2018

University of North Carolina Wilmington

Master of Science in

Computer Science and Information Systems

Proceedings

https://csbapp.uncw.edu/mscsis

MALWARE ANALYSIS: VIRTUAL NETWORK CONFIGURATION AND PROCESS DEVELOPMENT

Lex Dunlap

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2018

Approved by

Advisory Committee

Minoo Modaresnezhad, Committee Member	Lucas Layman, Committee Member

Jeffrey Cummings, Chair

Accepted By

Dean, Graduate School

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

As more data and business operations are being managed through computational infrastructure, maintaining defenses against malicious software, or malware, is becoming more important. Malware attacks are continuing to evolve so quickly that some industries may find it helpful to manage investigation into malware targeting their devices in-house. This project outlines the development of a secure virtual environment, designed to provide a place to conduct malware analysis which will garner details into the purpose of the malware and the vulnerabilities of the target's system which are being exploited. Additionally, the creation of a comprehensive tutorial was a part of this project, allowing interested parties to duplicate the environment. In order to ensure security, the project includes a series of documents outlining standard operating procedures which cover operator-based responsibilities and change controls. While additional technology-based controls would limit the burden of maintaining an isolated virtual network for the user, the current infrastructure maintains a network which successfully limits the scope and effectiveness of any malicious software being tested. Having the ability to analyze malicious artifacts in-house can provide institutions with a comprehensive understanding of their system's vulnerabilities, providing them with helpful knowledge which can be used to create valuable defenses against future attacks. Organizations will also be gaining an understanding of which file locations have been accessed and what data may have been compromised. This allows for organizations to be able to responsibly address the actual results of the attack.

INTRODUCTION

Information security, or the practice of protecting an institution's information resources, has drastically changed since the dawn of the computer. While some sectors have diligently attempted to continually update and enhance their information security measures, many companies have implemented bare minimum requirements, leaving them exposed to an array of possible exploitations. Recently, amidst leaks of private tax information, sluggish/dissatisfactory responses to security breaches, and concerns regarding how individual's private information is being shared between organizations, most people have been made acutely aware of the need to increase their own personal security measures and are starting to hold more corporations and organizations to the same standards. This shift in priorities comes at a time where there is a large discrepancy between the need for information security professionals and the number of people qualified to fill the positions [1].

Monitoring and maintaining robust security measures is easier said than done. As the technology involving information security continues to evolve and more robust measures become implemented and standardized, cyber attackers with malicious intent continue to develop more intelligent designs for infiltrating systems and data gathering. An integral component of the cyber security process is the responsibility of the Malware Analyst who is tasked with understanding what the malicious code that reaches a victim is designed to do. Malware Analysts also help to develop methods of mitigating the issue through reverse-engineering the code and recommending how to bolster network security to account for similar attacks in the future. Malware is software that is intended to either damage, monitor, or take control of an electronic device. The methods in which malware is installed upon a device varies from users unknowingly downloading malicious software, to visiting an unsafe website, to leaving their physical hardware unsecured. According to Malware Analyst Adam Kujawa, "They accomplish their task by using various tools and expert

level knowledge to understand not only what a particular piece of malware can do but also how it does it [2]."

While there are many resources available for individuals to learn the technical skills relating to Malware Analysis, few resources take you through all steps to configure a secure, isolated network, and assist in developing a starting point for actual analysis of the malicious artifacts. Providing interested individuals with resources, and security control considerations, would provide users with more context and may allow them to make better assessments as they are considering elements beyond simply using tools or developing a minimalist configuration with bare bones security backing. Within organizations, Malware Analysis is often aligned with network security and cyber forensics, as malware is often the cause of cyber incidents (e.g., cyber breaches). Ensuring that security concerns are an integral part of the configuration process can result in more nuanced conclusions on the part of Malware Analysts, especially within the context of a security incident or investigation.

The research question posed for the purpose of this project is to address the question of how to provide new learners of Malware Analysis with the tools to be the most effective at preventing, detecting, and deterring successful malicious activities while focusing on maintaining network and system security throughout the analysis process. This project entails the implementation of a virtual environment and the introduction of malicious software into the victim machines within that environment. Both basic static and dynamic analysis tools will be utilized in order to demonstrate how these tools work, and how the information gathered can be pieced together to develop helpful insights. Additionally, I will be considering how to incorporate the larger context of security and controls through the introduction of Standard Operating Procedures (SOPs) which should be followed at all times in order to maintain network segregation, consistency of analysis processes, and maintain a focus on general security practices.

BACKGROUND RESEARCH

In reviewing past works and other relevant materials, this research attempts to find justification in the benefits of Malware Analysis by providing an inclusive history of cyber tactics, technical skills, and sociological human behaviors that can affect malware development, deployment, and receipt. By providing a comprehensive tutorial, I hope that organizations and interested individuals will be able to review and develop the skills necessary to gain an understanding of networking, the technical security concerns that should considered when constructing a network for the purpose of testing or implementing any sort of unknown software, and gain experience analyzing malicious artifacts.

Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software outlines a highly detailed tutorial ranging from a basic static analysis of malware coding and components, to in-depth reverse engineering, disassembly, and debugging [3]. The book provides insight regarding how to set up a virtual environment, which tools you need, where and how to install said tools, and shares insight in regard to commonly faced bugs, and potential pitfalls that many users face. This book has been used and recommended by malware experts and is often considered one of the most comprehensive books on the topic [4]. While *Practical* gives an all-inclusive immersion into the world of malware analysis, it uses tools mostly available on the Windows platform only. It also uses the product, VMware, to set up the virtual environment where part of the dynamic analysis takes place. While VMware does possess many helpful tools, free versions of the software may not support all of the necessary functionality for proper network configuration and analysis. Another tutorial on Christophe Tafani-Dereeper's blog, simplifies the ideas presented in *Practical*, providing a complete step-by-step guide to setting up an isolated virtual network through VirtualBox and using more tools grounded in the Linux operating space [5]. While other tutorials may provide insight on basic construction of a virtual environment, and even some of the

malware analysis tools which could be useful in beginning to conduct analysis on your own, they do not focus on constructing or following security controls. By adding the utilization of a firewall to the project's tutorials and environment, and providing a central focus to following a set of procedures intended to limit the possible negative impact of escaped malware, users of the project can acquire an additional understanding of the implementation of cyber security practices within the context of malware analysis.

Project Overview

While Malware Analysts should be aware of a wide range of issues, spanning from the knowledge of human behavior to understanding the intricacies of the target's systems and their assets, the most integral part of a Malware Analyst's job is conducting an analysis of the malicious artifact itself. It is necessary to conduct this analysis in a properly configured environment in order to limit the possibility of malware spreading to other areas within an organization's system or infecting the host machine itself. Although there is no sure-fire way to guarantee that malware will not harm any devices during analysis, especially during dynamic analysis where Analysts actually run the software, there are reasonable steps to take in order to configure an environment which limits the possible damage of malware exposure. Additionally, by following user-based controls, Analysts can maintain a routine of safe practices through consistently following security-based procedures during the analysis process.

The completed project addresses the concerns of Analysts in need of a secure environment by developing a virtual system with additional security measures and producing documentation to assist future analysts in replicating the environment, as well as following security measures focused on limiting the possible exposure of malware to other systems. The completed project includes three main components. First, a technical deliverable that provides an infrastructure for running a comprehensive basic analysis of the malware. Second, a tutorial which comprises all steps to configure an environment which keeps security controls as a central focus. Finally, a series of formalized documents for Standard Operating Procedure SOP, which provides additional security controls, and standardization as it applies to the analysis process.

Project Details

The following section discusses the three main components of the framework developed including the design and execution of a technical tutorial, development of a virtual environment

on a dedicated machine, and a set of procedural standards which will dictate the processes for an analyst.

Using a mixture of available tutorials, a virtual network was designed to include an analysis machine and several victims on different operating systems. The analysis machine uses Ubuntu 16.04 64-bit operating system, as will one of the victim machines. After comparing the processes described in several tutorials, I recognized that utilizing the Ubuntu operating system as the basis for the analysis machine would be a more efficient implementation than using a Windows machine, mainly for the utilization of the iNetSim service, which functions as a robust network monitoring tool which is only compatible with Linux OS. I used Ubuntu 16.04 64-bit as it was the most recent version of the OS. The victim machine's OS is Windows XP 32-bit OS, which was selected for the purpose of testing malware analysis results against known outputs from artifacts which were built for Windows XP OS. While the main testing occurs with the Windows XP VM, the VirtualBox system used on the host machine has also downloaded copies of Windows 10 and Windows 7 which are configured for use as victim machines as well. Additionally, the tutorial instructions can be replicated for any OS, as long as users have access to change network interface/adapter settings within the VM.

There is some analysis software that can only be run on Windows machines making it valuable to have a Windows victim with a different set of analysis tools available. This virtual network is designed to be entirely isolated from both the host machine and the internet. The analysis machine is configured to also operate as the server and will be used to reproduce activities that occur online, including HTTP, SMTP, and DNS. The analysis machine uses iNetSim to handle the "internet" activity and requires remapping of its IP configuration and mapping. All network victim machines are remapped with unique IP addresses within the network and will be using the analysis machine as the gateway address [5]. iNetSim automatically creates a log of activity that

will be used during analysis. The analysis machines also require the use of Burp, a service that provides CA certificates and allows for SSL monitoring as well.

In order to ensure the validity of the malware analysis process and the developed environment, the malware analysis activities outlined in [3] were replicated and the expected results were compared to the outputs found within the virtual environment. Using a variety of tools (including; PEid, "Strings," UPX, Dependency Walker, PEView, Wireshark, and Netcat), data from the virtual environment has been collected, reviewed, and evaluated against the known output outlined in [3] to ensure accuracy of the results found within the environment. The project also provides users with SOPs and tutorials to garner a strong and strategic basis for gathering a broad swath of initial data and intelligence relating to the malicious artifact. Standardizing the analysis process should result in a secure method of analysis and formalized method of processing the data retrieved from the analysis.

TECHNICAL DELIVERABLE

The completed project is designed to offer a safe environment for the acquisition and analysis of malicious software. This was constructed on a University of North Carolina Wilmington (UNCW) owned laptop and designed to run on a series of Virtual Machines (VMs). VMs are connected through a virtual network in a strategic fashion in order to provide security and isolation through the process of acquiring the malicious artifact and safely running the malware in order to analyze the effects of the software.

HOST MACHINE

The host machine, as stated previously, was acquired through the University and has been dedicated to operating as a malware analysis device. Any access to the internet that occurs through the VMs will use the host machine, which is configured to access the internet through UNCW's network, using UNCW credentials. The IP address of the host machine is based on the DHCP

servers managed on the UNCW network and would change periodically. The host machine runs the 64-bit Ubuntu 16.04 LTS operating system, acquired from the Ubuntu website, and before beginning configuration for this project, the machine was wiped and given a fresh install of the operating system by booting from USB. The host machine has 15.5 GB of memory 475.4 GB of disk space and uses a quad-core Intel i7 processors at 2.6 GHz. The host machine made use of the following software for the purpose of developing and documenting the progress of this project:

- VirtualBox – used as the basis for creating a virtual environment and managing VM settings.
- Terminal – used to manage OS updates, software downloads, and some network configuration.
- Mozilla Firefox – Access to web browser for the purpose of updating documentation, gathering research, and managing project resources.
- Pinta – Screenshot modification including, highlighting, numbering, or cropping areas for the purpose of clarification in training or procedural documentation.
- Screenshot – Take visual documentation of the processes and outputs which occur throughout the development process.

FIREWALL VM

The firewall operates on a VM within the VirtualBox infrastructure located on the host machine. The operating system for the Firewall VM is FreeBDS, an open source OS descended from the original Berkeley Software Distribution (BDS) OS. The Firewall VM contains 512 MB of memory and runs on an Intel i7 processor at 2.6 GHz. Two network adapters are enabled through the VirtualBox settings. Initial configuration attempts tested using a bridged adapter, which failed, as the UNCW network does not allow for multiple IP address from the same user account, and the bridged adapter provides a unique IP address separate from the host's IP address. In order to

address this issue, Adapter 1 was attached to a Network Address Translation (NAT), which essentially adopts the host machine's IP address and uses that to access the network access which the host machine is connected to. Adapter 2 was set to an internal network named, "intent," which is the default internal network in VirtualBox.

The firewall uses pfSense, which was acquired from the pfSense website, and was installed through booting from an optical drive which was attached through VirtualBox. After the installation of pfSense to the VM, configuration occurred to assign each of the adapters to the appropriate field (WAN or LAN). Adapter 1 was set to WAN, and allowed for DHCP connection as managed by VirtualBox, this is a separate process from the VM accessing the UNCW network so using DHCP here did not cause any issues. Adapter 2 was set to LAN. The IP address for the LAN adapter was set to 172.16.1.1 with a subnet mask of 255.255.255.0.

Additional configuration was set up through a web portal which managed the rules and restrictions associated with the firewall. All access was denied, except for traffic coming from port 443. This denial included access from port 80. The purpose for implementing these restrictions was to ensure that no unexpected traffic was occurring without the user's knowledge. Based on the procedures outlined in SOP: 010, analysists are required to navigate to an approved site for malware acquisition, the approved sites will be located at an HTTPS address which will require access to port 443. The downloading of the artifact will also take place over this port, and any other activity should not be occurring. By blocking all unexpected ports, the environment limits the possibility of malicious actors or software from taking advantage of unprotected port traffic. An anti-lock rule was set in place to ensure that users were able to access the portal regardless of how strict the firewall settings were.

Figure 1 shows how the host machine, firewall, File Share VM, and Victim Machine are connected. Using two different internal networks, allows for the VM which will run the malware to never have a direct connection with the host machine, providing an additional layer of security.

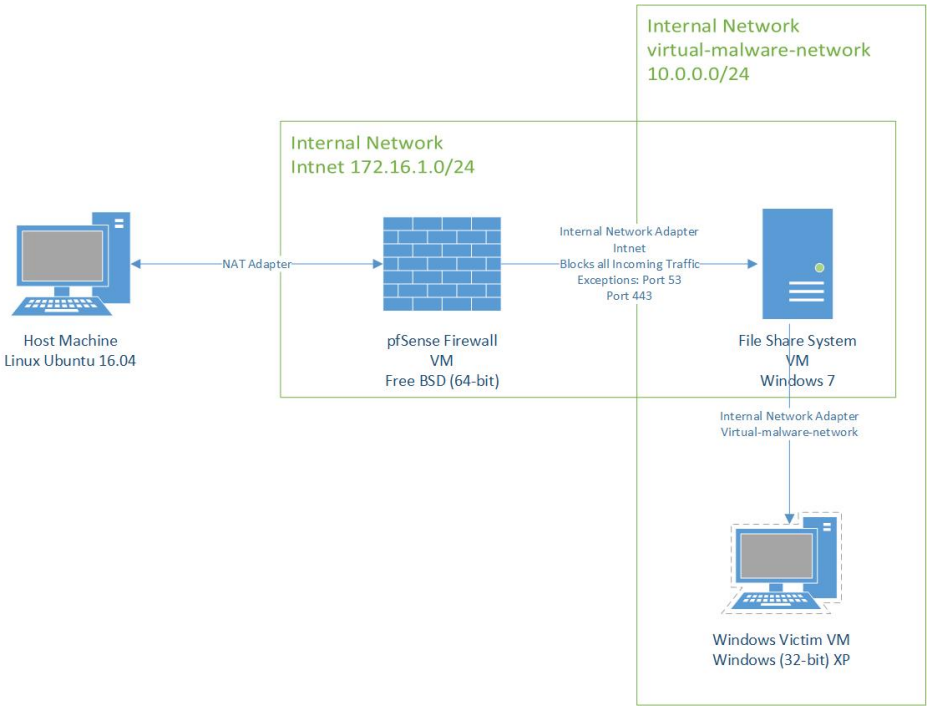


FIGURE 1 - NETWORK CONFIGURATION FOR FILE SHARE SYSTEM

FILE SHARE VM

A file share VM was created with the purpose of acquiring malicious artifacts from known/approved locations through the use of a firewall. The VM was then connected to an internal network (with all access to external networks shut off) in order to share files with the VM that will be running the malicious artifact for dynamic analysis. The file share VM runs Windows 7 32-bit OS as obtained through Microsoft's developer site. This VM has VirtualBox guest additions installed on the VM, which allows some graphical advantages, like expanding window size, as well as additional features. The VM contains 4096 MB of memory runs on an Intel i7 processor at 2.6GHz and the OS was installed through an added optical drive upon the first run of the VM. The

file share has 2 network adapters enabled, however, for security purposes, only one of these adapters should ever be enabled at one time.

Adapter 1 is set to attach to an internal network named, “intent,” the same as the pfSense firewall VM. Within the VM the IP of adapter 1 is set to 172.16.1.2 with a netmask of 255.255.255.0, and the gateway, as well as the DNS lookup IP address, as 172.16.1.1 (the IP address of the firewall). Using a web browser, when adapter 1 is enabled, the VM can navigate to https://172.16.1.1 and access the web portal for further configuration settings for the firewall.

Adapter 2 is set to attach to an internal network named, “virtual-malware-network.” This network will be used exclusively for the transfer of malicious documents from the file share VM and any victim machine which may be tasked with analyzing malicious artifacts. Adapter 2 should only be enabled when adapter 1 is disabled. The IP address of adapter 2 is set to 10.0.0.6 with a netmask of 255.255.255.0. The gateway and DNS lookup IP is set to 10.0.0.1. This step is not strictly necessary as there is not a need to have a common gateway for file sharing, however, this did assist in set-up configuration convention and was left in to the final project.

Although Internet Explorer 11 is pre-installed on the file share VM, Mozilla Firefox was installed as it operated with more efficiency. Additional configuration was conducted to share a folder, named “Shared Malware Folder” which was located on the file share VM’s desktop. This folder had granted read/write privileges to the user group: Everyone on the “virtual-malware-network.” The reason the user group was so broad was in part due to the expandability of the number and type of victim machines created and housed in VirtualBox. As the analysts are in charge of the host machine, and all subsequent VMs on the host machine, there is a relatively low risk of an unwanted user accessing the shared folder.

WINDOWS XP 32-BIT VICTIM VM

The victim machine uses a specific operating system for the purpose of testing malware designed for that operating system, however, the settings can be reproduced with any available operating system. This particular victim VM used Windows XP 32-bit as the OS. The VM has 4096 MB of storage and uses an Intel i7 processor at 2.6 GHz. There are two network adapters set up, but only one should be enabled at any one time. Additionally, this VM has VirtualBox guest additions installed in order to have better visual control and documentation.

Adapter 1 is set to connect to an internal network named, "malware-analysis-network." This internal network is used for the purpose of simulating a network for the malware. Additionally, it will be connected to an analysis machine which will monitor port traffic and serve CA certificates over port 443 (HTTPS) when called. The IP address is assigned to 10.0.0.33 with a netmask of 255.255.255.0. The gateway and DNS lookup IP is set to 10.0.0.1. The gateway and DNS lookup IP is set to the same IP address as the machine which will be configured to operate as the analysis machine.

Adapter 2 is connected to the internal network named, "virtual-malware-network." This network is used exclusively for transferring files from the file share VM, which acquired files through the firewall. Adapter 1 and adapter 2 should never be enabled simultaneously, to ensure segregation of networks and minimize escape vectors for malicious software. The IP address for adapter 2 is set to 10.0.0.7 and the gateway and DNS lookup IP is set to 10.0.0.1. Again, the gateway and DNS lookup are only configured for consistency with settings conventions throughout the project. Accessing folders in the Shared Malware Analysis folder mandates that the victim machine navigate to \\10.0.0.6 (the IP address of the file share VM) within File Explorer. Access may be restricted by prompting the analyst for username and password associated with the file share machine, however, the analyst has access to this information.

For the purpose of analysis, various software packages have been installed on the victim machine. All of the following packages are freeware developed for the purpose of malware analysis:

- ApateDNS – a tool used to monitor and capture traffic over port 53 (DNS)
- Dependency Walker – provides a detailed look into Doubly Linked Lists (DLLs) and Portable Executable (PE) files examine imports, access and other functionality
    - PE files - executable file types which can be ran in Windows machines, they contain valuable header information including strings, imports, exports, and file locations it needs access to.
    - DLL - give a list of libraries, located within the host machine, which need to be accessed in order to run the file. These libraries are kept in their respective locations on the host machine and simply accessed and ran when called within the program requires
- PEid – shows valuable information from PE files, including what the file was wrapped/compiled with
- PEview – shows information from PE files including imports
- Process Explorer – shows all processes occurring on a machine, helpful for monitoring the operations and activity of malicious files
- Process Monitor – shows all operations and provides details about the occurrence
- RegShot – takes a snapshot of the registry at two points of times and shows the changes made in registry locations
- Resource Hacker – a way of finding and exporting hidden binary files located within malicious files
- Strings – can list strings and imports found in malicious software

- WinMD5 – helpful for finding hash values based on the MD5 algorithm, a common convention in classifying malicious software
- Wireshark – monitors packet traffic over a network

UBUNTU ANALYSIS VM

The Ubuntu Analysis VM is a necessary component in the analysis process of malware. In addition to providing an additional end point in the virtual network, it operates as a gateway to the internet. Having all network traffic funneled through the Ubuntu analysis machine allows for monitoring systems, like iNetSim, used in this VM, to document and report on all network traffic on all ports. Additionally, iNetSim is able to offer fake packets back to the victim machine to fool the malware into thinking it has received the requested packets and information. The Ubuntu analysis machine also utilizes a tool called, Burp. Burp, provides certificates from the CA when requests occur over port 443 (HTTPS) which require certificate receipts. Utilizing both of these tools fulfills two main functions 1) that the malicious software does not suspect that it is operating in a virtual environment and continues to operate as it would in a live machine and 2) that there is a log of traffic which is stored and can be reviewed later.

The Ubuntu analysis machine uses the Ubuntu 16.04 LTS 64-bit operating system. This was acquired from the Ubuntu website and was loaded upon booting the machine from the optical drive. The memory on the Ubuntu analysis machine is 4096 MB and contains 12 GB of disk space. Only one network adapter is enabled for this VM, and it is attached to the internal network named, “malware-analysis-network.” The IP address of this machine is set to 10.0.0.1.

Additional configuration occurred for iNetSim, to ensure that iNetSim was provided with access to network features within the machine. This was handled in the inetsim.conf file and amended to ensure that the iNetSim program had access to all required services. Burp

configuration mandated that Burp monitor activity over port 443 on the localhost VM it is installed on.

In Figure 2 a diagram of the network configuration settings, including the network for the actual analysis of malware is shown. The segmented internal network titled, malware-analysis-network is isolated and none of the virtual machines on that network should ever need to access outside network resources.

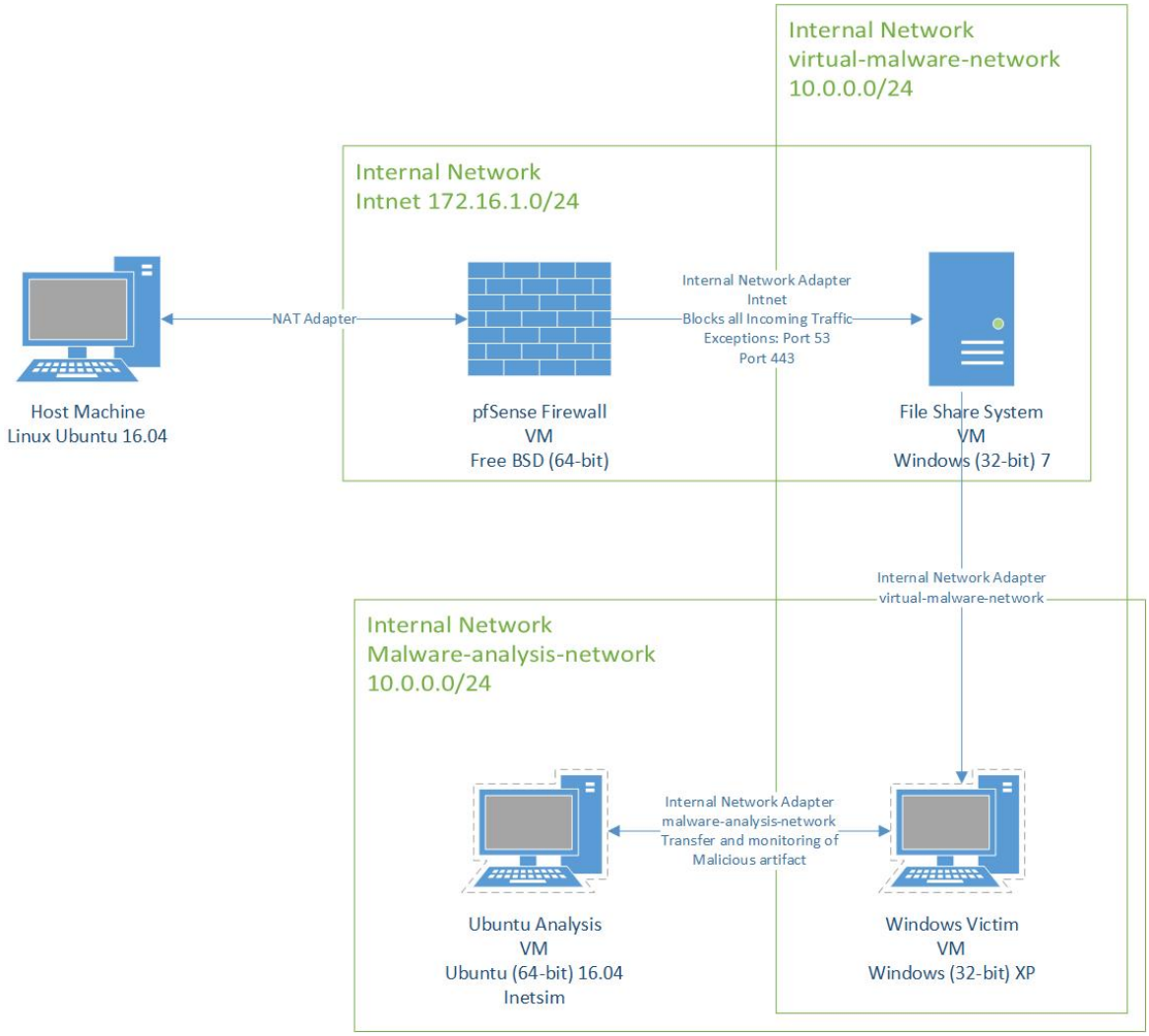


FIGURE 2 - INTERNAL NETWORK CONFIGURATION FOR MALWARE ANALYSIS

ONLINE TUTORIAL

The purpose of developing tutorials (Appendix A&B) is to provide individuals with all the tools and instructions to be able to duplicate the environment outlined within this paper on their own. In addition to being a teaching tool, the tutorials are designed to provide additional information to assist in troubleshooting for the possibility of some of the instructions changing over time. By adding some additional context, users of the tutorial should have enough information to quickly search for finding other compatible programs or finding replacement solutions.

The tutorials developed are posted on https://virtualnetworktutor.wixsite.com/malware-analysis and cover the entire creation process while providing additional context to assist in future users of the tutorial to acquire other similar software, or make their own adjustments at a later time if necessary.

The tutorials cover the entire process of environment creation and tips for analysis. Topics covered in the tutorial include:

- VirtualBox installation
- Creating the first VM
- Creating and configuring Ubuntu Analysis machine
- Creating the firewall VM
- Firewall configuration
- Network configuration
- Additional firewall settings
- Configuring shared folder
- Final settings adjustments and controls
- Tips for static analysis
- Tips for dynamic analysis

The tutorials include detailed instructions with additional troubleshooting tips at the end of each section. It is designed for users with some base level technical skills, but who may not be incredibly familiar with networking. Some terms are defined, and steps which are not inherently obvious as far as their intention are explained in additional detail.

Standard Operating Procedures

The Standard Operating Procedures (SOPs) are an integral part of the project but are not included as a part of the online tutorial. The online tutorial does include a section on advising how to come up with set of instructions which should be used as a security check list, similar to the SOPs in the project, but as each use case for malware analysis is different, and the network the host machine is connecting to will be different in each case, it is best for users, in conjunction with network administrators, to come up with a unique set practices designed specifically for the environment the Malware Analyst will be working in.

The SOPs are included as a part of this project for two purposes. The first, is to provide a structured system of instructions for use as a malware analysis tool in an organizational context. By developing standardized methods for how to go about an initial investigation of malware, users can control for consistency and the organization will have a base line of data gathered consistent for all malware gathered analyzed that can be used to decide on future investigatory measures.

The second, and perhaps more important, purpose of the SOPs is to include human security controls designed to limit the possible scope of escaped malware during any acquisition, transfer, or analysis of malicious artifacts. The procedures outlined in the SOPs will provide a structure that mandates users turn off networking capabilities during any analysis, use only necessary network adapters on VMs, and utilize a firewall during any access to online resources.

By mandating future analysts adhere to this very specific set of processes, organizations can mitigate some risk for users adding vulnerabilities to their network. While the technological components are set up in a way to maximize segregation and properly simulate other network requirements, the environment is not self-operating and requires an analyst to conduct additional adjustments to VirtualBox settings and network settings.

The SOPs address segregation and isolation of networks, changes to network configuration in the future, authorization for access to sites, and sets a broad set of expectations relating to what sort of data should be collected in order to review malware. The following are the SOPs developed and how the address the previously mentioned goals:

- SOP: 010 Malware Acquisition and Transfer (Appendix C) – mandates approval for website access, requires analysts to ensure that only network adapters that are required for the purpose of acquiring malicious software are enabled, ensures analysts are using only a clean snapshot of the VM, dictates that VMs be returned to their original clean state prior to completion of the acquisition and transfer process.
- SOP: 011 Malware Analysis (Appendix D) – mandates a broad range of data to be collected, dictates a logical order in which to approach an initial review of a malicious artifact, states that analysts must only have network adapters enabled which are absolutely required in the analysis process, requires that analysts return all VM states back to their original clean state prior to completion of the analysis process.
- SOP: 012 Malware Network Change Control (Appendix E) – sets up a required process to follow should any changes in the configuration be attempted, mainly updating all relevant procedures and processes that are affected so that future analysts and users are operating with accurate and up to date information.

TESTING AND RESULTS

While the completed project attempts to provide a secure environment developed for the purpose of analyzing malware, additional testing must be conducted in order to ensure that the environment is configured properly, is functioning securely, and is able to produce expected results from conducting an analysis. There are three main goals which have dictated the construction of this project; Security; Connectivity; and Analysis. Throughout the development of the processes listed in the SOPs and through the design and creation of the virtual machines and environment, there have been controls included to ensure the security of the host machine. Additionally, the controls attempt to limit and contain the spread of malware to any other machine while simultaneously maintaining connectivity within the virtual network where the analysis will be conducted. The SOPs are also designed to provide a baseline of data acquisition from analysis of the malware to ensure a minimum amount of data yielded from the analysis. Testing for each of these categories (Security; Connectivity; and Analysis) will include an examination of the effectiveness of each of these categories.

In the subsequent sections, a thorough inspection of the environment is conducted to ensure stability of the virtual network. First, security and connectivity tests are conducted by testing the effectiveness of the controls and SOPs stated in the capstone. This is followed by an evaluation of a number of malware labs to establish the stability of the environment to conduct both static and dynamic malware analysis.

SECURITY

There were six sets of controls which were tested for against both the Standard Operating Procedures, and the completed virtual environment:

1. Control: The host machine should have all networking capabilities shut down or disabled during the analysis process.

This control is implemented in SOP: 011 IX.A.1, “Before conducting any malware analysis turn off all networking capabilities of the host machine” (Appendix D). While this measure is dependent on the individual conducting the analysis to be vigilant in following the procedure, having this specifically outlined as a requirement in the SOP should provide a meaningful deterrent for unwanted network access during analysis. This does leave a vulnerability of human error to be considered as a flaw with the implementation of this security control. In SOP: 010 IV.A.8 the instructions, “8. Navigate to a pre-approved website in the “File Share System’s” browser,” are listed. This is an important step, which requires a high-level security administrator to have reviewed and approved the acquisition site prior to download or access. This ensures that the method of installation comes in a compressed file format, so as not to accidentally run the malware until it is intentionally run within the virtual environment. Additional measures could be added by limiting the host machine’s network access to a sandbox, providing it with a very limited segmentation of the network. This would allow for the host machine’s MAC address to only access the world wide web. For the scope of this project it did not seem necessary to implement this level of segregation, and the human controls implemented in the SOP were deemed sufficient. However, adding these measures in future work could be a valuable addition.

2. Control: Virtual machines should only have adapters connected to the relevant internal networks enabled.

This control is implemented in SOP: 011 IX.A.3&4 (Appendix D). The control requires the individual conducting the analysis to review all adapters on the victim VM and analysis VM before starting either virtual machine. Again, the main flaw with this implementation is that it is

dependent on a person's performance and attention to detail and could be improved with a technological solution.

3. Control: When acquiring malicious artifacts, or additional software, any connection to the internet must be made through a dedicated VM used only for File Sharing, which must be configured to connect to the internet through a secure firewall.

This control is implemented in SOP: 010 IX.A 4&5 (Appendix C). While the entire acquisition process including additional configuration standards are outlined in SOP: 010, the specific access to the dedicated VM for file sharing, and the VM containing the firewall are addressed in line A. 4&5 of SOP: 010. The technological controls are set so that the firewall has already been configured to be restrictive, only allowing for access across port 443 (HTTPS). This doesn't necessarily restrict the user conducting analysis from accessing a harmful site over this port, although the SOP clearly restricts accessing any URL other than an approved acquisition site. This approval is dependent upon a supervisor within Information Security who has the knowledge and skillset to make the assessment as to whether a site should be approved for acquisition or not. Additionally, the process requires the user to follow the SOP and start both the firewall VM and the file share VM, and to test the stability of the firewall prior to navigating to the acquisition site. This leads to vulnerabilities possible over port 443, and vulnerabilities occurring due to negligence or oversite on the part of the individual conducting the analysis.

4. Control: Virtual machine states with live malware should be discarded when not in use or when analysis has completed.

This control is addressed in SOP: 011 IX&IX.B.10 (Appendix D). This procedure mandates that anytime a victim machine is operating it must be restored to its previous clean-state snapshot when either a) it is not in use or b) after analysis has been completed. This ensures that no

individual will have access to a version of the VM which has active malicious software running on it and may use it without the proper security procedures. This is dependent upon individual acceptance and utilization of the protocol and could be improved by restricting access to the ability to save states on victim machines or creating additional snapshots on certain machines.

5. Control: Virtual machines should always be started from a clean state snap shot.

This control is addressed in SOP: 011 IX.A.4 (Appendix D). The implementation of this control requires users to verify that the current state of the VM is unchanged from the pre-analysis snapshot which was created before any malicious software had been inserted or ran on the VM. It also addresses the scenario in which the current state has been altered, by mandating users to boot from the previous snapshot, by selecting that version, and booting directly from the historic state. This control does require acceptance and utilization from the user and could be improved with a stricter set of technological controls.

6. Control: Technological controls should be in place to limit the possibility of human error to affect the safety of the analysis process.

This control is addressed through the utilization of a virtual environment, an isolated network, a secured firewall, and the basic infrastructure which is designed to have multiple barriers of entry and to limit the spread and scope of any software which may escape from the virtual environment. That being stated, there are, as listed above multiple concerns with the weight of dependency on human interaction within the virtual environment to ensure security. Adding additional and customized features to limit certain processes and minimize risk would help in addressing the limitations of the current controls.

CONNECTIVITY

While the host machine should not have any access to networking functionality, the virtual machines must exist in an environment which simulates an operational network infrastructure with access to all ports and an imitation of expected network responses when necessary. Testing occurred between the "Ubuntu Analysis" machine and the "XP 32-Bit Victim" machine. The network adapters for both of the VMs were attached to an internal network named "malware-analysis-network" and had been configured with the "Ubuntu Analysis" machine IP address set to 10.0.0.1 with a network mask of 255.255.255.0 and operating as the gateway and DNS lookup IP for the victim machine.

1. Process: Confirm network settings has all adapters configured properly (Appendix D).

The SOP: 011 requires users to ensure that all network adapters on the VM are configured appropriately (Appendix D). SOP: 011 also requires booting from a particular snapshot which has the internal settings already set to be attached to the virtual network without requiring additional contributions. This does not necessarily ensure that there will not be other settings or changes made to the current configuration in the future. If any settings which affect the connectivity of the virtual network are adjusted, the SOP needs to be updated to reflect those changes accurately, these changes are addressed in SOP: 012 which handles change control regarding the configuration settings of the virtual network.

2. Testing: Ping for response from other VMs on the network.

Testing results based on the virtual network between the victim VM and the analysis VM were ran to ensure that both machines could ping to each other.

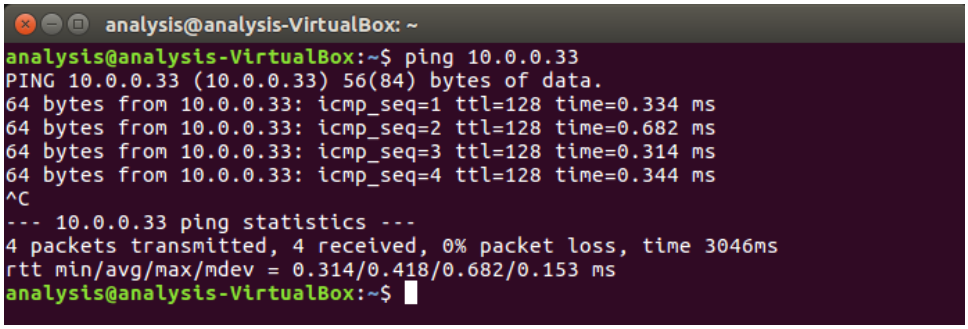


FIGURE 3 – CONNECTIVITY RESULTS FROM "UBUNTU ANALYSIS"

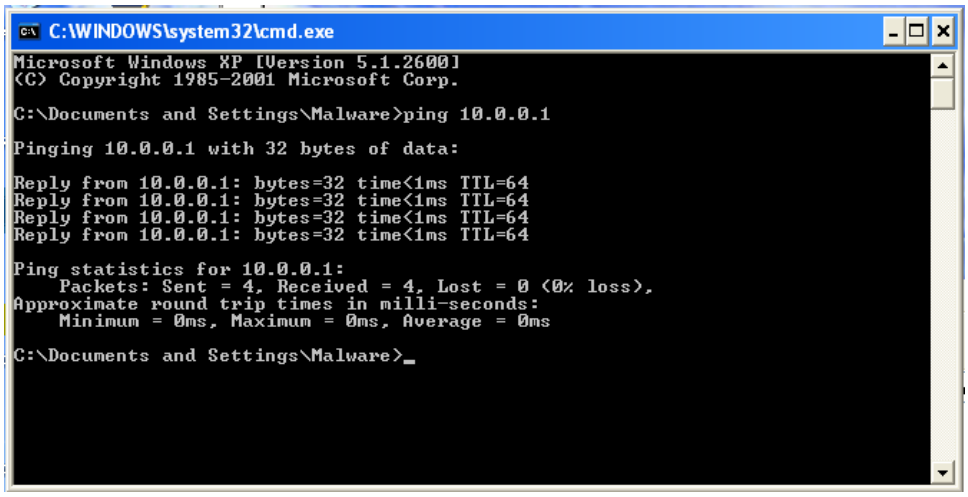


FIGURE 4 - CONNECTIVITY RESULTS FROM "WINDOWS XP 32-BIT VICTIM"

Both figures above show prompt responses from the other machine and provide a 0% loss. If any adjustments to the network are made in the future, testing to ensure that connectivity is occurring within the internal network should be conducted before creating new pre-analysis snapshots of the VMs.

3. Configuration settings/adjustment:

The configuration settings listed in the SOPs are static and presume that no network settings will be changed. In the future, however, there may be additional VMs added, more robust testing mechanisms implemented which may require a different network setting, or some other

purpose which will result in the SOPs currently written being no longer valid. In this case, users will need to reference SOP: 012, which outlines how to properly change and update the network configuration settings, including going back to SOP: 010, SOP: 011, and any other related procedure and ensuring those are updated with accurate, and secure, processes as well.

Analysis of Malware

While the primary goal of the capstone was to create a secure environment, procedures, and tutorial, a part of the testing conducted was to ensure that the analysis of malware within the developed environment yielded the expected results. Testing for the validity of the analysis process, goes beyond the original scope of the project, but is an important consideration when assessing the accuracy and usefulness of the tool developed. Using malware samples provided by www.practicalmalwareanalysis.com, the testing compares results gathered from the developed virtual environment and the posted results of the dynamic analysis [3]. The testing covers four separate malicious artifacts and confirms the results are matching the expected outcomes. Additionally, the Standard Operating Procedures, particularly SOP: 011, which covers Malware Analysis, are constructed in such a way as to provide a consistent method for obtaining preliminary data from malicious artifacts. Standardizing the analysis process results in analysts having a baseline of information to begin their investigation.

All testing occurred on the "XP 32-bit Victim" machine on an internal network named: "malware-analysis-network" with the "Ubuntu Analysis" machine operating on the same internal network. "Ubuntu Analysis" is configured with an IP of 10.0.0.1, used as the gateway and DNS lookup IP address on the "XP 32-bit Victim" machine.

1. Lab 03-01

The analysis results from *Practical Malware Analysis* for Lab 03-01 reviews the results of running the malicious artifact through a series of software designed to examine different functions of the malware. The following figures show the results gathered from the virtual network constructed for this project against the expected results taken from figures provided in *Practical*.

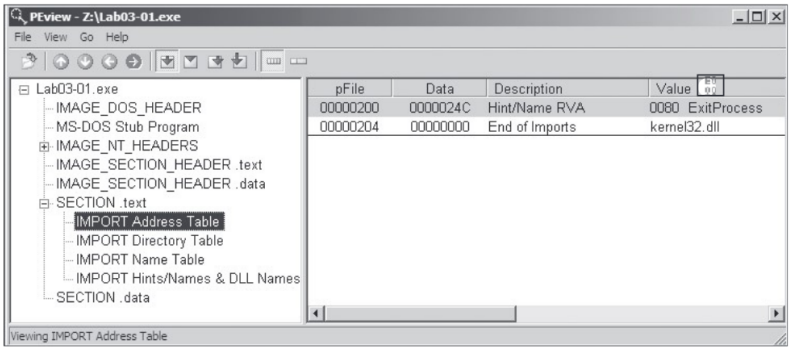


FIGURE 5 EXPECTED OUTPUT FROM PEVIEW

Figure 5 shows the expected imports found under SECTION.text > IMPORT Address Table. The figure shows both ExitProcess and kernel32.dll. These imports are minimal and do not express much insight into the funtionality off the malware, although the fact that there are so few imports may lead the analyst to consider that there is a possibility of the malware to be packed. When reviewing the output from the project's virtual environment in Figure 6 we see identical imports.

In Figure 7 we can see the expected output from running the strings.exe program. Strings of particular note would include the domain name, access to the VideoDriver, and registry locations. We can see that the expected results from the strings application matches the output gathered from the virtual environment.

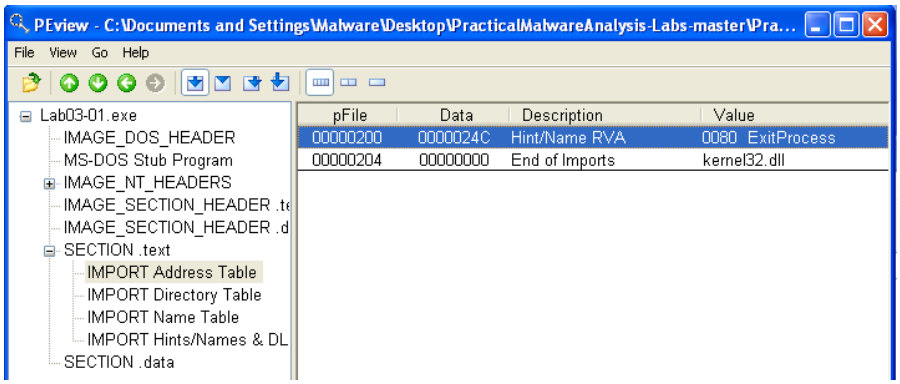


FIGURE 6 PROJECT OUTPUT FROM PEVIEW

```
StubPath
SOFTWARE\Classes\http\shell\open\commandV
Software\Microsoft\Active Setup\Installed Components\
test
www.practicalmalwareanalysis.com
admin
VideoDriver
WinVMX32-
vmx32to64.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
AppData
```

FIGURE 7 EXPECTED OUTPUT FROM STRINGS

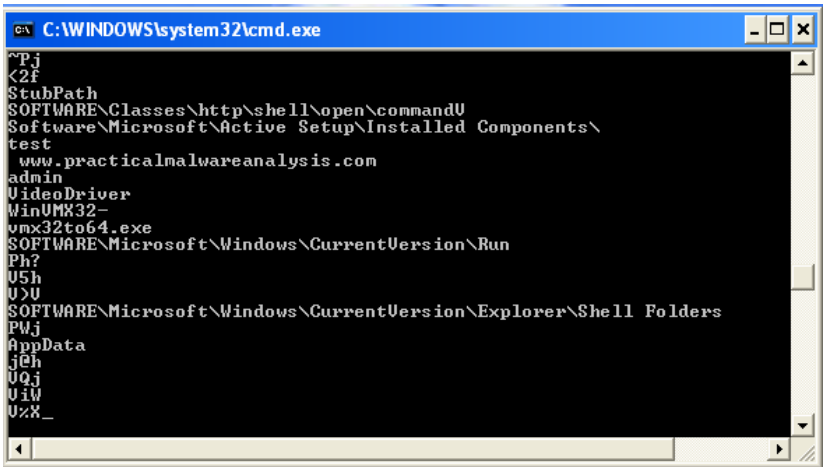


FIGURE 8 PROJECT OUTPUT FROM STRINGS

Figure 8 shows the result of a mutant file originating from the process of Lab03-01.exe. Viewing the output produced by the project's virtual environment demonstrates the same results, showing that it has created a mutated version of an original file. The results here do not occur until after the malicious software was run.

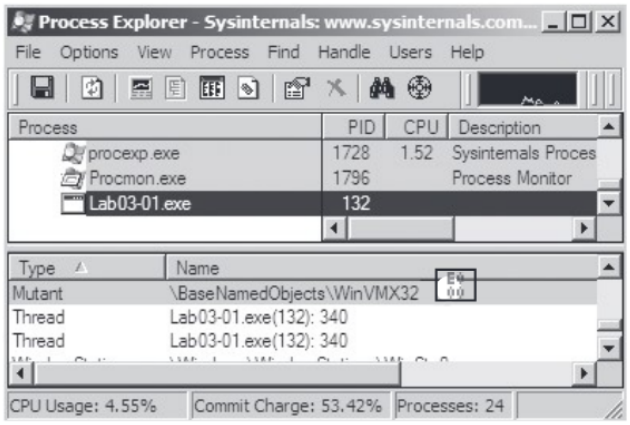


FIGURE 9 EXPECTED OUTPUT FROM PROCESS EXPLORER

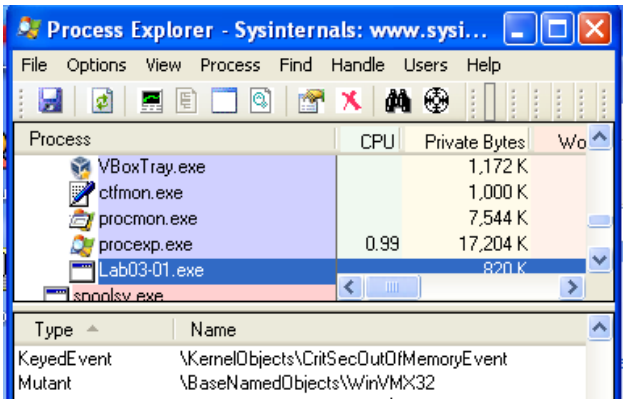


FIGURE 10 PROJECT OUTPUT FROM PROCESS EXPLORER

Reviewing the output listed in Figure 9 shows the operation WriteFile occurred in the directory vmx32to64.exe, which was one of the strings listed previously. Further exploration of this newly written file will determine that it is a copy of the malicious artifact. Additionally, the RegSetValue operation which creates the key named VideoDriver (again VideoDriver is as string found in one of the previous explorations) is used as a value to run the file created with the WriteFile operation in the previous line. Observing the output created by the project's environment proves to replicate the expected results.

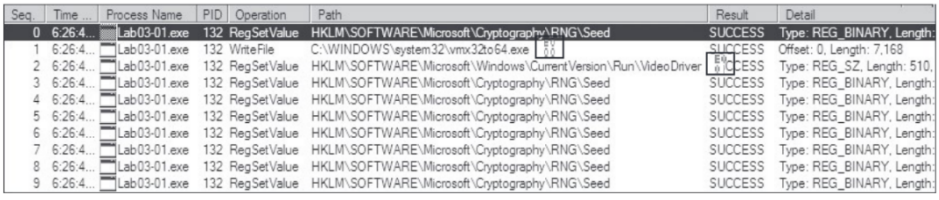


FIGURE 11 EXPECTED OUTPUT FROM PROCMON

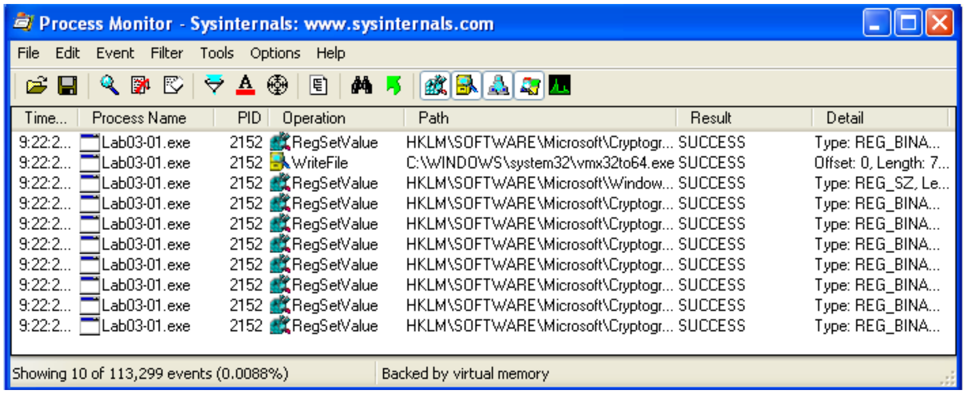


FIGURE 12 PROJECT OUTPUT FROM PROCMON

The expected results gathered from monitoring in Netcat show that there should be traffic occurring over port 443 (HTTPS). The project environment utilizes iNetSim and provides a more detailed and robust report which monitors traffic on all ports. The project's output does show traffic occurring over HTTPS, additionally port 53 (DNS) which is expected as it was attempting to ping the domain name listed in the strings in one of the previous outputs.

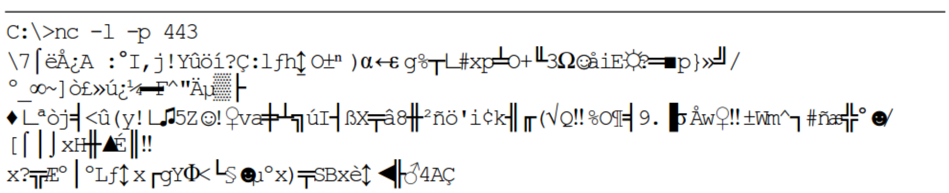


FIGURE 13 EXPECTED OUTPUT FROM NETWORK MONITORING

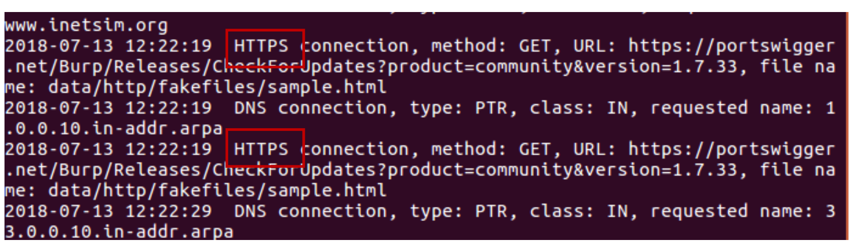


FIGURE 14 PROJECT OUTPUT FROM NETWORK MONITORING

2. Lab 03-02

The same process as above was conducted to ensure that analysis is being conducted properly. In the initial expected output shown in Figure 15, there is a list of exports, including Install and ServiceMain which may mean that the malicious software must be installed as a service. This output is of a DLL file, as opposed to an executable, which means there are additional steps to

running malicious artifact from the command line. The project's output shows an identical list of exports.

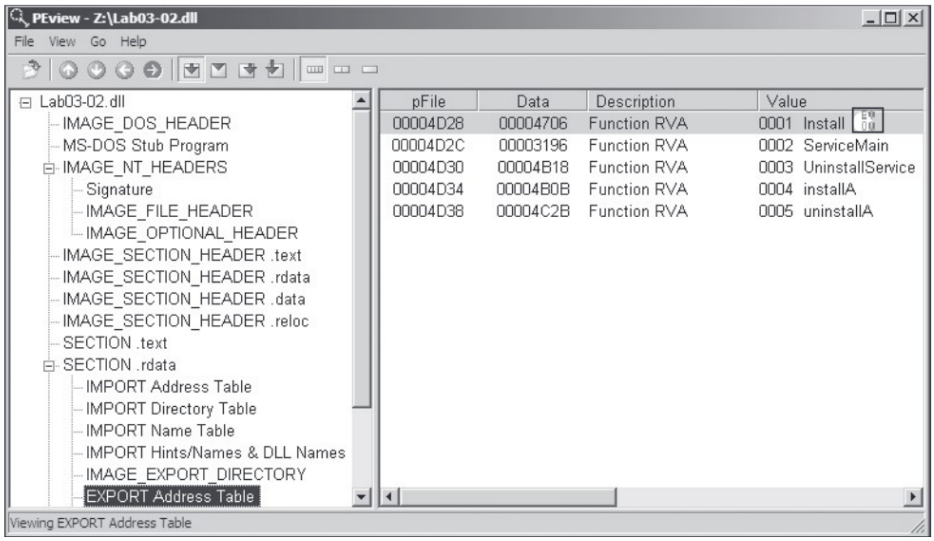


FIGURE 15 EXPECTED OUTPUT FROM PEVIEW LAB 03-02

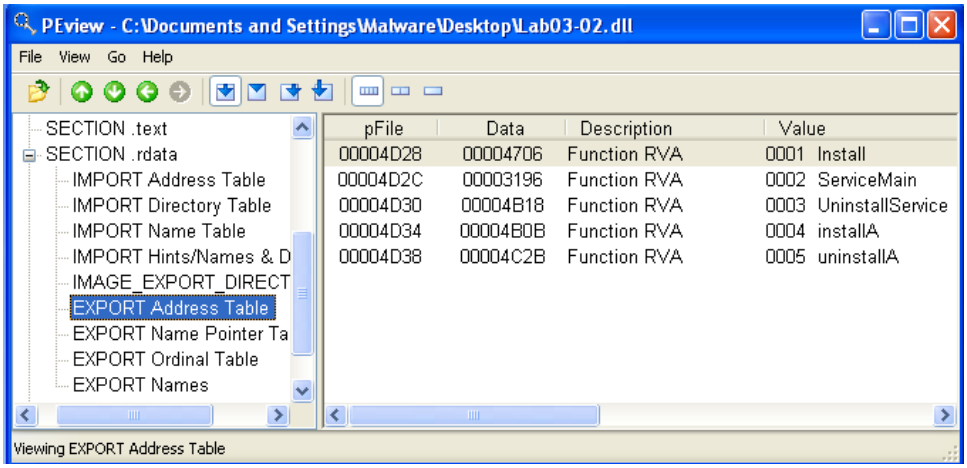


FIGURE 16 PROJECT OUTPUT FROM PEVIEW LAB 03-02

Figure 17 shows the expected strings and imported functions. The malware imports an array of functions ranging from networking capabilities to manipulation of services. Reviewing the Project's outputs demonstrates that there are all of the same outputs shown. The expected outputs have been curated to include only particularly relevant functions.

```
                    RegQueryValueEx
                    RegCreateKey
OpenService         RegSetValueEx
DeleteService       InternetOpen
OpenSCManager       InternetConnect
CreateService       HttpOpenRequest
RegOpenKeyEx        HttpSendRequest
                    InternetReadFile
```

FIGURE 17 EXPECTED OUTPUT FROM STRINGS LAB 03-02

FIGURE 18 PROJECT OUTPUT FROM STRINGS LAB 03-02

Further review of the strings output demonstrates the occurrence of a domain name and services including IPRIP. The strings collected from the project's output includes all of the items listed in the sanitized version of the expected output.

```
Y29ubmVjdA==
practicalmalwareanalysis.com
serve.html
dW5zdXBwb3J0
c2xlZXA=
Y21k
cXVpdA==
Windows XP 6.11
HTTP/1.1
quit
exit
getfile
cmd.exe /c
Depends INA+, Collects and stores network configuration and location
information, and notifies applications when this information changes.
%SystemRoot%\System32\svchost.exe -k
SYSTEM\CurrentControlSet\Services\
Intranet Network Awareness (INA+)
%SystemRoot%\System32\svchost.exe -k netsvcs
netsvcs
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
IPRIP
```

FIGURE 19 EXPECTED OUTPUT FROM STRINGS (2) LAB 03-02

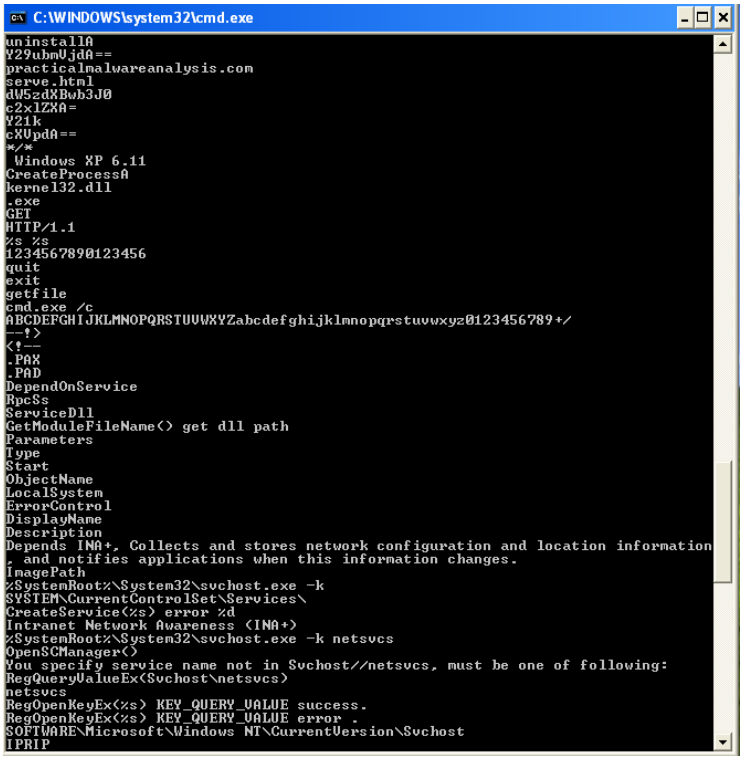


FIGURE 20 PROJECT OUTPUT FROM STRINGS (2) LAB 03-02

After properly installing the malicious DLL, further analysis through a comparison of the registry before and after running the malware provides additional information about the malware. Figure 21 shows a sanitized version of the comparison as presented by the RegShot software which describes keys and values which have been added. The project was able to produce the same results

as shown in Figure 22. Additional changes are normal as the expected output only shows relevant lines.

```
---------------------------------
Keys added
---------------------------------
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP
---------------------------------
Values added
---------------------------------
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Parameters\ServiceDll:
     "z:\Lab03-02.dll"
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\ImagePath:
     "%SystemRoot%\System32\svchost.exe -k netsvcs"
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\DisplayName:
     "Intranet Network Awareness (INA+)"
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Description:
     "Depends INA+, Collects and stores network configuration and location
information, and notifies applications when this information changes."
```

FIGURE 21 EXPECTED OUTPUT FROM REGSHOT LAB 03-02

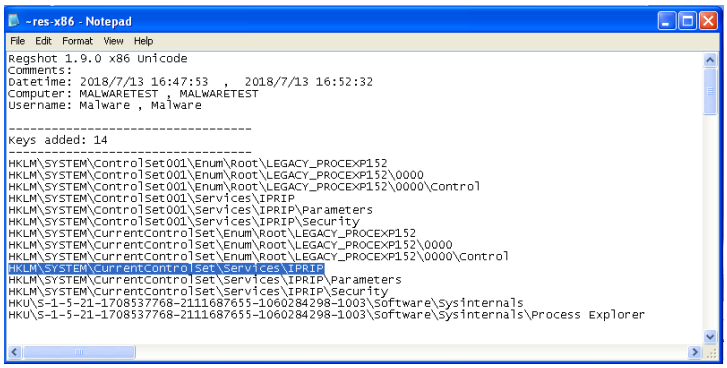


FIGURE 22 PROJECT OUTPUT FROM REGSHOT (A) LAB 03-02

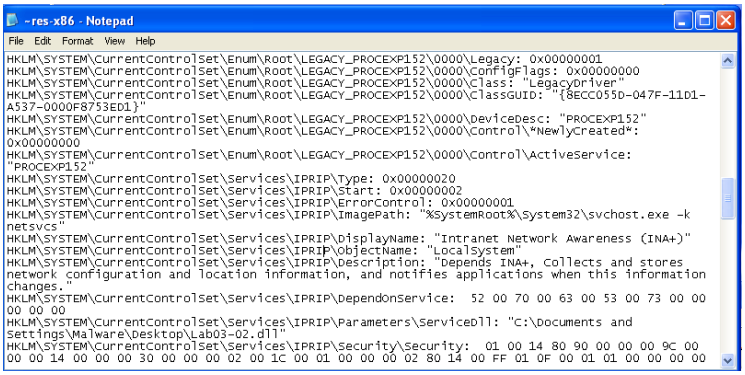


FIGURE 23 PROJECT OUTPUT FROM REGSHOT (B) LAB 03-02

Previous analysis showed that this particular artifact was installed as a part of the IPRIP service, due to registry locations and installation functions, to run the malware, the analyst must run the IPRIP services. Once the service has been started the analyst can confirm that the malicious element is in use through Process Explorer. Searching for the name of the malicious artifact results in selecting the associated process, in this case an orphaned instance of svchost.exe. The expected

output shows that INA is operating and that the malicious DLL has been loaded. Both of these results were duplicated in the project's virtual environment.

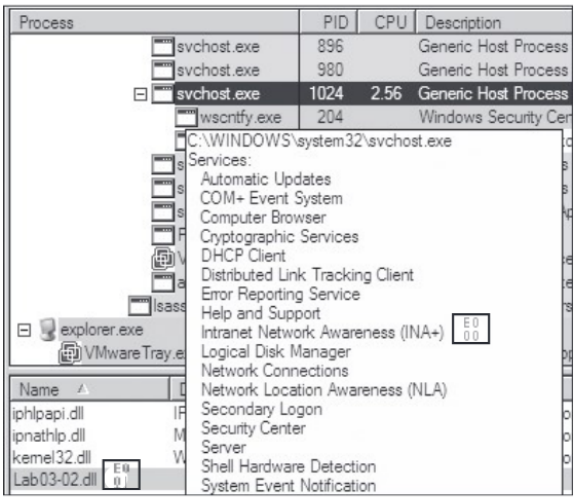


FIGURE 24 EXPECTED OUTPUT FROM PROCESS EXPLORER LAB 03-02

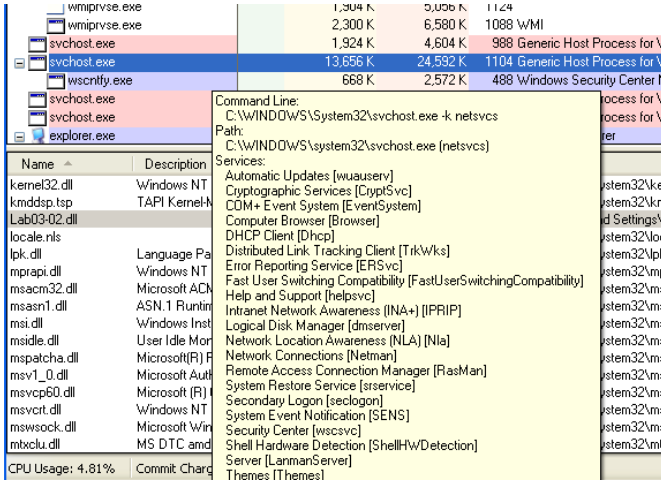


FIGURE 25 PROJECT OUTPUT FROM PROCESS EXPLORER LAB 03-02

Figure 26 shows the expected result of the malware utilizing port 80 (HTTP) and preforms a GET request. Using iNetSim, the project's network simulation was able to gather the proof that HTTP was being used and that a GET request was sent.

```
c:\>nc -l -p 80
GET /serve.html HTTP/1.1
Accept: */*
User-Agent: MalwareAnalysis2 Windows XP 6.11
Host: practicalmalwareanalysis.com
```

FIGURE 26 EXPECTED OUTPUT FROM NETCAT LAB 03-02

```
2018-07-13 15:13:01  HTTP connection, method: GET, URL: http://practicalmalw
areanalysis.com/serve.html, file name: data/http/fakefiles/sample.html
```

FIGURE 27 PROJECT OUTPUT FROM INETSIM LAB 03-02

3. Lab 03-03

The same process as above was conducted to ensure that analysis yields expected results. After running the malicious executable and observing activity in Process Explorer, the analyst can see the Lab03-03.exe process appear, create a child process, and then disappear, leaving the child process (svchost.exe) running. The expected output of a deeper analysis of the orphaned process shows relevant strings including the name of a log file, and notable strings which are typical of keyloggers. Replicating this process in the project's virtual environment produced identical results.

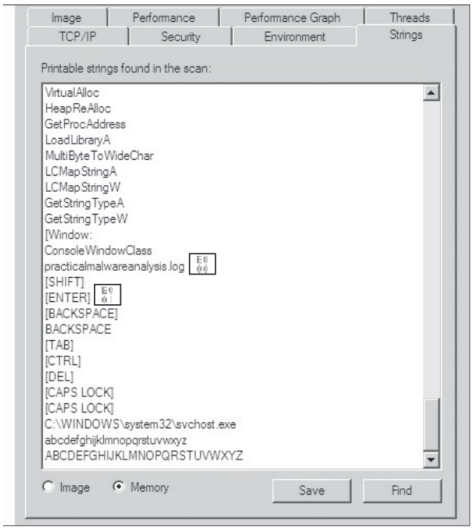


FIGURE 28 EXPECTED OUTPUT FROM PROCESS EXPLORER LAB 03-03

To confirm the suspicion of whether the malware is operating as a key logger further testing can be conducted. Opening a program such as Notepad while having Process Monitor running can instigate the logging and typing a few strokes should result in enough processes to obtain proof of

whether a log file has been created from the strokes the analysist typed. Filtering results based on the PID found in the previous test shows the CreateFile operation being used to create a logfile and a WriteFile operation being used at ever instance of a key stroke to the same file location. Running the same test in the project's environment yielded the same type of results.

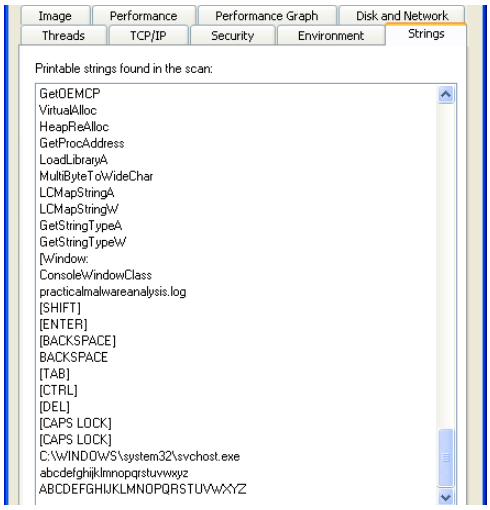


FIGURE 29 PROJECT OUTPUT FROM PROCESS EXPLORER LAB 03-03

Process Name	PID	Operation	Path
svchost.exe	388	CreateFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	QueryStandardInformationFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	WriteFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	WriteFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	WriteFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	WriteFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	WriteFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	CloseFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	CreateFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	QueryStandardInformationFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	WriteFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	CloseFile	C:\WINDOWS\practicalmalwareanalysis.log
svchost.exe	388	CreateFile	C:\WINDOWS\practicalmalwareanalysis.log

FIGURE 30 EXPECTED OUTPUT FROM PROCESS MONITOR LAB 03-03

12:35:...	svchost.exe	2696	RegCloseKey	HKCU	SUCCES
12:35:...	svchost.exe	2696	RegOpenKey	HKLM\SOFTWARE\Microsoft\CTF\	SUCCES
12:35:...	svchost.exe	2696	RegQueryValue	HKLM\SOFTWARE\Microsoft\CTF\EnableAnchorContext	NAME N
12:35:...	svchost.exe	2696	RegCloseKey	HKLM\SOFTWARE\Microsoft\CTF	SUCCES
12:35:...	svchost.exe	2696	CreateFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	QueryStandardI...	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	WriteFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	WriteFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	WriteFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	WriteFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	CloseFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	CreateFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	QueryStandardI...	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES
12:35:...	svchost.exe	2696	WriteFile	C:\Documents and Settings\Malware\Desktop\practicalmalwareanalysis.log	SUCCES

FIGURE 31 PROJECT OUTPUT FROM PROCESS MONITOR 03-03

4. Lab 03-04

The same process as above was conducted to ensure that analysis yields expected results. Examination of Lab 03-04 begins with a review of relevant imports and strings from strings.exe. Notable imports include networking capabilities as well as access and manipulation of registry locations and services. These imports and strings were similarly found during an analysis occurring within the project's virtual environment.

```
SOFTWARE\Microsoft \XPS
\kernel32.dll
 HTTP/1.0
GET
NOTHING
DOWNLOAD
UPLOAD
SLEEP
cmd.exe
 >> NUL
/c del
http://www.practicalmalwareanalysis.com
```

```
NT AUTHORITY\LocalService
 Manager Service
.exe
%SYSTEMROOT%\system32\
k:%s h:%s p:%s per:%s
-cc
-re
-in
```

FIGURE 32 EXPECTED OUTPUT FROM STRINGS LAB 03-04

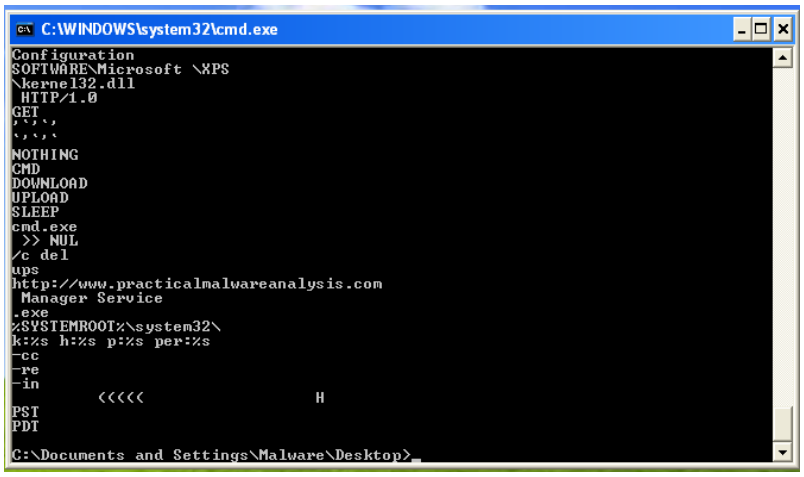


FIGURE 33 PROJECT OUTPUT FROM STRINGS LAB 03-04

Running this malware while observing it in Process Explorer shows the process starting, stopping and then results in the deletion of the file. To further explore the origin of this behavior the analyst will use a filter in Process Monitor based on the name of the process to find relevant operations. The expected output is an operation which calls itself "Process Create" and when reviewing the details of the operation, displays the command for deleting the originating file. Replicating the process in the project's environment yields duplicate results.

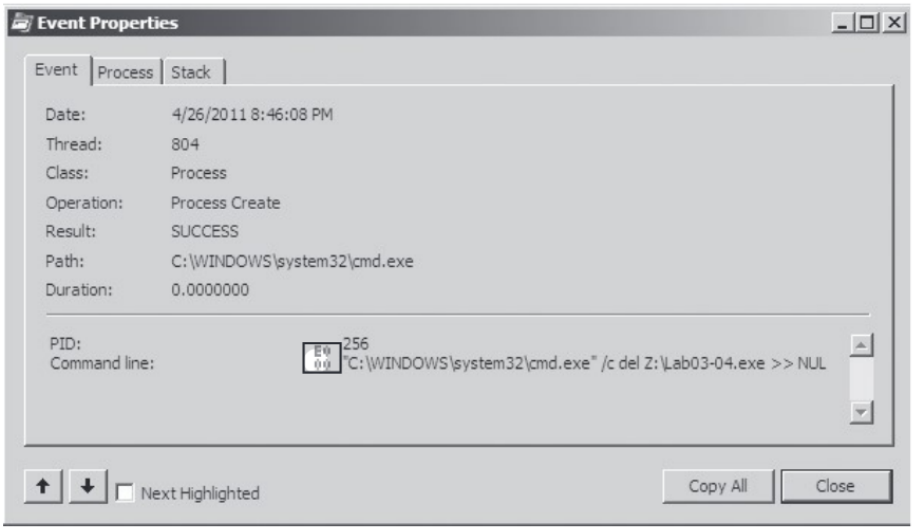


FIGURE 34 EXPECTED OUTPUT FROM PROCESS MONITOR LAB 03-04

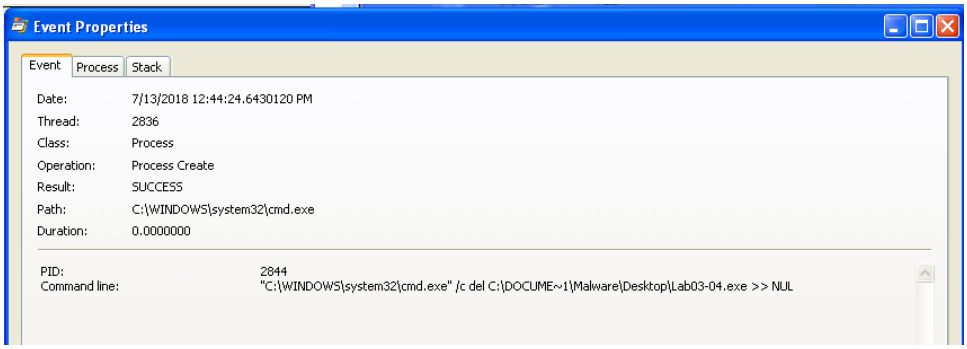


FIGURE 35 PROJECT OUTPUT FROM PROCESS MONITOR LAB 03-04

CONCLUSION

More personal, financial, and sensitive information is being stored on computers and the benefits to obtaining these private files has become more enticing to hackers and malicious entities. These realities translate into organizations having a vested interest in devoting more time and attention towards producing more robust solutions in security. Having the infrastructure and processes in place to conduct a comprehensive analysis could be a considerable benefit as conducting malware analysis in-house can lead to more customizable solutions for that particular network.

As organizations begin to seek out more security experts to assist in maintaining compliance, the need to minimize the gap between desired workforce and available talent becomes more obvious. Part of the reason there is a disparity between economic need for security professionals and the number of individuals who are trained in security is that there are many avenues of skill development within the software career path, many of which are not heavily publicized, incredibly popular, or even frequently offered as areas of study within universities or vocational institutions. This project helps to provide some basic solutions to both assisting organizations conduct additional security measures and introducing interested individuals to the concepts of malware analysis.

Providing documentation on how to produce a safe environment for malware analysis will allow for individuals to gain a comprehensive understanding of several fundamental concepts including, the importance of process (i.e. SOPs), the importance of technological security controls (i.e. firewalls), basic network configuration, and the importance of network segregation. The project is designed to communicate these ideals, and the environment is configured to have layers of security attached to it.

One of the main pitfalls of this project is the fact that there is a heavy burden placed on analysts who are subject to human error. While this is a concern, the SOPs which have been developed, if followed, should allow for a fairly straightforward utilization of the constructed tools without forcing too much critical thinking or placing too many expectations on the analyst.

FUTURE WORK

The future of malicious software is always advancing. While Ransomware was the most popular attack, and remains a threat to organizations to this day, the popularity of crypto mining is increasing steadily. Crypto mining is the process of hijacking processing power of a computer and using it to mine for crypto currency. While the act of crypto mining in and of itself is neither illegal or an undesirable pursuit, the act of using another individual's electricity and processing power is equivalent to stealing, especially considering the possibility of the crypto mining process taking up such a large percentage of a computer's resources that it is no longer able to conduct critical, or even non-critical, functions [6].

The project environment developed here provides a base in which analysis of crypto mining could be analyzed, however, there are features which are missing. Mainly, the SOPs which were developed, as well as the tutorial tips for conducting the analysis, do not address reviewing malware to look for signs or indications that the artifact is intending to crypto mine. This would be a valuable improvement as the increase of crypto mining attacks progresses.

Adding additional tools, especially those which map networks and the location of artifacts within those networks, would be especially helpful. Some crypto mining artifacts, similar to other more traditional malware, has the capacity to spread throughout a network. Utilizing network monitoring tools as a part of the analysis process and including it as a part of both the SOPs and tutorial would be a helpful addition.

Another issue, which was a prominent discovery during the testing process, was the reliance on human discretion to implement many of the security controls. While security professionals may have the best of intentions in conducting processes by the books and make minimal mistakes when handling their work, the fact is that humans make errors. This requires

even more human effort in the form of a supervisory role to ensure that processes are being conducted in the fashion in which the processes were designed. Additional technical features which can operate as the security control for some of the previous goals would be a welcome tool.

Developing a system which would simultaneously house the VMs, allow for the construction of multiple internal networks, and control for special circumstances, such as only allowing one network adapter to be in use at a time, would decrease the security burden on human analysts and result in a more efficient process. Another feature of this extended system would include a control to limit all network configuration settings, unless the administrator/owner of the system gave approval. That process could be added to the SOP to ensure that users are not making updates to the settings unless they are certain to be necessary and/or an improvement.

Additionally, there are ethical, economic, and environmental considerations that organizations must take into account in conjunction with implementing any cyber security measures. While not addressed within the scope of this project, adding insight into these considerations throughout the tutorials would provide learners of malware analysis with a more nuanced understanding of the area. In, “Doctrine for Cyber Security,” the authors discuss other considerations such as human engagement with technology (considered to be the primary cause of most malware coming into access with organizational data and devices) and legal and ethical considerations of both physical and digital assets [7]. This topic could be expanded on further and included in future iterations of this project.

One of the most important theories relating to the development and creation cycle of cyber threats is the concept of the cyber kill chain. Cyber kill chain discusses the process from concept to completion of Advanced Persistent Threats (APT), or threats where the perpetrator will continue to attack through different vectors until they have successfully completed their objective. The

cyber kill chain demonstrates the background work and other resources that hackers and bad actors must obtain in order to move further through the cyber kill chain [8]. In "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," the authors attempt to define the typical structure of components reviewed by Malware Analysists in addition to the chain of activity conducted by hackers. "Intelligence-Driven," describes intrusion indicators across three different types: (1) atomic, which includes the smallest unseparated information possible like IP addresses or email addresses; (2) computed, which might contain information like hash values or regular expressions; and (3) behavioral, which contains insights which are qualified from a combination of atomic and computed intrusion parameters. Kill chain phases during a computer network attack include:

1) Reconnaissance – acquiring information about potential targets.
2) Weaponization – creating weaponized files that are typically delivered as trojans masked as a legitimate file to targets.
3) Delivery – the transmission of the weaponized file to the victim.
4) Exploitation – either exploiting an already known vulnerability or operating feature.
5) Installation –installing the malware to maintain access to the victim machine.
6) Command and Control – establish a channel connection between victim machine and intruder machine.
7) Action on Objectives – obtain desired resources or tools from victim.

Hackers and bad actors spend time researching how businesses handle cyber security, they look for individuals who may have complaint with their organization, and they have an understanding of the types of accidental mistakes employees might be prone to fall victim to which could make them susceptible to downloading or acquiring malware.

This understanding of human behavior is a huge benefit to hackers, and in order to maintain proper defenses Malware and Security Analysists should be well-versed in the human components as well. The benefits to understanding the typical pitfalls employees often encounter can allow for better anti-virus and other automating tools to be developed which stops cyber kill chain at the attempted delivery stage, instead of needing to address it at the exploitation stage, or worse [9]. Analysts should also make special note of the fact that challenges can occur in the detection stages throughout an antivirus system, and as more systems and services are utilizing cloud-based infrastructure, these challenges can bring additional complications in malware detection and analysis [10]. Addressing how possessing a competency in these issues can add depth to the understanding of not only the process of analyzing malware, but also to developing comprehensive solutions that look to address solving the root of the attack, as opposed to simply patching a vulnerability. These are topics that are not covered within the scope of the project but could add valuable insight for future learners.

APPENDIX A (FIREWALL AND VIRTUAL NETWORK TUTORIAL)

Appendix A contains the list of all tutorials discussing the creation of the virtual environment. This document is long and has been split into the following subsections:

1. Introduction
2. Create Your First Virtual Machine in VirtualBox
3. Create Ubuntu Analysis Machine
4. Create Firewall
5. Firewall Configuration
6. Internal Network Configuration
7. Final pfSense Settings
8. Create Shared Folder
9. Final Configuration and Transfer Checklist

(Tutorials with tips on basic malware analysis use is in Appendix B)

1. INTRODUCTION: VIRTUALBOX DOWNLOAD AND INSTALLATION

WELCOME

This tutorial will be using VirtualBox to create and set up Virtual Machines, manage network segregation, and create a functional system for downloading, distributing and isolating malware so that it can be analyzed in a "secure" environment. While this tutorial attempts to provide you with sufficient layers of protection, anytime you interact with malware there is a possibility of escape from your environment and ultimately destruction or exposure of your host system or data. That being said, please be sure to exercise caution, double-check all of your steps, and if possible, use a computer that you have no problems reimaging should the worst occur.

While this tutorial attempts to be easy to understand and use, even with limited background knowledge of networks or malware tools, there may be missing explanations. If you encounter an unknown term, a quick google search or a Wikipedia article should be sufficient to fill in the gaps.

The host machine used for this tutorial utilizes a Linux operating system (Ubuntu 64bit), and the instructions written here are for that machine. If you are using a Windows machine, you should still be able to follow along, as most of the work happens within the virtual environment, which exist in the VirtualBox program.

Having relied on a variety of tutorials, books, and other resources to configure this network, you may encounter situations where these instructions may not work. The goal is to provide enough information about ***why*** we are doing each step so that if you encounter issues, you will at least know what sort of questions and research you need to do in order to find a solution.

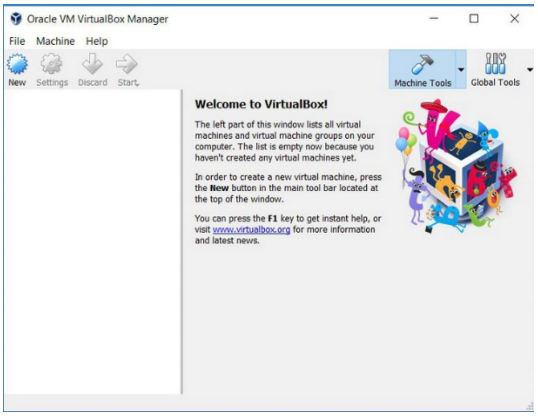
INSTALLATION

Note: Before getting started make sure that your operating system is up to date. Run apt-get-update from Linux terminal or if using Windows, check for updates.

Regardless of your operating system, you can go to https://www.virtualbox.org/ and click on the big box commanding you to download its latest version (see image below).

This will take you to a screen allowing you to select which operating system is on your host machine. The Windows link will automatically download an executable which will run you through the installation process. For Linux, you will have the option to download based on the version of Linux you are running (as well as processor) or you can download a version for all distributions as well. A list of Linux commands can also be found on this page to help with the installation of VirtualBox.

Once installed, you will be able to open VirtualBox which will have a similar screen as below:



If you're in, great! Let's move on to the next tutorial to get started with creating your first VM.

TROUBLESHOOTING

Troubleshooting: If you do encounter any issues during installation here are a couple of things to double check before banging your head against the wall.

- Do you have another version of VirtualBox already installed on your computer?
- Did you have too many other processes running during your VirtualBox installation?
- Is your computer up to date?
- Do you have any firewall or other constraints that are blocking the download of VirtualBox?

If the answer to any of the above questions is, "yes," uninstall all VirtualBox software from your machine, fix the issue, restart your computer, and attempt the installation again. Note: All of the Virtual Optical Disks, and Virtual Hard Disks take up space on your personal computer/host computer. Make sure before you start this project that you have the hardware capability and enough memory space to properly conduct the installation and configuration.

2. Create Your First Virtual Machine in VirtualBox

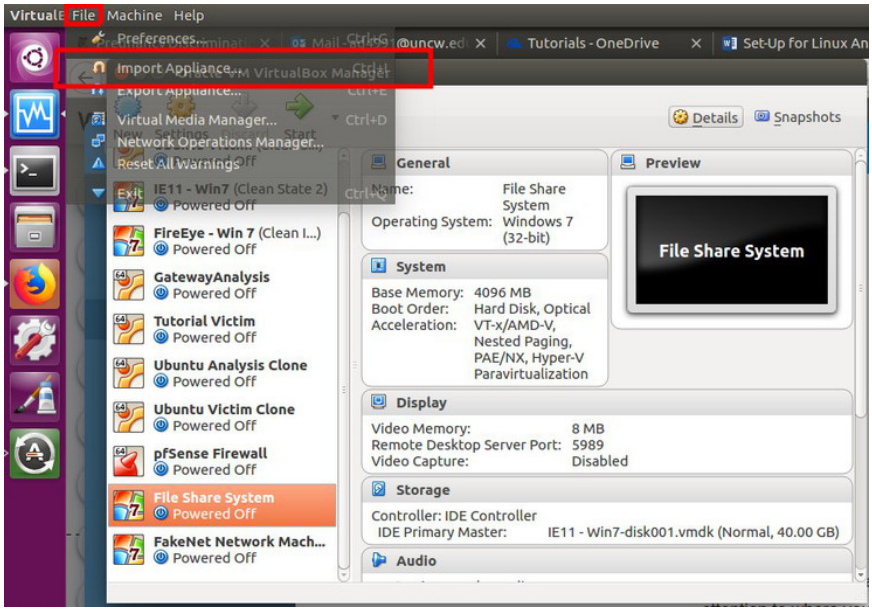
Background & Resources

Microsoft offers Windows 7 Virtual Machines (VM) configured for a variety of virtual environments. These VMs are authenticated for 30 days of use for developers who are testing web features on past versions of Windows and Internet Explorer/Edge. These VMs can be downloaded by going to the following website: https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/.

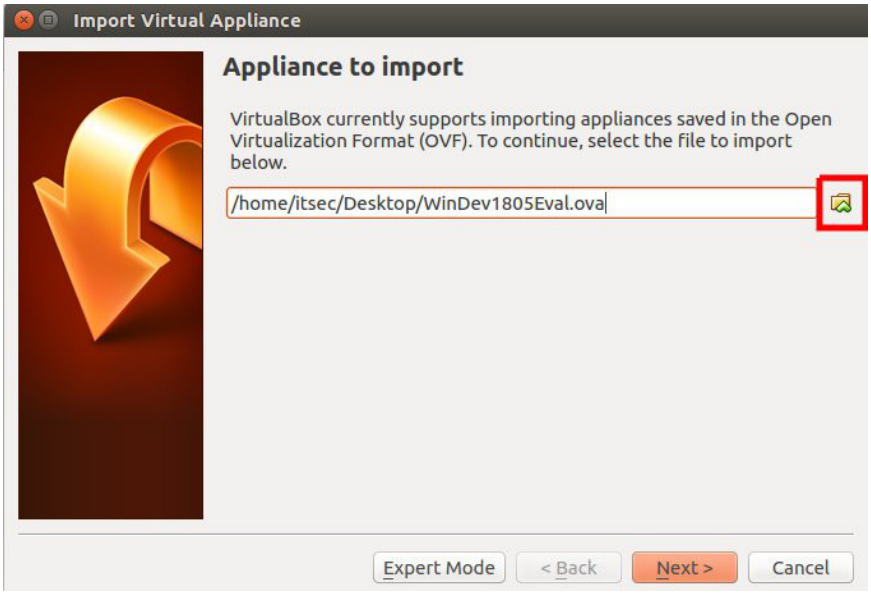
On this site, you will have the options of virtual machine and platform. For virtual machine, select one of the Windows 7 options. It does not matter which version of IE you choose to use (we will be installing another browser later anyway), but I used the most recent version, IE11 on Win7 (x86). For platform, select VirtualBox and click the DOWNLOAD .ZIP button.

VM Creation and Configuration

1. Open up your VirtualBox software (downloaded in the previous step).
2. On the top menu, click "File" and then select "Import Appliance"



3. Click the file icon that pops up on the next screen and navigate to the location where you saved your download



4. Click "Next", check the box marked "Reinitialize the MAC address of all network cards," then click "Import"

   Note: we reinitialize the MAC address for the VMs because we are creating a network which needs to have unique machines with unique identification features for them to be able to communicate with each other. Each device has a unique MAC address and making sure these are different will allow them to be recognized as unique machines within our virtual environment.

5. Once the previous step is complete, RIGHT click on the newly created VM in VirtualBox and select the "clone" option

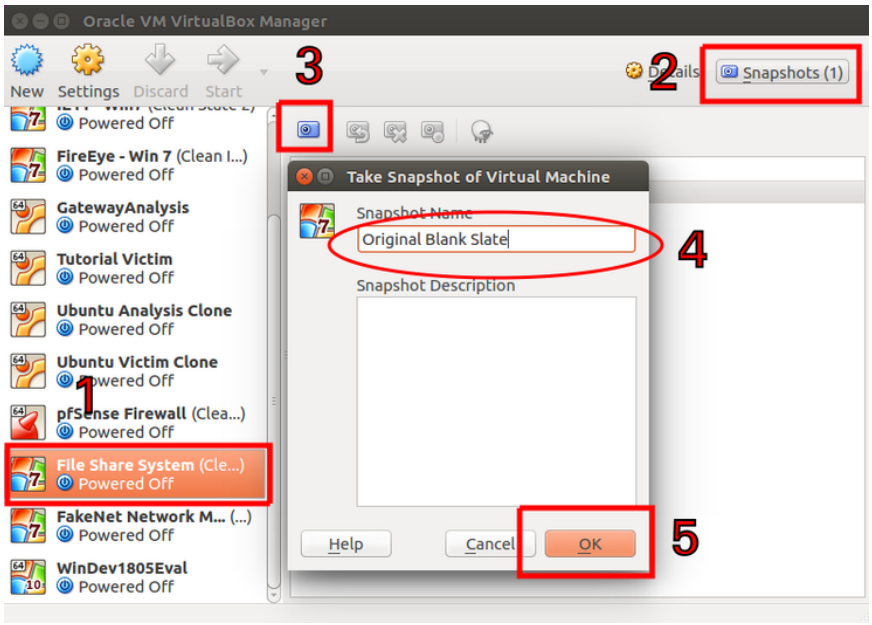
   Note: we want multiple computers on our network as each of them will be designated to handle a unique task, which will allow us to segregate.

6. Again, be sure to check the "Reinitialize MAC address of network cards," option, click "Next" then click "Import"

7. Create distinct names for each of the machines. I chose the names File Share System and FakeNet Network Machine. You should now have 2 VMs creating in your environment.

8. Before moving on, create a snapshot of both VMs. (1) Select the File Share System and (2) click the Snapshot icon, then (3) click the create snapshot icon. (4) Name it something like "Original Blank State" so you know this is after you downloaded it and have changed nothing. (5) Finally, click OK. **Note: this is important in case you accidentally download something funky before securing your network or machines.**

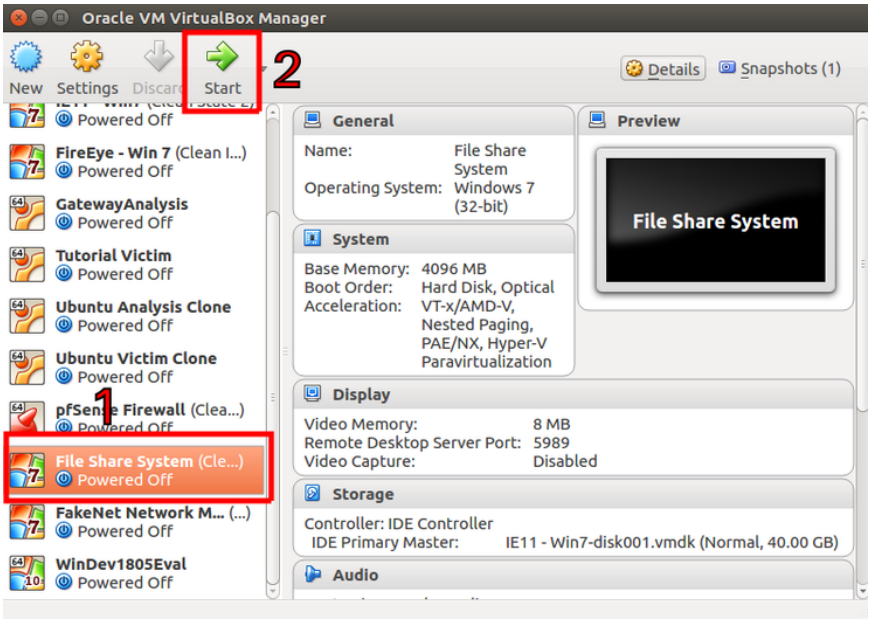
   Repeat the above process for the second VM (FakeNet Network Machine).



9. After that has completed, open the File Share System you created, by highlighting the VM and clicking the green arrow that says "Start" at the top of VirtualBox.

   Note: the default settings upon installation of this machine should have your VM connecting to the internet through your host machine, we'll change these settings later to connect to an internal network and a firewall.

   Additionally, we are using the File Share System NOT the FakeNet Network Machine to download all of our malware analysis tools from the internet. Doing this allows the FakeNet Network Machine to have never accessed the internet, as we can later use a shared folder on an internal network to transfer the files from the File Share System to the FakeNet Network Machine.



10. **Optional**: Once your VM has finished booting, you may want to install a different browser (in this tutorial, Firefox is used). The reason for doing this is not all websites will allow you to download from IE11due to the fact the TLS handshake in IE11 for Windows 7 has known bugs. Alternatively, you could attempt to fix IE11 with a patch, but a different browser tends to work better.

List of Malware Analysis Software to Download on the File Share System

Here is a list of other URLs and the name of the software that you should download on your File Share System. I will go into a more depth in a later tutorial where I discuss what each of these programs do and how to use it during malware analysis.

Note: the operating system downloaded for VirtualBox from Windows is a 32-bit OS, NOT 64-bit, so be sure to download the appropriate OS version if there are multiple options. Many of the programs work the same regardless of the OS, but there are some that do not. Also, if a link is broken, you may have to search for another location. All of this software is freeware and should be available somewhere online, although you are urged to exercise caution and go to reputable download sites like CNET or sourceforge.

Basic Analysis Tools

- WinMD5 - http://winmd5.com/
- PEid - http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml
- Dependency Walker (also used in dynamic analysis) - http://dependencywalker.com/
- PEview - https://www.aldeid.com/wiki/PEView
- Resource Hacker - http://angusj.com/resourcehacker/

Dynamic Analysis Tools

- Process Monitor (procmon) - https://docs.microsoft.com/en-us/sysinternals/downloads/procmon
- Process Explorer - https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer
- Regshot - https://sourceforge.net/projects/regshot/
- ApateDNS - https://www.fireeye.com/services/freeware/apatedns.html
- Netcat - https://sourceforge.net/projects/nc110/
- Wireshark - https://www.wireshark.org/download.html

Troubleshooting

Malware Analysis Tools: If you encounter any links that do not direct you to a viable website with a functioning download link, you should be able to conduct a search to find the software. All of this is freeware, offered to the public without cost, and most of the listed software can be found hosted on different websites.

Windows VM: If you are having difficulties importing the Windows appliance, try closing VirtualBox, restarting your computer, and re-importing the Windows VM while no other programs are running on your host machine. If that fails again, go back to the Microsoft Developer's page (https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/) to download the Windows 7 VM and attempt a fresh download and try to import the new copy into VirtualBox.

VM Internet Connection Issues: If your File Share System fails to connect to the internet, check the following:

- Make sure your host machine is connected to the internet and that you can access all of the links on your host machine. If not, fix your network connectivity issues on the host machine before continuing.
- Look at the network settings in VirtualBox for the File Share System VM. Select the VM and click on "Settings" then choose the "Network" tab on the left.
    - Make sure the "Enabled Network Adapter" option is selected
    - Make sure the "Attached to: " selection is set to "NAT." If changes were made, click the "OK" button to save the setting updates.

      Note: NAT stands for Network Address Translation, which means that the VM is adopting the host machine's IP address and assigning it to itself. This tutorial was created using a University's network resources, which required the use of NAT during later configuration steps when working with the firewall, as the University had network security controls in place to restrict additional DHCP IP addresses from being assigned outside of the University's control. NAT was the solution to this problem and can sometimes assist in troubleshooting other connectivity circumstances.

If you have worked your way through these issues and are still encountering problems, you may have to do some internet searching. Reference the software and operating system you are using in your search, i.e. connectivity issues with VirtualBox VM on Windows 7. This will tend to yield the most succinct descriptions for your issue

3. Create Ubuntu Analysis Machine

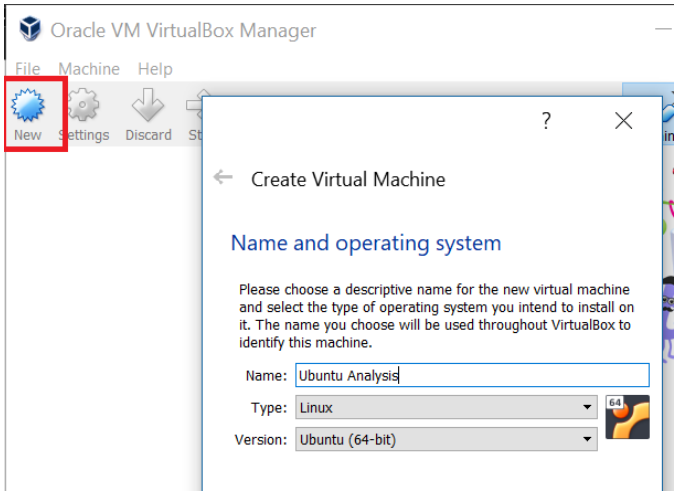
Background & Resources

During this section of the tutorial you will be creating and configuring the Ubuntu Analysis Machine. This VM's purpose is to be used as a monitoring tool, and a network simulation tool. Through the use of software including iNetSim, which monitors activity on all ports within your network, and Burp, software used to simulate CA certificates, required for accessing HTTPS sites, you will be able to properly simulate network activity to fool malware into behaving as usual, allowing you to review the activity happening.

To obtain a copy of the Ubuntu OS, which you will need before beginning this tutorial visit http://cdimage.ubuntu.com/netboot/16.04/ and select the option for amd/intel 64-bit processors. This tutorial relies heavily upon the work of Christophe Tafani-Dereeper. You can find his blog post here: https://blog.christophetd.fr/malware-analysis-lab-with-virtualbox-inetsim-and-burp/.

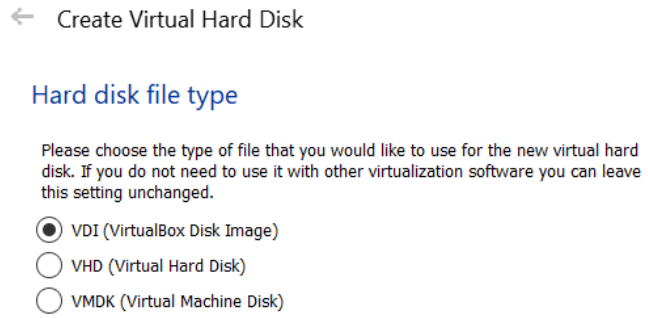
Set-up Ubuntu Analysis Machine

1. Click "New"
    a. Name VirtualBox Machine – Ubuntu Analysis Machine
    b. Select OS – Linux
    c. Select Version – Ubuntu 64-bit

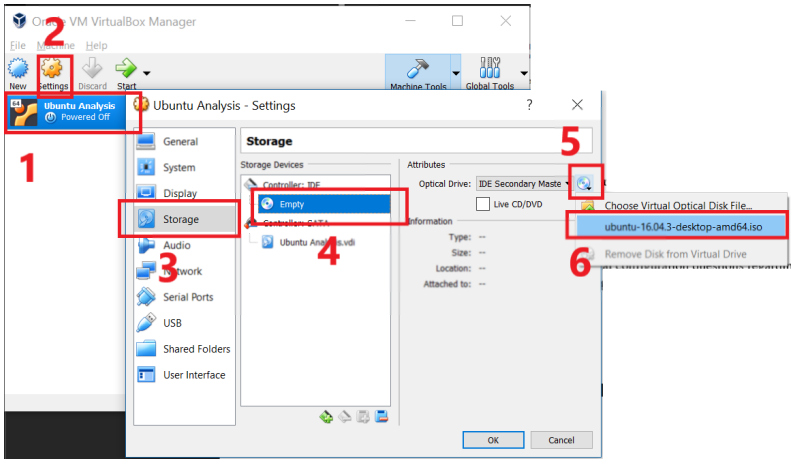


    d. Click next

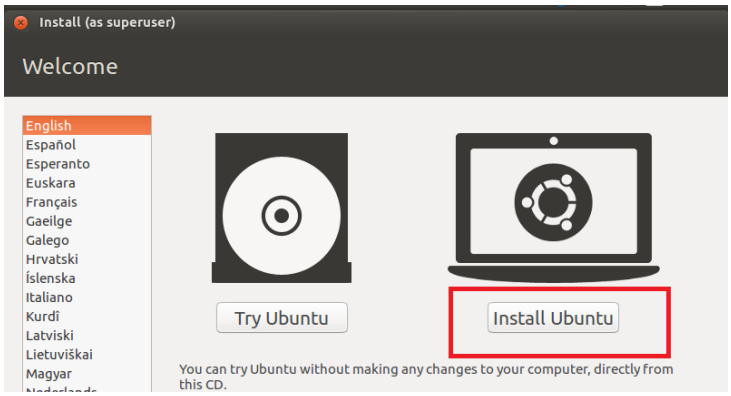
2. Choose Memory (4096MB) and click next
3. Create Virtual Hard Disk and click next
4. Select VDI option



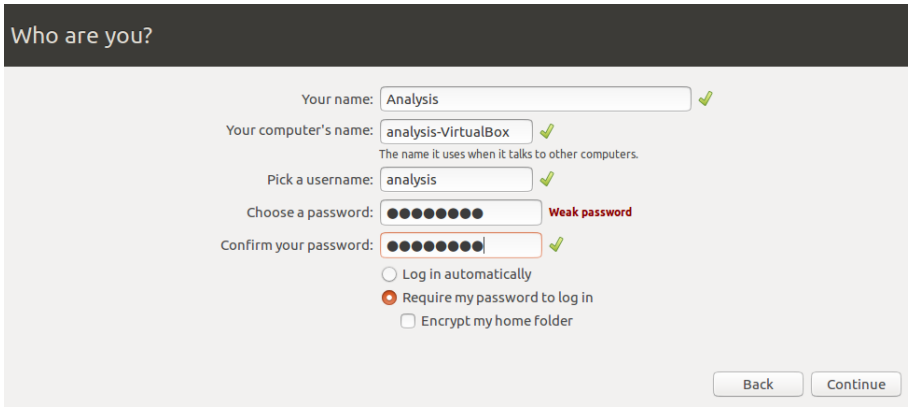
5. Dynamically allocated and click next
6. Select hard disk size (12G) click create
7. Highlight the machine and click settings in the main VirtualBox portal
8. Click the Storage tab on the left menu bar
9. Select the word Empty under the Controller: IDE section
10. On the right menu click on the disk icon and select the OS



11. Click OK to save
12. Highlight the Ubuntu Analysis machine in Virtual Box and press the green start arrow
13. Choose to install the full version of ubuntu



e. Select the option to install updates while installing, this will save you time later
f. Select the option to erase disk and install Ubuntu
g. The installation setup will ask you several configuration questions regarding your preferences including time zone, language, and username and password



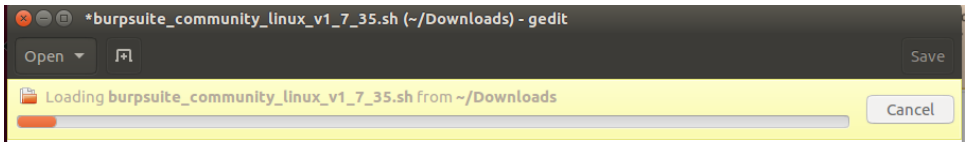
h. Work through these questions and complete the installation

14. After installation has completed you will be asked to restart the virtual machine. Restart and press enter if prompted.
15. First thing you'll do is update your computer. Open the Terminal (click top left icon and start typing terminal in the search box to find the application)
16. Open Terminal and type: sudo apt-get update
17. Type: sudo apt-get install virtualbox
18. Type: sudo su
    i. If prompted provide password
19. Type: echo "deb http://www.inetsim.org/debian/ binary/" > /etc/apt/sources.list.d/inetsim.list
20. Type: wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc | apt-key add -
    j. Note: Don't forget the dash at the end!!!
21. Type: apt update
22. Type: apt install inetsim
    k. Note: if any above instructions fail, look at the official inetsim installation instructions here: http://www.inetsim.org/packages.html
23. You've now completed your basic set up

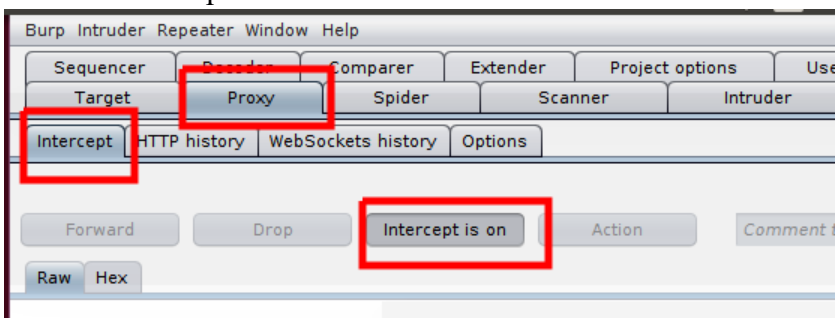
24. Power down your virtual machine by clicking the power icon and selecting shut down

INSTALL AND CONFIGURE BURP

1. Power on your analysis machine
2. Open mozilla firefox
3. Navigate to https://portswigger.net/burp/communitydownload
4. Download the 64-bit Linux version
5. Go to the downloads file in your VM and double click the .sh file from burpsuite_community

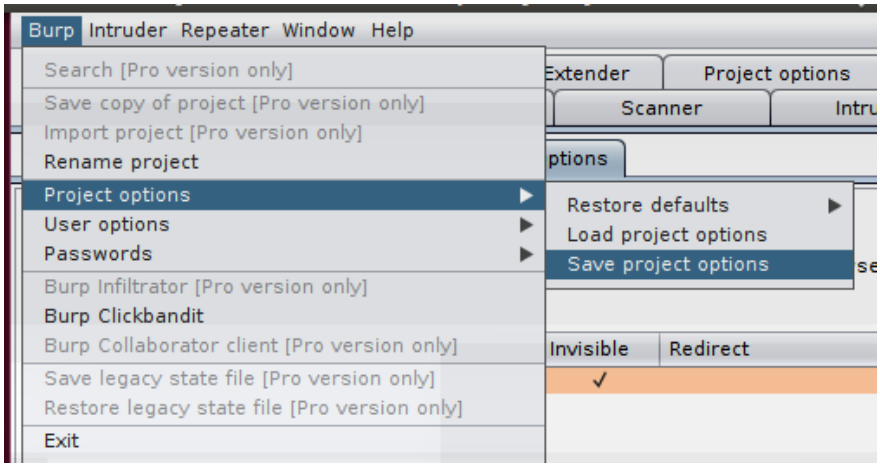


6. Allow it to finish downloading
7. Open terminal and type: cd /BurpSuiteCommunity
8. Type: sudo /home/analysis/BurpSuiteCommunity/BurpSuiteCommunity
9. Select: Temporary project
10. For the initial set up use Burp defaults > click start Burp
11. Navigate to Proxy
    a. Turn off Intercept



    b. Select the options tab
    c. Click on the only listed Proxy Listener and click the edit button
    d. Bind to port 443
    e. Bind to address all interfaces

f. Navigate to Request Handling tab and ensure that the "Support invisible proxying" option is enabled
    i. Click OK
g. Save the settings to use in the future by clicking Burp > Project options > Save project options



Network Configuration

1. In the VirtualBox Manager select the analysis machine and click settings
2. In the left toolbar select Network
3. In the dropdown next to “Attached to:” select internal network
4. Create a name for your new network (I chose virtual-malware-network)
5. Complete this process for each of your virtual machines within the network
6. Start the analysis machine and open Terminal
7. Type: ifconfig
    l. Verify that the first file printed is enp0s.
    m. If it is not, replace the word enp0s3 with the file name listed for all future steps
8. Type: cd /etc/network
9. Type: sudo nano interfaces
    n. Provide password
    o. Note: I am using nano because it is a very simple command line text editor. If you prefer using vim, emacs, or any other editor then you should use that instead.
10. Scroll to the bottom of the document and add the last four lines so that your document looks like:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto enp0s3
iface enp0s3 inet static
 address 10.0.0.1
 netmask 255.255.255.0
```

    p. Then Type: ctrl+o
    q. Make sure that the file name is interfaces

```
File Name to Write: interfaces
^G Get Help      M-D DOS Format   M-A Append       M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend      ^T To Files
```

r. Hit enter

11. Go back to Terminal and type: sudo ifup enp0s3
    s. This puts your updated settings into effect
        i. Note: if this causes an error, open up your interfaces file and look for typos.
12. Keep the analysis machine on, turn on the ubuntu victim, and open terminal
13. Open a new terminal in the analysis machine and type: mkdir analysis
14. Type: mkdir analysis/test-analysis
15. Type: cp /etc/inetsim/inetsim.conf analysis/test-analysis
16. Type: sudo cp –r /var/lib/inetsim analysis/test-analysis/data
17. Type: sudo chmod –R 777 data
18. Type: cd analysis/test-analysis
19. Type: sudo nano inetsim.conf
20. Change the service_bin_address from 10.0.0.1 to 0.0.0.0 and delete the #

```
##########################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address    0.0.0.0
```

21. Open a new terminal window and type: sudo systemctl disable systemd-resolved.service
22. Type: sudo service systemd.resolved stop
23. Back in the configuration document comment the line that says "dns_defualt_ip 10.0.0.1" by deleting the #

```
##########################################
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip          10.0.0.1
```

24. Change #https_bind_port 443 to: https_bind_port 8443

```
##########################################
# https_bind_port
#
# Port number to bind HTTPS service to
#
# Syntax: https_bind_port <port number>
#
# Default: 443
#
https_bind_port         8443
```

25. Save the configuration file
26. In terminal type: sudo inetsim –data data –conf inetsim.conf

27. This should run inetsim
28. Type: ctrl + C to stop running inetsim
29. This should result in a statement telling you the location of the log report

Troubleshooting

If you encounter any difficulty in the installation of Ubuntu onto your VM, double check the following:

- Make sure you have enough space on your host computer for the VM
- Make sure the OS you downloaded from the ubuntu website managed the type of processor you selected within VirtualBox
- If you are still encountering problems, restart the host machine and make sure you are not conducting any other processes during the installation process

If you encounter any challenges with the installation or configuration of iNetSim or Burp, navigate to their installation archives at https://www.inetsim.org/downloads.html and https://portswigger.net/burp/communitydownload respectively.

4. Create Firewall

Background & Resources

This section of the tutorial will be used to provide an added layer of protection between the File Share VM and any websites you attempt to access from the File Share VM. The File Share VM will be used to acquire any malicious artifacts from the internet, or to download any additional software at this point. By adding in a firewall, which will restrict access to the VM from remote servers online, we can limit the potential harm caused as well. The firewall and these rules are not a foolproof way of eliminating all threats, as you will still be able to download malware, but it can limit some unknown threats.

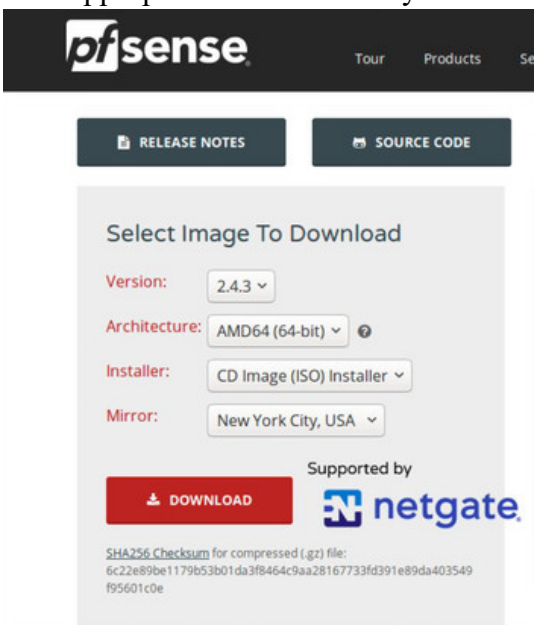
This tutorial steps through the installation and set-up process for establishing a pfSense firewall, which is a free to access tool found at pfsense.org/download.

Initializing VM in VirtualBox for Firewall

1. Download the most recent version of PFsense from https://www.pfsense.org/download/ on your host machine

2. Use the following configuration:
    - Architecture: AMD 64 **Note: The architecture isn't hugely important, but you should remember which option you select as it will affect the configuration process of the VM you will be making later in VirtualBox.**

- Installer: CD Image ISO **Note: the ISO is a readable drive format that your VirtualBox VM can use to install the pfSense software**
- Mirror: Select the appropriate time zone for your location



3. Extract the file onto your computer (Hint: Be sure you know where it is saved. During the configuration process you may want to keep it saved somewhere on your desktop for ease of access)

4. Open VirtualBox and click New and use the following configuration:

    - Name: PFSense Firewall (or whatever you prefer)

    - Type: BSD

    - Version: FreeBSD (64-bit) **Note: if the architecture you selected for download does not use 64-bit, select another appropriate option for your version**

    - This does not need to have as much memory as the virtual computers. Using 512 MB will be plenty, but if you're running low on space you can go down to 256MB

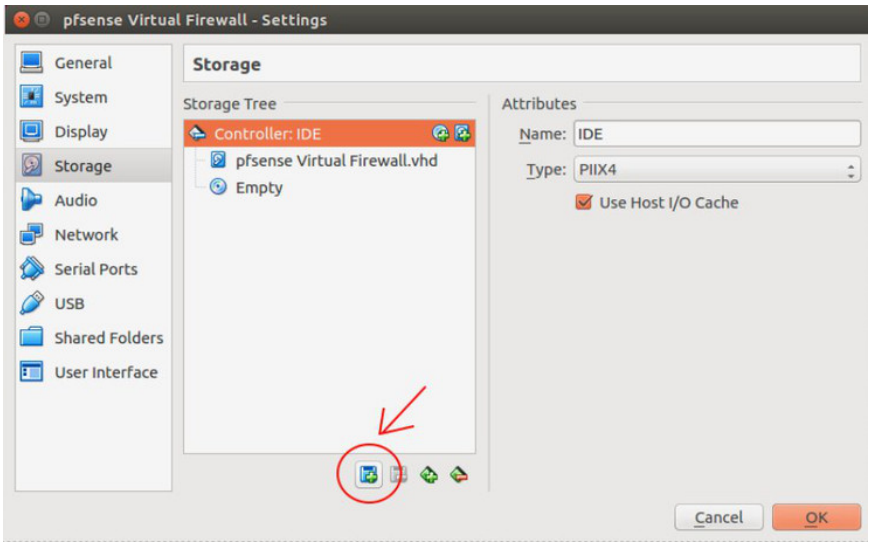
5. Select "Create a virtual hard disk now" and click the "Create" button

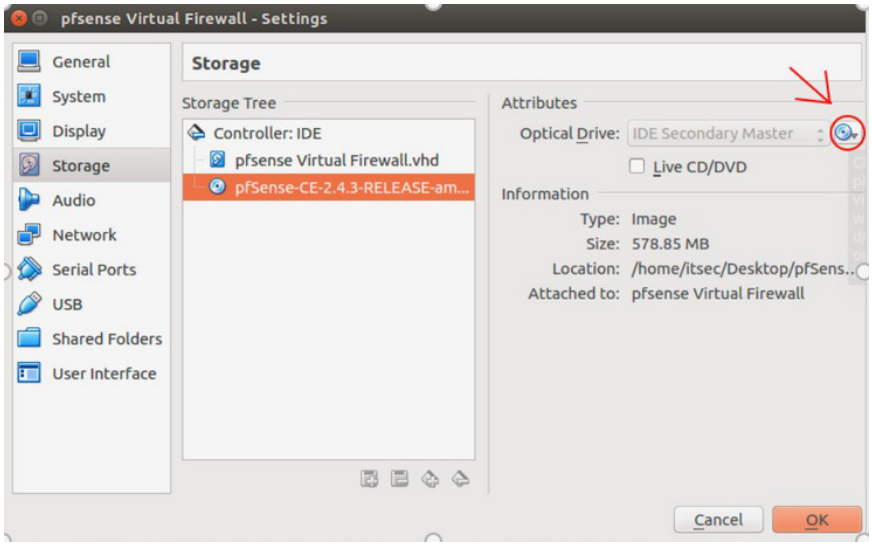
6. Select "VDI (VirtualBox Disk Image)" **Note: remember how we select the CD Image ISO above, the VirtualBox Disk Image is going to allow us to read the ISO file and boot from the pfsense download once it is attached to the VM**

7. Select "Dynamically allocated"

8. Choose File location and Size (I left my file location at the default pathway)--> Again the size does not have to be as large as the virtual machines, but 4 GB is recommended, although 2GB can suffice. Go ahead and click "Create" when you're ready.

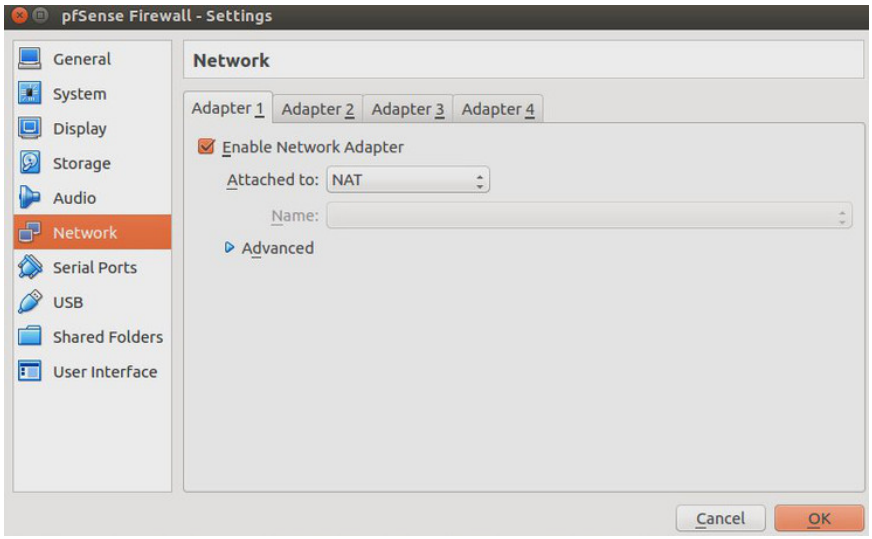
9. Select the newly created VM and click "settings" in VirtualBox
10. Go to the Storage tab on the left panel
11. Make sure "Controller: IDE" is selected
12. Hit the Add Optical Drive" button (see image below)





13. Go to the location where the pfSense ISO is saved and select the ISO file.
14. Select the pfSense Firewall VM in VirtualBox and then click "Settings" in VirtualBox
15. Click the "Network" in the toolbar to the left and ONLY change the following (Note: names of adapters and adapter types may differ on your computer)

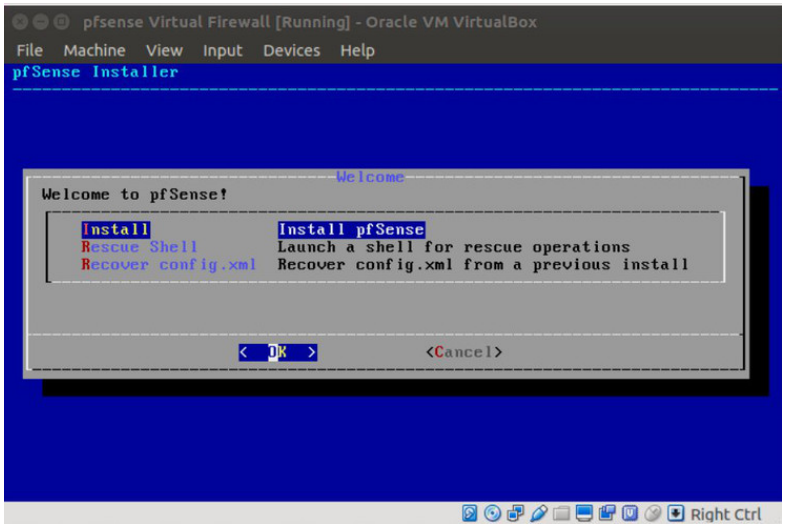
16. Attached to: NAT **Note: We went over how a NAT functions in the previous tutorial if you're interested in learning more about it.**



17. In the same Network setting screen, click the "Adapter 2" tab at the top of the screen
18. Check the Enable box, and update Attached to: Internal Network
19. Note: the default internal network in VirtualBox is intnet, which is what we'll use for now.

pfSense Installation within VM

1. In VirtualBox, select your pfSense firewall and click "Start"
2. Accept any of the user agreement software conditions by pressing enter on your keyboard
3. Click enter to install pfSense

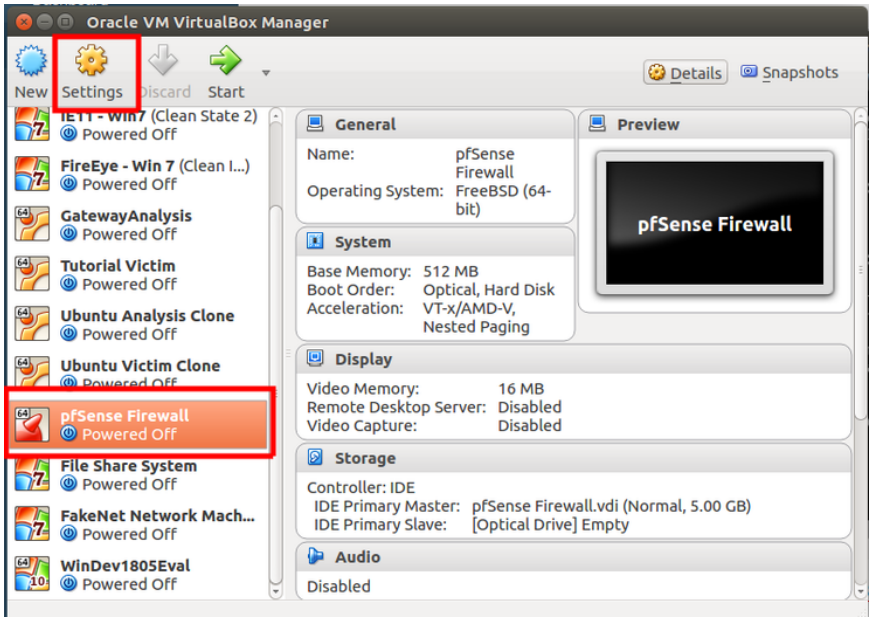


4. Select the default keymap and Auto Guided Disk Setup

5. You do not need to open the shell unless you want to customize any additional configuration settings which is not required for this tutorial

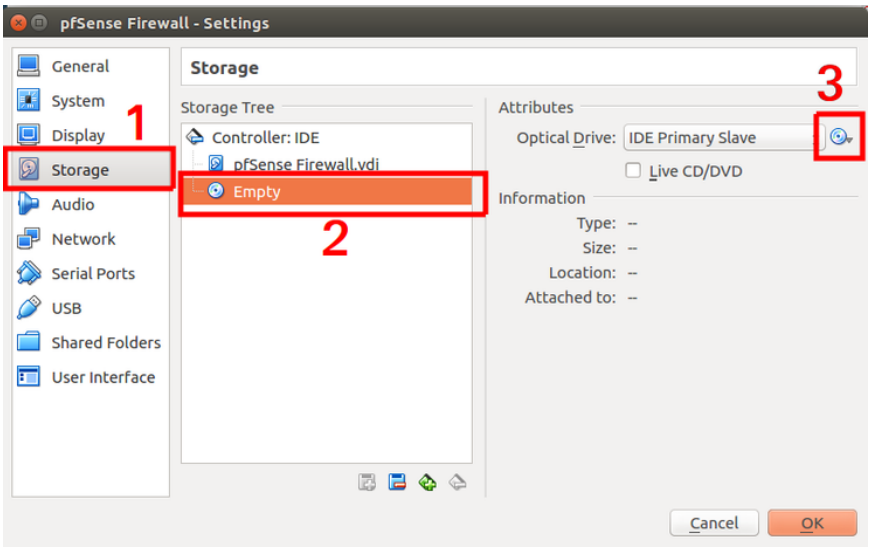
6. After the installation is complete, the VM will automatically reboot. At this point we can shut down the VM by clicking "Machine" and then "ACPI Shutdown"

7. In the VirtualBox main menu select your pfSense VM and click the settings icon



8. Follow the steps below to remove the ISO: (1) Click on the storage tab on the left and (2) select the optical drive under the Storage Tree part of the screen (see below). Finally, (3) click the disk icon next to the Optical Drive selection under Attributes and select "Remove from Drive" **Note: If you do not remove the ISO, the next time you start the**

pfSense VM it will continue to try and install the software on a loop, so ejecting it is an important step



Now that the firewall has been installed we can move on to the next step: configuration.

TROUBLESHOOTING

If you had trouble with the installation of the pfSense software from the website:

- I remember at one point when I clicked on the download button on the pfSense site it took me to a long list of links instead of automatically the download that I requested. If this happens to you, look over the link names, most of the naming conventions include the architecture type in the name, which is what you should use to select
- Remember that once the download is completed the ISO can only be used, or is designed to be used in a VM environment, so opening it on your host machine won't work
- If the download fails, I would attempt to restart my computer, and if it continues to fail, it may be a temporary issue with the pfSense site. I recommend waiting a few hours before trying again

If you had trouble configuring the pfSense VM or installing the software:

- Make sure that the VM you selected when you went to create the new VM has the compatibility to run on the architecture of the pfSense software you downloaded, if they're not the same it won't operate
- Make sure that the Optical Drive was properly added, and that the ISO was mounted appropriately
- Make sure that once the download was completed that you shut down the VM, and then remove the ISO from the optical drive
- If you're still encountering issues, try downloading a fresh copy of the pfSense software and try the install on the VM again with the fresh download

5. Firewall Configuration

Background

Now we are going to set up the configuration of the firewall. Some of this we are going to do within the VM and some we are going to do in the VM that we will configure to be on the same internal network (the "File Share System VM created in an earlier tutorial). (Another option is to do all of the configuration settings within the pfSense VM but should only be attempted by those who have experience with pfSense).

Assign Network Adapters and IP Addresses

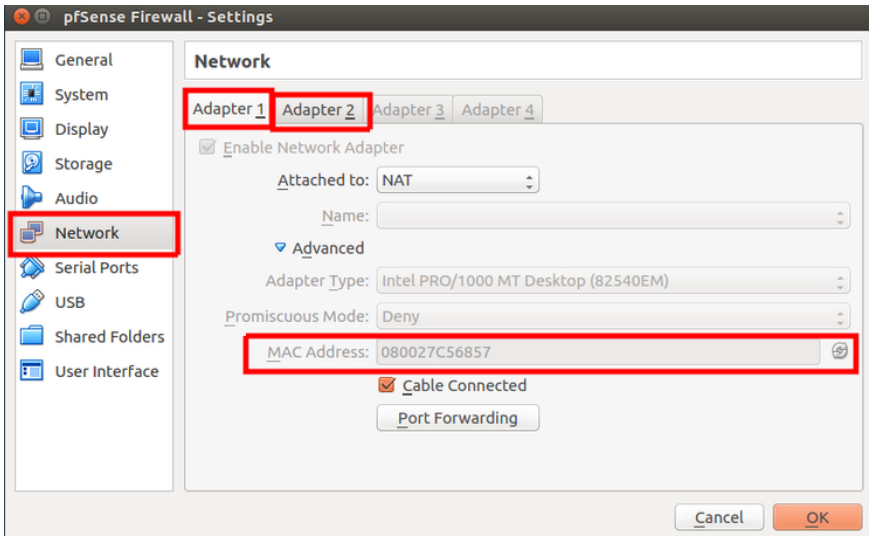
1. Check the MAC addresses assigned to each of the network adapters in the pfSense VM.

   Note: The MAC address is a unique ID that most electronic devices have. The MAC address of the adapters will dictate how the pfSense interfaces will be assigned

2. In the VirtualBox main menu, highlight the pfSense VM and select "Settings"

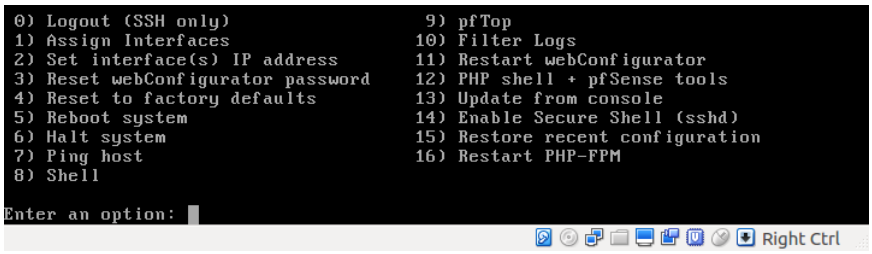
3. Click on the "Network" tab on the left

4. Select the "Advanced" option and review/write down the MAC address for both Adapter 1 (NAT - and WAN interface) and Adapter 2 (internal network - and LAN interface)



5. In the VirtualBox main menu, select the pfSense VM and select "Start"

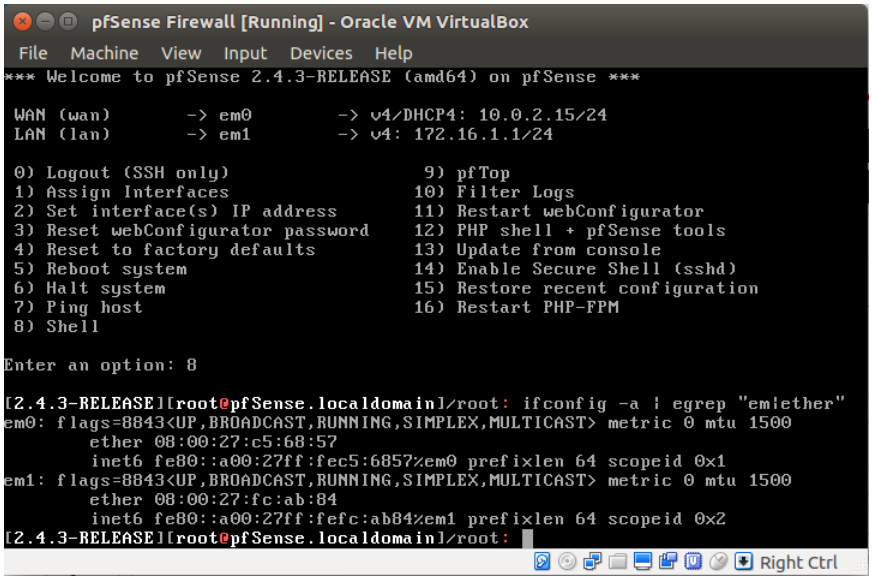
6. Wait a minute until the menu screen pops up, it should look like this:



Note: If it is attempting to reinstall the software go back into VirtualBox and ensure that the ISO is not attached to the optical drive in the pfSense VM (more detailed instructions in previous tutorial)

7. Select option 8 (Shell) and press enter on your keyboard

8. Type the command: **ifconfig -a | egrep "em|ether"** and your output should look similar to the following:



9. Take note of which MAC address is connected to which em. If you followed the steps above in the configuration process, the MAC associated with the NAT will be connected to em0 and the MAC associated with the internal network will be em1

10. Next, we'll be assigning the interfaces, so type "exit" and you should see the main menu again. Type 1 (Assign Interfaces), if you're asked about needing to connect VLANs you can say no "n"

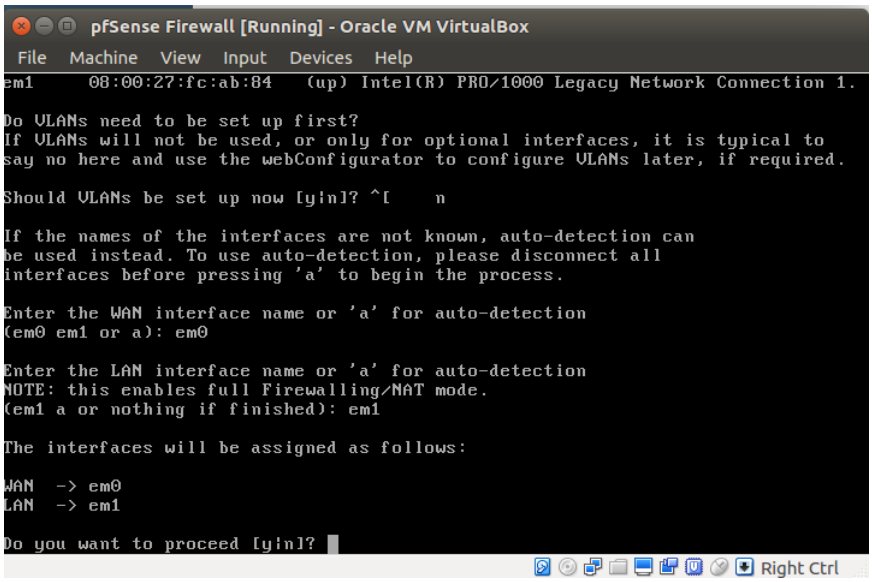
11. Next, it will ask about the WAN interface. Type the correct em that was associated with your NAT adapter, which should be em0.

Note: Again, the NAT basically adopts the host machines IP address within the network as its own IP in order to access WAN resources. The firewall we are establishing will eventually allow us to control what sort of traffic we accept by

filtering out all traffic except for approved destinations or port traffic, more details later.

12. Next type the em associated with the LAN which should be em1.

    Note: the other network adapter we had set up with the internal network. This network connects only the pfSense firewall machine and the File Share System machine to each other. The File Share System is not connected to the Wide Area Network, it is only connected to the filtered results from the WAN accepted through the firewall, or that will be the case once configuration has been completed.

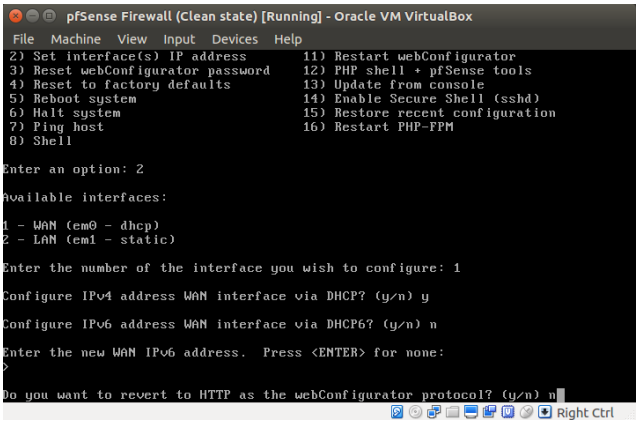


13. When asked if you want to proceed, type "y" and wait for the configuration to take place.

14. The next step is to set the interface IP for the WAN and LAN. After the menu screen appears select option 2 (Set interface(s) IP address).

15. The prompt will ask which interface to set. The WAN IP should already be configured to DHCP, if it is not, type 1. When asked if you would like to configure the IPv4 address WAN interface via DHCP, type "y". When asked the same for IPv6 you can type "n" and then hit enter when asked for the new WAN IPv6 address. You should say, "n" to reverting to HTTP protocol

    Note: There are two major differences between HTTP protocol and HTTPS protocol, which in itself is just the standard for how to initiate a transfer of information over the web. Point one being, that they occur on different ports, HTTP on port 80, and HTTPS on port 443. This will be something we need to take note of during the final configuration of the firewall. The other main difference is that HTTPS is transferred with encryption while HTTP is not.



16. Press <Enter> to continue, this should take you back to the main menu

17. Select option 2 (Set interface(s) IP address) again, so that we can configure the LAN IP

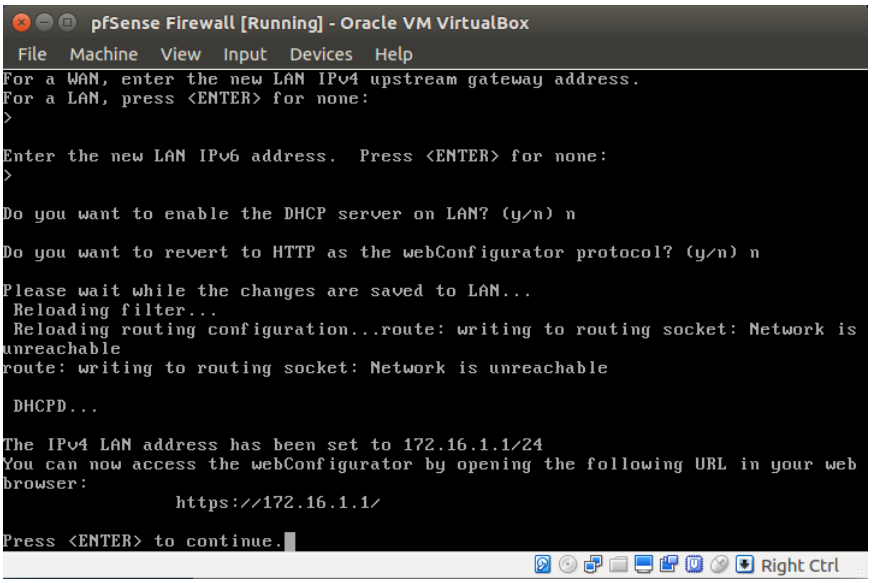
18. Select the second interface (2 – Lan (em1 – static)). Set the IP address to 172.16.1.1 (note that this can be set to anything, just make sure you remember it, because this IP will be used as a gateway on our other VM in a moment). It will then ask for the subnet value, choose 24.

    Note: This means that the network includes all IP addresses from 172.16.1.1 to 172.16.1.224. If you were to choose the subnet value of 16, your range would change from 172.16.1.1 to 172.16.255.254

19. Leave the upstream address blank and press enter. Do the same for IPv6, leave blank

20. Type "n" for enabling DHCP server as you do not want to enable DHCP for the LAN interface Enabling DHCP would destroy some of our manual configuration settings we have implemented.

21. Type "n" for revert to HTTP protocol



22. See that web address posted at the bottom of the screenshot? That allows you to access the online web configuration portal. We will be able to access that from the File Share System VM after we complete the configuration process for that.

23. Close pfSense VM. In VirtualBox, create a snapshot of the current state of the VM, "Clean state"

TROUBLESHOOTING

If you had any issues during the configuration process here are some things to double check:

- Make sure that the pfSense network settings are configured properly in VirtualBox. Adapter 1 should be connected to NAT, adapter 2 should be connected to an internal network named, intnet
- Make sure that you're connected to the internet on your host machine and that your host machine is having no connectivity issues

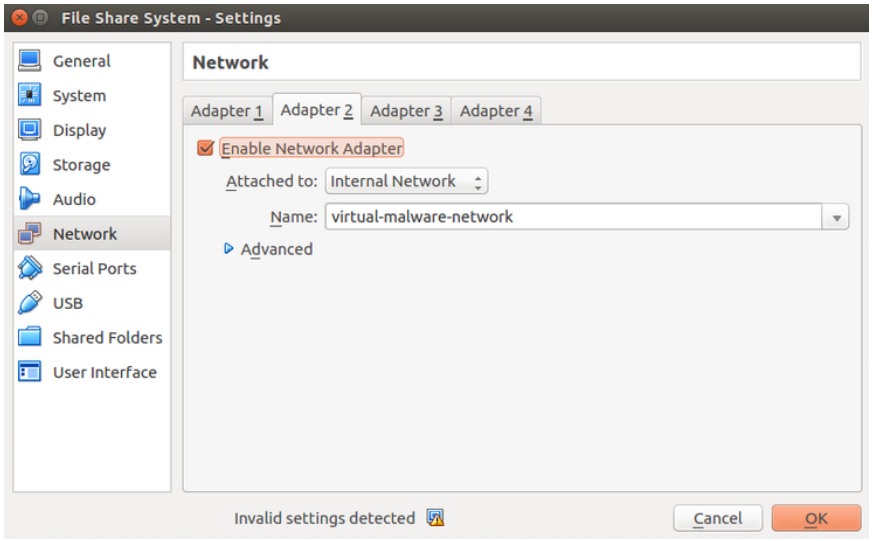
6. Internal Network Configuration

Background

Now that we've set up the configuration of the pfSense VM, we need to make sure that the machine which we'll be using to access the internet (The "File Share System" VM) is properly configured on the internal network. This will connect the File Share System to the firewall, and the firewall is connected to the internet and will filter only accepted traffic through to the File Share System. We will also be setting up the connection between the File Share System VM (which will be used to acquire but not to run malware) and the FakeNet Machine, which will access the malware from a shared folder in the File Share System.

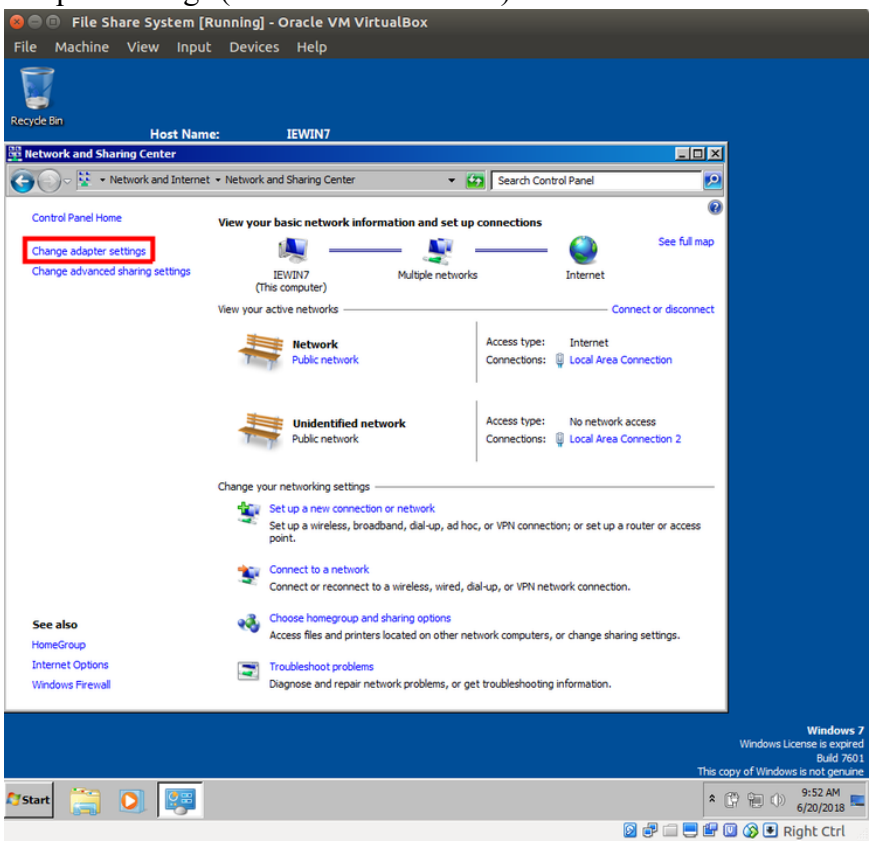
Initial Set-Up

1. On the VirtualBox main page select the File Share System VM and click the "Settings" option
2. Navigate to the "Networks" tab on the left side of the screen and under the Adapter 1 tab select "Internal Network." This should default to selecting the network intnet, or press the down arrow and select intnet, the same network put on Adapter 2 of the pfSense VM
3. Navigate to the Adapter 2 tab and select "Internal Network"
4. Type "virtual-malware-network" in the name box

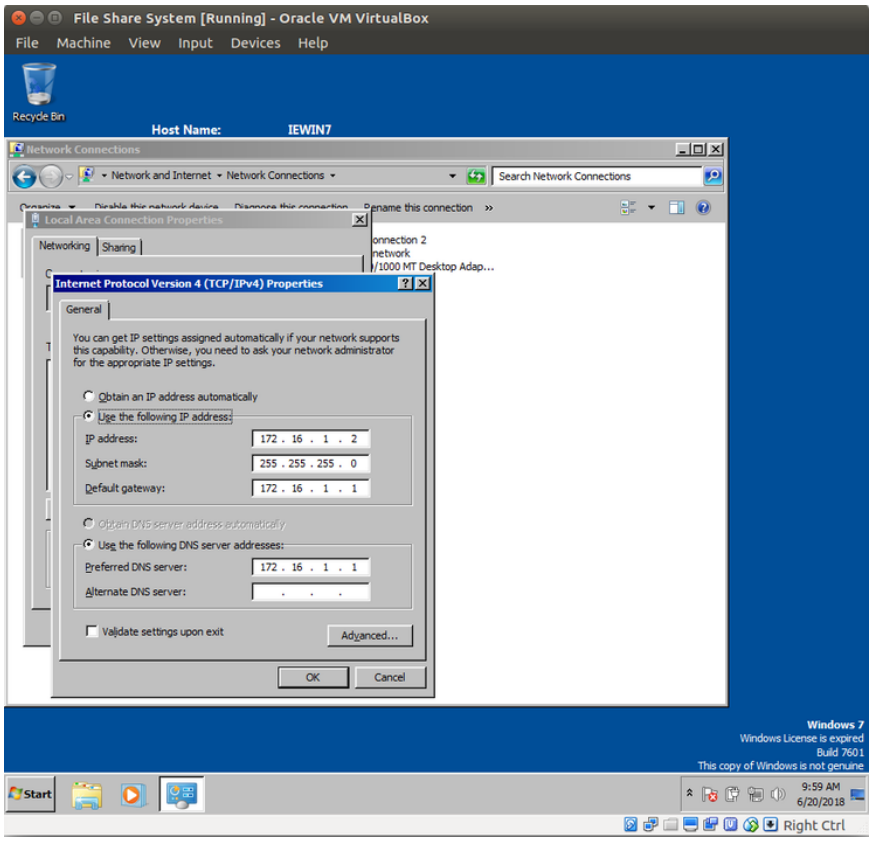


5. Clock "OK" to save your settings and head back to the VirtualBox main page
6. Select FakeNet Network Machine VM and click Settings
7. Navigate to the "Networks" tab on the left of the screen
8. Under the Adapter 1 tab select Attached to: "Internal Network"
9. If you click the down arrow on the right side of the Name box "virtual-malware-network" should be an option
10. Select the "virtual-malware-network" and click OK
11. Back at the VirtualBox main page we are going to want the pfSense VM, the File Share System VM and the FakeNet Network Machine VM all up and running.

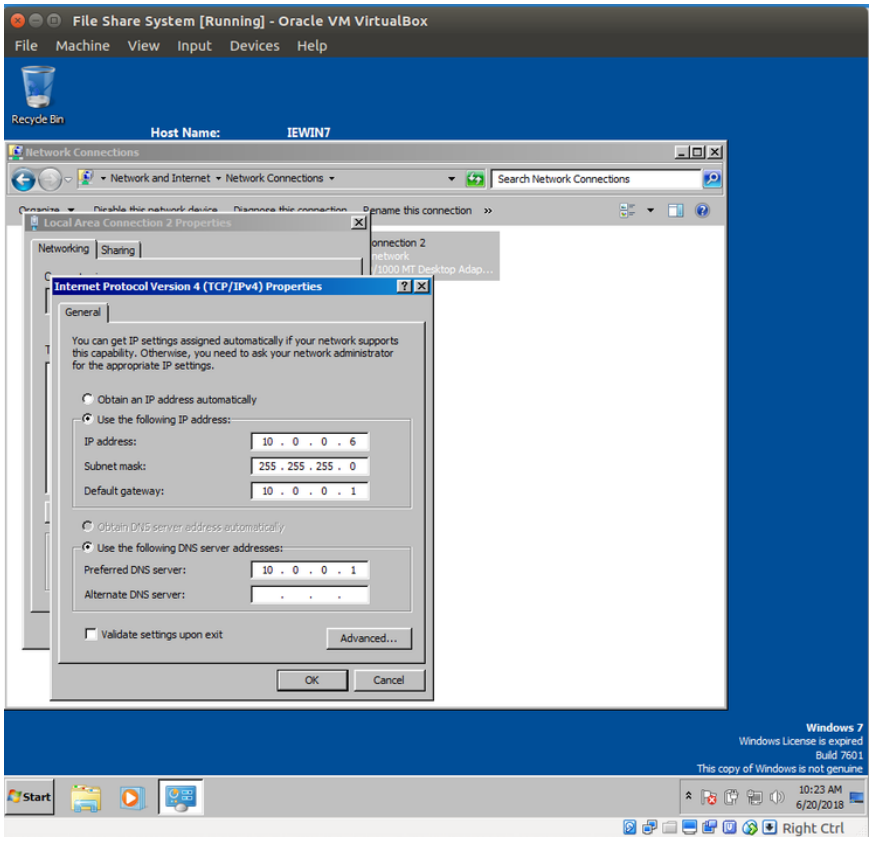
12. After everything has completed booting, we will select the File Share System first
13. Open up the control panel > Network and Internet > Network and Sharing Center > Change adapter settings (link found on the left)



14. You should see two options showing up. Local Area Connection, and Local Area Connection 2
15. Right click on the first Local Area Connection and select properties
16. Highlight Internet Protocol version 4 and select properties
17. Select "Use the following IP address"
18. I used the IP address 172.16.1.2 (you can use any IP within the range you determined in the previous section based on the subnet value and the pfSense LAN IP)
19. I used 24 subnet value again for this one, which translates to 255.255.255.0 as a subnet mask **Note: This should be consistent with the mask value you set up during the pfSense configuration in the previous tutorial**
20. Default gateway is going to be the IP of the pfSense machine, here we used 172.16.1.1
21. Select "Use the following DNS server address"
22. The preferred DNS server should also be the pfSense IP (172.16.1.1)



23. Click OK, then click Close
24. Next, we'll configure LAN 2. Right click on the name and select properties. Then, highlight IPv4 and click properties
25. Here we are setting up a unique network just for the File Share System and the FakeNet Network Machine to communicate
26. Select Use the following IP address. We are going to want to use a different range for this network for clarity. I made the IP address for this one 10.0.0.6, subnet mask: 255.255.255.0, Default gateway: 10.0.0.1.
27. Select the "Use the following DNS server address" option and put preferred DNS server 10.0.0.1 **Note: gateway and DNS server aren't really needed in this section for this tutorial, I had this set up as an expansion of the original process, but it doesn't hurt anything to have it configured this way**



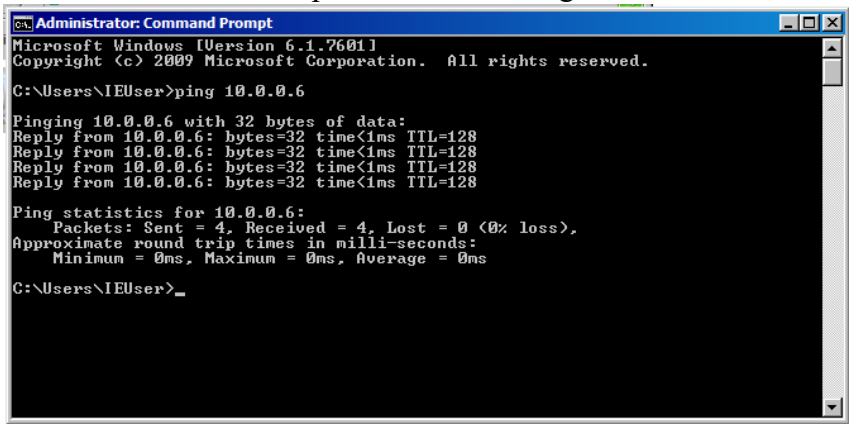
28. Now navigate to the FakeNet Network Machine, open up the control panel > Network and Internet > Network and Sharing Center > Change adapter settings (found on the left)
29. Here there should only be one LAN option. Right click, choose properties, highlight IPv4 and click "Properties"
30. Select Use the following IP address. I made the IP address for this one 10.0.0.5, subnet mask: 255.255.255.0, Default gateway: 10.0.0.1.
31. Select the "Use the following DNS server address" option and put preferred DNS server 10.0.0.1
32. Click OK and then Close

Testing Connections

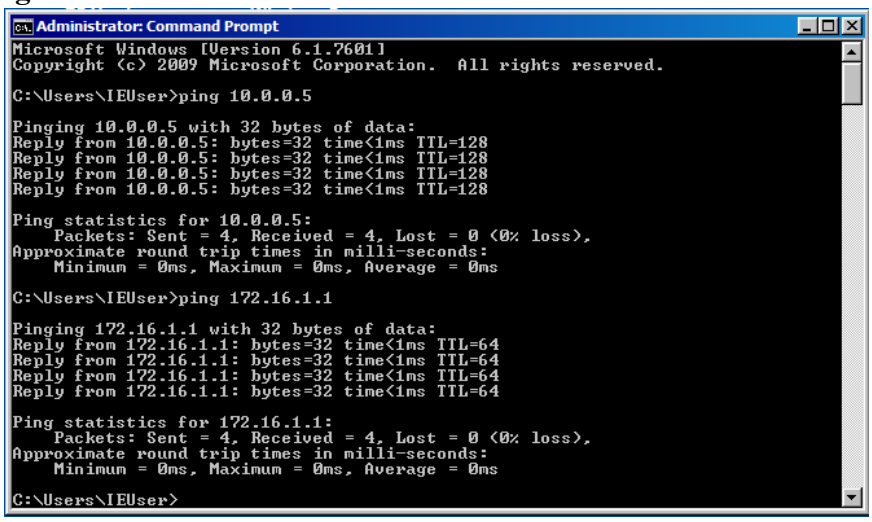
Let's make sure that all of our VMs are talking to each other properly now. If at any point you don't have a connection go back and try working through the steps again to troubleshoot any possible issues. Again, make sure you have all three VMs up, including the FakeNet Network Machine, the File Share System, and the pfSense VM. Make sure that the correct internal network names are all associated with the correct adapter and associated IP addresses.

1. Navigate to the FakeNet Network Machine and open the command prompt application

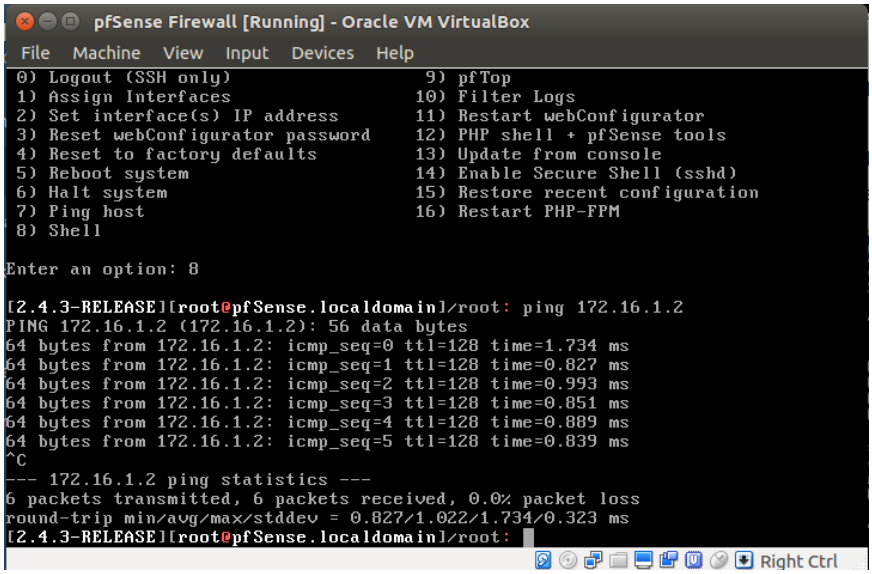
2. Type: ping 10.0.0.6
3. You should receive a similar output to what the image shows below



4. Let's go to the File Share System and open command prompt
5. Type: ping 10.0.0.5
6. You should receive a similar output to the previous one
7. Type: ping 172.16.1.1
8. Again, you should get an output of the reply from with the number of bytes and the time
   Note: Occasionally, during testing after the completed firewall configuration was set up, I noticed that I would not get a ping from 172.16.1.1 on the File Share System, but the firewall was still operating functionally. If you aren't getting a response here, check the next step, pinging FROM the pfSense VM. If that works, you're good to go



9. Navigate to the pfSense VM
10. At the menu screen, select option 8 to open shell
11. Type: ping 172.16.1.2
12. You should get similar outputs to the one above. Type: ctrl + c this stops the pinging



Troubleshooting

I briefly mentioned some troubleshooting tips earlier, but if you're having trouble with getting connections between the VMs double check these settings:

- Make sure that the pfSense VM network settings in VirtualBox are set to Adapter 1: NAT, Adapter 2: Internal Network named: intnet
- Check that the File Share System VM network settings in VirtualBox are set to Adapter 1: Internal Network named: intnet, Adapter 2: Internal Network named: virtual-malware-network
- Check the Windows network configuration settings for the File Share System in the adapter settings section of the control panel and double check that the values for IPv4 match the screenshot and instructions posted above
- Make sure that the FakeNet Network Machine VM network settings in VirtualBox are set to Adapter 1: Internal Network named: virtual-malware-network
- Check the Windows network configuration settings for the FakeNet Network Machine in the adapter settings section of the control panel and double check that the values for IPv4 match the screenshot and instructions posted above

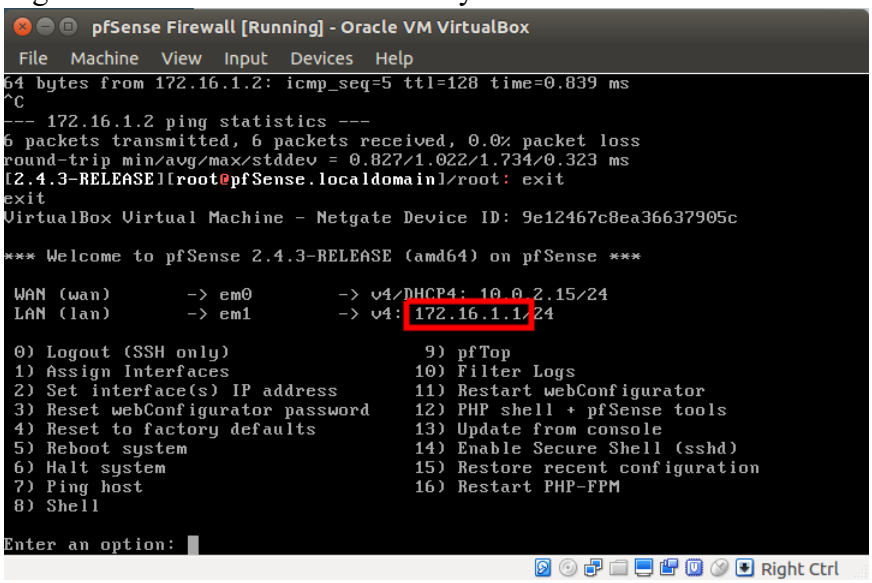
7. Final pfSense Settings

Background

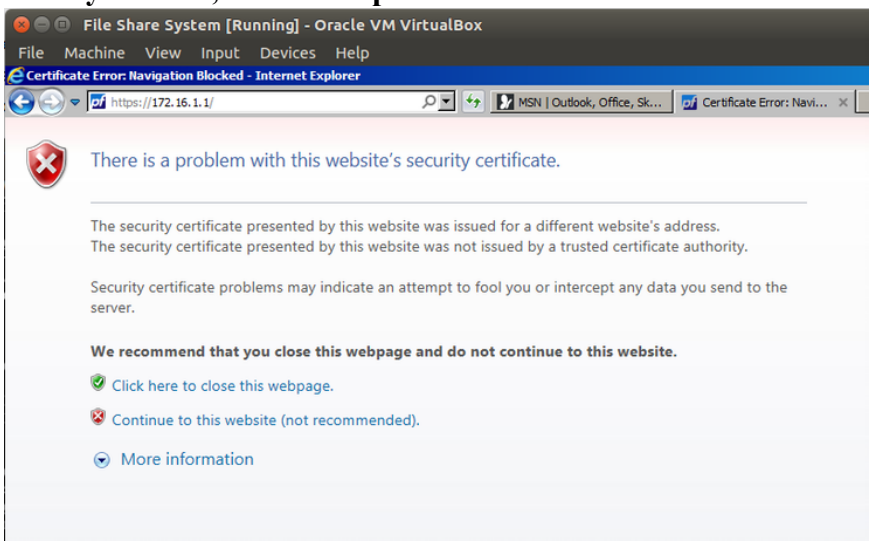
We are going to go into the web Configuration portal for pfSense to finalize our configuration settings. This includes creating the rules which will limit access to the VM, by allowing only traffic over certain ports to be viable and blocking all other traffic. Make sure you have both the pfSense VM running and the File Share System VM running.

Creating Firewall Rules

1. Take note of the LAN interface IP address on the pfSense VM. You can find this information listed right before the main menu populates, if you are strictly following this tutorial it should be 172.16.1.1
2. Everything before the /24 is the IP address you need

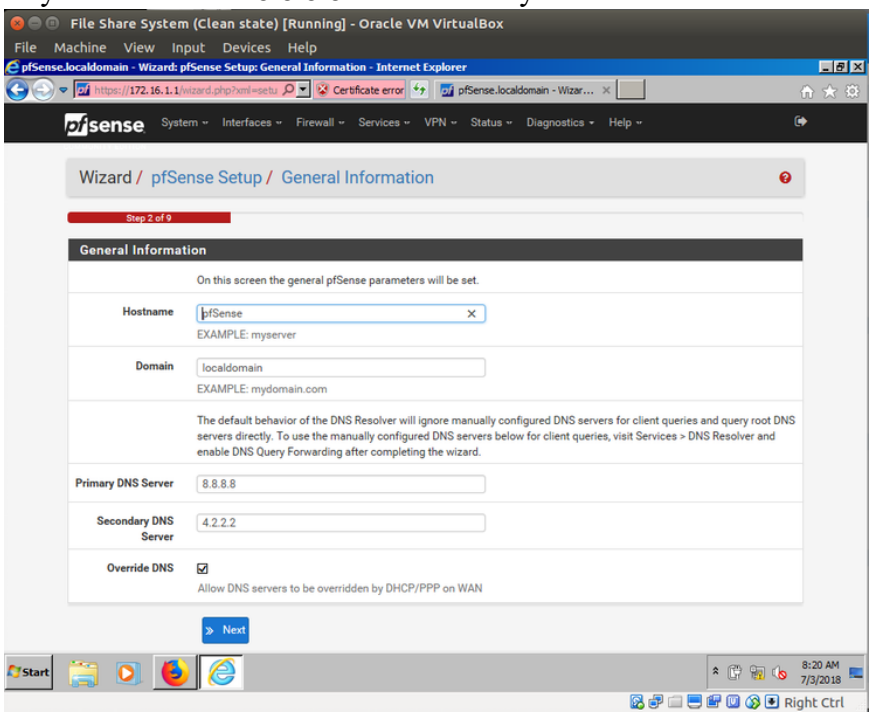


3. Now navigate to the File Share System VM, open a browser window and type http://172.16.1.1 (or whatever the LAN IP is for your system)
4. **Note: you may be blocked from accessing the site due to a certificate error. You are basically connecting to the pfSense VM from your File Share System VM, which never obtained a certificate from the CA (Certificate Authority) as it is a local machine that we just finished setting up, so as long as you didn't put anything malicious on your VM, it's safe to proceed**

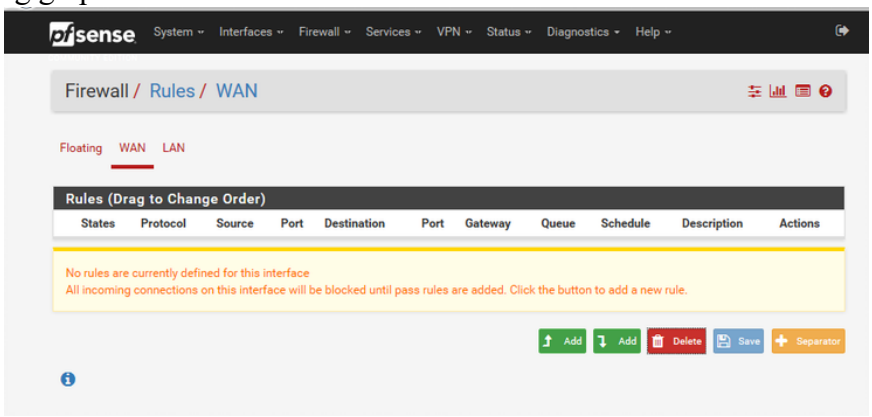


5. Just click the red shield and continue to this website (not recommended) **Note: I'm assuming you're using the VM installed from Microsoft's website, which only has Internet Explorer on it. If you aren't using IE, this screen will look a little different. We will be able to download another browser**

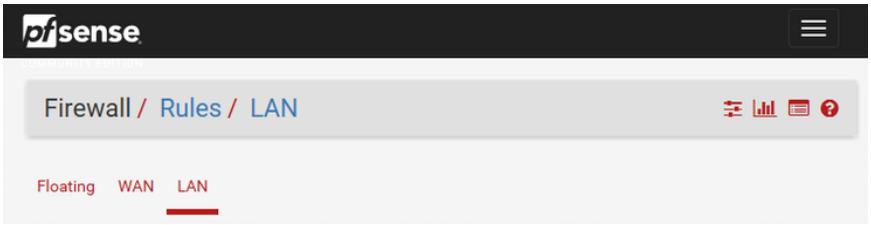
6. To sign in for the first time your credentials are username: admin password: pfsense
7. Start the setup wizard to get the basic configuration set up
8. We are going to leave all settings set to their default state unless noted in one of the following steps
9. Set primary DNS server to 8.8.8.8 and secondary DNS to 4.2.2.2



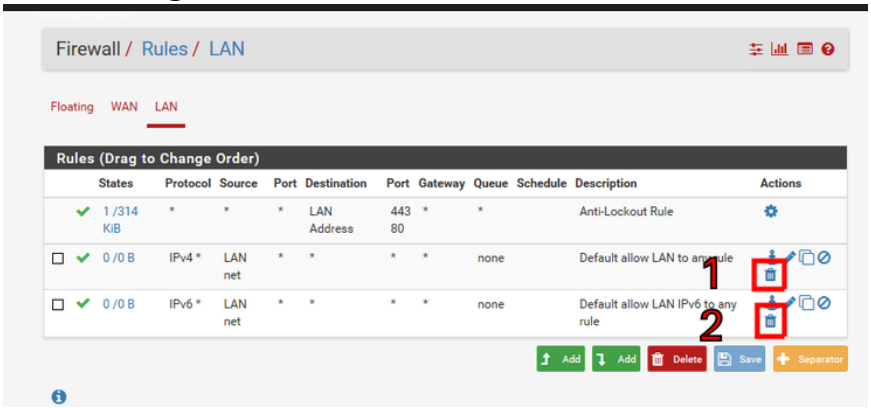
10. Select the appropriate time zone, you can leave the time server hostname as is
11. In step 4 make sure both the Block RFC1918 Private Networks option and the Block bogon networks options are deselected
12. Continue working through the set-up wizard, leaving default values as they are. **Note: Step 6 asks you to make a new password for access, you should fill this out and make sure you remember your password**
13. After the set-up has completed and pfSense has reloaded with the updated changes click Firewall > Rules
14. There should be no rules listed under the WAN page you landed on, if there are any, click on the setting icons and deselect the option. Having no rules listed should look like the following graphic



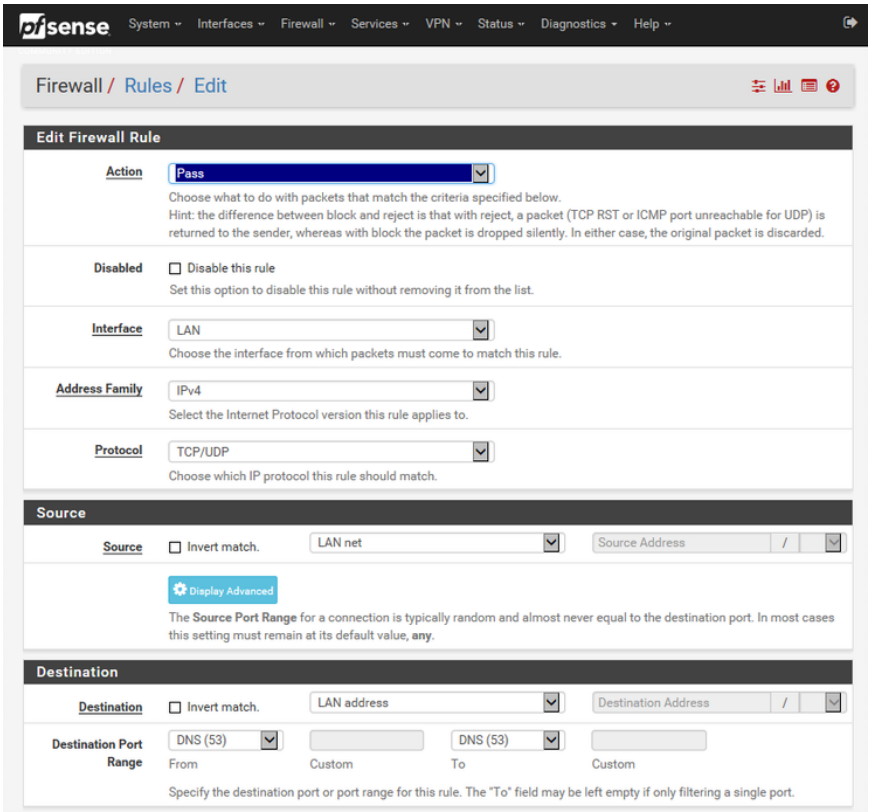
15. Click on the LAN link



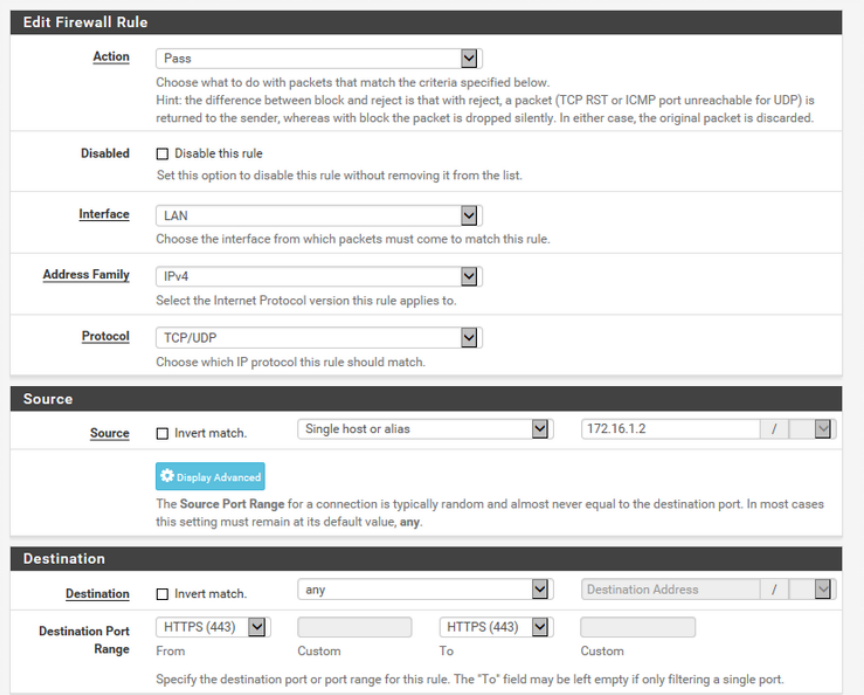
16. Click on the trash icon, deleting the two rules at the bottom of the screen **Note: these rules are designed to let all traffic through on the LAN. What we are going to do now, is start with allowing no traffic, and then build up the type of traffic we want to have come through**



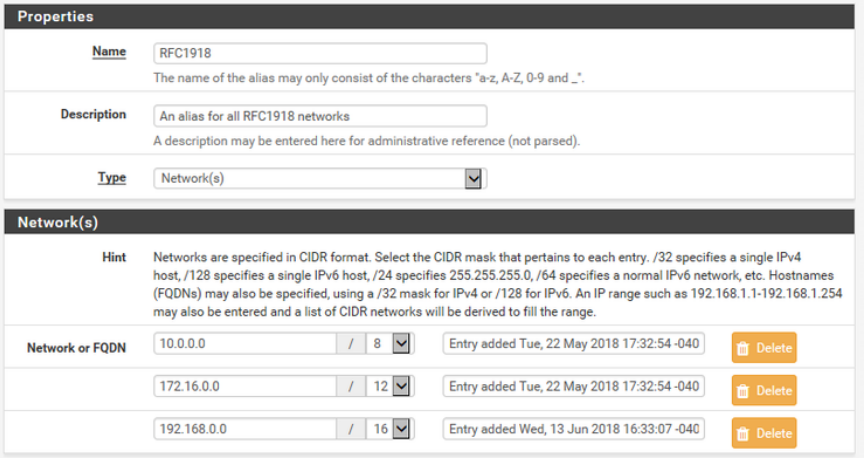
17. Navigate back to the Firewall > Rules page and press the green "Apply Changes" button at the top of the screen **Note: None of the changes you make to the firewall in the rules section are implemented until this button has been pressed**
18. Just to test, in a new tab attempt to navigate to a known website you trust, I used https://www.netflix.com
19. You should not be able to load the page. If your traffic is currently being blocked continue forward, if not, double check that there are no rules set up on your pfSense Firewall and that you have applied all changes
20. Click on the add button. Choose the one with the upward facing arrow which will add the rule to the top of the list (selecting the downward facing arrow will put it at the end of the list)
21. Fill out the form so that Action: Pass; Protocol: TCP/UDP; Source: LAN net; Destination: LAN address; Destination port range: From DNS (53) To DNS (53); Description: DNS Rule **Note: Port 53 (DNS) is the information that allows the translation between the URL you are inputting to the actual IP address of the server housing the information you're requesting**



22. After filling out this form click "Save" at the bottom of the screen,
23. Click the green "Add" arrow with a downward facing arrow and fill out the following; Action: Pass; Protocol: TCP/UDP; Source: Single host or alias: 172.16.1.2; Destination: any; Destination Port Range From: HTTPS (443) To: HTTPS (443). Everything else can be left as default values **Note: HTTPS is a secure way of transferring the data you're requesting through an encrypted connection**



24. After filling out this form click "Save" at the bottom of the screen, once the page reloads be sure to press the "Apply Changes" button
25. Next, navigate to the Firewall > Aliases section and click on IP and click Add
26. Fill out the form with the setting listed below



27. Click Save at the bottom of the screen and then Apply Changes
28. You should now be able to open a new tab and try to access a known website (using https, not http), most popular sites should be using https. You can try accessing Netflix again, https://www.netflix.com

TROUBLESHOOTING

If you're struggling to access the Internet from the File Share System VM after following the instructions check these things:

- Double check the network settings in VirtualBox for the pfSense VM. Adapter 1 should be enabled and set to: NAT; Adapter 2 should be enabled and set to Internal Network named: intnet
- Double check the network settings in VirtualBox for the File Share System VM. Adapter 1 should be enabled and set to Internal Network named: intnet
- In the File Share System VM the adapter settings must be configured properly, refer to this tutorial for more details
- Your host machine must be connected and able to access the internet
- The pfSense LAN IP address and the address in the File Share System VM browser window must match

If you're still experiencing technical troubles go to pfSense's troubleshooting page https://www.netgate.com/docs/pfsense/routing/connectivity-troubleshooting.html.

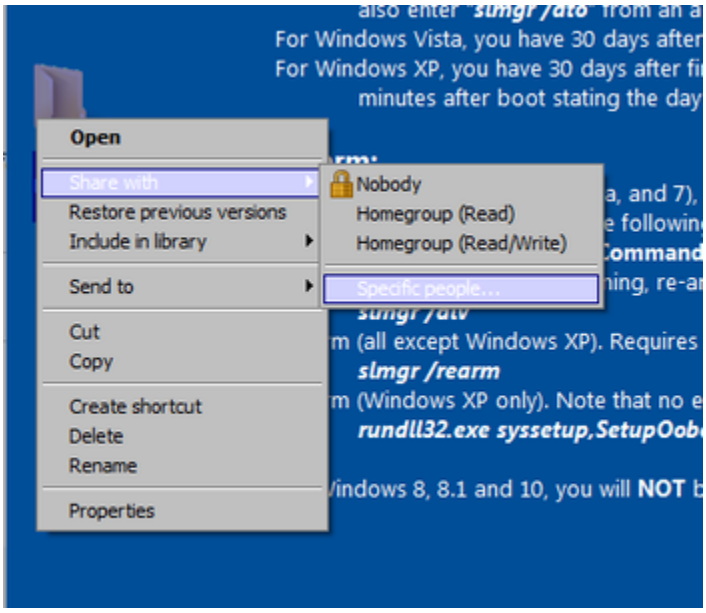
8. Create Shared Folder

Background

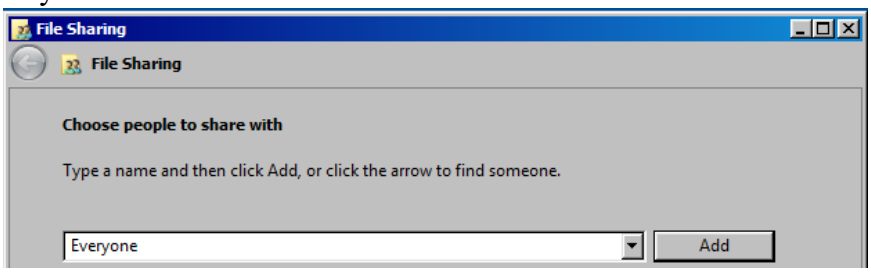
For this step we are going to want the File Share System VM and the FakeNet Network Machine VM up and running. This folder is going to allow the malware to get from the VM connected to the Internet to transfer files via an isolated network to the machine where we can safely run and test the malware.

Create and Share Access to a Shared Folder for File Transfer

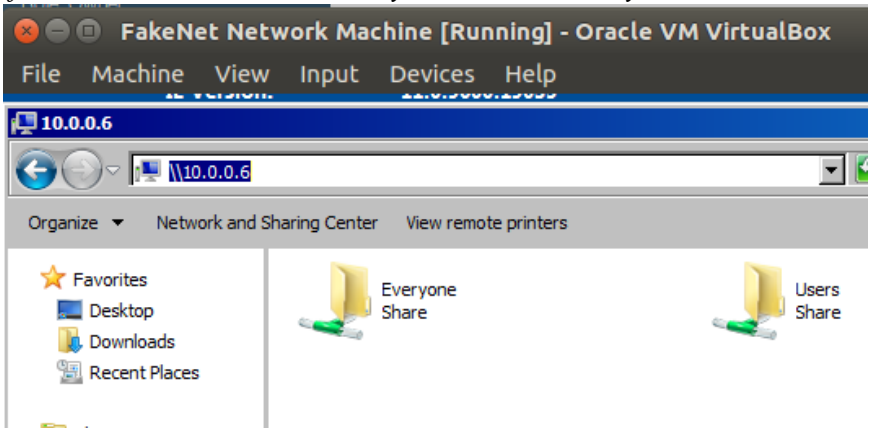
1. Open up the network settings in VirtualBox for the File Share System VM
2. Navigate to the Adapter 2 tab, make sure it is enabled and that it is attached to an Internal Network named: virtual-malware-network
3. Click Ok
4. Open up the network settings in VirtualBox for the FakeNet Network Machine VM
5. The Adapter 1 setting should be enabled and attached to Internal Network named: intnet
6. Boot up both the File Share System VM and the FakeNet Network Machine VM
7. Navigate to the File Share System VM and access the folder we created in an earlier tutorial to be used as a shared folder with the FakeNet VM, I called mine Shared Malware Folder. If you skipped that step or haven't made one yet, you can create a new folder now
8. Right click the folder and click "Share with" and "Specific people"



9. Type Everyone in the bar and click "Add"



10. Select permissions and change to read/write
11. Click "Share" at the bottom of the page
12. Click Done
13. Open up the FakeNet Network Machine and open File Explorer
14. In the search bar type: \\10.0.0.6 (***Note**: don't forget the \\. Also, if you used a different IP address for LAN 2 on the File Share System IP address, you'll need to use that instead.*)



15. Click on the Folder labeled Everyone > IEUser (or whatever username is associated with your File Share System) > Desktop (or whatever file location you used to save your shared folder

Troubleshooting

If you're having trouble accessing the shared folder from the FakeNet Network Machine VM double check the following:

- In the network settings in VirtualBox for the File Share System VM make sure Adapter 2 is enabled and set to Internal Network named: virtual-malware-network
- In the network settings in VirtualBox for the FakeNet Network Machine VM make sure Adapter 1 is enabled and set to Internal Network named: intnet
- Make sure the adapter settings within the VM are accurate and consistent with the previous tutorial found here

9. Final Configuration and Transfer Checklist

Final Configuration Settings

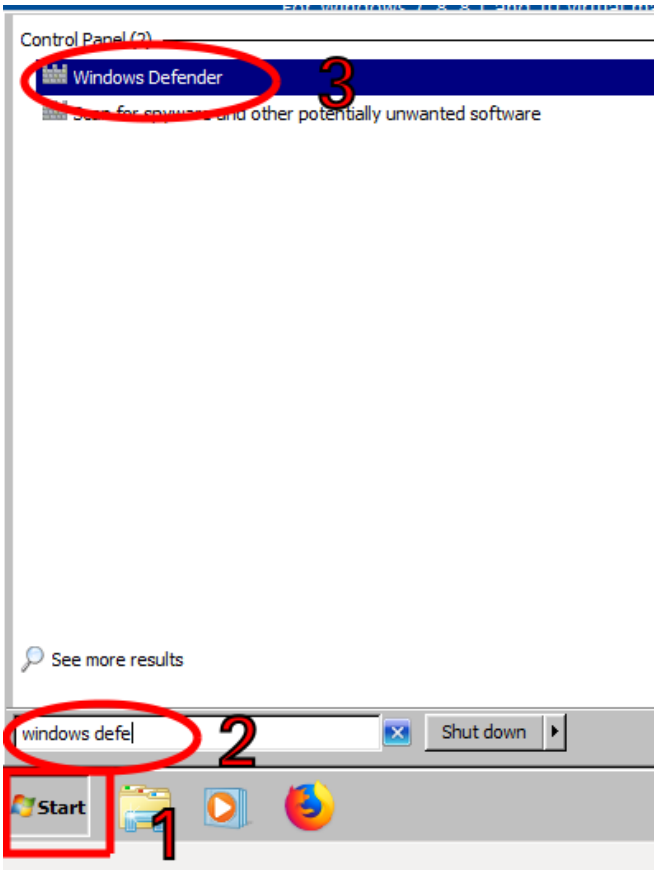
There are a few steps you'll want to take to prep your FakeNet Network Machine before you actually start the process of analyzing your malware. In addition to changing your security settings so that Windows does not automatically get rid of your malware before you have a chance to look at it, you'll also want to come up with a procedure that you'll follow each and every time you start the malware analysis process.

This provides you with a way to ensure that you are consistently approaching malware analysis safely. Instead of relying on your memory to make sure you're going through the steps appropriately you can print out or pull up your procedure and use it as a guide to ensure you do the same thing every time and minimize the risk of causing harm to your host computer.

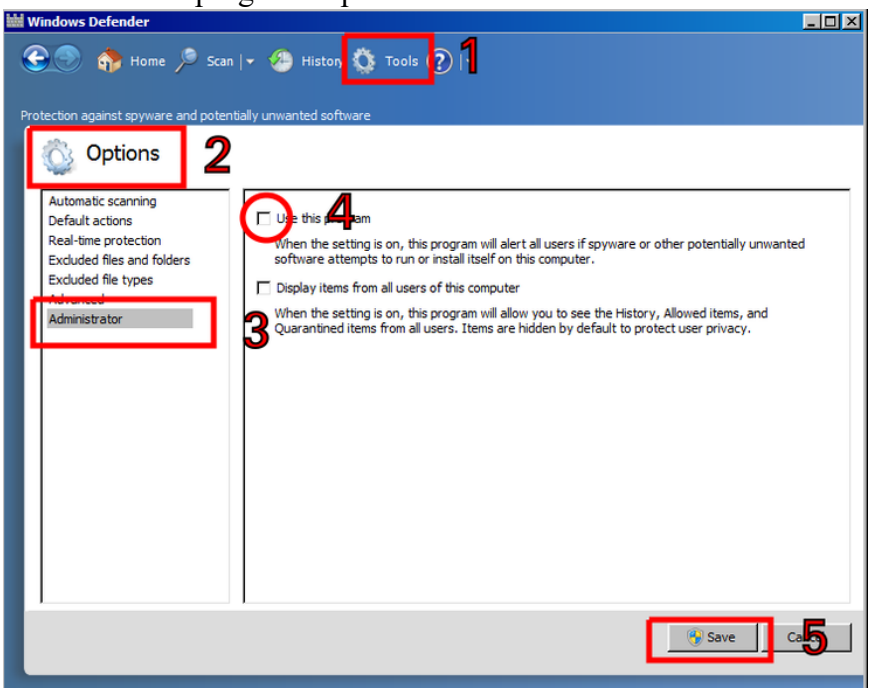
Disable Security Functionality

Let's start with the process of eliminating Windows security features as an obstacle in our analysis process:

1. Start the File Share System VM in VirtualBox
2. Once finished loading, click the "Start" button and type Windows Defender, click on the application in the search results

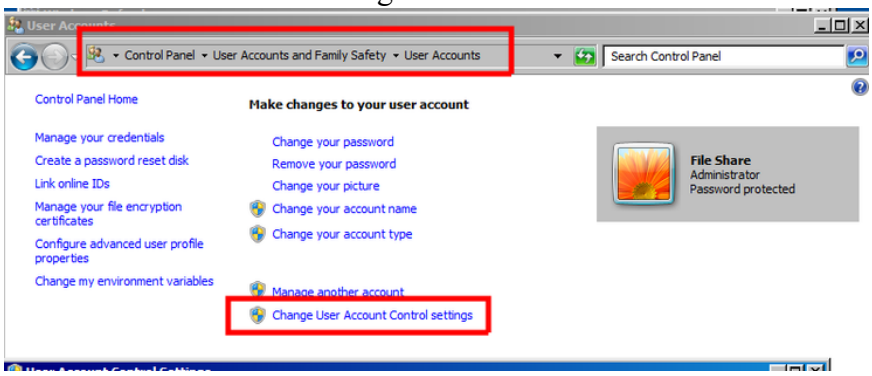


3. Click the "Tools" icon, then "Options", and select the "Administrator" tab on the left tool bar
4. Deselect the "Use this program" option and click "Save"

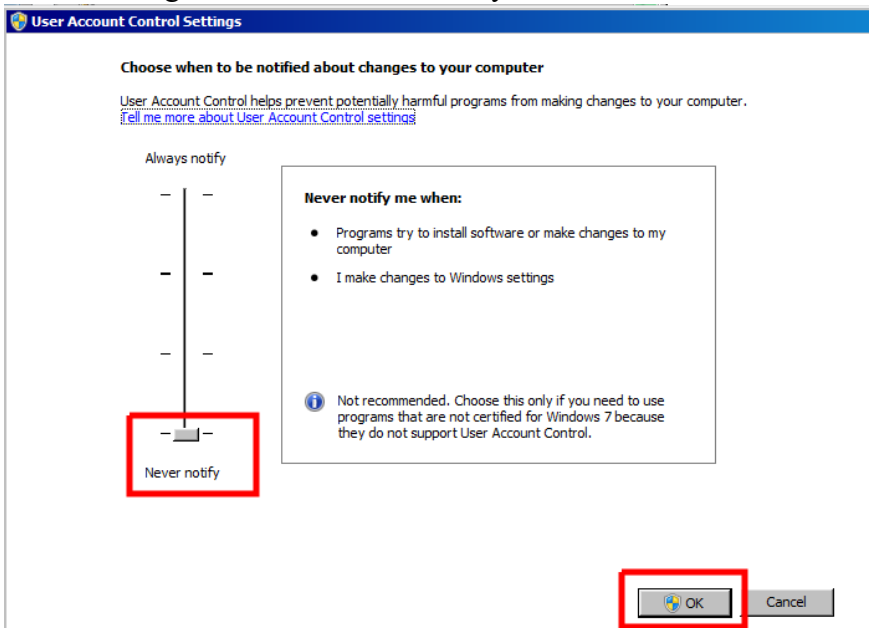


You'll probably also want to turn of user account notifications of security settings.

1. Go into control panel > User Accounts and Family Safety > User Accounts and select Change User Account Control Settings

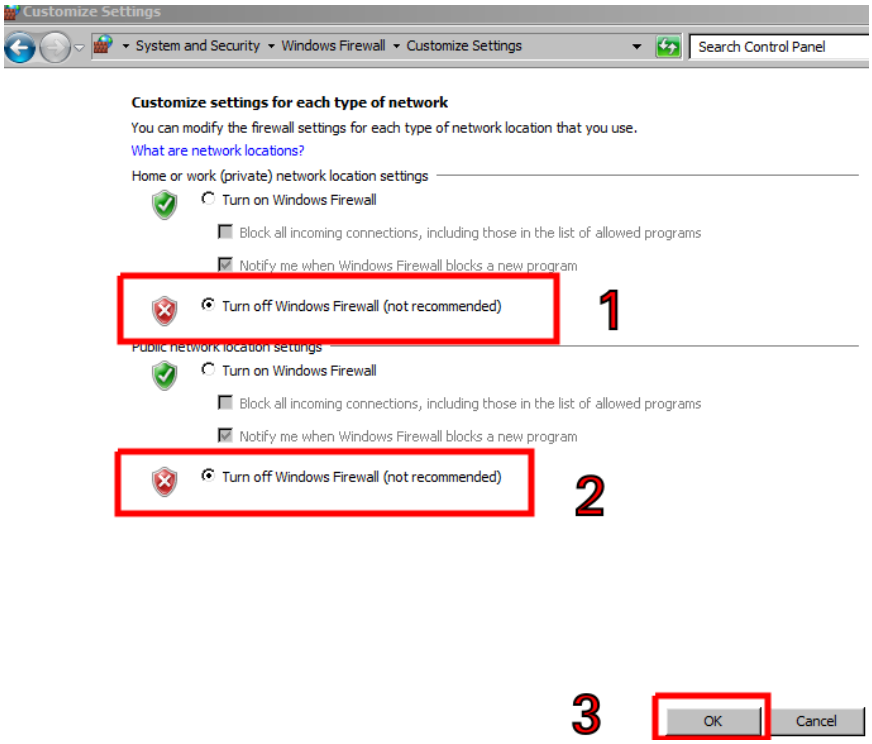


2. Make sure the settings are set to "Never Notify" and click OK



You'll also want to turn off the firewall.

1. Search for firewall in the start menu search and open "Windows Firewall"
2. Select the "Turn Windows Firewall on or off" link on the left, and turn both public and private firewalls off



Now that the network configuration steps have been completed, what we're going to do is come up with a list of must-dos each time we access the Internet or need to transfer documents between our File Share VM and the FakeNet VM.

Malware Transfer Procedure

1. Open VirtualBox.
2. Change network settings in VirtualBox for File Share System VM to disable Adapter 2 and that Adapter 1 is set to an Internal Network named: intnet.
3. Open pfSense VM network settings in VirtualBox.
4. Double check that Adapter 1 is enabled and set to NAT, and that Adapter 2 is set to an Internal Network named: intnet.
5. Start pfSense VM.
6. Start File Share VM.
7. Log in to pfSense web portal through the File Share machine to ensure it is up and running.
8. Navigate to pre-approved website for acquisition of compressed folder containing malicious artifact.
9. Download artifact directly into the Shared malware folder.
10. As soon as download has completed, power off the File Share VM.
11. Power off pfSense VM.
12. Go into File Share System VM network settings and disable Adapter 1.
13. Enable Adapter 2 in File Share's network settings and make sure it is set to an Internal Network named: virtual-malware-network
14. Go into the FakeNet Network Machine VM network settings and ensure that Adapter 1 is set to an Internal Network named: virtual-malware-network
15. Start File Share VM.

16. Start FakeNet VM.
17. Access the shared folder from FakeNet VM.
18. Move the downloaded artifact in the shared folder, without opening or editing it any way, onto the local FakeNet machine by dragging it onto the desktop.
19. Once it has been copied onto the desktop, delete the file in the shared folder and ensure that there is no copy of the malicious artifact in the shared folder.
20. Power off the File Share Machine by pressing the x in the top corner and selecting "Power off the machine" and selecting the check box that says, "Restore current snapshot 'Clean state' before clicking OK.
21. Open malicious artifact and run any applicable testing or review of the artifact.
22. Power down FakeNet machine by pressing the x in the top corner and selecting "Power off the machine" and selecting the check box that says, "Restore current snapshot 'Clean state' before clicking OK.

If at any time the transfer or analysis process is interrupted or must be stopped before completion of all steps, user MUST destroy all progress and return both the File Share and FakeNet VMs to their original clean state before leaving the workstation.

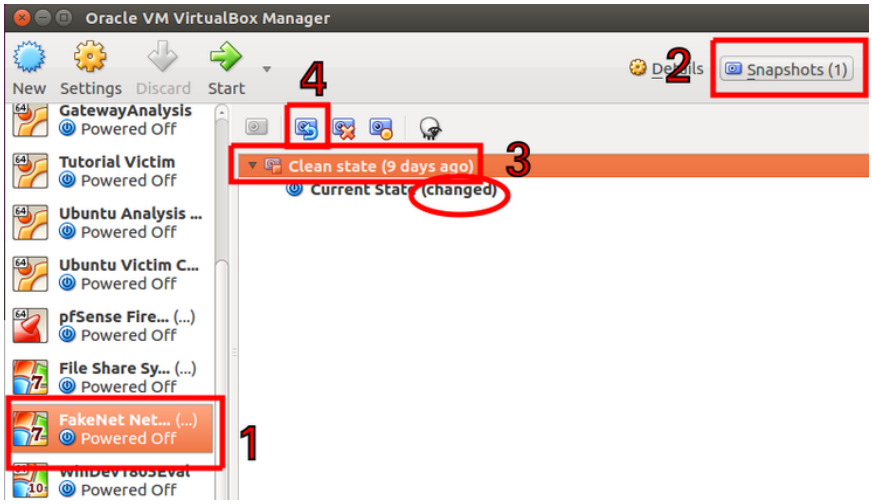
Note: These are simple reminders, in only some detail, intended to help you not forget steps to ensure the acquisition process remains segmented and secure. If you will be sharing the task of analyzing malware with others you should make a more detailed set of instructions and include a failsafe at multiple points to assist in case something goes wrong.

APPENDIX B (MALWARE ANALYSIS TUTORIAL)

BASIC STATIC MALWARE ANALYSIS

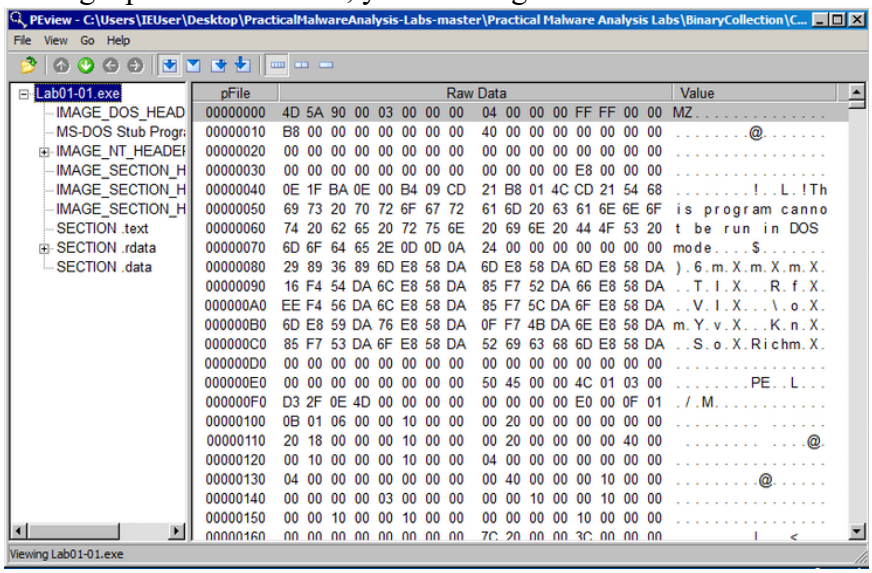
Malware analysis is complex and requires lots of critical thinking. It takes a long time to build up the skills and knowledge which serves as the foundation to conduct high level analysis. That being said, I attempt to provide some basic structure for how to begin the analysis process. As you are working through analyzing your own malware, take notes, improve your processes and add in tools you find useful through your review process. This tutorial uses malware and strategies that can be found in the comprehensive book, Practical Malware Analysis, a book that I highly recommend for further investigation into malware analysis. The malware is available through their website (https://www.practicalmalwareanalysis.com).

1. First thing, you'll want to acquire and transfer the malicious file(s) to your FakeNet machine. If you're downloading them from online, you'll want to use the File Share System VM we set up in the previous tutorials and transfer them to your FakeNet VM through the shared folder we set up previously. Once you have the malware on your FakeNet Network Machine you can continue progressing through this list
2. In VirtualBox click on the FakeNet Network Machine VM (make sure its current state is "Clean state" **Note: You can determine whether it is in a Clean state, or changed from that state by clicking on the snapshots tab and reviewing whether you see (changed) next to the current state**
3. If the current state is changed, select the "Clean state" option and select the "Restore" icon to revert the machine back to its "Clean state"

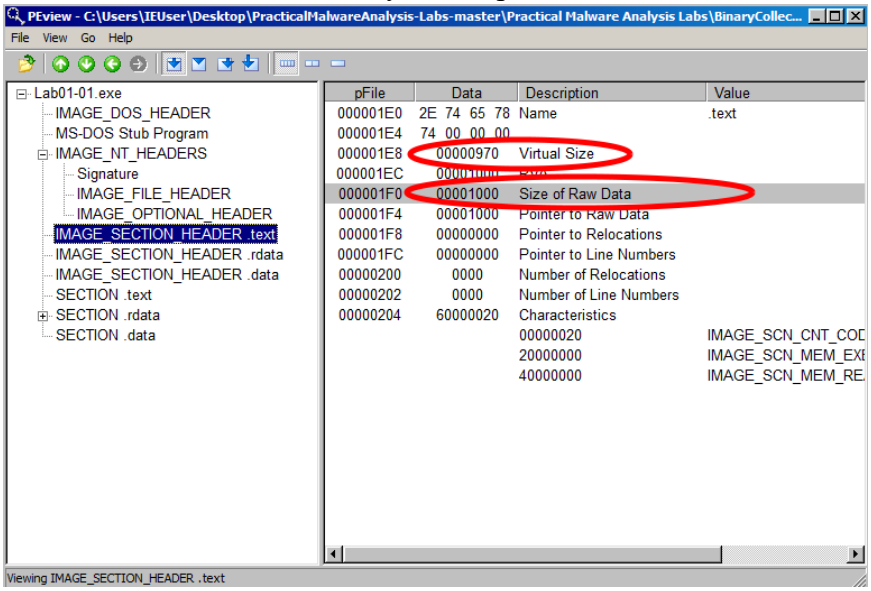


4. **Note: You can choose to create a snapshot of the Current State, which you may want to do if you would like to further run analysis on malware on that machine. For safety and consistency, however, I would get rid of any copies of your FakeNet Network Machine which have malware on it**

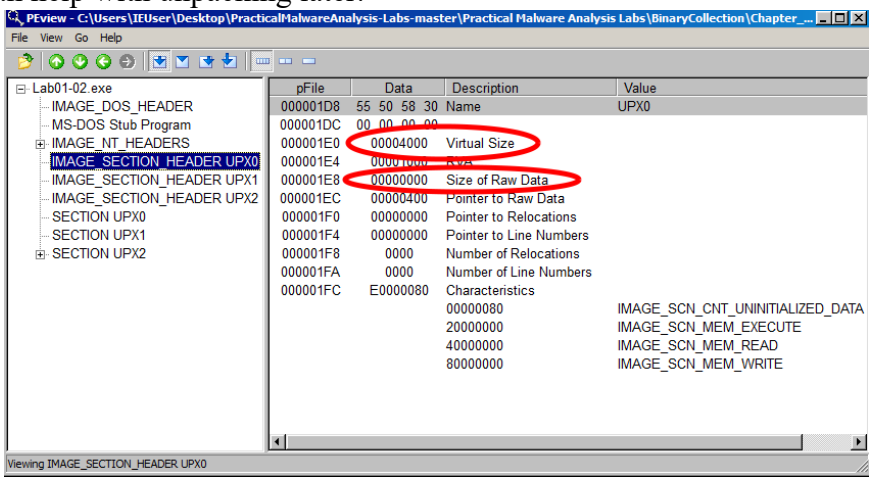
5. Select the FakeNet Network Machine VM and select "Start"
6. Open PEview
7. Navigate to the location of the malware you are wanting to analyze and select the file you want to review
8. After selecting a piece of malware, you should get a similar view to this:



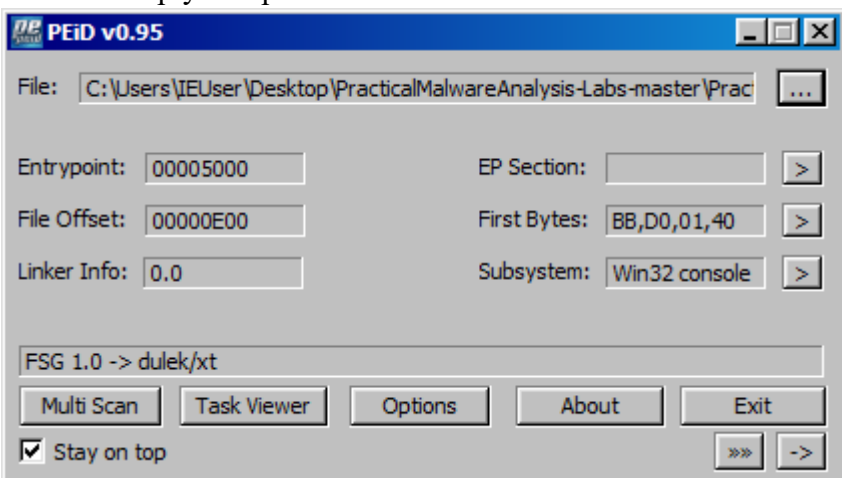
9. This is going to give you a look at a lot of information, one of which is the string values found in the PE header **Note: PE means Portable Executable, which is a file type. This file type contains headers which have valuable information including strings, and imported functions which can be used in determining the function of the malware**
10. The first thing we want to determine is whether the file is packed or not. The first thing to review is whether the IMAGE_SECTION_HEADER sizes are consistent between Virtual Size and Size of RAW DATA. There is likely to be some difference, but they should be similar in size, the example below indicates a Virtual Size of 970 and a Raw Data Size of 1000, so this is an indicator that it may not be packed



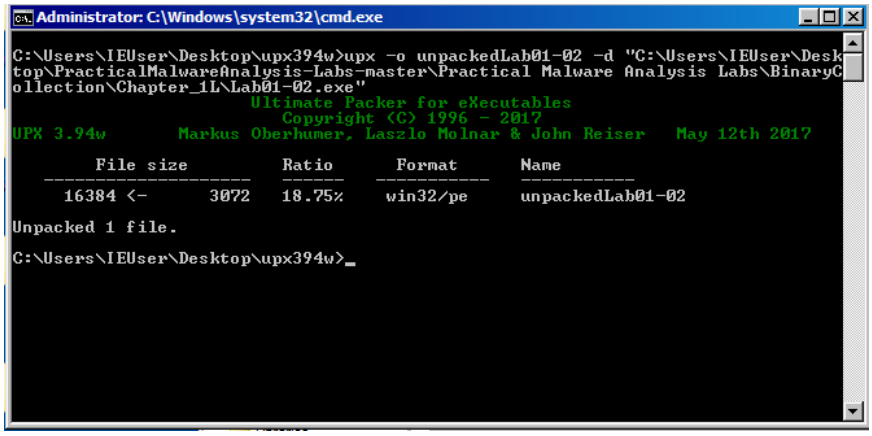
11. The example below shows a file with a virtual size of 4000 but a raw data size of 0. This is an indication that the file is packed. The highlighted IMAGE_SECTION_HEADER UPX0 is titled with UPX, which is type of packer/compiler. Knowing that the file uses UPX can help with unpacking later.



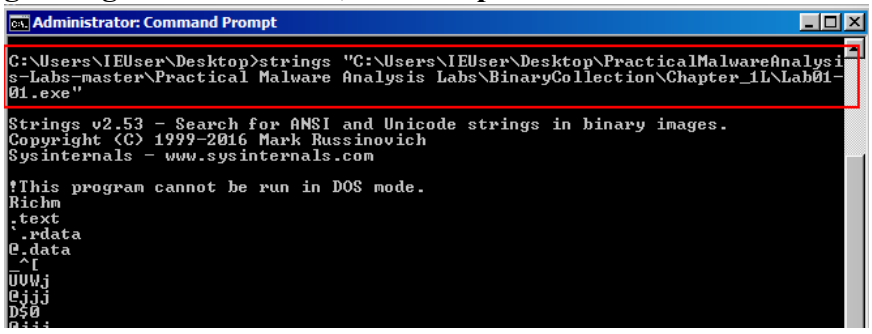
12. Note that using PEid can also assist in determining what the file is wrapped with. If it is wrapped in a compiler PEid will tell you what type, if it is able to determine it. PEid, just like any other step in the analysis process, is not foolproof. We have to review it as if it were another clue and make sure the surrounding information backs up our hypothesis. The image below shows that FSG is used to pack the program. PEid may show that they are unable to find what the file is wrapped in, or it may accurately or inaccurately indicate that it is simply compiled with Microsoft Visual C++.



13. If the file you're reviewing is packed, you can attempt to unpack the malware. If you can identify it as being packed with UPX we can attempt to unpack it. You'll want to download UPX onto your File Shared System and transfer it to your FakeNet Network Machine first (https://upx.github.io/). You can open up Command Prompt and type: upx -o newFileName -d originalFileName **Note: You must be in the file location that the upx.exe is located in to run the upx command.**

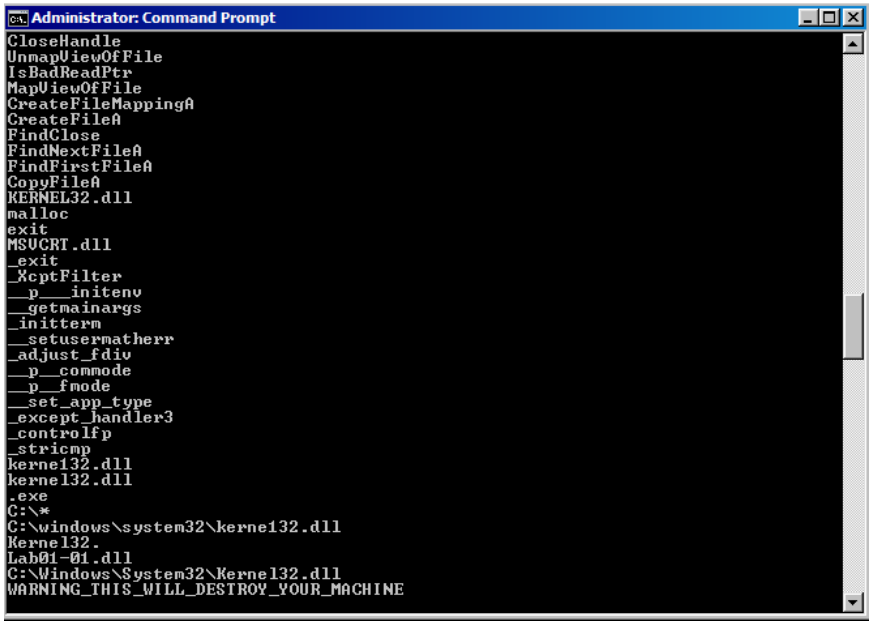


14. After this you can run the analysis on the newly created file as usual. **Note: the output file will save in the current open directory you are using in Command Prompt unless otherwise specified. For me it is in the upx394w folder located on my Desktop.**
15. We then want to review strings and imports. PEview, the program we began with has a column named 'Value' which shows string values included in the PE header. We can use this to look for helpful hints in determining the function and type of file we are looking at. Another way to review the strings in the PE header is to use the Strings program (offered by Microsoft) you can run it in Command Prompt by writing "strings 'file_location' as seen below **Note: Remember you need to be in the same folder location as the Strings application is when you are running the strings command in Command prompt. For me this was on my desktop (see below). If you need to learn some Command Prompt commands (like change directory (cd) or listing files in a folder (ls) you can find a list of Windows 7 commands at this site: https://www.lifewire.com/list-of-windows-7-command-prompt-commands-4107370. Also, note that you can use File Explorer to navigate to the file location of the executable you're analyzing and press SHIFT + right mouse click and select "Copy as path" you can paste that into Command Prompt (after navigating to the correct directory with the strings application, and after typing out 'strings ' and simply pressing the right mouse button, that will paste the file location**

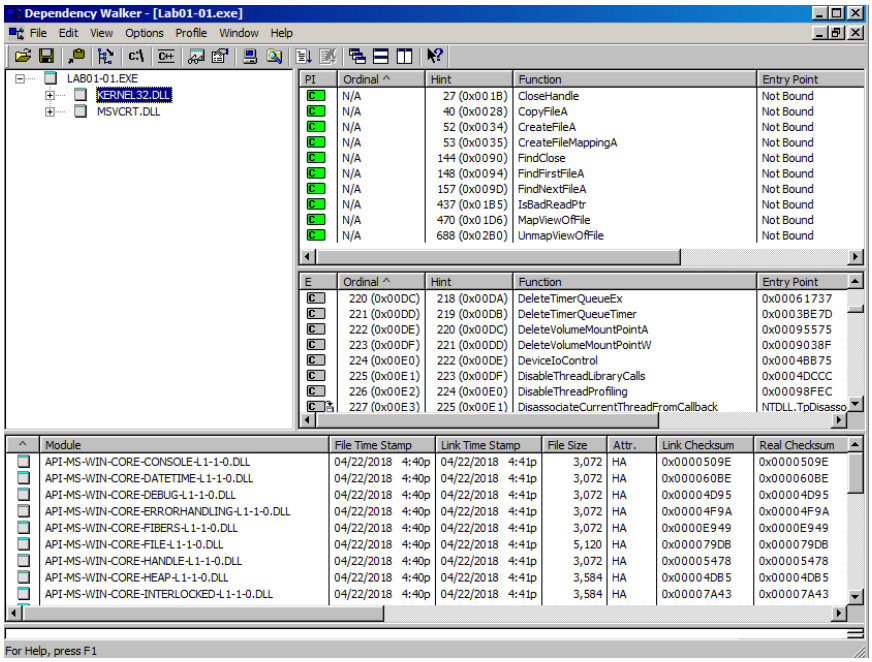


16. You can note that most of the strings are gibberish, but the first line says that !This program cannot be run in DOS mode. This is just a standard input in dll (dynamically linked lists) files. **Note: Dynamically linked lists are files which store imports that are already on the host machine. When needed, the dll calls the function and it is run in the location it is already stored in.**

17. Further down in the list of strings we received from our Command Prompt statement, we see a list of imported functions that denote certain activities. Strings like "FindNextFile" "CopyFile" and "CreateFile" shows this program has the capacity to read and manipulate files.

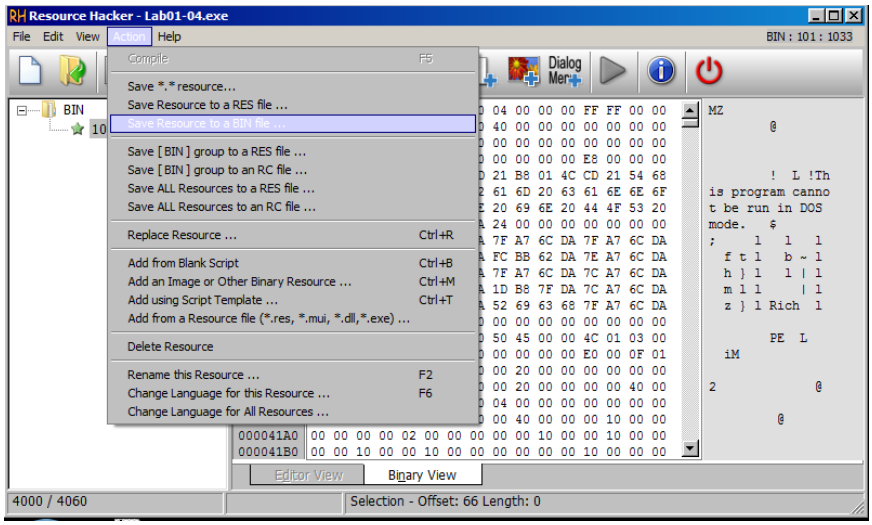


18. Another way to find imports is using Dependency Walker. Clicking on the dll shows all the functions imported reflecting the capabilities of the artifact you're reviewing. The data output in Dependency Walker matches the functions we reviewed in the strings output above.

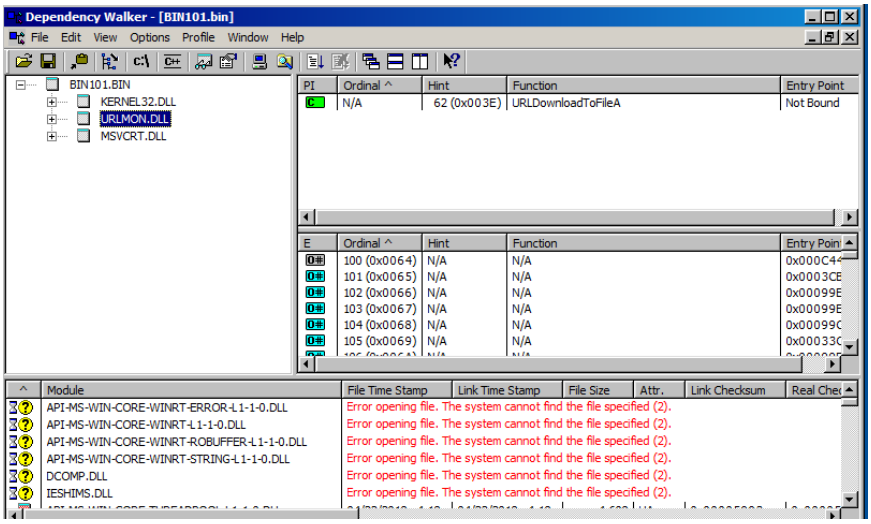


19. We can also use Resource Hacker to review resources for PE-formatted binaries. This tool is able to capably extract any binary resources without needing to run the malware.

The file below shows one example of an extracted binary, which is actually an executable, as it has the same header format as the previous examples "!This program cannot be run in DOS mode." Not all malware has binary files saved this way. For additional review select "Action" > "Save as BIN file" and review further to determine the network functions it accesses.



20. You can see that URLDownloadToFile is an import which is a part of the binary file we saved to review. This function can be used to access and download more malware onto the host machine.



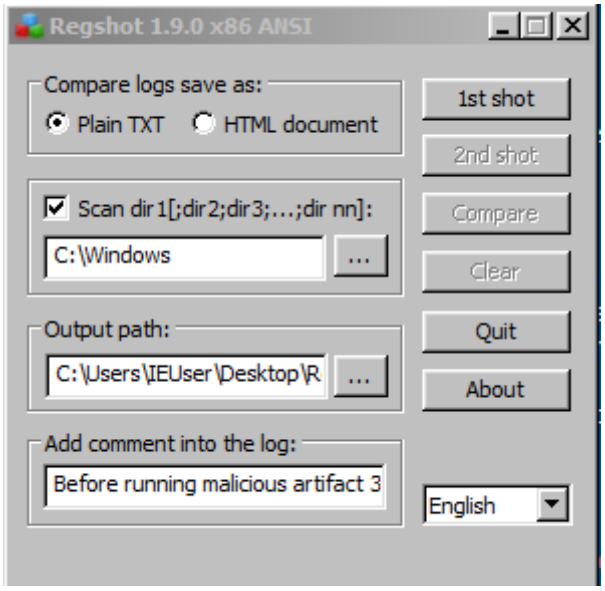
SUMMARY

Hopefully this gives you a bit of help starting a process of reviewing malware to determine what the malware's intent is. After determining the malware is not packed, or unpacking the malware if at all possible, review the malware for strings and imported functions which may communicate the intent of the malware. Use Resource Hacker to look for any additional files, if those files happen to be a PE, you can conduct further analysis with the software we reviewed in this section.

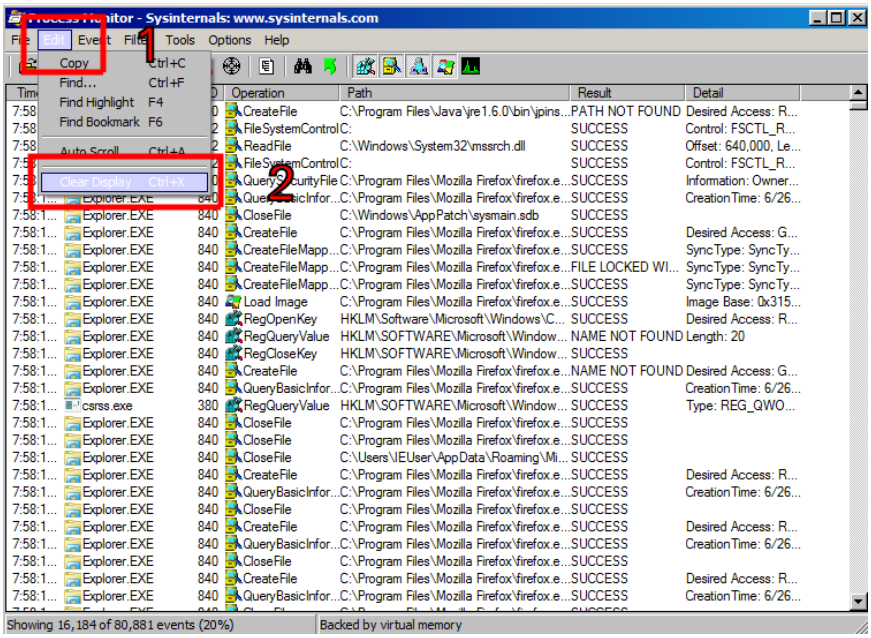
Basic Dynamic Malware Analysis

Every time you review malware you'll want to begin with the static analysis. This will give you a good starting to point to understand of the type of malware you're looking at, and its capabilities. You should keep those in mind as you review the malicious software during dynamic analysis, where you run the malware in a virtual environment and monitor the changes that take place within the machine and network.

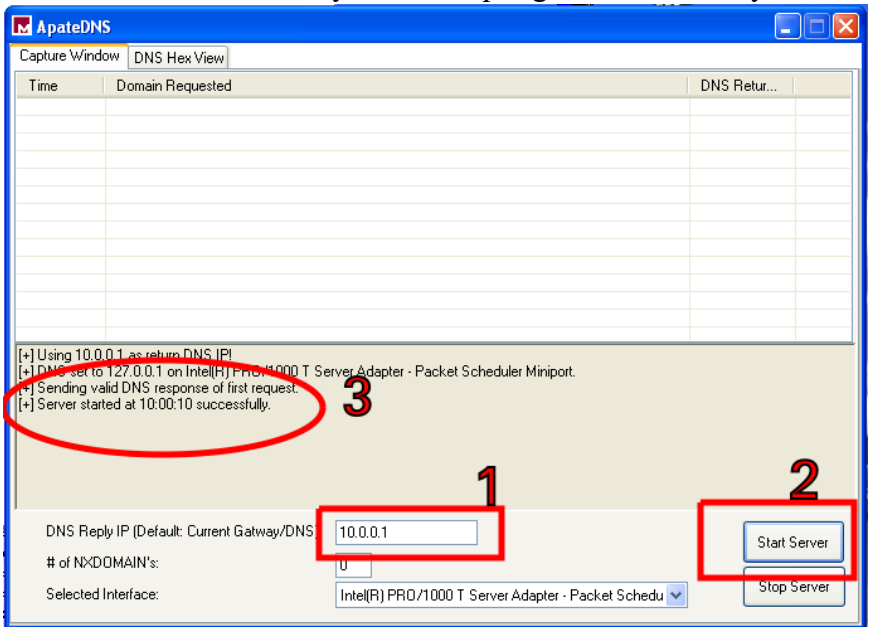
- Click on the FakeNet Network Machine in VirtualBox and press Start.
- In VirtualBox select the Ubuntu Analysis machine and press Start.
- Begin your analysis with some of the Basic Static Malware Analysis Tips that were covered in the previous tutorial. You'll want to know: whether the malware is packed; what sort of strings and imports it uses; if there are any additional binary files to be discovered.
- First, you'll want to open Regshot and save a "1st shot" of the current registry. Select the option to "Shot". After we run the malware we'll take the 2nd shot and compare the two.



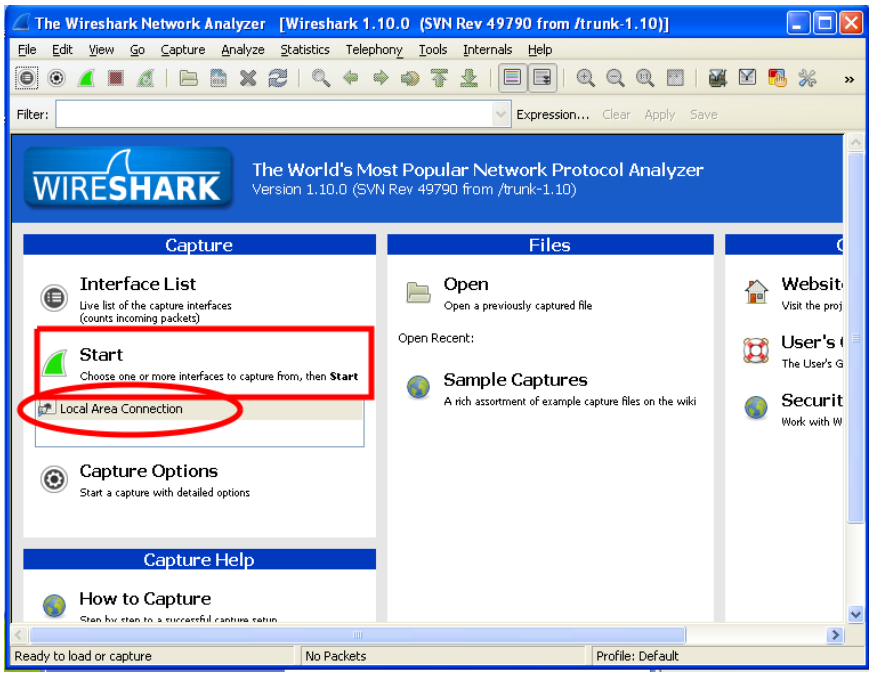
- Next, you'll start Process Monitor, procmon.exe. A bunch of current processes should show up on the screen right away. We only care about the new processes that occur after the malware has been run or executed. So, to start we'll clear all the current procmon events by selecting Edit > Clear Display



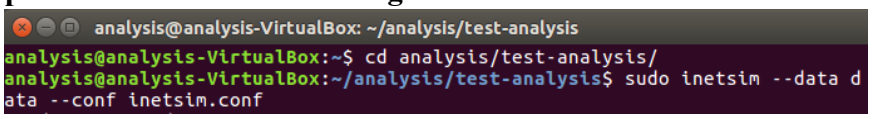
- Open Process Explorer
- Open ApateDNS. Fill out DNS Reply IP with the Ubuntu Analysis Machine IP address (10.0.0.1). Click Start Server. You should see a comment about stating that the server has started successfully in the dialogue box. ApateDNS functions as a tool which manages traffic on port 53 (DNS). The traffic occurring over this port translate web addresses to the IP address of the server where the information is housed on. By utilizing ApateDNS, we can find which websites, if any, are attempting to be reached by the malicious artifact.



- Open Wireshark. Make sure the Local Area Connection is selected (it should be the only option). Click Start.



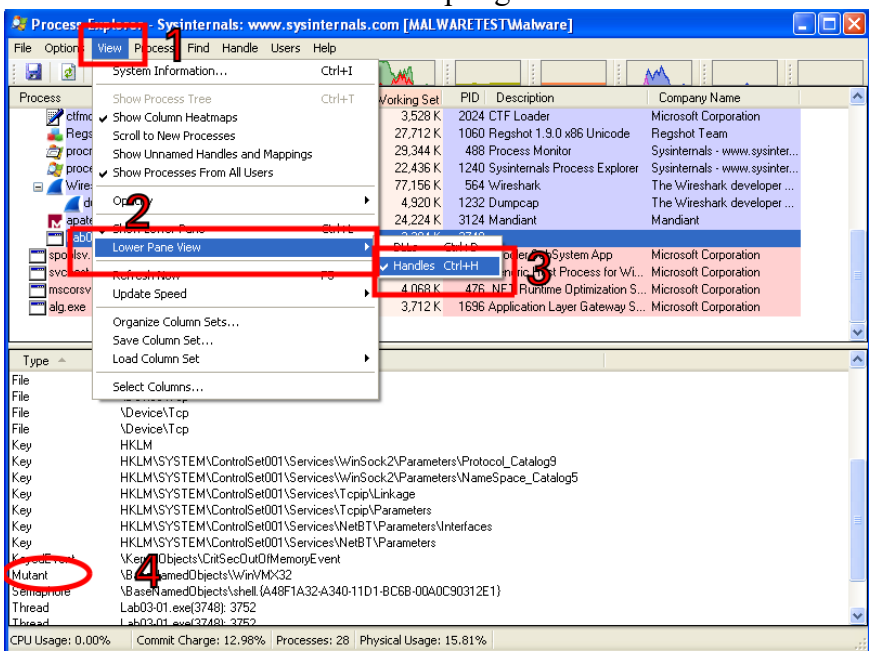
- Navigate to the Ubuntu Analysis machine. Open up terminal and type: **cd analysis/test-analysis** then type: **sudo inetsim -- data data --conf inetsim.conf** This will start monitoring traffic on all ports and produce a report of any attempt to use the network based on the malware. **Note: Don't have any other browser windows open during this time, as that will add traffic which is not the direct result of the malware. Additionally, Microsoft may attempt to reach out to the network, possibly for the purpose of updating. You can ignore those calls. We will be comparing our notes from ApateDNS with the inetsim logs later.**



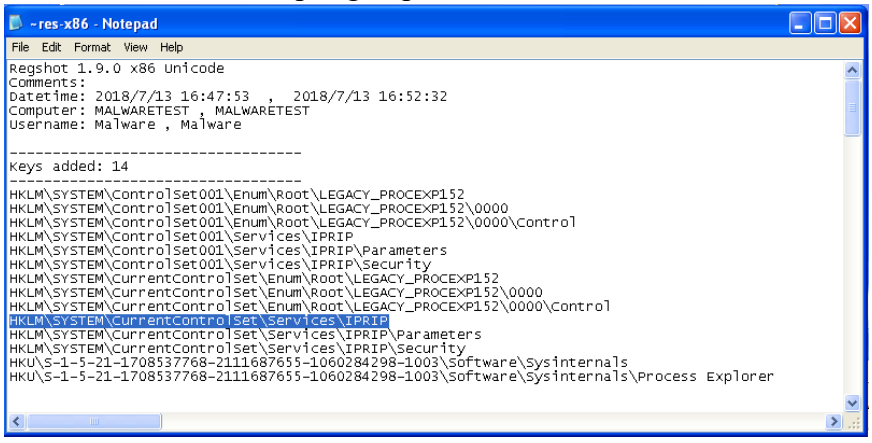
Note: Before continuing on to the next step, read through the following instructions. A few things should happen within quick succession, otherwise you may not have the best information to work with. Read through and make sure you know the steps you need to take before progressing.

- Make sure you have your Process Explorer window front and center before going to run the malware. If the malware is an executable (.exe) you should be able to simply double click to run it. If it is a .dll (Doubly Linked List) you may have to conduct additional research to determine how to install or run the malware. When you're ready (and after reviewing the following steps first), run the malware.
- Pay attention to the Process Explorer screen. Does the malware pop up as a process and then immediately disappear? Does the process not show up at all? Does the executable file disappear (delete itself) after being run? All of these are important observations that can contribute to knowing what to further research next. If, however, the malware shows up as a process and remains active, you can select it in Process Explorer. Then navigate to View > Lower Pane > Handles. This shows you an array of valuable information about

what activity the malware has conducted. Noting the file locations that the malware accesses, any events that occur, and whether any files, or mutants were created can give us hints as to what the malware was attempting to do.

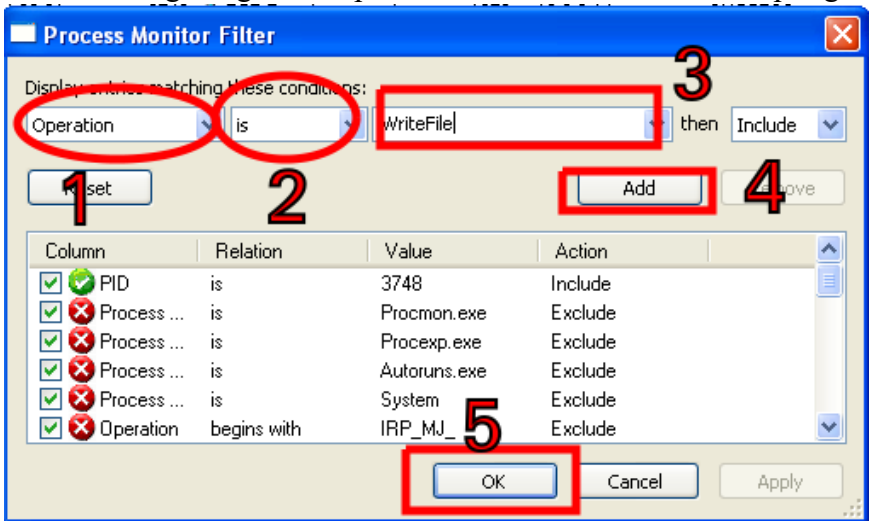


- After noting the behavior of the malware in Process Explorer, but before analyzing the malicious software, navigate back to RegShot, click 2nd Shot (make sure you're scanning your hard drive) and then click Compare. You can review the comparison later, but I'm going to explain how this is helpful now. This shows all the values, keys, and files that were added and modified. Modifications made in the services folder may show what sort of access the malware is attempting to procure.

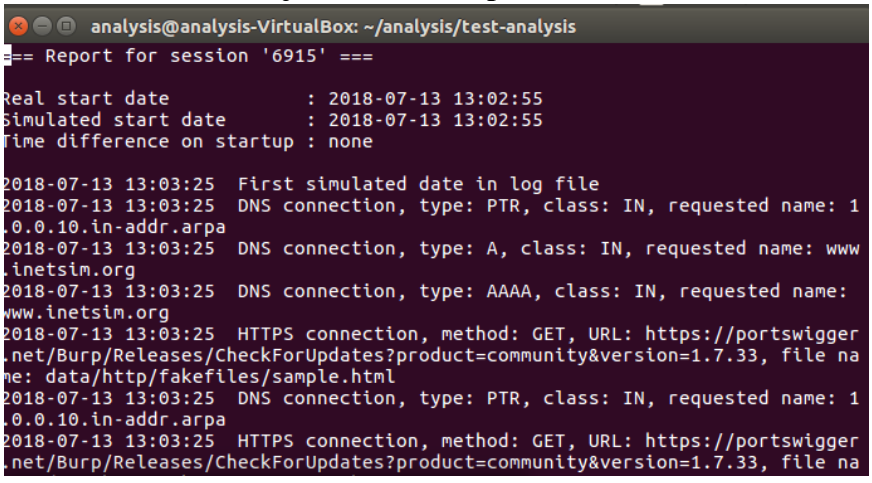


- Looking at Process Monitor we see a lot of activity, so we need to filter it out. Select Filter > Filter. Here you can filter by a whole bunch of different factors. To start it's best to filter by the Process Name (the name of the executable or DLL) or PID (you can find the PID through basic dynamic analysis, or looking at the Process Explorer line with the malicious process). You can add more filters, for example filtering by Operation: CreatFile, or even through excluding certain operations. It may be helpful to start with

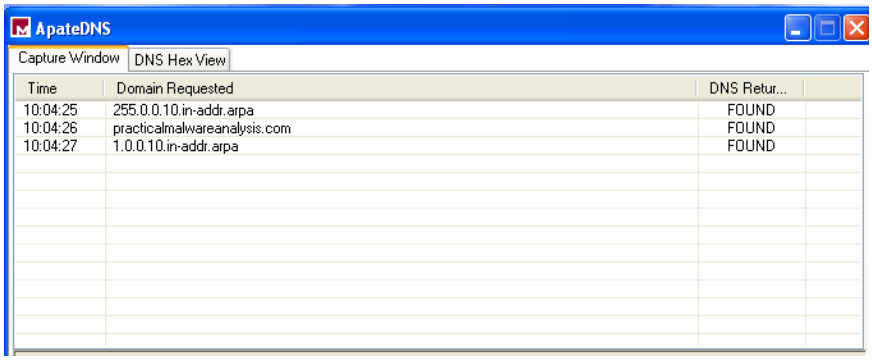
just filtering based on the PID, or process name to start, and slowly looking at a narrower scope after reviewing the general operations the malware was attempting to perform.



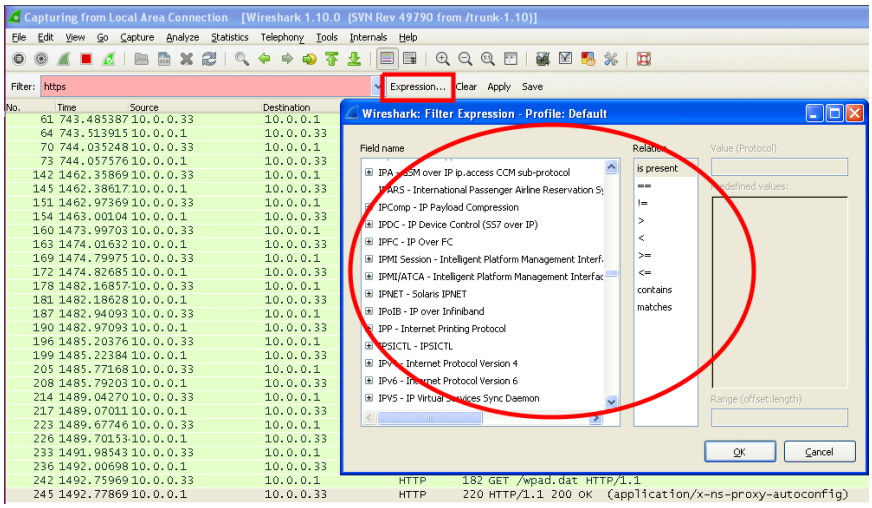
- Navigate back to the Ubuntu Analysis machine and the terminal which is operating the inetsim monitor. Type: **ctrl + C** this should result in the stopping the simulation. The terminal should also have information about where the inetsim report was saved to type: **sudo vi 'file location'** be sure to include the .txt at the end of the file name. This opens up the log in Vim, you can search for strings in this view by typing: **/'search term'** for example, if I am interested in whether my computer attempted traffic over port 80 (HTTP) I will type: **/HTTP** and if I want to scroll through all occurrences of traffic I can type: **n** to find the next location. If you want to search for something new, you can just type: / again and it will search your new term. **Note: if you want to search for HTTP traffic and exclude HTTPS just include a space after HTTP.**



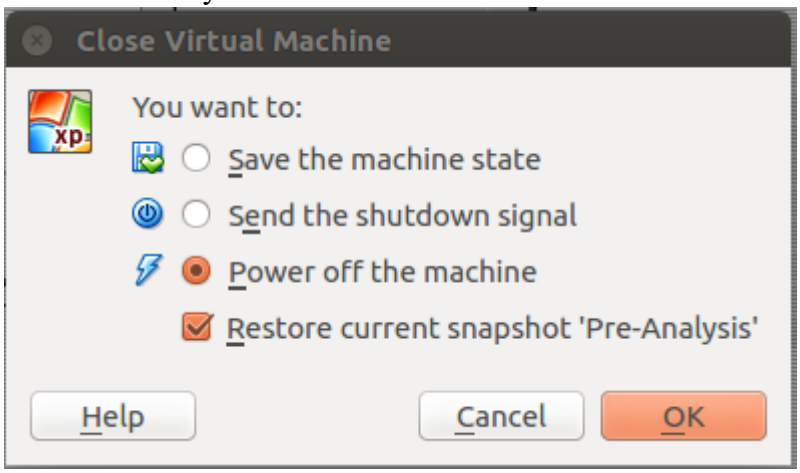
- Go back to the VM running the malware and open up ApateDNS. ApateDNS will provide you with the desired location of the malware. It may be used as a location to grab and then download additional malicious add-ons to the machine, or it may be reporting information to that location, or setting up a connection between that server and your machine.



- Open Wireshark and review all traffic occurring on the LAN network. You can filter by http traffic, tcp, dns or any other filter expression which you can find by clicking on the "Expression" tab. Wireshark gives you more detail about the packet transfer happening over the virtual network.



- Whenever you're done with your analysis, you should close it out by "x"-ing out of the window and selecting to power off the machine and restore the current snapshot. You should do this for both of your Virtual Machines



SUMMARY

These tips are organized in a way to provide some structure to beginning your malware analysis. As you develop more skills and obtain a broader knowledge of how to navigate to deeper understanding of malicious software, you'll be able to use a more intuitive sense of how to review and further delve into malware analysis.

APPENDIX C (SOP FOR ACQUISITION OF MALWARE SAMPLE)

	University of North Carolina Wilmington Information Technology Services	Standard Operating Procedure
UNCW	IT Security	010

MALWARE ACQUISITION AND TRANSFER

Outline the proper steps to obtaining malicious artifacts, transferring those artifacts to an isolated virtual machine, and restoring the virtual machine to its original clean state.

I. General Information

Malware, or malicious software, is a program that infiltrates a computer or system and has the capacity to cause harm to a system or network. Malware may affect the integrity of data, may monitor your activity and share that with an external entity, or can limit the functionality of your hardware. One way to limit the effectiveness of malware is to conduct a proper analysis of the software. By doing this, malware analysts are able to see which vulnerabilities within the system are being targeted and can recommend solutions to assist in limiting the possibility of having a similar breach occur in the future.

II. Scope

This procedure applies to any malware acquisition from an external source and must be conducted prior to any malware analysis.

III. Applicable Definitions

A. Virtual Machine (VM)

A virtual computer which can be configured to conduct processes in an isolated environment, minimizing risk to the home network.

B. VirtualBox

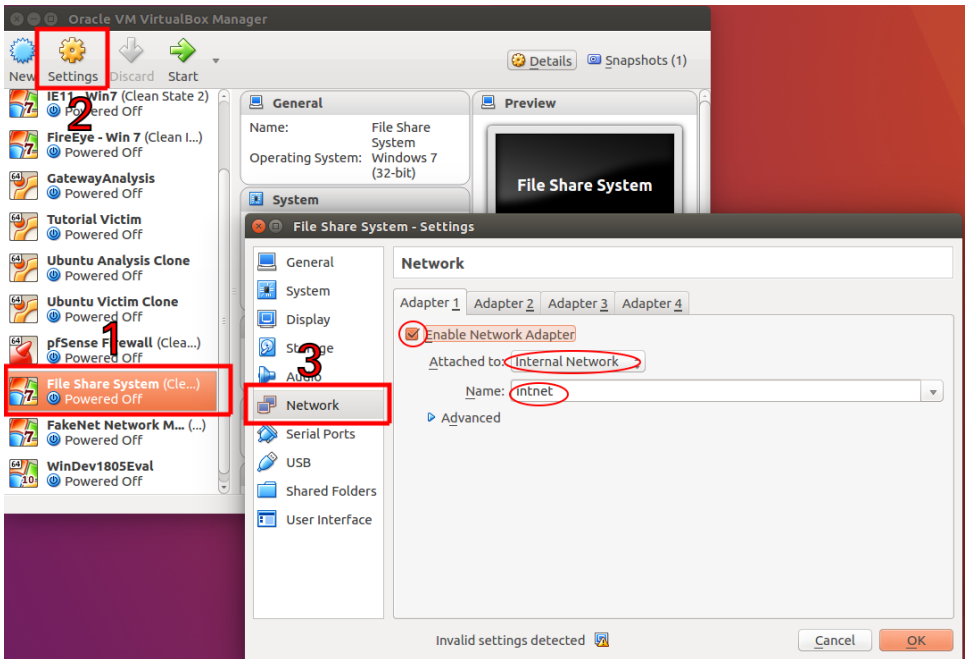
Software that allows you to create, restore, and destroy VMs and virtual networks. The version being used in this document is 5.1.34_Ubuntu and all instructions are compatible with this version of VirtualBox.

IV. Procedures

***If at any time the transfer or acquisition process is interrupted or must be stopped before completion of all steps, OR there is a failure of success of the firewall or shared folders, OR there is a failure to complete one of the steps, OR you notice suspicious or unexpected behavior from the host machine user MUST destroy all progress and return both the File Share System and FakeNet VMs to their original clean state before leaving the workstation.**

**** If at any time you notice suspicious or unexpected behavior from the host machine contact the endpoint administrator to have the host machine reimaged. DO NOT CONNECT THE HOST MACHINE TO THE UNCW NETWORK OR THE INTERNET. Reset all UNCW accounts used on the host machine.**

A. Acquisition of a malicious artifact from an external source online
   1. Open VirtualBox software
   2. Select the VM named “File Share System” and open Settings > Network
      a) On the Adapter 1 tab double check that “Enable Network Adapter” is selected and that it is attached to an Internal Network named “intent”



      b) Navigate to Adapter 2 tab and deselect the “Enable Network Adapter” option if it is checked. If it is already unchecked, leave it
      c) Check to make sure Adapter 3 and 4 are also unchecked
      d) Click OK
   3. Select the VM named “pfSense Firewall” and open Settings > Network
      a) Double check that Adapter 1 is selected for “Enable Network Adapter” and that it is attached to: NAT
      b) Click on the Adapter 2 tab and ensure that the “Enable Network Adapter” option is selected, that it is attached to an Internal Network named “intent”
      c) Check Adapter 2 and 3 to make sure that both are NOT enabled
      d) Click OK
   4. Select “pfSense Firewall” and click “Start”
   5. Select “File Share System” and click “Start”
   6. Wait until both VMs have completed booting and loading

7. Using the Windows interface that opened after starting the “File Share System” open a web browser and navigate to https://172.16.1.1

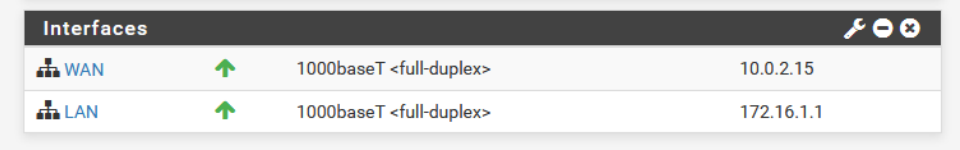
a) If a window pops up informing you of an insecure or untrusted network, select the option to continue on to the site

b) Log in to the pfSense portal

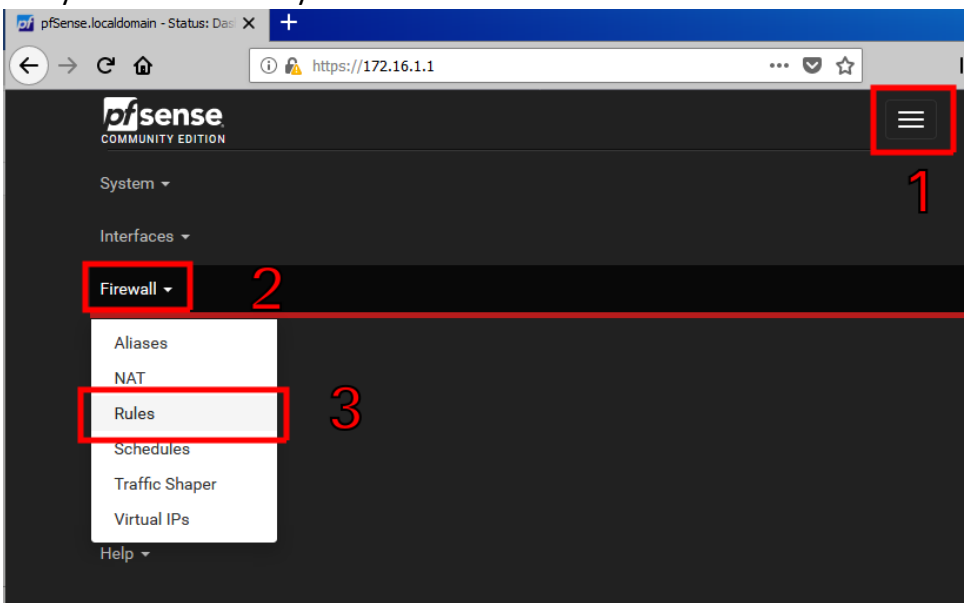
(1) Username: admin

(2) Password: malware123

c) After logging in, make sure that both WAN and LAN interfaces are up



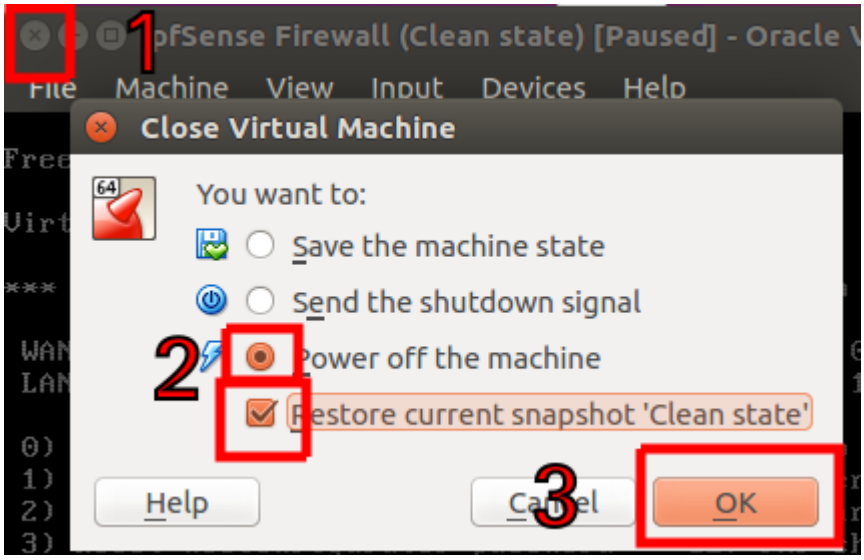
d) Navigate to rules page, then click the LAN tab and ensure that the “DNS Config” rule is listed at the top of the page and the “Malware Analysis” rule is directly under it and that all other rules are disabled



8. Navigate to a pre-approved website in the “File Share System’s” browser

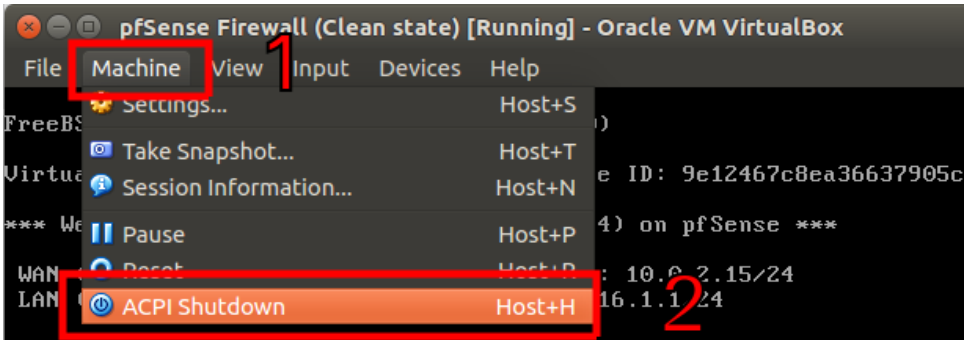
a) Initiate the download of the malicious artifact, choose to select the destination for the download to the folder named “Shared MW Folder” (it should download to this location by default) **DO NOT OPEN THE DOWNLOAD AT THIS TIME! – IF OPENING OCCURS (OR ANY OTHER SUSPICOIUS ACTIVITY OR BEHAVIOR) EXIT OUT OF THE VM WINDOW VIA THE ‘X’ AND SELECT “POWER OFF MACHINE” AND CHECK THE**

“RESTORE CURRENT SNAPSHOT ‘CLEAN STATE’” IMMEDIATELY



b) After download has occurred, and no suspicious behavior is observed, and the file or document has **NOT** been opened, power off the machine by clicking Start > Shut down

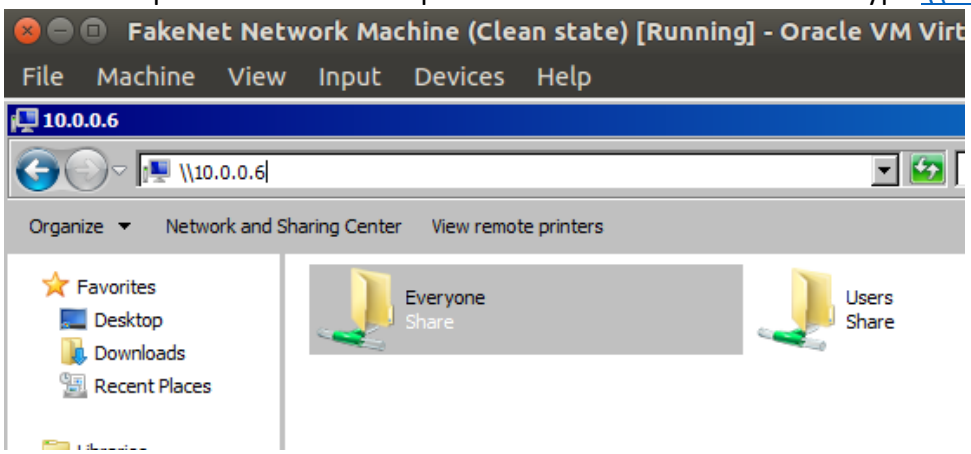
9. Power off the “pfSense Firewall” by selecting Machine > ACPI Shutdown



B. Transfer of malicious artifact to “FakeNet Network Machine” VM

1. Select “File Share System” and select Settings > Network
   a) Deselect “Enable Network Adapter” in Adapter 1 tab
   b) Select “Enable Network Adapter” in Adapter 2 tab
   c) Ensure that the Adapter 2 tab is attached to an Internal Network named “virtual-malware-network”
   d) Click OK
2. Select “FakeNet Network Machine” and select Settings > Network
   a) Make sure that Adapter 1 is enabled and attached to an Internal Network named, “virtual-malware-network”
   b) Check Adapters 2, 3, and 4 to make sure they are NOT enabled
   c) Click OK
3. **Disconnect the host machine from the internet**
4. Select “File Share System” and click Start
5. Select “FakeNet Network Machine” and click Start

6. In the window displaying the Windows interface of the “FakeNet Network Machine” open Windows File Explorer and in the file location bar type \\10.0.0.6



a) Open the folder Everyone > IEUser > Desktop > Shared Malware Folder

b) Grab the file containing the malicious artifact and drag it to a location on the Desktop

c) Right click on the file, still located in the “Shared Malware Folder” and delete the file

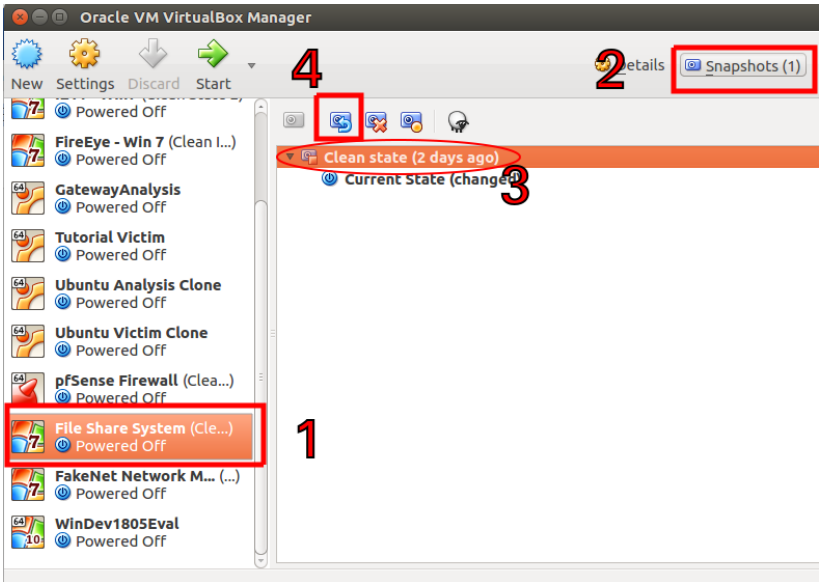
7. Immediately power down the “File Share System” by clicking the ‘X’ in the top corner of the window and selecting “Power off the machine” option while checking the “Restore current snapshot ‘Clean state’”

8. In VirtualBox immediately restore the “File Share System” to a clean state

a) Highlight the “File Share System” VM and click on Snapshots

b) Click the “Clean state” snapshot and then click the restore icon

c) Deselect the “Create a snapshot of the current machine state” option before clicking Restore



9. Refer to Standard Operating Procedure #xx for instructions regarding analyzing malware **THROUGHOUT THE ANALYSIS PROCESS YOU MUST REMAIN**

AT THE WORKSTATION AT ALL TIMES. YOU MUST CONTINUE TO MONITOR THE VM AND HOST MACHINES FOR SUSPICIOUS BEHAVIOR. IF AT ANY TIME YOU NOTICE SUSPICIOUS ACTIVITY, OR YOU MUST LEAVE YOUR WORK STATION, CLOSE OUT OF THE VM BY SAVING MACHINE STATE AND RESTORE PREVIOUS CLEAN STATE SNAPSHOTS. IF HOST MACHINE HAS BEEN AFFECTED BY MALWARE CONTACT AN ADMINISTRATOR AND REIMAGE HOST MACHINE. DO NOT CONNECT HOST MACHINE TO THE UNCW NETWORK OR INTERNET.

10. Upon completion of analysis, or needing to stop the analysis process for any reason, close down the "FakeNet Network Machine" by clicking the "X" in the top corner and selecting "Save machine state"
11. In VirtualBox immediately restore the "File Share System to a clean state
    a) Highlight the "FakeNet Network Machine" VM and click on Snapshots
    b) Click the "Clean state" snapshot and then click the restore icon
    c) Deselect the "Create a snapshot of the current machine state" option before clicking Restore

V. References to Other Applicable SOPs

Title	SOP #
Analysis of Malicious Software in Isolated Environment	xxxx

VI. Responsibilities

Title	Responsibility
Information Security Administrator	Operations

VII. Revisions

Date	Description
06/25/2018	Creation of SOP document

APPENDIX D (SOP FOR ANALYSIS OF MALWARE)

	University of North Carolina Wilmington Information Technology Services	Standard Operating Procedure
UNCW	IT Security	011

MALWARE ANALYSIS

Outline the proper steps to conduct a preliminary analysis of malicious artifacts on an isolated virtual network while maintaining network segregation and security.

VI. General Information

Malware, or malicious software, is a program that infiltrates a computer or system and has the capacity to cause harm to a system or network. Malware may affect the integrity of data, may monitor your activity and share that with an external entity, or can limit the functionality of your hardware. One way to limit the effectiveness of malware is to conduct a proper analysis of the software. By doing this, malware analysts are able to see which vulnerabilities within the system are being targeted and can recommend solutions to assist in limiting the possibility of having a similar breach occur in the future.

VII. Scope

This procedure applies to any malware which has been properly acquired and transferred to the appropriate virtual machine as defined in SOP for Malware Acquisition and Transfer.

VIII. Applicable Definitions

A. Portable Executable (PE)

Includes valuable information which can be read by Windows OS Loader and can be reviewed with PE analysis software.

B. Virtual Machine (VM)

A virtual computer which can be configured to conduct processes in an isolated environment, minimizing risk to the home network.

C. ApateDNS

Software which monitors traffic over port 53 and collects data on URL information being sent over that port.

D. Wireshark

Software which monitors packets being transferred over a network.

E. PEview

Software responsible for looking at details within a PE.

F. PEid

Software responsible for looking at details within a PE.

G. Process Monitor

Software which shows every operation occurring on the machine. Users can use filters to view only operations which are relevant to a particular process or may filter based on other criteria.

H. Process Explorer

Software that provides detailed analysis of processes and key functions and operations occurring within the processes.

I. Burp Community Suite

Software used for simulating online security certificates from the Certificate Authority (CA).

J. iNetSim

Software used to monitor traffic on all ports within a network.

K. Dependency Walker

Provides detailed explanation of DLL functions and imports.

L. Resource Hacker

Software which allows users to access and analyze extracted binary files within malware, allowing these files to be exported and further reviewed.

M. Dynamically Linked Lists (DLL)

Used to grab libraries already stored on the host machine and runs the required libraries in their home location without having to load them into the executable.

N. VirtualBox

Software that allows you to create, restore, and destroy VMs and virtual networks. The version being used in this document is 5.1.34_Ubuntu and all instructions are compatible with this version of VirtualBox.

IX. Procedures

***If at any time during the analysis of the malware the analyzer must break or take leave of the malware analysis machine they must close out of any open virtual machines through clicking the "x" in the top corner and selecting "Power off machine" and checking the "Restore previous snapshot" option.**

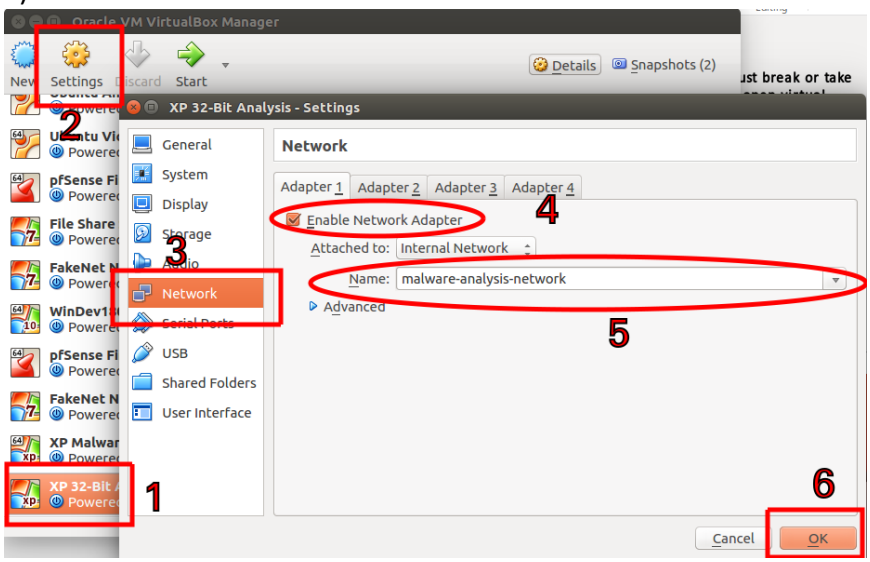
**** Familiarize yourself with all steps before beginning the analysis process.**

A. Basic Static Analysis

1. Before conducting any malware, analysis turn off all networking capabilities of the host machine
2. Open VirtualBox software
3. Select the VM victim that matches the operating system the malware is designed to target and click the "Settings" option on the top bar
    a) Navigate to the Network tab
    b) Confirm that the Adapter 1 is "Enabled," that it is attached to an "Internal Network," and that the name of that network is "malware-analysis-network"

c) Ensure that Adapter 2, 3, and 4 are **NOT** "Enabled"

d) Click OK

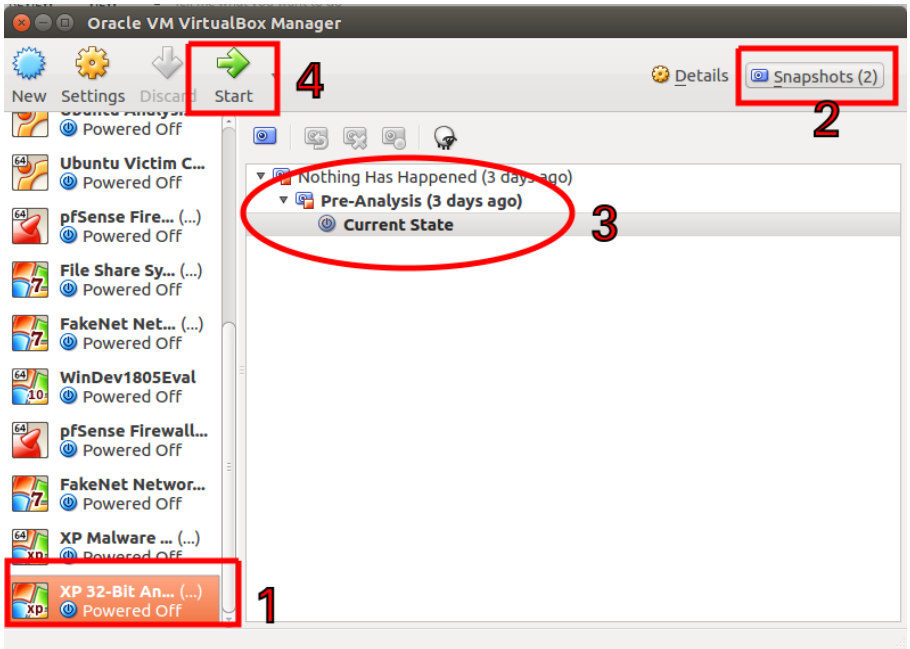


4. With your victim VM still selected, navigate to the Snapshots section

a) Ensure that the "Current state" is unchanged from the Pre-Analysis snapshot

b) If it is changed, boot from the Pre-Analysis snapshot by selecting that option

c) Click the "Start" icon in the top tool bar



5. Select the VM named "Ubuntu Analysis Machine" and open Settings > Network

a) Double check that Adapter 1 is selected for "Enable Network Adapter" and that it is attached to an "Internal Network" named: "malware-analysis-network"

b) Click on the Adapter 2, 3 and 4 and ensure that all of them are **NOT** "Enabled:

c) Click OK

d) Navigate to the snapshot section and select the "Pre-Analysis" snapshot

e) Click Start

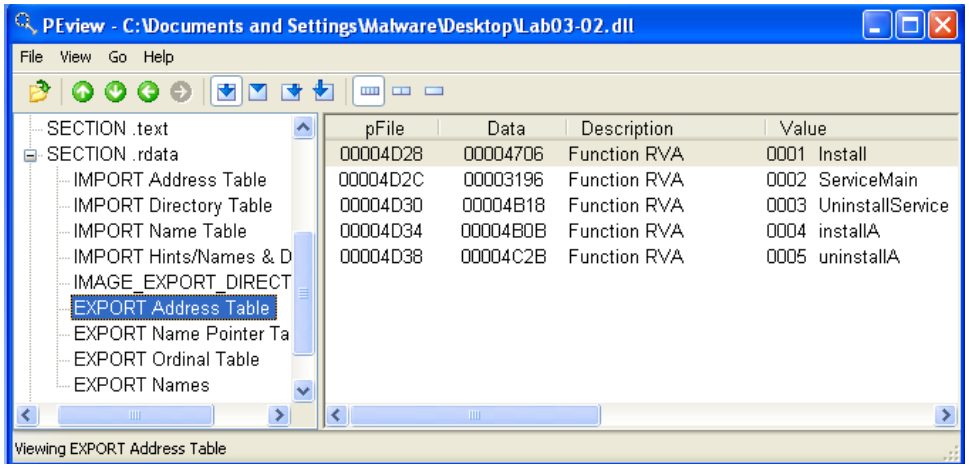
6. Open PEView

a) Navigate to the location of the malware you wish to analyze

b) Select the IMAGE_SECTION_HEADER option and review the virtual size vs the raw data size

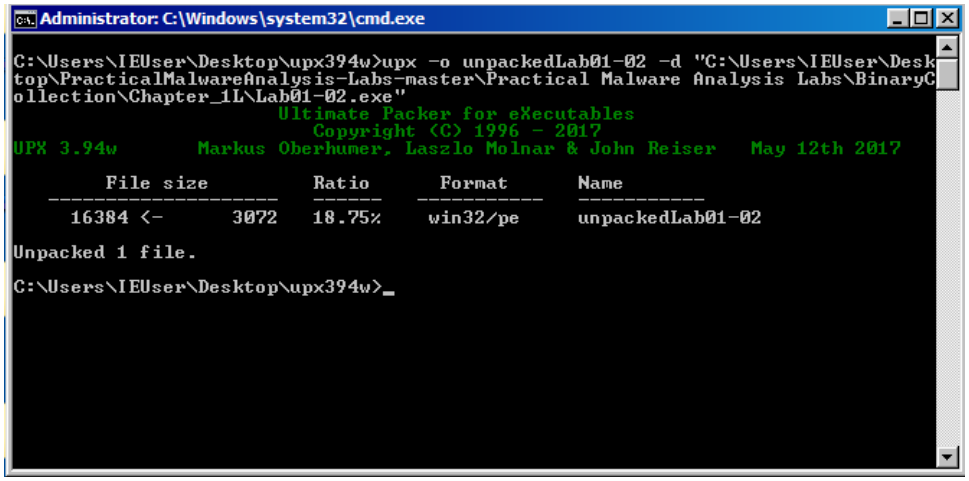
(1) If these two sizes are similar, the malware is most likely not packed, otherwise it is probably packed

c) If possible, expand the SECTION option to review imports, select the imports to review any processes being imported



7. Open PEid to see what the value was wrapped or compiled in (Note: UPX wrapping can be unwrapped by downloading (https://upx.github.io/) via the same process as the SOP covering Malware Acquisition and Transfer)

a) If unpacking via UPX you can open Command Prompt, navigate to the directory containing the UPX download and type: **upx -o newFileName -d originalFileName**



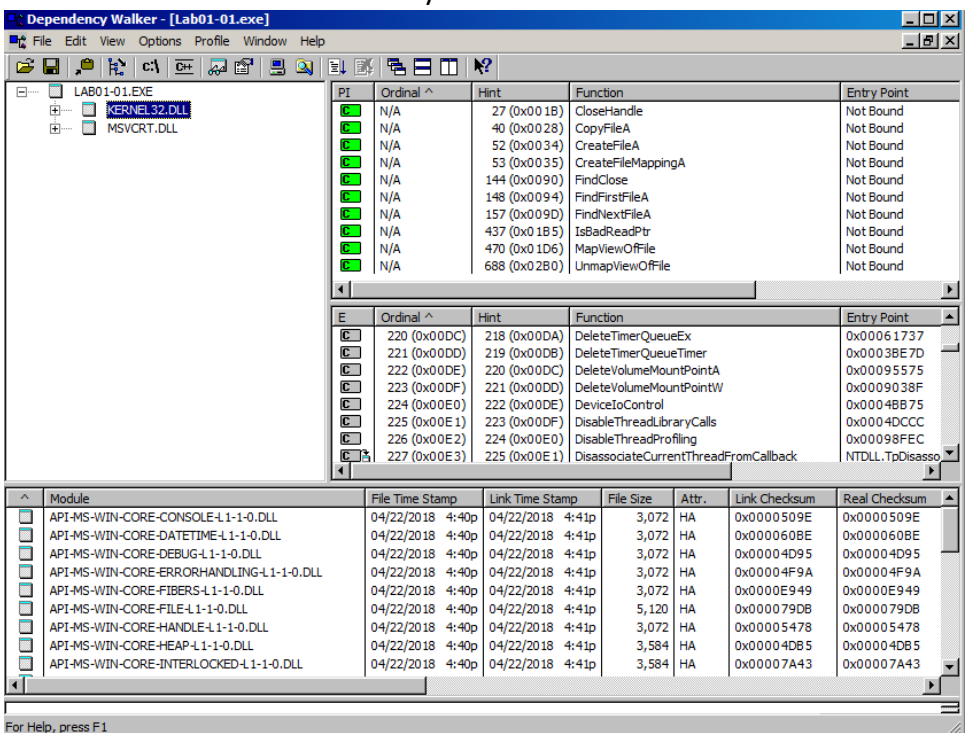
8. Open Command Prompt and navigate to the Desktop (or the location of the "strings.exe" file

a) Type: **strings** in the command prompt, accept any terms and conditions

b) Type: **strings fileName**

c) Review all strings and imports, take special note of WriteFile CreateFile, URLs, Drivers, access to network services, or other areas of access the malware attempts to import

9. Open Dependency Walker (Depends), open the malicious artifact, review the functions associated with any DLLs inherent in the malware

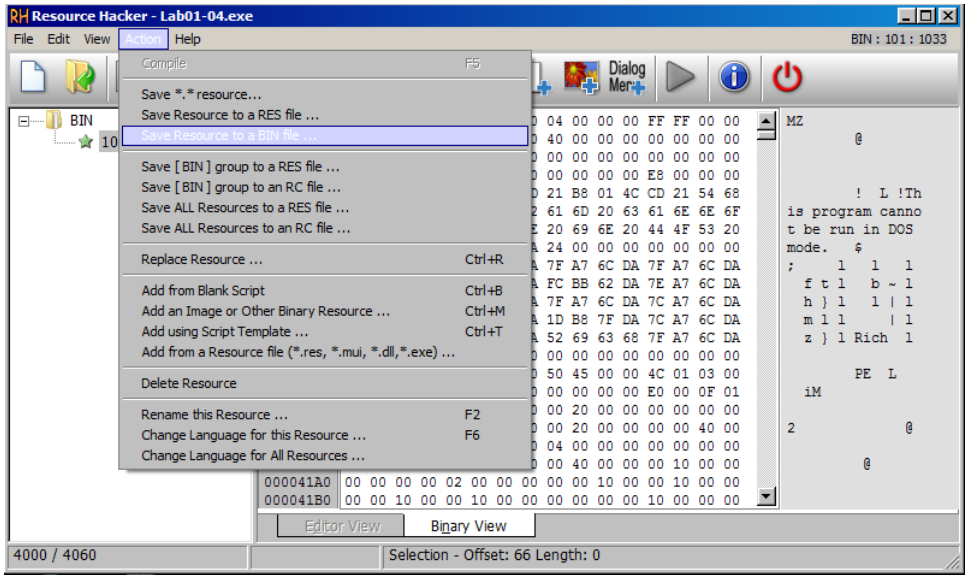


10. Open Resource Hacker, navigate to the file location of the malicious artifact, there may or may not be an extracted binary file for review

a) If there is a binary file for review select Action > Save as BIN file

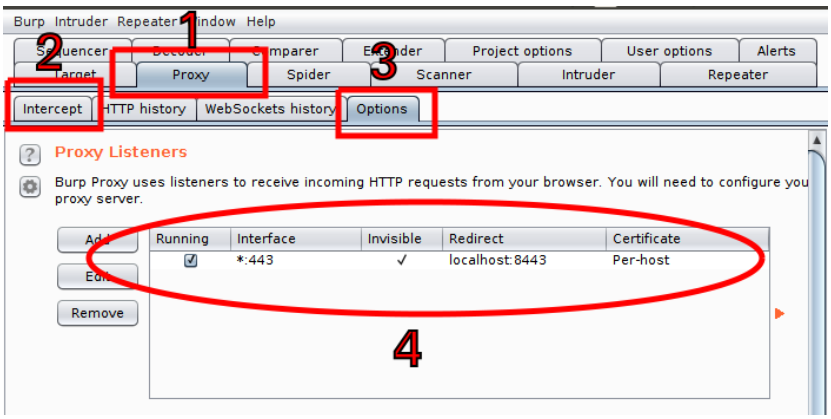
b) Conduct an analysis on this file by opening Dependency Walker and reviewing imports associated with the DLLs within the extracted

binary file



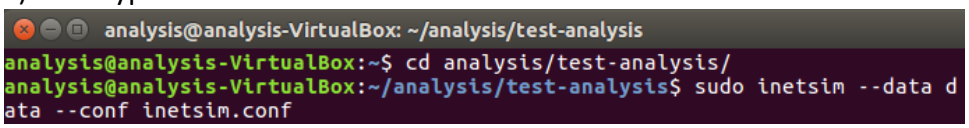
B. Basic Dynamic Analysis

1. Open Process Explorer
2. Navigate to the Ubuntu Analysis VM
   a) Open a terminal window and type: **cd BurpSuiteCommunity**
   b) Type: **sudo /home/analysis/BurpSuiteCommunity/BurpSuiteCommunity**
   c) The Burp program should open a GUI program
      (1) Select “Temporary Project” and click Next
      (2) Select “Load from configuration file” select the burpSettings.json file and click “Start Burp”
      (3) Click the “Proxy” tab
      (4) Deselect the “Intercept is on” button so that the button reads “Intercept is off
      (5) Navigate to the sub-tab “Options”
         (a) Ensure that the “Running” checkbox is enabled
         (b) That Interface*.443 is selected, that it is invisible, that it redirects to localhost:8443 and the Certificate is Per-host



d) Open a new terminal window and type: **cd analysis/test-analysis**

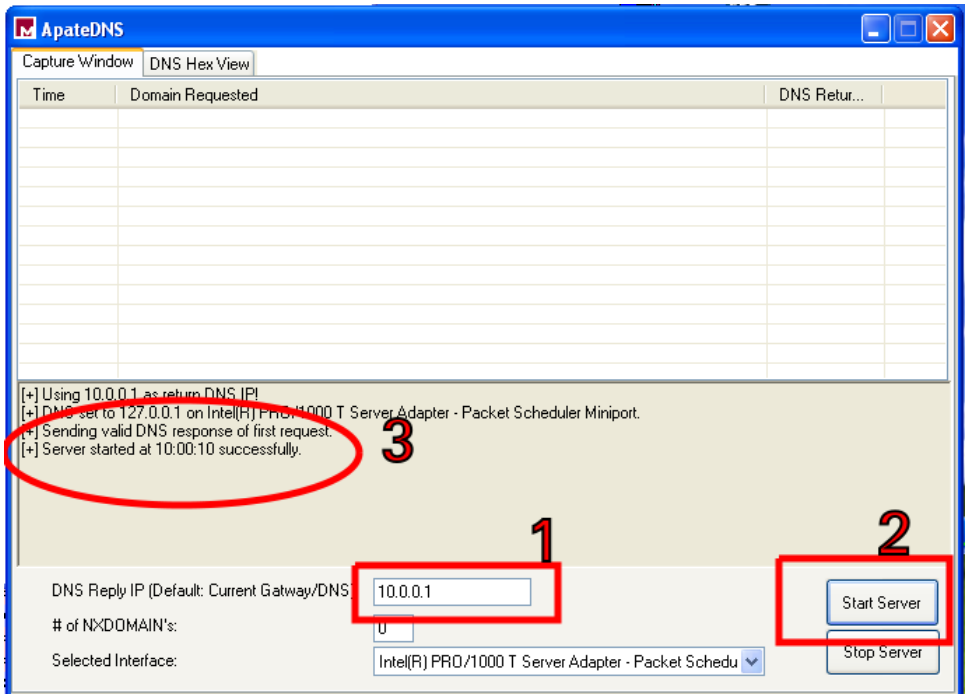
e) Type: **sudo inetsim --data data --conf inetsim.conf**



3. Navigate back to the victim VM

4. Open ApateDNS and type 10.0.0.1 into the DNS reply IP and click “Start server”

a) Ensure that the server started successfully



5. Open Wireshark

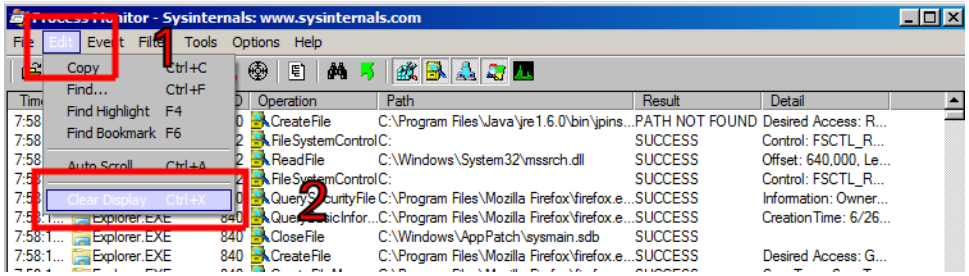
a) Select the Local Area Network and click the “Start” button

6. Open RegShot, select the “Scan dir” option and ensure that it is grabbing the entirety of the hard drive

a) Select 1st Shot and then select the “Shot” option

7. Open Process Monitor (procmon)

a) Click Edit > Clear screen



8. Make sure that Process Explorer is open and visible

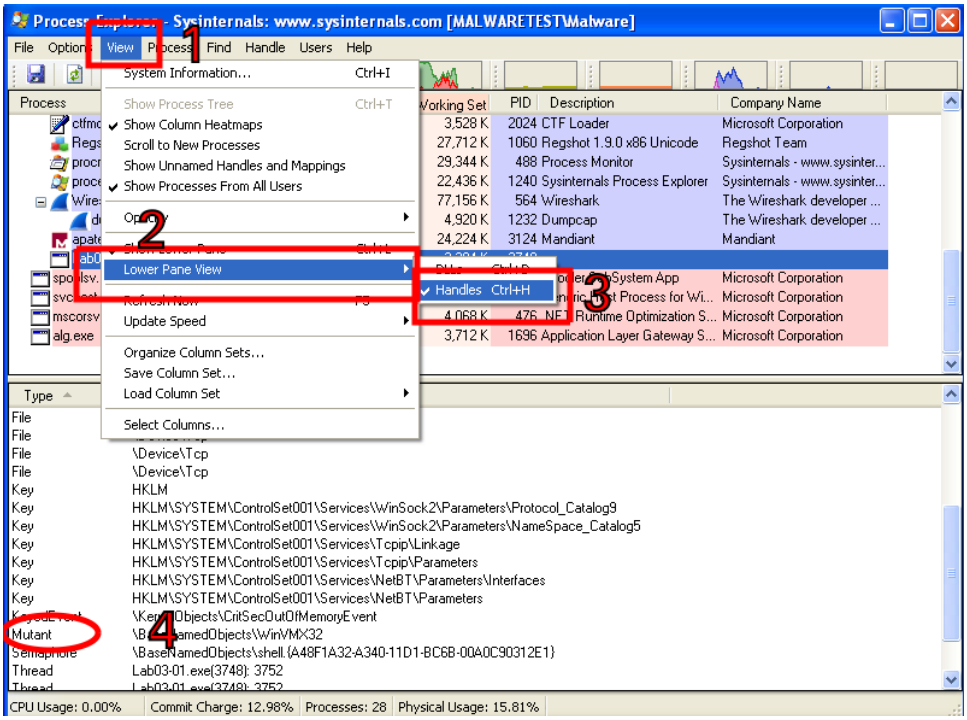
9. Run the malware **Note: Ensure that the host machine is not connected to the UNCW network, or any external network before running malware**

a) Look at the Process Explorer and note the behavior of the process

(1) If it appears as though the process starts, and remains running for more than 10 seconds, navigate to RegShot and click 2nd Shot and select “Shot” **Note: This step should be completed quickly so as not to collect unnecessary data**

(2) You can click “Compare” once the scan has completed to review added or altered files, keys, and values

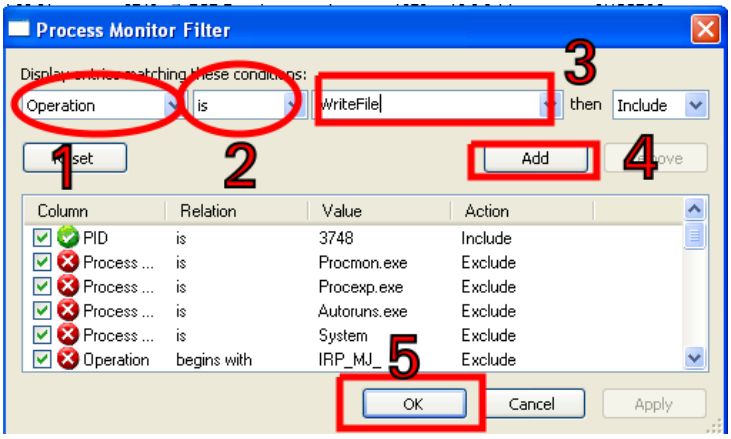
(3) Go back to Process Explorer and select the process and select View > Lower Pane View > Handles. Review any Mutant files, accessed Keys/Files



(4) Take note if the file seems to fail to run at all, if it runs and stops its process, or if the file runs and deletes itself

b) Open Process Monitor and create a filter by selecting Filter > Filter with the (“Process Name” “is” “malwareFileName”), select Add, then select OK

(1) Review all the processes, and if desired, add additional filters by selecting Filter > Filter



(2) Take note of operations which access services, instantiates TCP calls, creates or writes to a file, or seems to add or adjust anything additional

c) Open ApateDNS and review any URLs which the malware attempted to access

d) Open Wireshark and review the packets attempted to be transferred through the virtual network

e) Navigate to the Ubuntu Analysis Machine

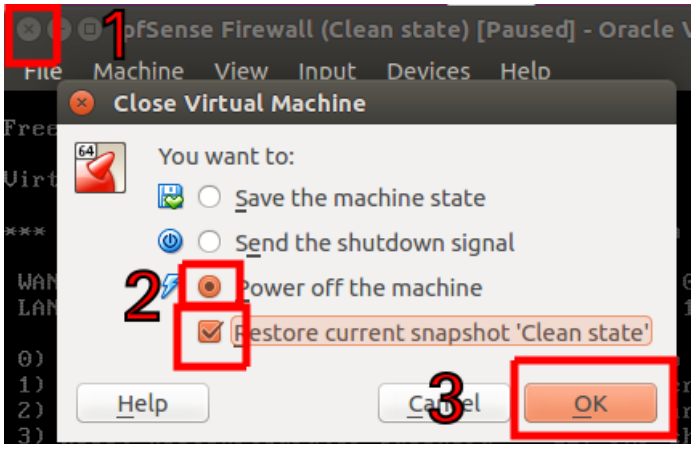
(1) Select the terminal window which is running iNetSim

(2) Type: **ctrl + C** take note of the output which explains where the report which was just generated is located

(3) Type: **sudo vi 'fileLocation/filename.txt'**

(4) Review which ports were accessed

10. In the victim VM, click the "X" in the top corner of the window and close out of the VM, select "Power off the machine" and check the box marked "Restore previous snapshot"



11. Conduct the same process for the Ubuntu Analysis machine

12. Review processes in the host machine and ensure it has been unaffected by the analysis

VI. References to Other Applicable SOPs

Title	SOP #
MALWARE ACQUISITION AND TRANSFER	xxxx

VII. Responsibilities

Title	Responsibility
Information Security Administrator	Operations

VIII. Revisions

Date	Description
07/15/2018	Creation of SOP document

APPENDIX E (SOP FOR UPDATING NETWORK CONFIGURATION FOR MALWARE ANALYSIS)

	University of North Carolina Wilmington Information Technology Services	Standard Operating Procedure
	IT Security	012

UPDATING NETWORK CONFIGURATION FOR MALWARE ANALYSIS

Outline the proper steps to update, change, or add network adapter settings on the virtual or host machine associated with conducting malware analysis.

X. General Information

Malware, or malicious software, is a program that infiltrates a computer or system and has the capacity to cause harm to a system or network. Malware may affect the integrity of data, may monitor your activity and share that with an external entity, or can limit the functionality of your hardware. One way to limit the effectiveness of malware is to conduct a proper analysis of the software. It is important to maintain network segregation and maintain all documentation to ensure security throughout the analysis process. Providing network segregation limits the scope of potentially affected devices in the case of malware escape.

XI. Scope

This procedure applies to any and all network settings related to the host machine managing the malware analysis environment, or any changes in the network settings (apart from enabling or disabling adapters) which occur within a virtual machine.

XII. Applicable Definitions

A. Virtual Machine (VM)

A virtual computer which can be configured to conduct processes in an isolated environment, minimizing risk to the home network.

B. VirtualBox

Software that allows you to create, restore, and destroy VMs and virtual networks. The version being used in this document is 5.1.34_Ubuntu and all instructions are compatible with this version of VirtualBox.

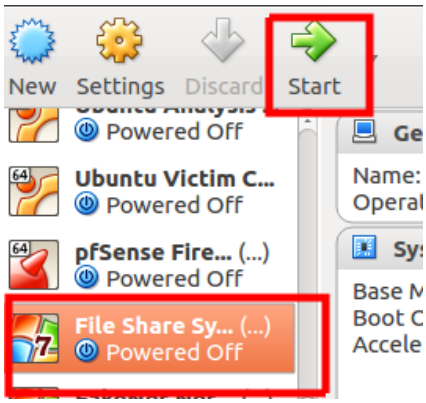
XIII. Procedures

***Changes to network configuration must be to fulfill a specific purpose and must be approved by an Information Security supervisor. Changes must maintain network segregation, and all considered security controls. Changes must be made to SOP 010 and 011 in order to provide**

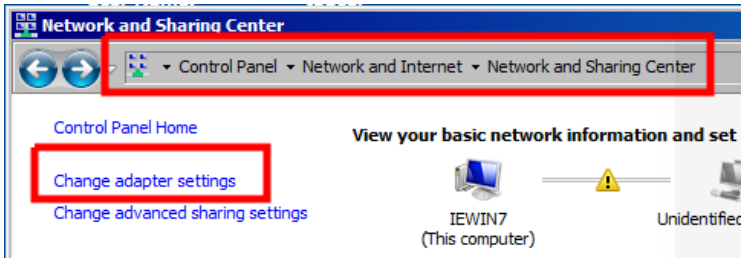
accurate information for network settings checks to settings before beginning the analysis or acquisition and transfer processes.

C. Updating Network Settings in a Windows Based VM

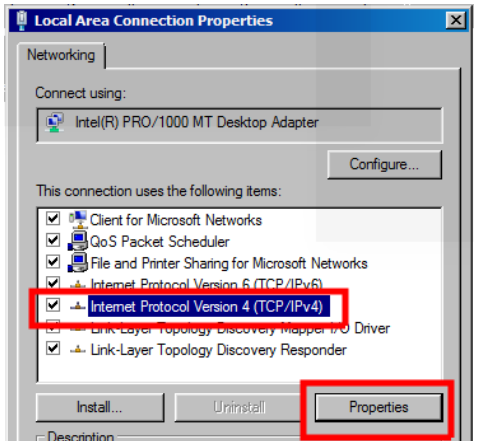
1. Open VirtualBox
2. Select the Windows machine you want to adjust network settings for
3. Press start



4. Navigate to Control Panel > Network and Internet > Network and Sharing Center and click the "Change adapter settings on the left tool bar



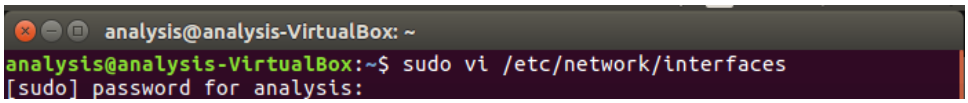
5. Right click the adapter you want to update and select properties
6. Select the Internet Protocol Version 4 option and click on Properties



7. Make the relevant changes and click "OK"
8. **Note: Changes must ensure proper network segregation and have been approved by an Information Security supervisor**
9. Update SOP: 010 and SOP: 011 to reflect updated network security checks

D. Updating Network Settings in an Ubuntu VM or Host Machine

1. Open VirtualBox
2. Select the Ubuntu operating system you want to change network settings on
3. Click Start
4. Open up a terminal session
5. Type: sudo vi /etc/network/interfaces



6. Password for analysis is: analysis
7. Change the network settings as have been approved
8. Type: :q
9. **Note: Changes must ensure proper network segregation and have been approved by an Information Security supervisor**
10. Update SOP: 010 and SOP: 011 to reflect updated network security checks

VI. References to Other Applicable SOPs

Title	SOP #
MALWARE ACQUISITION AND TRANSFER	010
MALWARE ANALYSIS	011

VII. Responsibilities

Title	Responsibility
Information Security Administrator	Operations

IX. Revisions

Date	Description
07/19/2018	Creation of SOP document

WORKS CITED

[1] Bureau of Labor Statistics, "Information Security Analyst: Occupational Outlook Handbook," 9 March 2018. [Online]. Available: https://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm. [Accessed 5 April 2018].

[2] A. Kujawa, "So You Want To Be A Malware Analyst," Malwarebytes Labs, 22 November 2017. [Online]. Available: https://blog.malwarebytes.com/security-world/2012/09/so-you-want-to-be-a-malware-analyst/. [Accessed 15 April 2018].

[3] M. Sikorski and A. Honig, Practical Malware Analysis : A Hands-On Guide to Dissecting Malicious Software, No Starch Press, 2012.

[4] M. Branscombe, "Practical Malware Analysis: Book review," ZDNet UK Book Reviews, 29 August 2012. [Online]. Available: https://www.zdnet.com/article/practical-malware-analysis-book-review/. [Accessed 15 April 2018].

[5] C. Tafani-Dereeper, "Set up your own malware analysis lab with VirtualBox, INetSim and Burp," 5 June 2017. [Online]. Available: https://blog.christophetd.fr/malware-analysis-lab-with-virtualbox-inetsim-and-burp/. [Accessed 22 January 2018].

[6] N. Avital, "Addressing the Threat of Cryptomining Malware," *Bitcoin Magazine,* 17 July 2018.

[7] D. K. Mulligan and F. B. Schneider, "Doctrine for Cybersecurity," *Daedalus,* vol. 140, no. 4, p. 70+, 2011.

[8] E. M. Hutchins, M. J. Cloppert and R. M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," in *6th Annual International Conference on Information Warfare and Security*, Washington, 2010.

[9] A. Hahn, R. Thomas, I. Lozano and A. Cardenas, "A multi-layeredandkill-chainbasedsecurity analysis frameworkforcyber-physicalsystems," *International Journal of Critical Infrastructure Protection,* vol. 11, pp. 39-50, 2015.

[10] A. Sujyothi and S. Acharya, "Dynamic Malware Analysis and Detection in Virtual Environment," *I.J. Modern Education and Computer Science,* vol. 3, pp. 48-55, 2017.