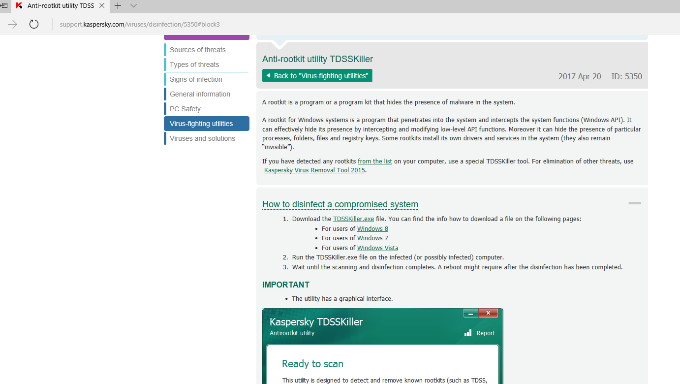


**Evaluation**

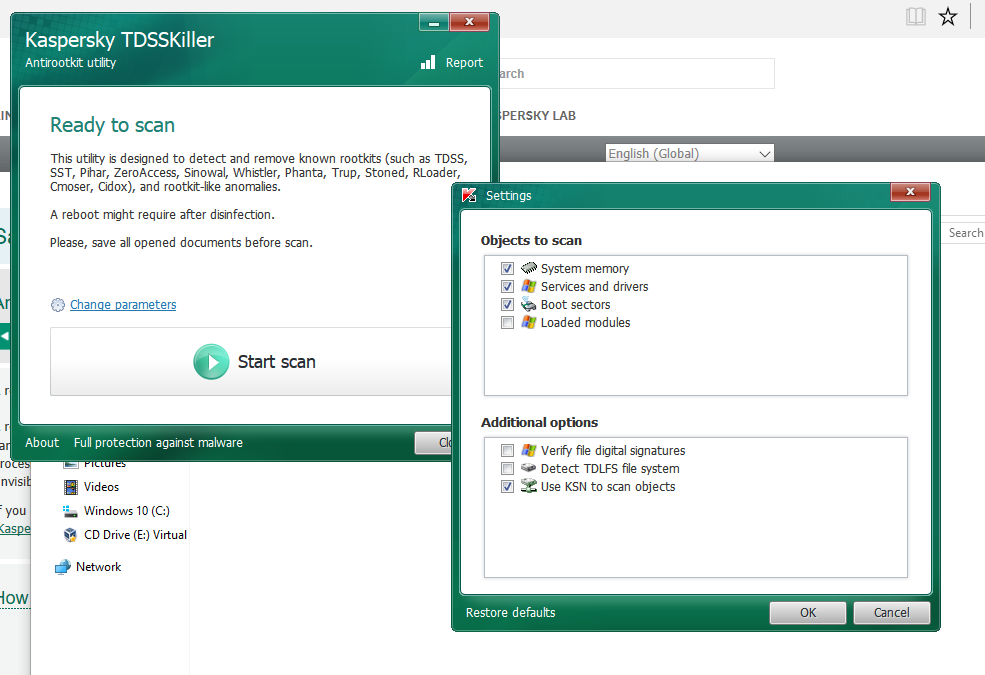
I feel the purpose of this project served a dual purpose. One is to apply software that can write protect USB flash drives to prevent malware from accessing and duplicating itself onto it. Chances are, that the USB drive will most likely be used on another machine, in which the virus can again duplicate or attack the new system. Another feature used is completely disabling the USB port itself. This is done by manipulating the registry folder USBSTOR by using the “Start REG\_DWORD”. The other purpose was to show how a single, small program can manipulate the systems registry, without the user fully aware, by a single click. I’d like to add that the Thumbscrew program didn’t successfully disable the port, but the intention was clear.

**Project 2-2**

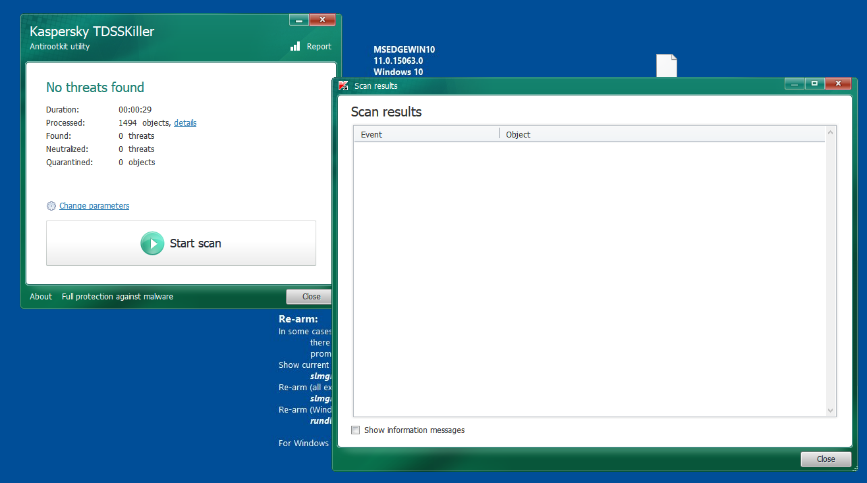
Part 1-2



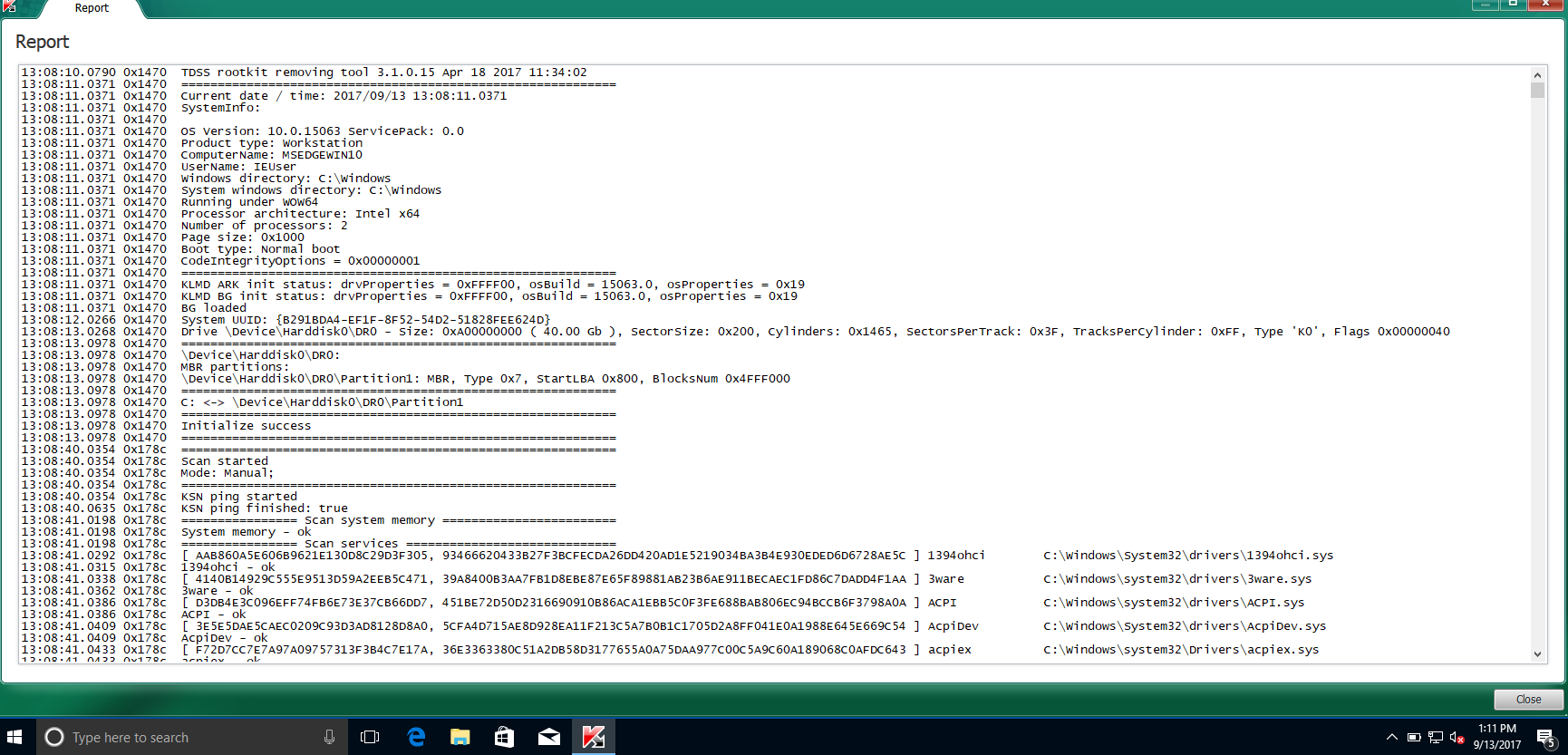
Parts 3-7



Parts 8-12



Parts 13-14

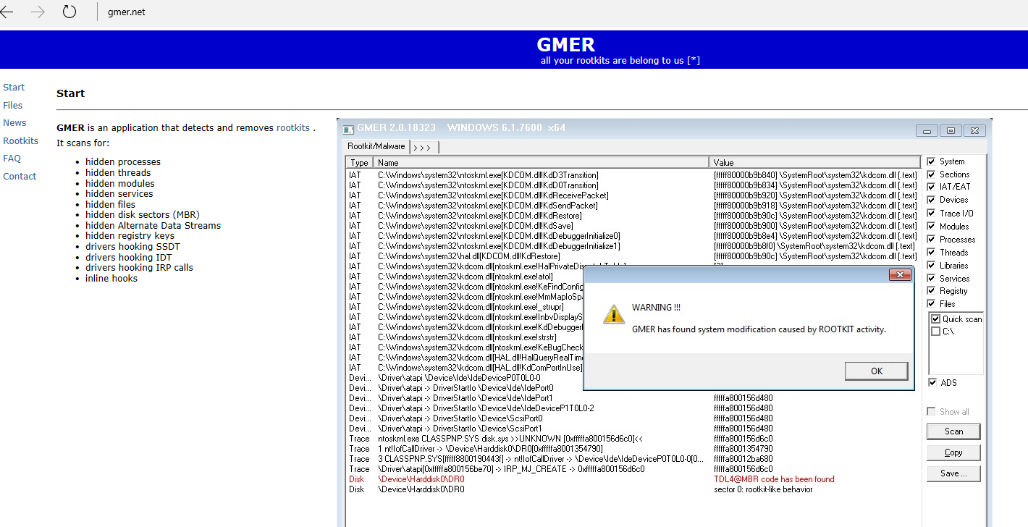


**Evaluation**

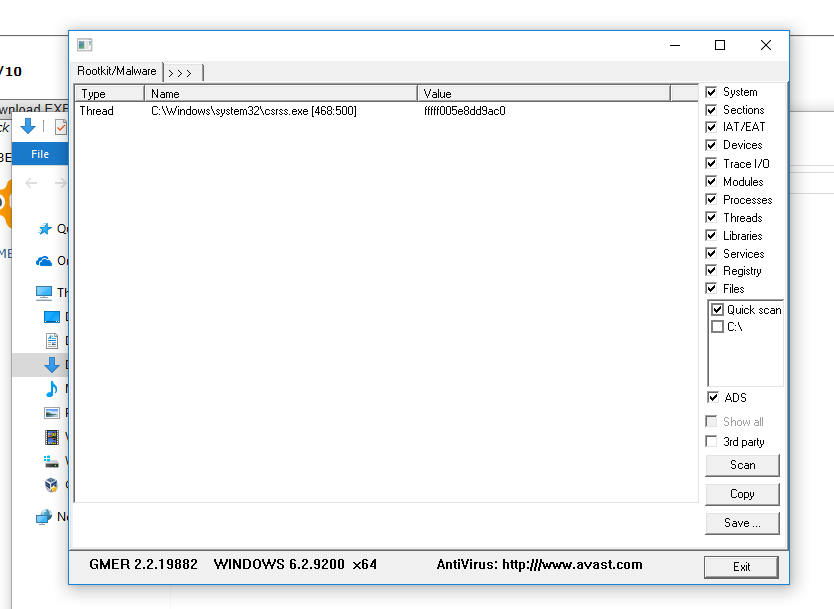
The Kaspersky TDSS is a tool used to detect and eliminate the TDSS family of rootkits. These rootkits exclusively target Windows based operating systems that execute malware, delivers adware, and blocks programs from running. Although it isn’t as robust as some other rootkit detectors, it executed its required duties well and is relatively lightweight in function. If you have a rootkit that’s on TDSSkillers rootkit list, it will find and quarantine it. After the install, parameters are set and in our case, the loaded features box need to be selected to scan. TDSSkiller rebooted my system and loaded the features needed for the scan. The scan checks your files, gives details with found rootkits, and provides a detailed account of the scan itself.

**Project 2-3**

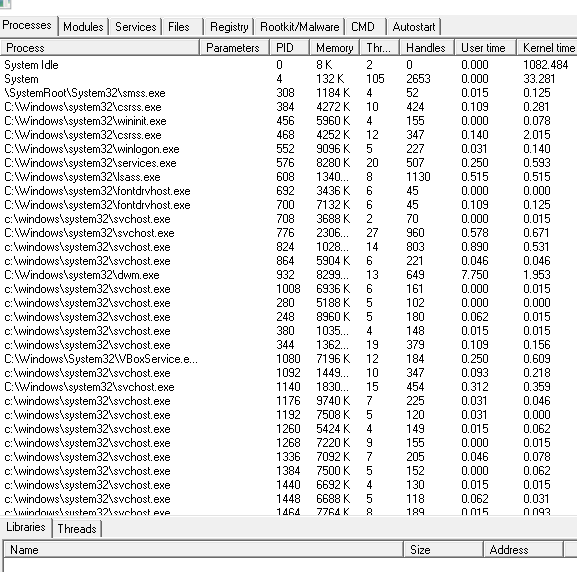
Part 1



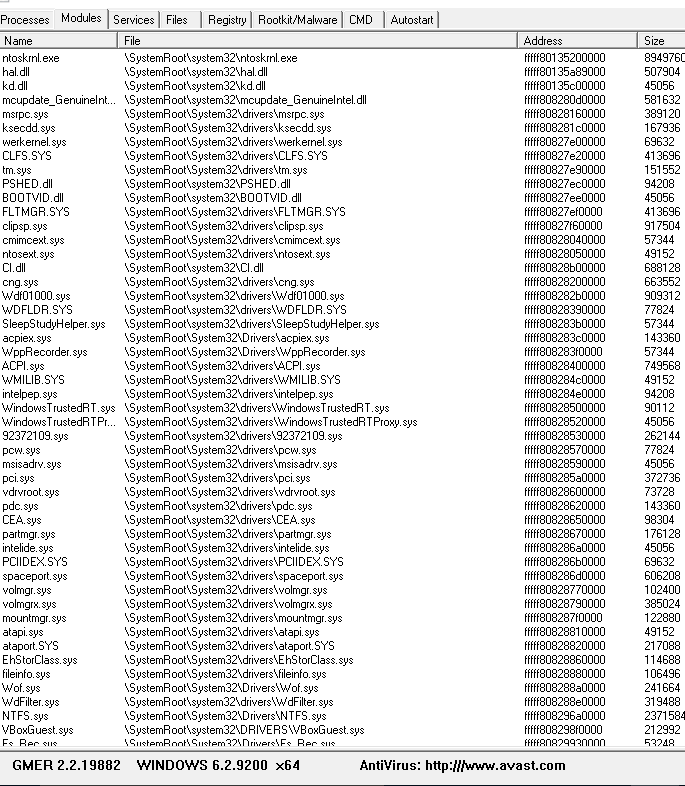
Parts 2-4



Parts 5-6



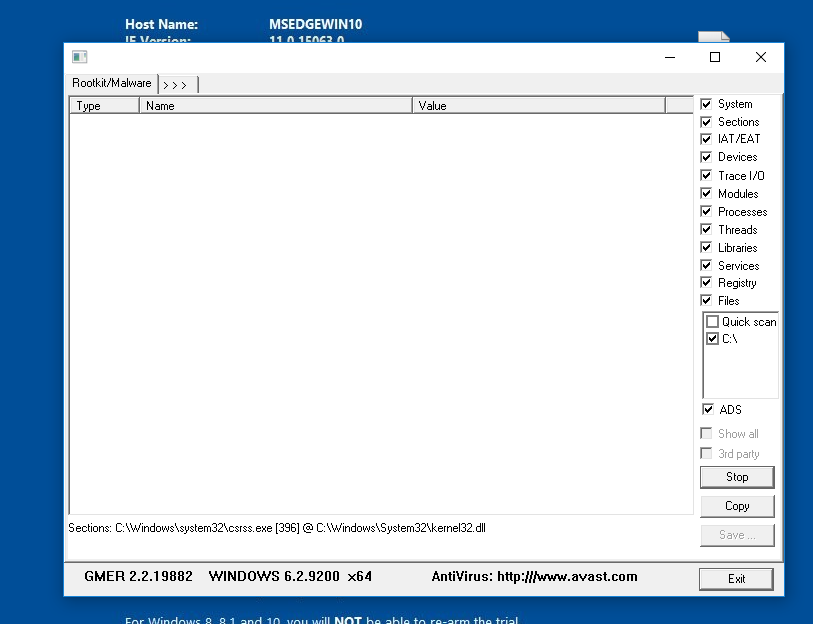
Part 7



Part 8



Parts 9-13



**Evaluation**

Much like TDSSkiller, GMER scans for rootkits. However, the scan is increased in depth and universal for what it attemps to locate. For instance, the following is located:

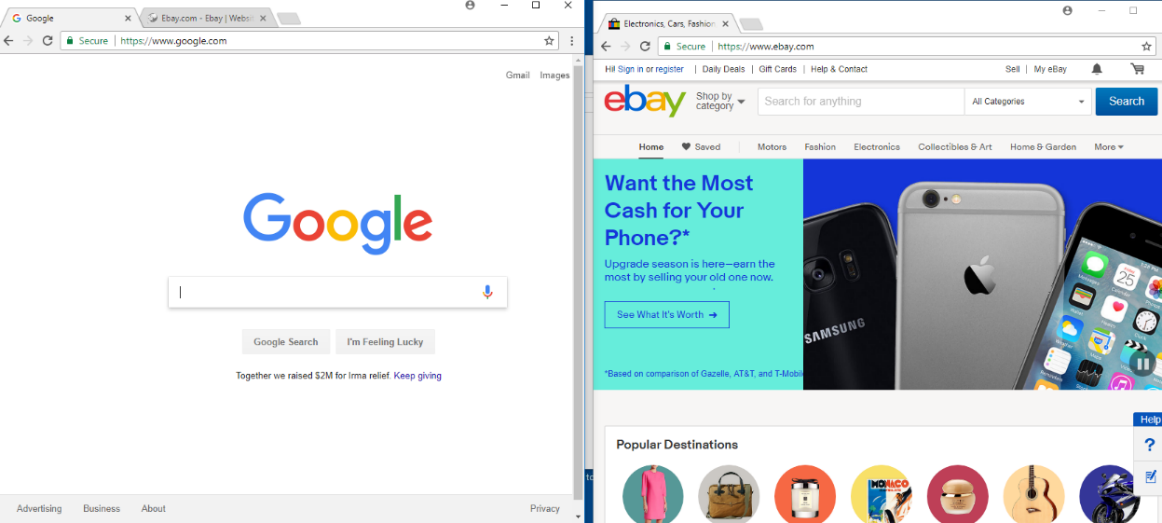
* hiddenprocesses
* hiddenthreads
* hiddenmodules
* hiddenservices
* hiddenfiles
* hiddendisk sectors (MBR)
* hiddenAlternate Data Streams
* hiddenregistry keys
* drivers hooking SSDT
* drivers hooking IDT
* drivers hooking IRP calls
* inline hooks

(list taken directly from www.GMER.net)

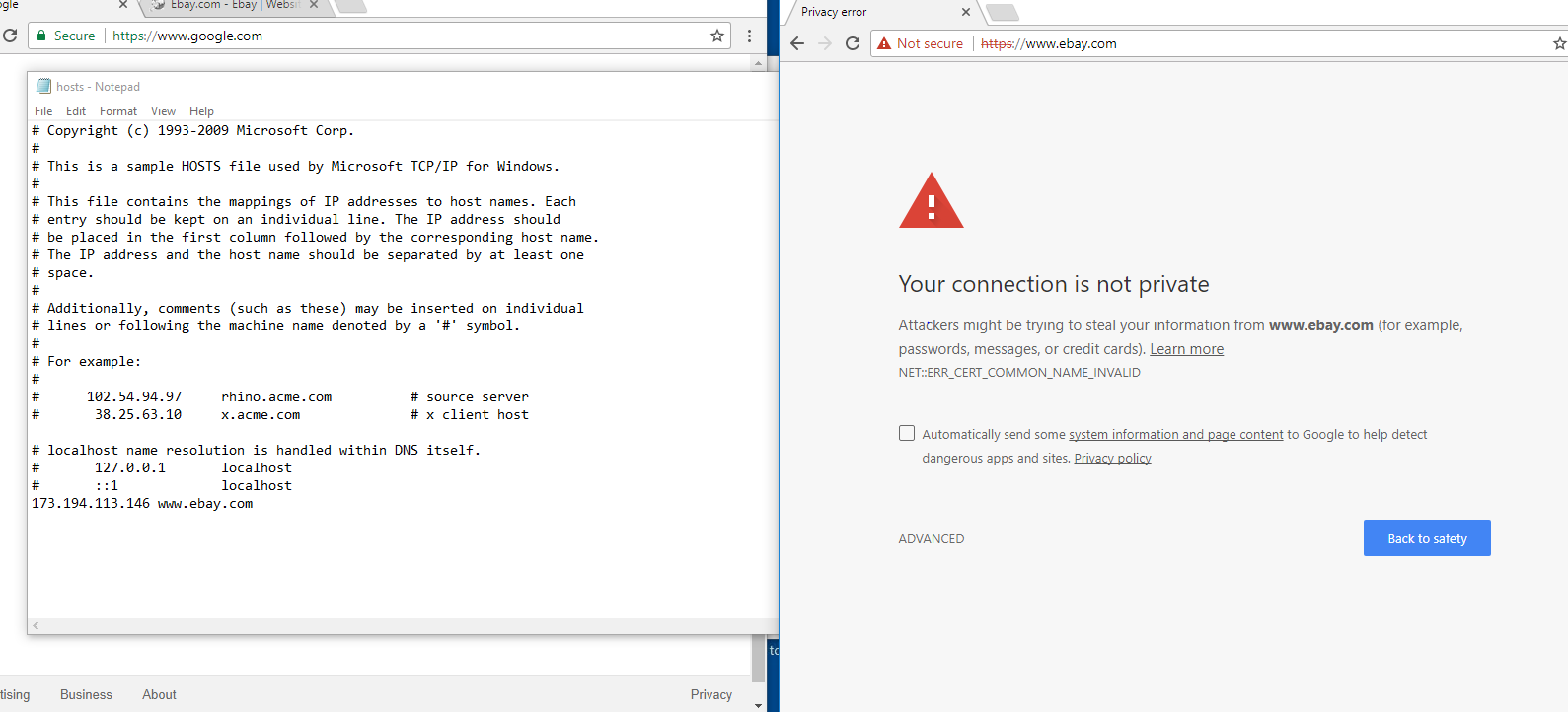
As you can see, GMER is a much more robust scanning tool than that of the lightweight TDSSkiller. In this Project, after the download and install, GMER scans all running processes. As the processes are scanned, hidden processes are outlined in red for easy indication for the user. Then, modules are scanned which displays all current modules that are loaded onto the devise (In my case, my VM). Much like the Processes scan, GMER scans the services that are running, with outlining hidden services in red. Scanning for hidden processes and services is the prime target for GMER, due to the nature of rootkits being hidden. Finally, a full scan of the system with Rootkit/Malware selected, C:\ for me, preforms a full scan of the selected drive which can take up to 30 minutes, depending on computer capabilities. After the scan, any hidden resources are displayed. It’s worth mentioning that GMER is such an effective tool, some malware builders have taken precautions to code disabling features into the rootkit if the GMER exe is executed. To prevent this, a random name is assigned to the exe of GMER at download to make it undetectable by the rootkits.

**Project 3-4**

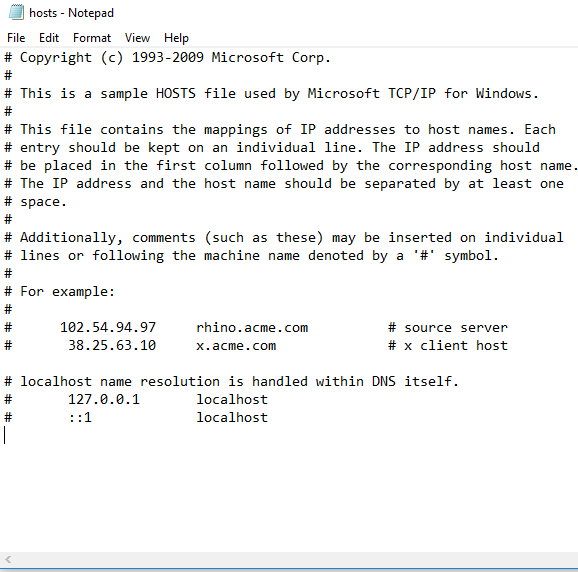
Parts 1-2



Parts 4 -11



Parts 12-14

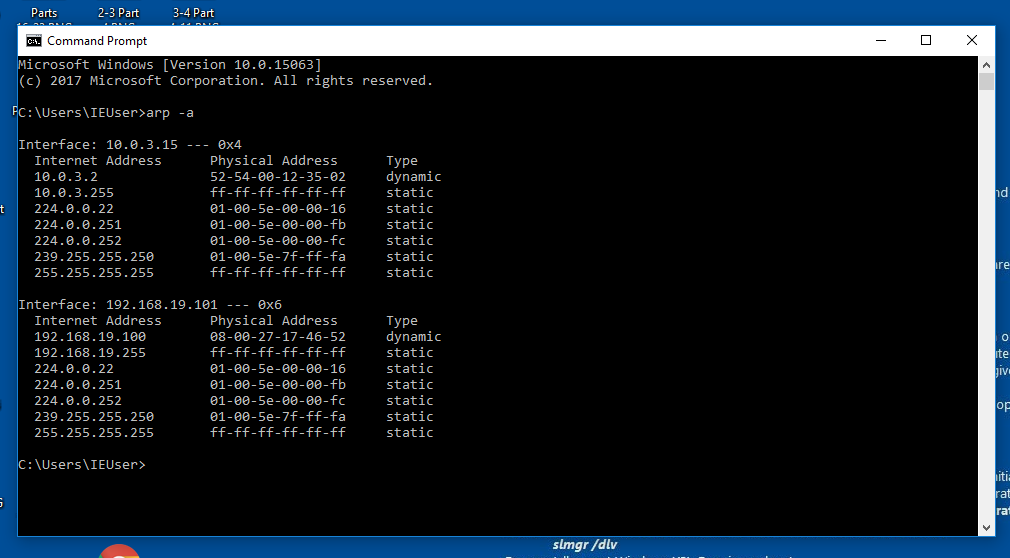


Evaluation

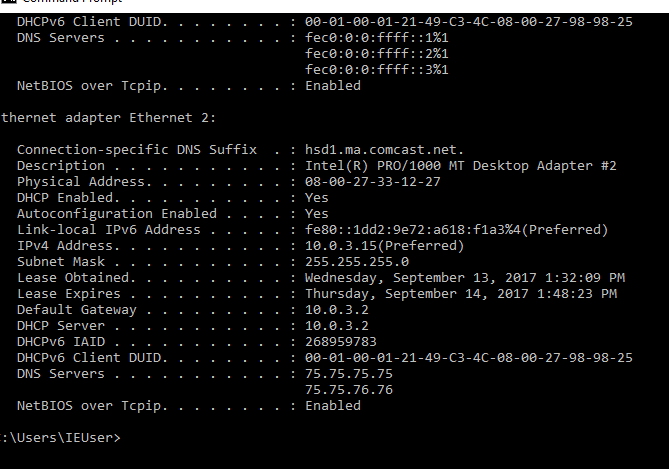
This short Project shows how relatively easy it is to assign a fraudulent IP address to an assigned website name. This would enable hackers to redirect your imputed domain name to a website they may seem like your intended destination, but in truth brings you to a website that could record important login information or worse. This is done my altering the local host file by assigning IP addresses to their own fake websites. In my situation, I switched the IP address between google.com and Ebay.com. Interestingly, it seems my browser discovered the problem with the COMMON\_NAME\_INVALID. This was probably a result of a browser security patch that introduced a common website/IP address list that was cross checked before the site was accessed with the bogus information in the host file. I was pleasantly surprised that this was picked up but I’m also aware that hackers will most definitely find a way to bypass this security feature, if they haven’t already.

**Project 3-5**

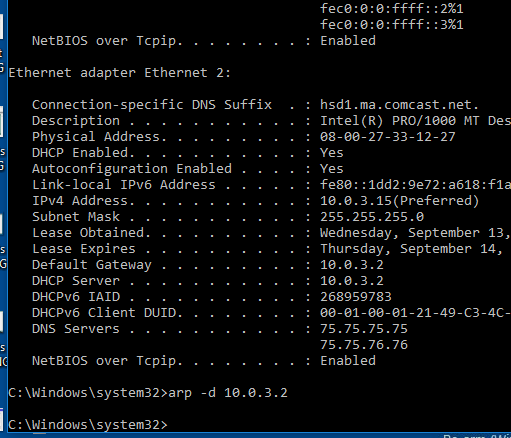
Parts 1-3



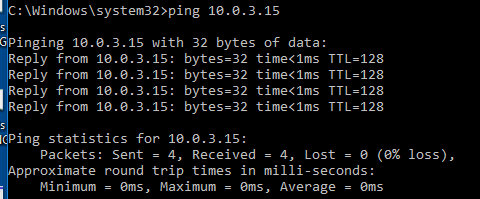
Parts 4-5



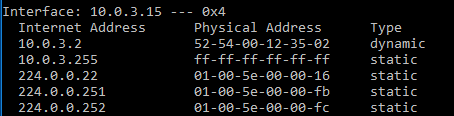
Part 6



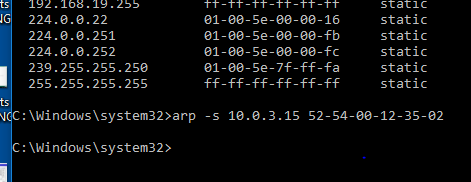
Part 7



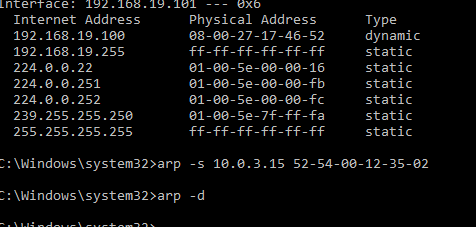
Part 8



Part 9



Part 9-11



Evaluation

The ultimate goal of ARP poisoning is to redirect and intercept traffic intended for one sending computer. The results are then sent back to the attacker’s computer instead of the sender. Interceptions include data packets, modified packets, and disruptions. This can be done due to the lack of authentication for ARP. MAC address is switched via the cmd command line, deleting the current cache of ARP, creating an automatic entry of the victims IP, and assigning the IP to the attackers MAC address. The network will the send the data to the assigned MAC address. In order for this to be successful, the attacker must have access to the network must be made to perform the ARP -a (a relatively easy feat). This overrides the firewalls of a standard “ping” command and displays the MAC address regardless of firewalls installed. The attacker must have some form of administrative rights as well.