

# EXPLORING CYBER THREAT HUNTING THROUGH INTERACTIVE LABS

by

Michael Webber

A thesis submitted to the faculty of  
The University of North Carolina at Charlotte  
in partial fulfillment of the requirements  
for the degree of Master of Science in  
Cybersecurity

Charlotte

2021

Approved by:

---

Dr. Jinpeng Wei

---

Dr. Bei-Tseng “Bill” Chu

---

Dr. Thomas Moyer

©2021  
Michael Webber  
ALL RIGHTS RESERVED

## ABSTRACT

MICHAEL WEBBER. Exploring Cyber Threat Hunting through Interactive Labs.

(Under the direction of DR. JINPENG WEI)

As we continue to see the impact of cyber attacks and cyber threats throughout the world, the need for cyber threat hunting skills is more and more important to the cybersecurity industry and provides a compelling case for expanding the workforce capable of performing these tasks. In this regard, this thesis is intended to help provide more opportunities for students to learn and practice these cyber threat hunting skills. The labs developed are intended to leverage a safe environment and provide an instructor with a toolset to challenge students to better find and analyze cyber threats in an interactive environment. The labs have specific goals and objectives, are built on modern Windows Operating Systems, and leverage free tools and software to make the work more attainable to more people and also create sustainability beyond the labs while avoiding some student costs. The labs demonstrate different examples of malware functions interacting with a victim computer through a command and control (C&C) model and to demonstrate various capabilities and payloads. The student is intended to leverage the tools and techniques of malware analysis to identify and analyze the malware in order to understand how the malware operates and ultimately understand how the malware interacts with the C&C operator. This skill set is extensible to a broader set of malware analysis and cyber threat hunting activities and is anticipated to continue to be a skillset which is needed to address the challenges we see today and in the future.

## DEDICATION

This thesis work is dedicated to my wife, Lisa, who has been so supportive and helped motivate me to achieve this goal. This work is also dedicated to my parents, Wil and Rose, who show through their example how hard work and determination can achieve many things.

## ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Dr. Jinpeng Wei of the Department of Software and Information Systems at UNC Charlotte. Throughout this endeavor, I was encouraged by Dr. Wei both by his continuous research efforts and through his advice and guidance in my area of focus.

I would also like to acknowledge both Dr. Tom Moyer and Dr. Bei-Tseng “Bill” Chu of the Department of Software and Information Systems at UNC Charlotte for serving on my thesis committee. Thank you both, very much.

To my friends who have accommodated my distractions and availability, especially my fellow Rotarians, fellow volunteers supporting our school band programs, fellow CyberCamp and InfraGard members, this accomplishment would not have been possible without your support. Thank you.

To my parents and siblings, thank you for all you have done and continue to do, your support means the world to me.

And last, but certainly not least, to my wife, Lisa, and my children, Chris and Sammy thank you for putting up with me during this time! I really appreciate you. I think this is something we all share as it takes time and energy from everyone. Thank you, you are the best!

## TABLE OF CONTENTS

LIST OF TABLES	VIII
LIST OF FIGURES	IX
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Lab Concepts and Cyber Hunting Skills	1
Chapter 2: Technical background	3
2.1 Lab Build	3
2.2 Lab Specifications	3
2.3 Malware Overview	4
Chapter 3: Methodology	7
3.1 Design Goals/Objectives	7
G1: Tailored for Different Skill Levels	7
G2: Accessible/Free	7
G3: Safely Sustainable/Extensible/Customizable	8
3.2 Methods to Achieve Education and Skill Level Goal (G1)	8
3.2.1 G1 Bloom’s Taxonomy	8
3.2.2 G1 Threat Hunting Skill Set	9
3.2.3 G1 Questions and Manuals	11
3.3 Methods to Achieve Accessible and Free Goal (G2)	11
3.4 Methods to Achieve Sustainable, Extensible, and Customizable Goal (G3)	13
Chapter 4: Results	14
4.1 G1 Results	14
4.1.1 Goals and Objectives of Blueshift Lab	15
4.1.2 Goals and Objectives of Declination Lab	17
4.2 G2 Results	18

4.2.1 Tools used by Blueshift Lab by Goals	19
4.2.2 Tools used by Declination Lab by Goal	24
4.3 G3 Results	33
4.4 Additional Technical Background	36
Chapter 5: Conclusion	50
5.1 Summary of results and alignment to Goals/Objectives	50
5.2 Possible future work - Automation of variable changes, Additional Labs to build...	51
References	52

## LIST OF TABLES

Table 1: Tools and software.....	4
Table 2: Bloom's Taxonomy .....	9
Table 3: Threat Hunting Skill Set .....	9
Table 4: Technique Categories by Lab .....	11
Table 5: Overview of G1 Results by Lab .....	14
Table 6: G1 Blueshift Lab Results.....	15
Table 7: G1 Declination Lab Results.....	17
Table 8: Overview of G2 Results by Lab .....	18
Table 9: G2 Blueshift Lab Results.....	19
Table 10: G2 Declination Lab Results.....	24



## LIST OF FIGURES

Figure 1 - List of Functions .....	5
Figure 2 - Example Lab Questions .....	15
Figure 3 - Process Explorer notepad.exe .....	20
Figure 4 - Procmon notepad.exe keylogger file information .....	20
Figure 5 - Wireshark .....	21
Figure 6 - ApateDNS .....	22
Figure 7 - Wireshark with ApateDNS.....	22
Figure 8 - FAKENET-NG .....	23
Figure 9 - IP Resolution for DNS name.....	24
Figure 10 - Declination Wireshark Capture.....	25
Figure 11 - Example OllyDbg Attach.....	26
Figure 12 - OllyDbg Showing contacts.starlightlabs.org.....	26
Figure 13 - Entry HEX location in GHIDRA .....	27
Figure 14 - Entry Function Diagram in GHIDRA .....	27
Figure 15 - String Search for WSOCK32.DLL::recv .....	28
Figure 16 - String Search for send .....	28
Figure 17 - Key Following Recv: the key string “#KCMDDC51#...” is used to call Decrypt() after recv() is called.....	29
Figure 18 - Key Preceding Send: the key string “#KCMDDC51#...” is used to call Encrypt() before send() is called.....	30
Figure 19 - Encryption Key in Use in OllyDbg.....	31
Figure 20 - Main Selector Function .....	32
Figure 21 - Hosts File Values .....	34
Figure 22 - Keylogger Exfiltration Target.....	35
Figure 23 - List of Functions .....	36

Figure 24 .....	37
Figure 25 - Fun Manager .....	38
Figure 26 - Fun Manager Send MessageBox.....	38
Figure 27 - Fun Manager Chat.....	39
Figure 28 - Process Manager .....	39
Figure 29 - Remote Registry .....	40
Figure 30 - Remote Shell .....	40
Figure 31 - Uninstall Applications.....	40
Figure 32 - Hosts File .....	41
Figure 33 - Services Startup.....	41
Figure 34- Registry Startup.....	42
Figure 35 - Remote Scripting.....	42
Figure 36 - Files Manager.....	43
Figure 37 - Passwords.....	43
Figure 38 - MSN Functions .....	44
Figure 39 - Spy Functions - WebCam, Microphone, Keylogger .....	44
Figure 40 - Spy Functions - Remote Desktop.....	45
Figure 41 - Network Active Ports.....	46
Figure 42 - Network Shares .....	46
Figure 43 - LAN Computers .....	47
Figure 44 - Misc Functions - Print Manager and Clipboard .....	47
Figure 45 - Computer Power.....	48
Figure 46 - Restart Socket (Client and Server).....	48
Figure 47 - Server Actions and Update Server .....	49
Figure 48 - Update Server.....	49
Figure 49 - DDoS Functions .....	50

## **Chapter 1: Introduction**

### **1.1 Overview**

The capabilities and effectiveness in defending against cyberattacks is dependent on the development of cyber hunting skills[1][2][13][14]. We continue to see an increase in cyber-attacks [5][6][7][8][9][10][11][12][13][14][15][16][17][18][20][21][22][23] which have different impacts and some significant attacks are not detected by even the most sophisticated, automated systems. Over the past year, we have seen supply chain attacks against pivotal IT tools like Solarwinds® [17] and Kaseya® [15], both of which individually created threat vectors into thousands of other businesses. We have seen an individual government require tax software which contained malware [16]. We continue to see a deficit in skilled labor in the area of cybersecurity and more specifically in the area of cyber threat hunting [24][25]. Although advancements in Artificial Intelligence and Automated Information Security are likely to grow and improve, we will still need to cultivate the individual's cyber hunting skills in our industry to defend our systems and environments from cyberattack.

### **1.2 Lab Concepts and Cyber Hunting Skills**

Developing and improving cyber hunting skills leveraging free tools is the focus of this research and fundamental in continuing to develop the workforce to support cybersecurity. The research project created labs built with free software and tools to introduce and hone Cyber Hunting skills. In order to fulfill the free software and tools goal, the labs have some fundamental requirements including: (1) the labs must be able to

support the analysis of the malware, so the malware must be able to be run without a dependency on internet access and (2) the labs must have all the required elements to be able to perform the analysis. In support of these fundamental requirements, the lab environment allows the student to spin up multiple virtual machines (VMs) to be a helpful tool for analysis, for example a Ubuntu VM to run Netcat or INetSim to provide command and control capabilities to interact with malware [4].

The overall basis for cyber threat hunting is focused on 8 Threat Hunting Skills[1] which are: (1) incident detection and analysis, (2) threat intelligence, (3) security data analysis, (4) forensic analysis, (5) malicious code analysis, (6) analytical models, (7) penetration testing, and (8) vulnerability testing. The labs created are focused on threat detection and malicious code analysis with specific analysis activities further categorized into 4 groups: Basic Static Analysis, Basic Dynamic Analysis, Advanced Static Analysis and Advanced Dynamic Analysis. The students operating the labs will utilize specific techniques and tools with different levels of complexity to identify and analyze the threat(s) they find within the labs.

## **Chapter 2: Technical background**

### **2.1 Lab Build**

The labs are built using Oracle Virtual Box virtual machines with a Microsoft Windows Operating System. The labs run stable virtual machines with virtual networking to provide students with an effective cyber threat hunting environment while also providing security options to prevent the malware from escaping the lab environment. The labs include additional virtual machines which are optional and can be started by the student to perform functions and introduce capabilities to be used individually or in coordination with other virtual machines to analyze and interact with the malware. These virtual machines contain software and tools which are available to the general public at no cost. More details on these tools is provided in later sections.

### **2.2 Lab Specifications**

The lab objectives are designed to provide real-time examples of cyber threat hunting to demonstrate specific ways to identify and analyze malware. Each lab objective has both a technical goal as well as a level of complexity or difficulty which is intended to match up with the various levels of student skills and knowledge which is explored further in this thesis. The lab environment is running Oracle Virtual Box built on virtualbox-5.2\_5.2.22-126460. The operating system of the virtual machine running the malware is

Microsoft® Windows® 8 and Windows® 10. The tools and software available in the environment are listed in Table 1.

*Table 1: Tools and software*

Tools and software		
apateDNS	PEiD	PEView
IDA Free	Regshot	SysinternalsSuite Process Explorer
Netcat	FlareVM	SysinternalsSuite Process Monitor
OllyDbg	WinPcap	Dependency Walker
GHIDRA	WinSCP	INetSim
FAKENET-NG	Wireshark	other tools as available

The labs are built to provide an environment for the student to practice and demonstrate the use of these tools in a secure way by providing a hands-on lab with an active threat and providing an instructor and student lab manual to assist both parties in accomplishing the goals of the labs.

### **2.3 Malware Overview**

The malware in the lab includes a Remote Access Trojan (RAT), also known as a Remote Administration Tool or backdoor, with support of reverse shell and remote desktop experience along with Command and Control (C&C) capabilities. The method of the C&C interaction varies from lab to lab as in some cases, the victim machine can be analyzed independently from the C&C machine, but in other labs, the C&C machine is part of the lab activities.

The Primary Functions of the malware include File System Management, Remote Shell Commands, Camera and Sound Spying, Keystroke Logger, Remote Control, amongst other functions. The full list of functions available through the malware are shown in

Figure 1 and more detail of this malware can be found in Section 4.4. For now, we will look at some of the primary functions which are of interest in the malware capabilities.

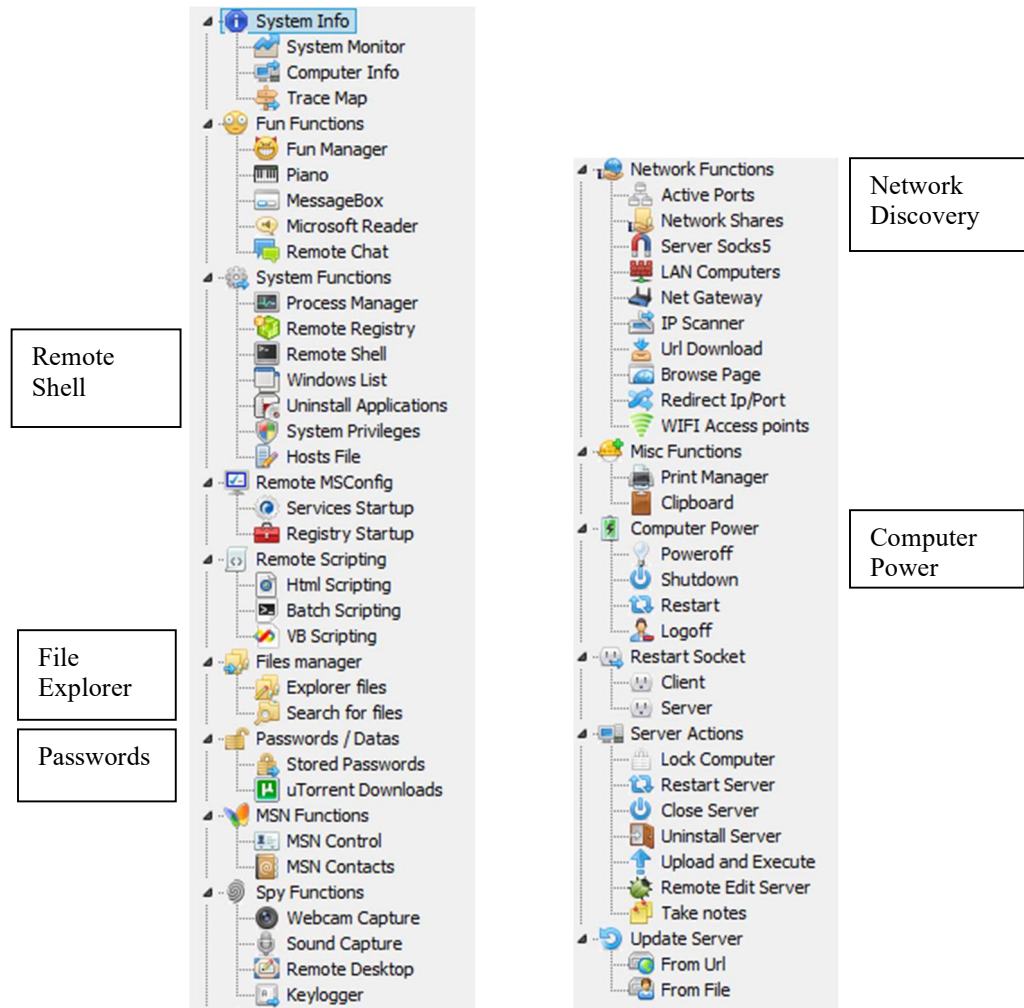


Figure 1 - List of Functions

This malware has many different attack methods including enumeration of the computer system components, file system access, remote access shell, interactive chat, audio surveillance, video surveillance, keylogging, screen capture, and many others (Figure 1). In addition, the malware can persist after restart and hide visibility of its files. These

capabilities are useful in both the variability and difficulty of the labs which can be developed.



## **Chapter 3: Methodology**

### **3.1 Design Goals/Objectives**

#### **G1: Tailored for Different Skill Levels**

The first goal of each lab is to align the activities to meet different skill levels in order to provide students of different backgrounds the opportunity to learn cyber threat hunting skills appropriate to their knowledge and ability. By creating labs with varying skill levels, the labs will be appropriate for a broader set of students, both from an interest level as well as the depth of skill which an individual student brings. The expectation is to help increase the number of students who can participate as well as the extent to which students can grow and learn throughout the activities. In practical application the instructor will need to decide which labs will be used for which students. This is important as the intent is to challenge the student without making the labs so difficult that they discourage the student from future efforts. From the Goals and Objectives are developed questions to be answered by the student during the lab. These questions provide a starting point for the student and help move the student through different tools to achieve the intended results and the Goals and Objectives of the lab.

#### **G2: Accessible/Free**

The second goal of the labs is to use software and tools which are free and available to the general public. This goal is intended to facilitate development for the student which can be applied outside the classroom experience and helping to minimize the financial burden for the student. This approach is beneficial both on the part of the instructor or

institution as well as the ongoing use and practice of the student. The student is more likely to be able to build and leverage these tools after their training completes if the tools are freely available. In some cases, the tools are open source and students could even provide improvements by contributing to the open source project or provide feedback for future application development by others.

### **G3: Safely Sustainable/Extensible/Customizable**

The third goal of the labs is to be sustainable, extensible and customizable. This goal will help to provide longevity of the learning environment. The labs can be *sustained* through utilization of stable, well-known and established operating systems, virtual machine platforms, and software. The labs are *extensible* through the use of additional, optional virtual machines and tools which can be implemented by the student to provide additional capabilities. The labs are *customizable* through modification of key elements of the malware to provide the instructor with the option to make changes which can be reflected in assignments and alter the outcomes of specific results. It is important that these customizations do not negatively impact the lab environments.

## **3.2 Methods to Achieve Education and Skill Level Goal (G1)**

### **3.2.1 G1 Bloom's Taxonomy**

Each Lab Objective incorporates Bloom's Taxonomy [26][27], the Cyber Hunting Skill Set, and Malware Analysis Techniques to create effective lessons which have different

difficulty skill levels. Bloom's Taxonomy is a framework which provides six categories of educational goals.

*Table 2: Bloom's Taxonomy*

<b>Remembering</b>	Identify Recall Select Label Recognize Tell List Match Name
<b>Understanding</b>	Classify Demonstrate Infer Relate Translate Compare Explain Interpret Show Contrast Illustrate Outline Summarize
<b>Applying</b>	Use Respond Organize Choose Solve Carry out Apply Build Model Provide Develop Select Utilize
<b>Analyzing</b>	Assume Classify Dissect Analyze Compare Distinguish Categorize Contrast Examine Conclude Discover Inspect
<b>Evaluating</b>	Appraise Assess Award Choose Criticize Defend Disprove Estimate Interpret Judge Rate Support Justify
<b>Creating</b>	Create Design Assemble Generate Build Change Choose Combine Formulate Elaborate Modify Compose Invent Improve Predict Plan

### 3.2.2 G1 Threat Hunting Skill Set

Threat Hunting Skill Set incorporates multiple skill areas. The Lab objectives align to this skill set:

*Table 3: Threat Hunting Skill Set*

Threat Hunting Skill Set
Incident detection and analysis Threat intelligence Security data analysis Forensic analysis Malicious code analysis Analytical models Penetration testing Vulnerability analysis

Within the skill set of malicious code analysis, the Labs leverage principles of malware analysis [28] including:

- Basic Static Analysis - Perform Analysis on the malware while it is not actively running and without viewing the actual instructions of the malware. This is a common starting point for malware analysis, but in the labs the malware must first be identified which can move Basic Static Analysis later in the sequencing.
- Basic Dynamic Analysis – Perform Analysis with malware running to observe the malware and begin to understand the activity of the malware. It is important to note that this requires a safe, secure, and contained environment.
- Advanced Static Analysis – Perform analysis through disassembler program to view actual instructions of the malware and gain an understanding of what the program is designed to do.
- Advanced Dynamic Analysis – Perform analysis through a debugger program to see the malware in a running state to further examine dynamic changes while the malware is running. This method allows the threat hunter to see data in active registers, in memory, and discover details which may only be visible while the malware is running.

Each lab is mapped to the categories of the techniques which are demonstrated within the lab. The following table shows the mapping for five labs developed in this thesis. For example, the lab named “Apogee” covers basic static and basic dynamic techniques.

*Table 4: Technique Categories by Lab*

Analysis Technique	Apogee	Blueshift	Celestial	Declination	Eclipse
Basic Static	X	x	X	x	x
Basic Dynamic	X	x	X	x	x
Advanced Static				x	x
Advanced Dynamic			X	x	x

### 3.2.3 G1 Questions and Manuals

Student Lab Manuals (Available upon request) are scoped for the student and the Instructor Lab Manual (Available upon request) provides details for the instructor to align the activities to the expected results and provide consistency across the lab experiences. These lab manuals enable the instructor to provide the appropriate level of assignments to the student skill level and to reference key information which match to the malware characteristics in the lab. The manuals include questions intended to both help the student progress through the different techniques but also help the instructor evaluate the efforts of the student in the labs.

### 3.3 Methods to Achieve Accessible and Free Goal (G2)

Free and Open Source Software are available and effective. These tools are often not the most up-to-date and missing features which paid tools may have, but the concept of Free and Open Source has been around for many years and the tenets of the model are to have

access to software without cost being prohibitive. This applies to the field of cybersecurity and the tools leveraged for these labs are available as free and open source. Free, Open Source, Freeware and Shareware are all terms which are used and sometimes misunderstood [29][30]. Free Software Foundation (FSF) is based on 4 pillars: the software is free to use for any purpose, free of cost to use, free to study, and free to redistribute copies. Open Source Initiative (OSI) has specific terms and conditions. Examples of Open Source licenses are Apache, BSD, GNU, MIT and Mozilla [30], but open source software is generally free to use and redistribute. Freeware Software is free to use, but cannot be modified and must be free of cost when redistributed. Shareware software is free to use, but is limited in functionality or in features. In general, the software associated to these labs will fall under one of these 4 Free and Open Source categories.

In addition to leveraging free software to perform the labs, the malware selection process is also an important aspect of the lab methodology of accessible/free. The malware was selected through a process whereby multiple malware samples were reviewed. These malware samples were found using freely accessible websites, like github, hybrid-analysis.com, and others. Multiple different malware samples were evaluated, but this particular malware software provided many different functional capabilities as well as a built-in user interface to customize and deploy many different executable files with variability in many aspects. This was a driving factor in the selection of the malware for the labs.

### **3.4 Methods to Achieve Sustainable, Extensible, and Customizable Goal (G3)**

To achieve G3, Sustainable, Extensible, and Customizable, the labs incorporate multiple methods to modify and enhance the labs. Parameters for configuration and customization are available through modification of the malware binary as well as through the interaction with the C&C operator system. The lab re-deployment options allow for reconfiguration of the virtual machines to change data points to help address sharing of information between students and sessions and to provide an opportunity for labs to maintain contemporary or present-day references. The lab information which can be readily changed includes:

- Name of the executable to be deployed to primary VM
- Entry in Hosts file of the primary VM
- DNS Name of the C&C client (reverse client-server malware)
- IP Address of the C&C client (reverse client-server malware)
- Port Number used in communication with C&C client (reverse client-server malware)
- Registry entries
- Date of File Creation

The steps to change these values is part of the binary creation process, but some changes can also be made through use of the Dark Comet RAT after deployment through dynamic interaction.

## Chapter 4: Results

### 4.1 G1 Results

In this research, we developed five labs: Apogee, Blueshift, Celestial, Declination, and Eclipse. Each lab has specific goals and objectives which align to a category of techniques.

*Table 5: Overview of G1 Results by Lab*

Overview of G1 Results by Lab		
Lab Name	Technique Category	Goal and Objective
Apogee	Basic Static Basic Dynamic	1. Identify network traffic generated by malware including traffic destination DNS name and IP address. 2. Examine the running processes on the computer to determine which process is sending the malware network traffic.
Blueshift	Basic Static Basic Dynamic	1. Determine what type of malware information is being sent through the network traffic. 2. Provide the path, filename, and some example data attempting to be exfiltrated by the malware.
Celestial	Basic Static Basic Dynamic Advanced Dynamic	1. Establish which protocol and network service is being used by the malware to attempt to exfiltrate data. 2. Begin collecting data sent by the malware for potential future analysis.
Declination	Basic Static Basic Dynamic Advanced Static Advanced Dynamic	1. Identify the Malware running on the victim computer through basic static and basic dynamic analysis techniques. 2. Show the location (hex instruction) of the start/entry function of the Malware through advanced static analysis. 3. Identify the location (hex instruction) of the message handler which is listening for communication from the C&C Client (operator) 4. Discover which Encryption Key in use by the malware through inspecting the malware code debugger analysis and comparing to disassembler functions. 5. [Optional] Select and analyze 2 primary functions available to the malware C&C Client based on Advanced Static Analysis and/or Advanced Dynamic Analysis. Find the cross reference (xref) call instructions for each of the 2 chosen functions.
Eclipse	Basic Static Basic Dynamic Advanced Static Advanced Dynamic	1. Find Start of Malware function location and Initial Network Communication function location in disassembler program. 2. Identify and Analyze 2 primary functions available to the malware C&C Client based on Advanced Static Analysis. 3. Work with the C&C Client to send active commands to the victim computer while observing the interaction and then find



		the function location in the malware which represents the attack primary function. 4. Use Python Script in tools to decrypt malware communication attempting to be exfiltrated to C&C Client.
--	--	--

From these goals and objectives, questions are added to the Student Manual to help the student choose appropriate tools to perform tasks and gather information to complete the goals and objectives. The following figure provides some example questions.

Example Questions from Labs
Is any suspicious network traffic being sent by the computer? (Determine the destination DNS name and IP address)
How is the DNS name resolved to the IP address?
Determine which active process is sending the malware network traffic.
Establish which protocol and network service is being used by the malware to attempt to exfiltrate data.
What are some keystrokes which were captured in the logs?

*Figure 2 - Example Lab Questions*

As a student looks to determine suspicious network traffic, the expectation from the instructor is for the student use tools like Wireshark, ApateDNS, FAKENET-NG, or other tools with opportunities to observe network traffic to identify something out of the ordinary. Examples of the results of this work are found in the G2 results (Section 4.2).

#### 4.1.1 Goals and Objectives of Blueshift Lab

*Table 6: G1 Blueshift Lab Results*

Lab Name	Technique Category	Goal and Objective
Blueshift	Basic Static Basic Dynamic	1. Determine what type of malware information is being sent through the network traffic. 2. Provide the path, filename, and some example data attempting to be exfiltrated by the malware.

Looking in more detail at the Blueshift lab, the student is expected to perform tasks associated with Basic Static and Dynamic Analysis to determine information about the network traffic being generated by the malware and what type of information the malware is attempting to exfiltrate. Through Basic Dynamic Analysis techniques, the student will look for network traffic intended for unexpected destinations. Some expectation of general knowledge of the operating system is necessary and the lab uses different tools (refer to section 4.2.1) to analyze the network traffic and isolate the malware as well as the information attempting to be exfiltrated.

To assist the student in accomplishing the objectives, questions are provided in the lab manual to help the student choose tools as well as help the instructor evaluate the success of the student's efforts. Some example questions from this lab include:

- Is any suspicious network traffic being sent by the computer? (Determine the destination DNS name and IP address)
- Determine which active process is sending the malware network traffic.
- What is the path of the keylogger log files?
- What are some keystrokes which were captured in the logs?

The skill level of this lab is considered to be a fairly simple and straightforward demonstration of Basic Static and Dynamic Analysis techniques because the Goals and Objectives require limited information focusing on the network traffic which is being generated, then identifying the process responsible for the traffic and further looking into the behavior of the process to find the payload of the malware function. This lab does not require more advanced skills in static or dynamic analysis and although more advanced techniques could also achieve the results, the focus is on the basic techniques and

emphasize starting with observation to focus the work to be completed and sets the stage for more advanced efforts.

#### 4.1.2 Goals and Objectives of Declination Lab

*Table 7: G1 Declination Lab Results*

Lab Name	Technique Category	Goal and Objective
Declination	Basic Static Basic Dynamic Advanced Static Advanced Dynamic	<ol style="list-style-type: none"><li>1. Identify the Malware running on the victim computer through basic static and basic dynamic analysis techniques.</li><li>2. Show the location (hex instruction) of the start/entry function of the Malware through advanced static analysis.</li><li>3. Identify the location (hex instruction) of the message handler which is listening for communication from the C&amp;C Client (operator)</li><li>4. Discover which Encryption Key in use by the malware through inspecting the malware code debugger analysis and comparing to disassembler functions.</li><li>5. [Optional] Select and analyze 2 primary functions available to the malware C&amp;C Client based on Advanced Static Analysis and/or Advanced Dynamic Analysis. Find the cross reference (xref) call instructions for each of the 2 chosen functions.</li></ol>

The skillset for the Declination lab is expected to be significantly higher than the Blueshift lab and it becomes apparent as the goals and objectives are focused well beyond the network traffic and process behavior at a basic level, but instead expect the student to quickly find and identify the malware and progress into advanced analysis through disassembly and debugging techniques. Example questions included in the Declination lab include:

- What instruction point is the Entry or Start of the Malware?
- What is the encryption key used in the C&C communications?
- What function is used by the malware to identify the C&C communications?

In this lab, the student looks for specific instructions within the code of the malware and looks for variable values to determine the Encryption key which is in use by the malware. This lab expects the student to be able to differentiate between the important functions and processes and the insignificant functions to identify where in the code the Encryption key might be used and then find the key within the malware. The approach taken through this activity set is intended to provide opportunities for the student to debug the active malware by pausing the malware, setting breakpoints and testing theories based on the observations of advanced static analysis working closely with advanced dynamic analysis techniques. The expectation is for the student to work through a logical analysis building on discoveries and knowledge until the student completes the tasks which is covered in more detail in section 4.2.2.

## 4.2 G2 Results

Each lab has specific tools which can be used by the student, at the student's discretion. The expectation is that the student will achieve the lab objectives through a combination of these tools used to complement and build on another tool's results.

*Table 8: Overview of G2 Results by Lab*

Overview of G2 Results by Lab		
Lab Name	Technique Category	Tools
Apogee	Basic Static Basic Dynamic	Process Explorer, Process Monitor, Wireshark, ApateDNS, FAKENET-NG, PEView, Strings
Blueshift	Basic Static Basic Dynamic	Process Explorer, Process Monitor, Wireshark, ApateDNS, FAKENET-NG, PEView, Strings
Celestial	Basic Static Basic Dynamic Advanced Dynamic	Process Explorer, OllyDbg, Process Monitor, Wireshark, ApateDNS, INetSim

Declination	Basic Static Basic Dynamic Advanced Static Advanced Dynamic	Process Explorer, IDA Free, GHIDRA, Process Monitor, Wireshark, FAKENET-NG, ApateDNS, OllyDbg, C&C Client (optional)
Eclipse	Basic Static Basic Dynamic Advanced Static Advanced Dynamic	Process Explorer, IDA Free, GHIDRA, Process Monitor, Wireshark, FAKENET-NG, ApateDNS, OllyDbg, Python Script (optional), C&C Client (optional), InetSim

#### 4.2.1 Tools used by Blueshift Lab by Goals

*Table 9: G2 Blueshift Lab Results*

Malware Analysis Tools	Basic Static	Basic Dynamic
Process Explorer		x
Process Monitor		x
Wireshark		x
ApateDNS		x
FAKENET-NG		x
PEView	x	
Strings	x	

As an example of the tools in use by the Blueshift Lab, the student is expected to apply Process Explorer (Procexp.exe) and Process Monitor (Procmon.exe) to observe the active processes on the system. While looking through Process Explorer, the student may notice the malware msdsc.exe is running and is not a common program. Also, the student may

observe a notepad.exe program is running with a subprocess of the same malware name, msdsc.exe. This is unusual for notepad.exe, which is a basic text editor.

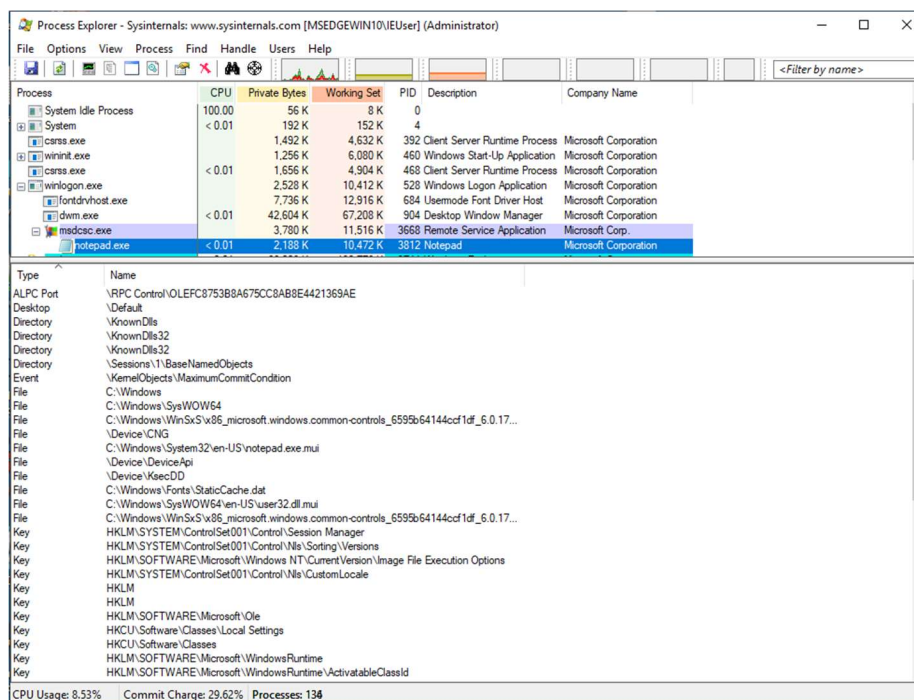


Figure 3 - Process Explorer notepad.exe

Process Monitor is helpful in further understanding what the malware process is sending (payload) in the network traffic as shown in Figure 4, which shows a Notepad process accessing a log file. This is a significant clue into the payload of the malware.

Time	Process Name	PID	Operation	Path	Result	Detail
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKCR\Unknown	SUCCESS	Query: HandleTags, HandleTags: 0x0
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKCU\Software\Classes\Unknown	NAME NOT FOUND	Desired Access: Maximum Allowed
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKCR\Unknown\NeverShowExt	SUCCESS	NAME NOT FOUND Length: 12
8:37.3...	NOTEPAD EXE	892	CloseFile	C:\Users\IEUser\AppData\Roaming\idlog	SUCCESS	
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePa...	SUCCESS	Query: HandleTags, HandleTags: 0x0
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePa...	SUCCESS	Desired Access: Query Value, Enumerate Sub Keys
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePa...	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformation, Length: 0
8:37.3...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePa...	NAME NOT FOUND	Desired Access: Query Value
8:37.4...	NOTEPAD EXE	892	ReadFile	C:\Windows\System32\notepad.exe	SUCCESS	Offset: 91,136; Length: 4,096; I/O Flags: Non-cached, Paging I/O, Synchronous I
8:37.4...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\OOBE	SUCCESS	Offset: 62,464; Length: 32,768; I/O Flags: Non-cached, Paging I/O, Synchronous
8:37.4...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\OOBE\LaunchU...	SUCCESS	Query: HandleTags, HandleTags: 0x0
8:37.4...	NOTEPAD EXE	892	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\OOBE	NAME NOT FOUND	Desired Access: Query Value
8:37.4...	NOTEPAD EXE	892	CreateFile	C:\Users\IEUser\AppData\Roaming\idlog\2021-11-14-1.doc	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point
8:37.4...	NOTEPAD EXE	892	OpenFile	C:\Users\IEUser\AppData\Roaming\idlog\2021-11-14-1.doc	SUCCESS	Offset: 759,296; Length: 8,192; I/O Flags: Non-cached, Paging I/O, Synchronous
8:37.4...	NOTEPAD EXE	892	ReadFile	C:\Windows\System32\efsvnt.dll	SUCCESS	Offset: 418,816; Length: 4,096; I/O Flags: Non-cached, Paging I/O, Synchronous
8:37.4...	NOTEPAD EXE	892	ReadFile	C:\Windows\System32\efsvnt.dll	SUCCESS	Thread ID: 1496; User Time: 0.0000000; Kernel Time: 0.0000000
8:37.4...	NOTEPAD EXE	892	ReadFile	C:\Windows\System32\efsvnt.dll	SUCCESS	Offset: 747,008; Length: 4,096; I/O Flags: Non-cached, Paging I/O, Synchronous
8:37.4...	NOTEPAD EXE	892	Thread Exit	C:\Windows\System32\efsvnt.dll	SUCCESS	Thread ID: 4724; User Time: 0.0000000; Kernel Time: 0.0000000
8:37.4...	NOTEPAD EXE	892	Thread Exit	C:\Windows\System32\efsvnt.dll	SUCCESS	Offset: 414,720; Length: 28,672; I/O Flags: Non-cached, Paging I/O, Synchronous
8:37.4...	NOTEPAD EXE	892	Thread Exit	C:\Windows\System32\efsvnt.dll	SUCCESS	Thread ID: 5528; User Time: 0.0000000; Kernel Time: 0.0000000
8:37.4...	NOTEPAD EXE	892	Thread Exit	C:\Windows\System32\efsvnt.dll	SUCCESS	Thread ID: 456; User Time: 0.0000000; Kernel Time: 0.0000000
8:37.4...	NOTEPAD EXE	892	ReadFile	C:\Windows\System32\efsvnt.dll	SUCCESS	Offset: 463,872; Length: 14,848; I/O Flags: Non-cached, Paging I/O, Synchronous
8:37.4...	NOTEPAD EXE	892	ReadFile	C:\Windows\System32\efsvnt.dll	SUCCESS	Offset: 678,424; Length: 16,384; I/O Flags: Non-cached, Paging I/O, Synchronous

Figure 4 - Procmon notepad.exe keylogger file information

But first, the student should apply a network tool like ApatеDNS, Wireshark, FAKENET-NG or combinations of tools to identify a destination IP and hostname of the traffic generated by the victim computer. Simply running Wireshark on the victim computer provides some immediate information on the loopback interface as shown in Figure 5.

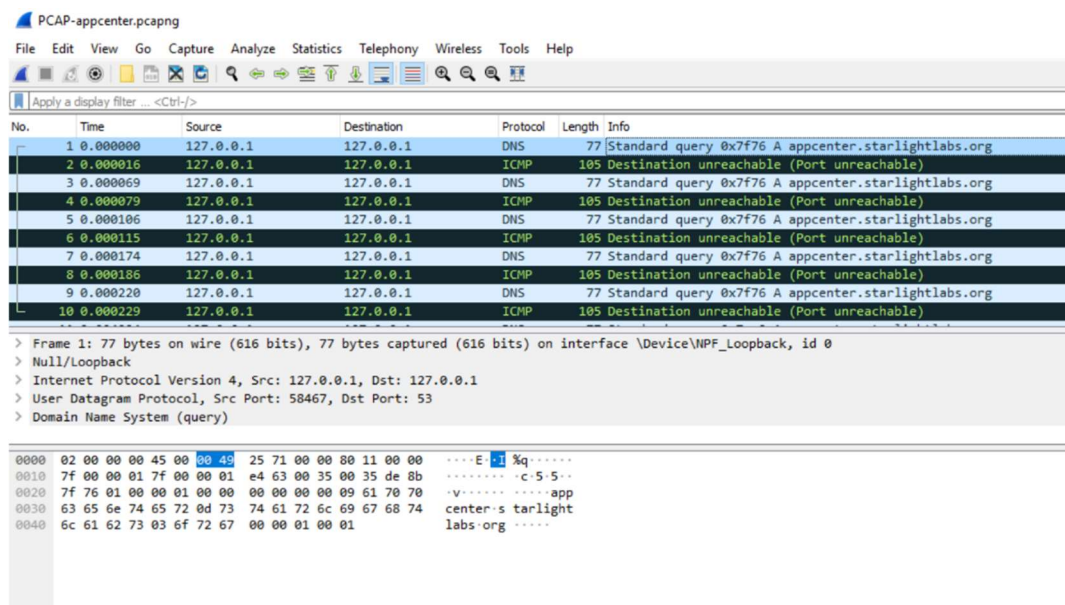


Figure 5 - Wireshark

Implementing ApatеDNS on the victim computer also identifies network traffic generated to a DNS hostname as shown in Figure 6.

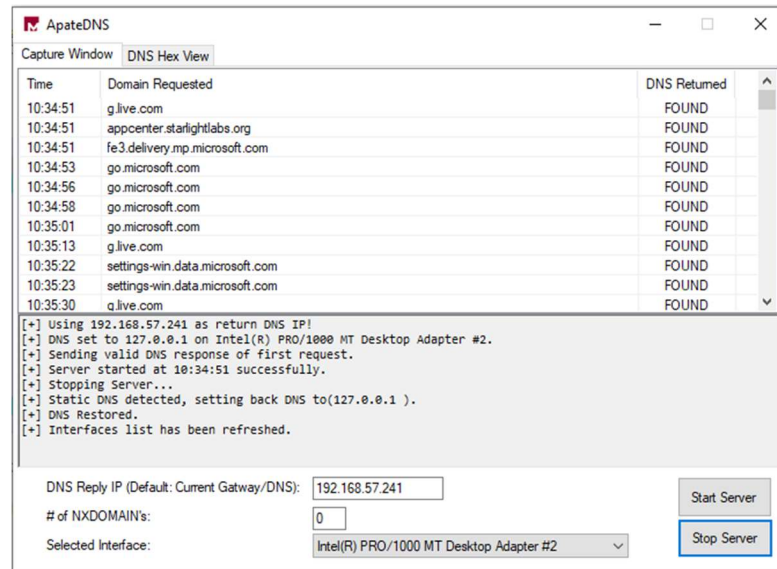


Figure 6 - ApatеDNS

And if Wireshark is run in conjunction with ApatеDNS, then the Wireshark window becomes more standard as ApatеDNS provides a method for the communication to be sent rather than ICMP errors without ApatеDNS as shown in Figure 7.

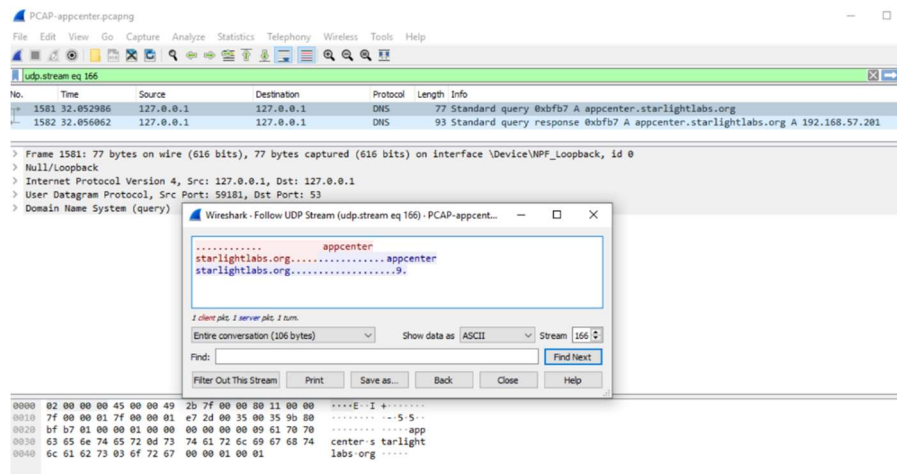
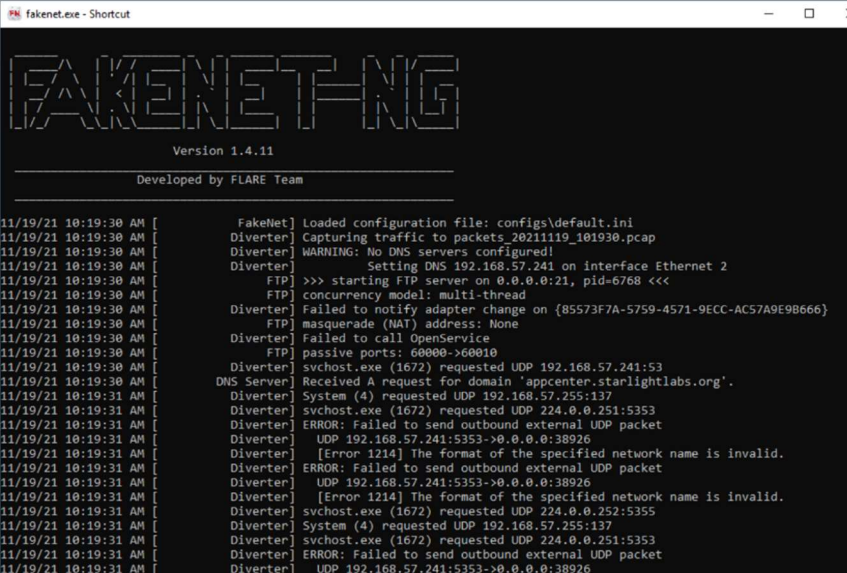


Figure 7 - Wireshark with ApatеDNS



As an alternative toolset, FAKENET-NG when launched on the victim computer quickly identifies the DNS request for an unusual hostname as seen in Figure 8.



```
fakenet.exe - Shortcut
FAKENET-NG
Version 1.4.11
Developed by FLARE Team

11/19/21 10:19:30 AM [ FakelNet] Loaded configuration file: configs\default.ini
11/19/21 10:19:30 AM [ Divertor] Capturing traffic to packets 20211119_101930.pcap
11/19/21 10:19:30 AM [ Divertor] WARNING: No DNS servers configured!
11/19/21 10:19:30 AM [ Divertor] Setting DNS 192.168.57.241 on interface Ethernet 2
11/19/21 10:19:30 AM [ FTP] >>> starting FTP server on 0.0.0.0:21, pid-6768 <<<
11/19/21 10:19:30 AM [ FTP] concurrency model: multi-thread
11/19/21 10:19:30 AM [ Divertor] Failed to notify adapter change on {85573F7A-5759-4571-9ECC-AC57A9E9B666}
11/19/21 10:19:30 AM [ FTP] masquerade (NAT) address: None
11/19/21 10:19:30 AM [ Divertor] Failed to call OpenService
11/19/21 10:19:30 AM [ FTP] passive ports: 60000->60010
11/19/21 10:19:30 AM [ Divertor] svchost.exe (1672) requested UDP 192.168.57.241:53
11/19/21 10:19:30 AM [ DNS Server] Received A request for domain 'appcenter.starlightlabs.org'.
11/19/21 10:19:31 AM [ Divertor] System (4) requested UDP 192.168.57.255:137
11/19/21 10:19:31 AM [ Divertor] svchost.exe (1672) requested UDP 224.0.0.251:5353
11/19/21 10:19:31 AM [ Divertor] ERROR: Failed to send outbound external UDP packet
11/19/21 10:19:31 AM [ Divertor] UDP 192.168.57.241:5353->0.0.0.0:38926
11/19/21 10:19:31 AM [ Divertor] [Error 1214] The format of the specified network name is invalid.
11/19/21 10:19:31 AM [ Divertor] ERROR: Failed to send outbound external UDP packet
11/19/21 10:19:31 AM [ Divertor] UDP 192.168.57.241:5353->0.0.0.0:38926
11/19/21 10:19:31 AM [ Divertor] [Error 1214] The format of the specified network name is invalid.
11/19/21 10:19:31 AM [ Divertor] svchost.exe (1672) requested UDP 224.0.0.252:5355
11/19/21 10:19:31 AM [ Divertor] System (4) requested UDP 192.168.57.255:137
11/19/21 10:19:31 AM [ Divertor] svchost.exe (1672) requested UDP 224.0.0.251:5353
11/19/21 10:19:31 AM [ Divertor] ERROR: Failed to send outbound external UDP packet
11/19/21 10:19:31 AM [ Divertor] UDP 192.168.57.241:5353->0.0.0.0:38926
```

Figure 8 - FAKENET-NG

Regardless of the tools utilized by the student, the DNS name specific to this lab is appcenter.starlightlabs.org and has an IP address of 192.168.57.152 (Figure 9). If the DNS name is found first, the student can use the ping utility to find the IP address. The

student may wonder why that name is resolvable and could check the Hosts file on the computer (Figure 9).

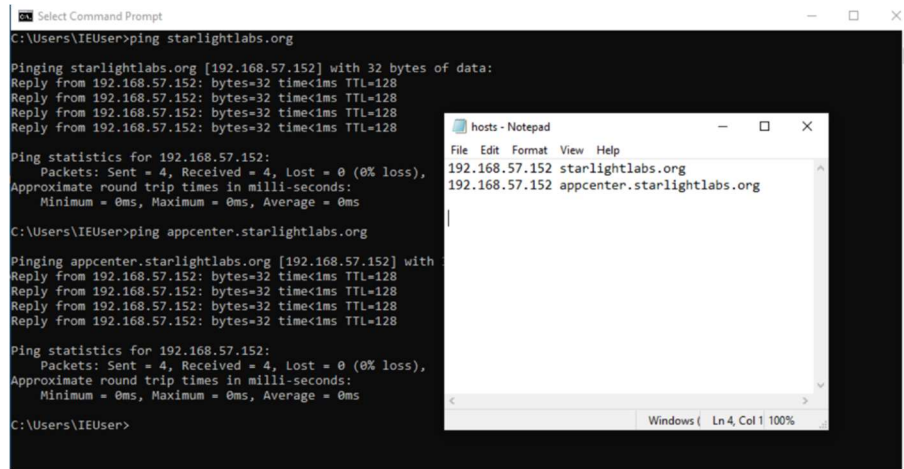


Figure 9 - IP Resolution for DNS name

#### 4.2.2 Tools used by Declination Lab by Goal

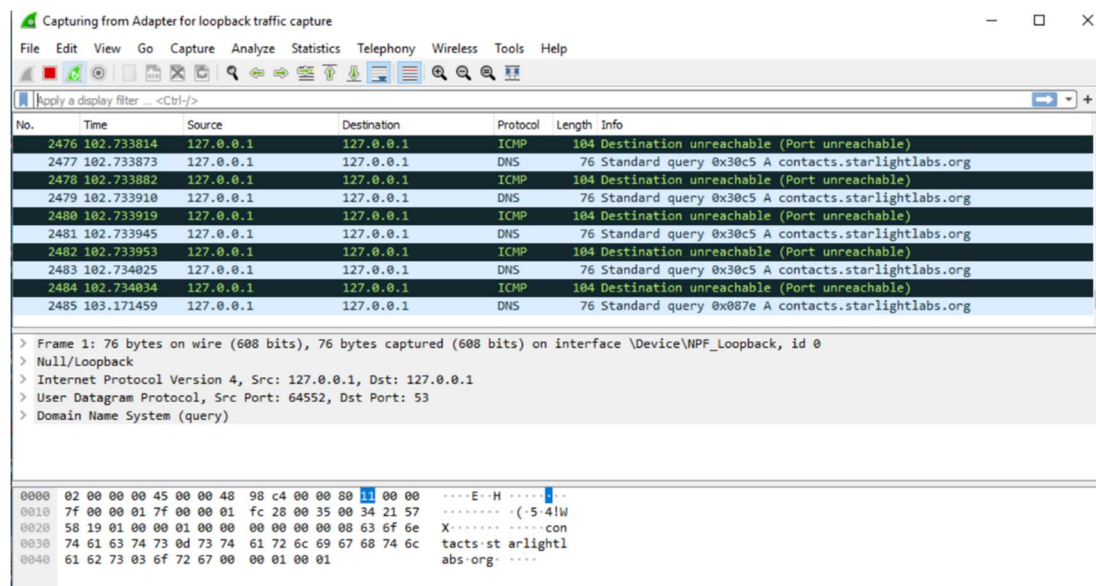
Table 10: G2 Declination Lab Results

Tools	Basic Static	Basic Dynamic	Advanced Static	Advanced Dynamic
Process Explorer		X		
IDA Free	X		X	X
GHIDRA	X		X	
Process Monitor		X		X
Wireshark		X		
FAKENET-NG		X		
ApateDNS		X		
OllyDbg				X

The Declination lab tools provide a series of different insights into the behavior of the malware and the student is considerably more involved in the analysis of the malware functions through disassemblers and debugging toolsets. The student can choose from a

few different tools to achieve the goals and objectives and a combination of tools will be necessary to complete the tasks. The Basic Static Analysis and Basic Dynamic Analysis in the Declination Lab is all contained in the first goal and objective: Identify the Malware running on the victim computer through basic static and basic dynamic analysis techniques.

An example screenshots of the first objective lab results are in the following Wireshark capture.



*Figure 10 - Declination Wireshark Capture*

But this lab quickly moves into more advanced analysis tools and expects the student to practice disassembly and debugging techniques. The tools available include IDA Free and GHIDRA for Advanced Static Analysis as well as IDA Free and OllyDbg for Advanced Dynamic Analysis. The student is expected to find specific instructions and values within the malware code as demonstrated in the following figures.

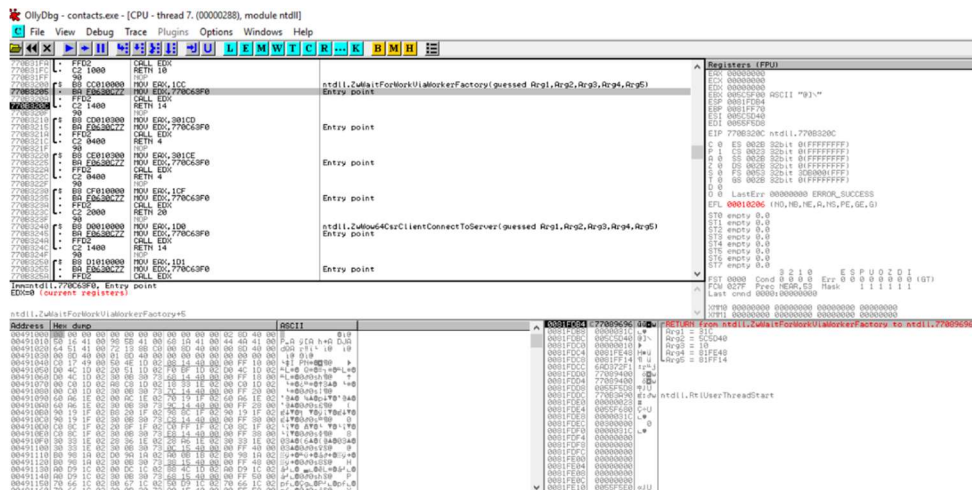


Figure 11 - Example OllyDbg Attach

Simply attaching to the malware using OllyDbg is captured in Figure 11 above. And working further through the application code, the student may find the reference to [contacts.starlightlabs.org](https://contacts.starlightlabs.org) within the debugger as shown in Figure 12 below. OllyDbg provides a strong tool for walking through the malware code in an effort to achieve the goals and objectives of the lab.

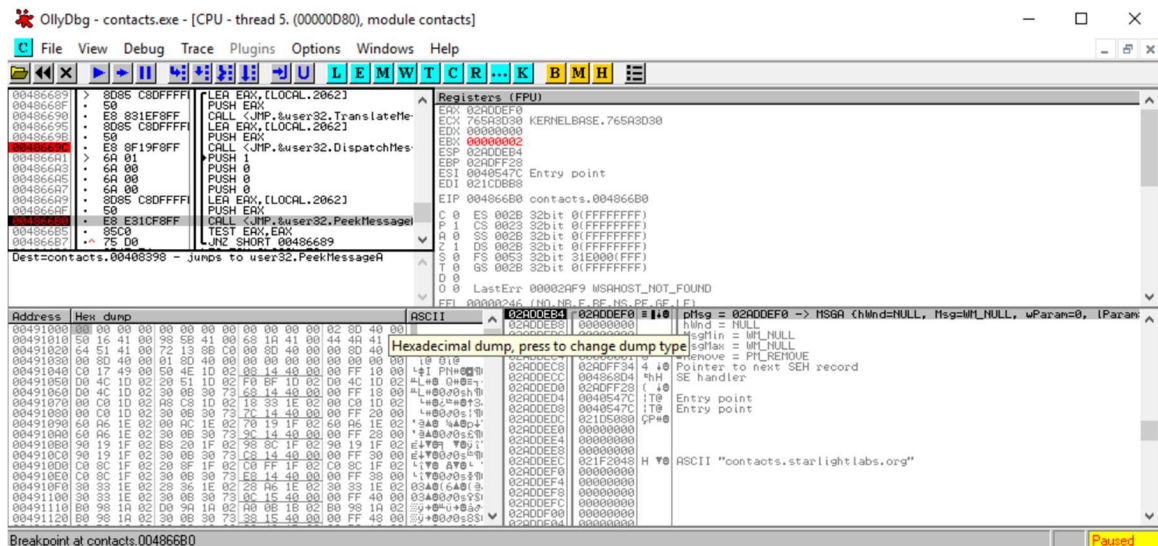


Figure 12 - OllyDbg Showing [contacts.starlightlabs.org](https://contacts.starlightlabs.org)

For Advanced Static Analysis, GHIDRA provides an interactive disassembler as shown in Figure 13, identifying the entry hex location for the malware code. This is one of the objectives of the lab. In addition, visualizing the function is an important aspect of advanced static analysis and the following Figure 14 shows the same Entry function diagram.

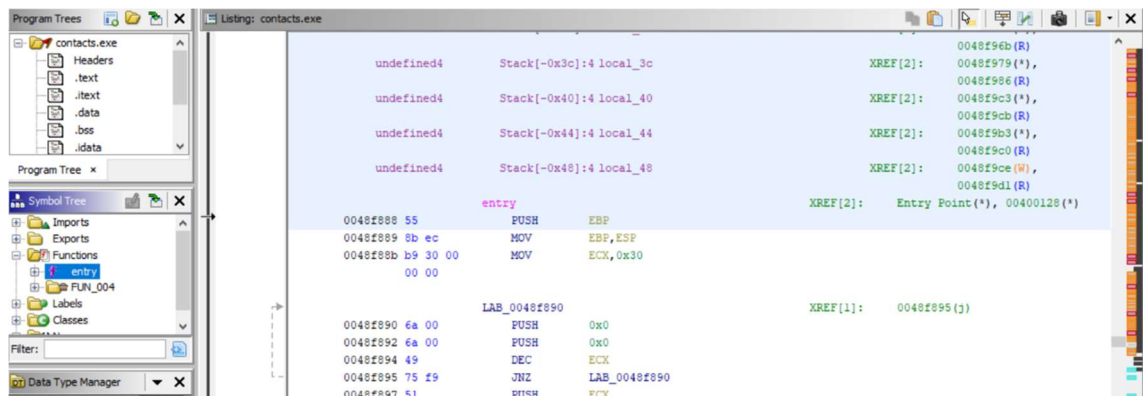


Figure 13 - Entry HEX location in GHIDRA

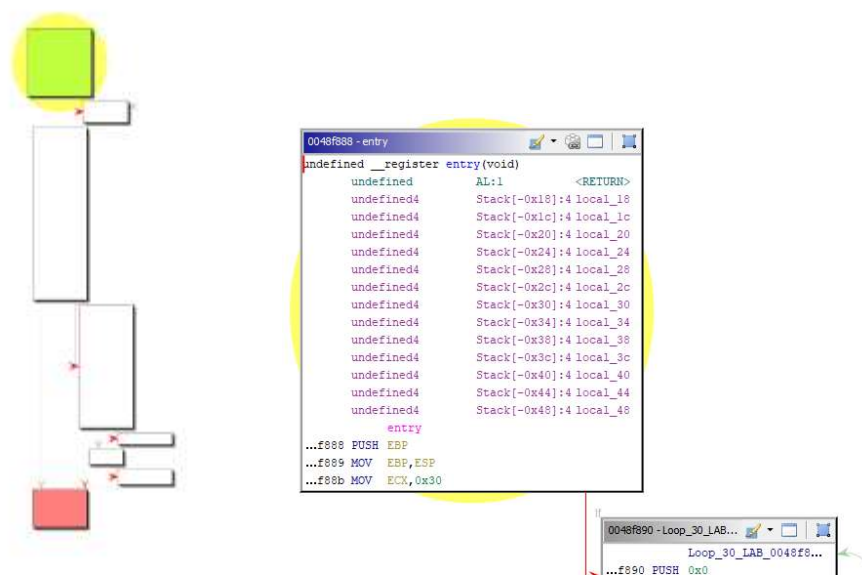
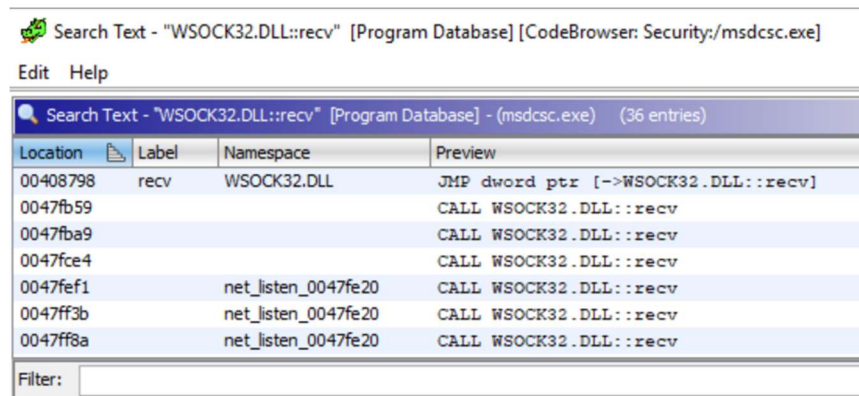


Figure 14 - Entry Function Diagram in GHIDRA



The communication between the victim computer and the C&C operator is encrypted and the student can continue to leverage tools like GHIDRA and IDA Free through advanced static analysis techniques and possibly advanced dynamic analysis techniques to find this additional information. After determining the start function of the code, the student can follow the code branches to find where the victim computer listens for C&C operator commands. Tools like GHIDRA and IDA Free have visual graphing to help determine these branches and follow the logic. Also, the tools can search for different types of commands like send and recv commands to find locations where the code will be listening and sending commands as shown in the following examples.



Search Text - "WSOCK32.DLL::recv" [Program Database] [CodeBrowser: Security/msdcsc.exe]

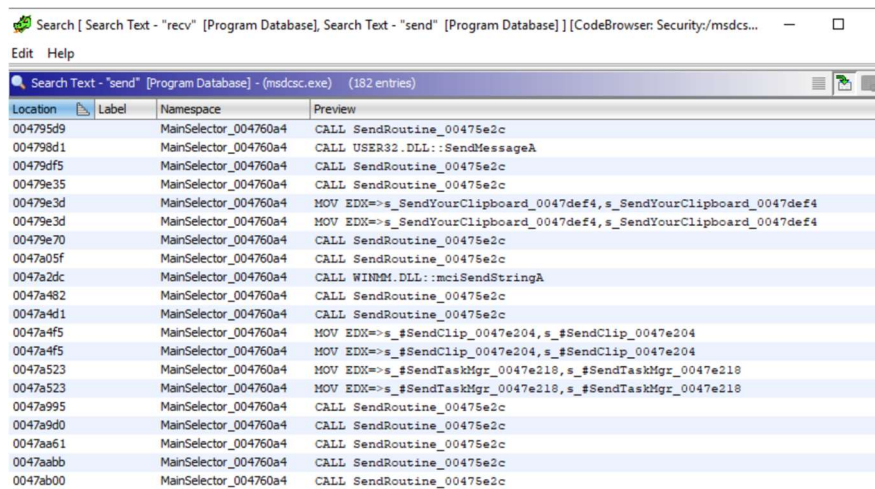
Edit Help

Search Text - "WSOCK32.DLL::recv" [Program Database] - (msdcsc.exe) (36 entries)

Location	Label	Namespace	Preview
00408798	recv	WSOCK32.DLL	JMP dword ptr [->WSOCK32.DLL::recv]
0047fb59			CALL WSOCK32.DLL::recv
0047fba9			CALL WSOCK32.DLL::recv
0047fce4			CALL WSOCK32.DLL::recv
0047fef1		net_listen_0047fe20	CALL WSOCK32.DLL::recv
0047ff3b		net_listen_0047fe20	CALL WSOCK32.DLL::recv
0047ffa8		net_listen_0047fe20	CALL WSOCK32.DLL::recv

Filter:

Figure 15 - String Search for WSOCK32.DLL::recv



Search [ Search Text - "recv" [Program Database] Search Text - "send" [Program Database] ] [CodeBrowser: Security/msdcsc.exe]

Edit Help

Search Text - "send" [Program Database] - (msdcsc.exe) (182 entries)

Location	Label	Namespace	Preview
004795d9		MainSelector_004760a4	CALL SendRoutine_00475e2c
004798d1		MainSelector_004760a4	CALL USER32.DLL::SendMessageA
00479df5		MainSelector_004760a4	CALL SendRoutine_00475e2c
00479e35		MainSelector_004760a4	CALL SendRoutine_00475e2c
00479e3d		MainSelector_004760a4	MOV EDX=>s_SendYourClipboard_0047def4,s_SendYourClipboard_0047def4
00479e3d		MainSelector_004760a4	MOV EDX=>s_SendYourClipboard_0047def4,s_SendYourClipboard_0047def4
00479e70		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047a05f		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047a2dc		MainSelector_004760a4	CALL WINRM.DLL::mciSendStringA
0047a482		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047a4d1		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047a4f5		MainSelector_004760a4	MOV EDX=>s_SendClip_0047e204,s_SendClip_0047e204
0047a4f5		MainSelector_004760a4	MOV EDX=>s_SendClip_0047e204,s_SendClip_0047e204
0047a523		MainSelector_004760a4	MOV EDX=>s_SendTaskMgr_0047e218,s_SendTaskMgr_0047e218
0047a523		MainSelector_004760a4	MOV EDX=>s_SendTaskMgr_0047e218,s_SendTaskMgr_0047e218
0047a995		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047a9d0		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047aa61		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047aabb		MainSelector_004760a4	CALL SendRoutine_00475e2c
0047ab00		MainSelector_004760a4	CALL SendRoutine_00475e2c

Figure 16 - String Search for send

In looking closer at examples of these send and receive calls, the student can find some consistent requests before or after that reference the encryption key.

```

...67c5 MOV EAX,[PTR_DAT_00494af0]
...67ca MOV EAX=>DAT_00499f58,dword ptr...
...67cc PUSH EAX
...67cd CALL WSOCK32.DLL::recv
...67d2 MOV dword ptr [EBP + local_c],...
...67d5 CMP dword ptr [EBP + local_c],...
...67d9 JG Decrypt_Received_LAB_004867e7

While Loop

004867e7 - Decrypt_Received_LAB_...
Decrypt_Received_L...
...67e7 LEA EAX=>local_2048,[EBP + 0xf...
...67ed LEA EDI=>local_2020,[EBP + 0xf...
...67f3 CALL FUN_00405e3c
...67f8 MOV EAX,dword ptr [EBP + local...
...67fe LEA EDI=>local_2044,[EBP + 0xf...
...6804 CALL FUN_00409bd0
...6809 MOV EDI=>local_2044,dword ptr ...
...680f LEA EAX=>local_8,[EBP + -0x4]
...6812 CALL FUN_004057d4
...6817 LEA EAX=>local_204c,[EBP + 0xf...
...681d MOV EDI,dword ptr [PTR_PTR_s_#...
...6823 MOV EDI=>PTR_s_#KCMDDC51#-_004...
...6825 MOV EAX,dword ptr [EBP + local...
...6828 CALL Decrypt01_004616b4
...682d MOV EDI,dword ptr [EBP + local...
...6833 LEA EAX=>local_8,[EBP + -0x4]
...6836 CALL FUN_004055c8

```

Figure 17 - Key Following Recv: the key string “#KCMDDC51#...” is used to call Decrypt() after recv() is called



Figure 18 - Key Preceding Send: the key string “#KCMDDC51#...” is used to call Encrypt() before send() is called

The commands which are sent must first be encrypted at the victim computer; while the commands which are received must be decrypted afterwards. The code references these encryption processes and without fully investigating the encryption algorithms, indications of the encryption key itself can be identified within the code. This activity requires the analyst to follow along with the code through advanced dynamic analysis to see the key. In this process, the full key can be observed in use as shown below. IDA



The screenshot displays a debugger interface with three main panels:

- Assembly View (Left):** Shows a list of assembly instructions. The instruction at address 0040672C is highlighted: `CALL [JMP, wsock32.recv]`. A blue arrow labeled "recv Command" points to this instruction.
- Registers (FPU) View (Right):** Shows the state of various registers. The `ES` register is highlighted with the value `00000000`. A blue arrow labeled "Encryption Key" points to this value.
- Memory View (Bottom):** Shows a hex dump of memory starting at address 00406740. The first few bytes are `01B8F740`, `0139F918`, `ASCI "B46084040FDE5E484694F904507440743EC8B121737"`, and `ERR:01B8F740`.

31

In addition, as the student is looking through the code for the rcv commands and looking for a location where the C&C operator's commands could be managed, the student can find many additional strings and values represent the many functions of the application.

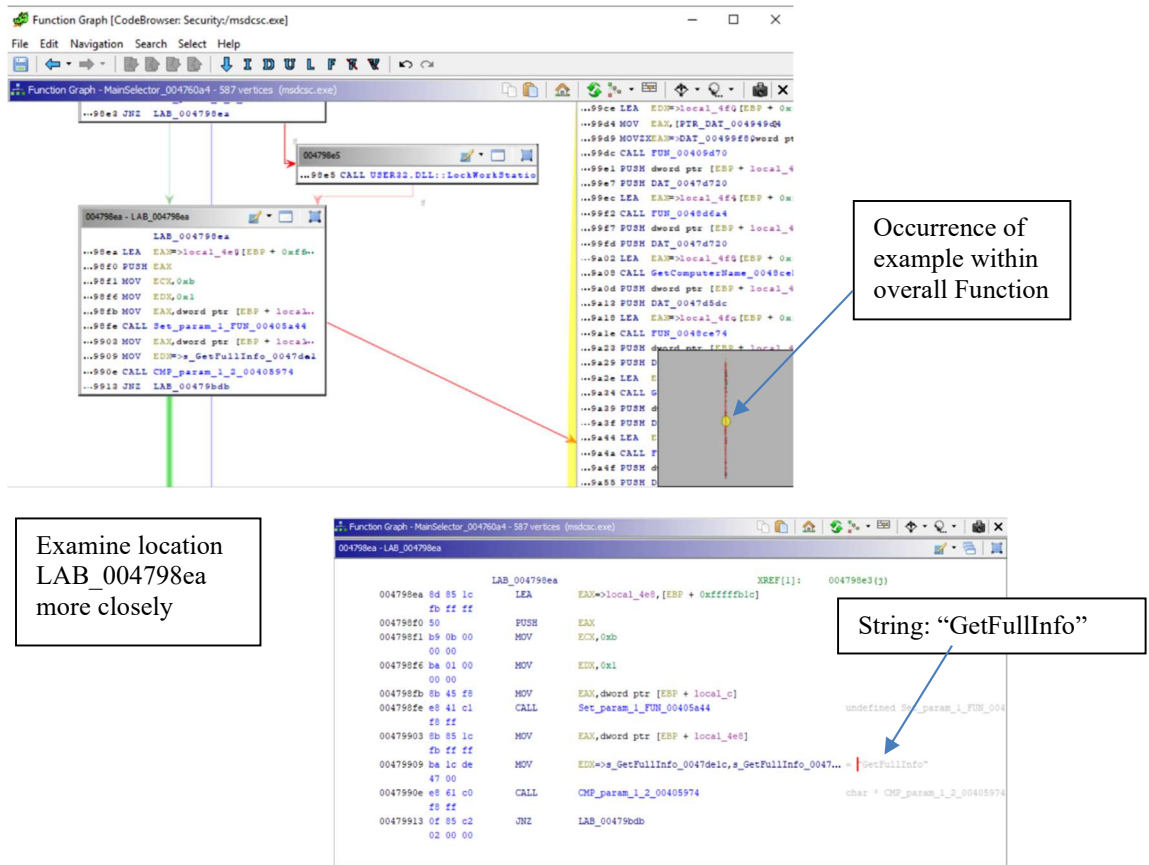


Figure 20 - Main Selector Function

As shown in Figure 20, the function which is renamed to MainSelector is a long function. This function tests multiple values to determine which action to take. In the analysis of the code, the student can find strings like "GetFullInfo" and this example shows what code will be executed when this string is matched. As the malware tests each value with the command it received from the C&C operator, it determines what action should be taken. In the above case, the function requests more information about the computer. This same process can be done to map functions within the malware.

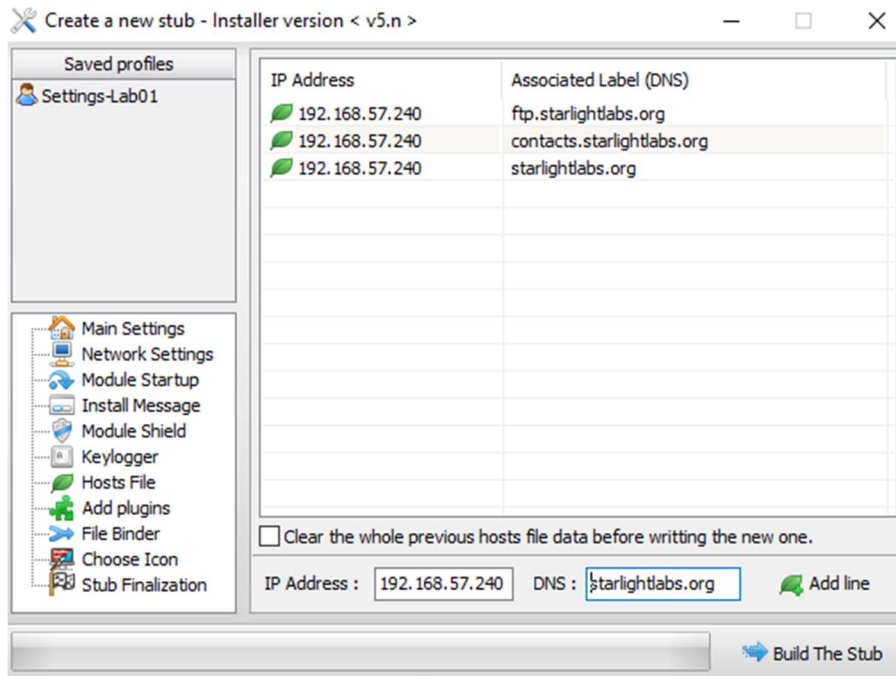
The tools leveraged for the Declination Lab are going to be determined by the student's preferences as well as the student's interest in exploring and learning more about all the tools available. The instructor can provide additional suggestions or hints within the Student Lab Manual, if desired, to direct students to utilize specific tools, if that is needed. The instructor's discretion on providing more assistance with the tools is an important aspect for the student's success in using the labs.

### **4.3 G3 Results**

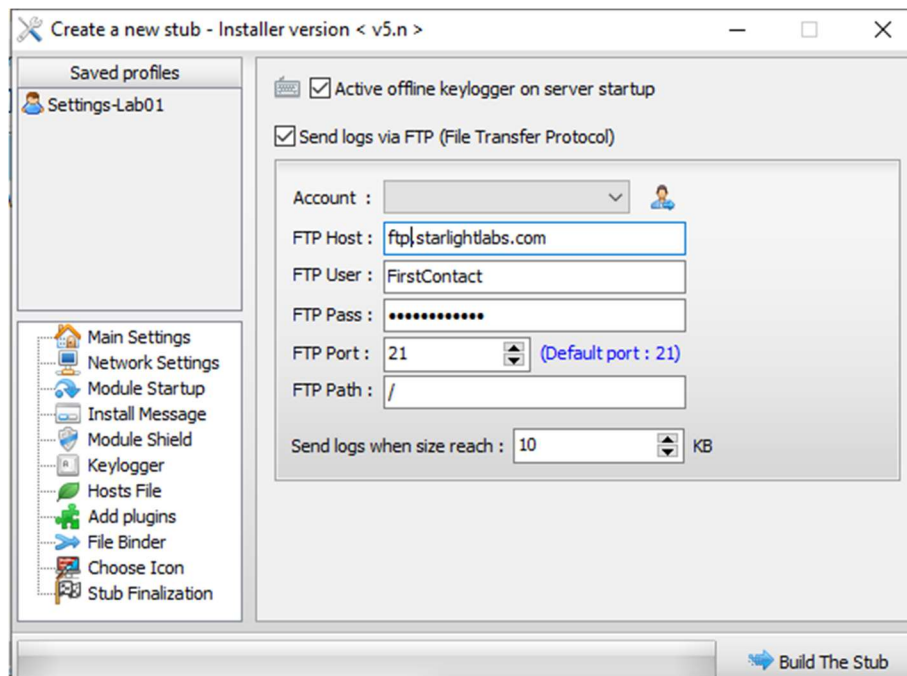
Overall, the demonstration of the G3 results can be summarized in the building out of 5 separate labs. Each lab was modified in multiple ways and demonstrate the labs are sustainable, extensible and customizable. Consider the malware executable, which is modified to create each lab instance. IP addresses have been changed as well as DNS names. Hosts files have been modified to reflect the different changes. Persistence has been enabled to provide a consistent repeatable experience after operating system restarts. This is only a sampling of the changes which are available to the instructor for many more labs as well. To demonstrate in more specific manner the results of G3, we can continue to focus on the 2 labs, Blueshift and Declination.

In the Blueshift Lab, as it is currently captured, we have an example with the G3 results of both what to do and what not to do in the development of future labs. The malware executable was created with the msdsc.exe filename which is the default filename of the malware. This filename, if not changed, could be searched on the internet to quickly find the source of the malware. In other labs, the malware executable is renamed, like in Apogee, to security.exe and in Declination to contacts.exe. The malware network traffic

in Blueshift is generated toward a DNS name of `appcenter.starlightlab.org` while in the Apogee Lab, the malware reaches out to `ftp.threathuntinglabs.com`. This change of the domain name are all configurable during the build out of the malware executable. Other examples of the malware buildout are captured in the below figures demonstrating a set of hosts file entries and a keylogger target DNS name as found in the Declination Lab.



*Figure 21 - Hosts File Values*



*Figure 22 - Keylogger Exfiltration Target*

In addition to the examples provided in the Blueshift Lab, the Declination Lab also incorporates an example of the G3 results to adjust the skill level of the lab. This could be useful to accommodate a student performing the lab who may not be achieving success. The example can be seen in the optional Goal and Objective:

5. [Optional] Select and analyze 2 primary functions available to the malware C&C Client based on Advanced Static Analysis and/or Advanced Dynamic Analysis. Find the cross reference (xref) call instructions for each of the 2 chosen functions.

If the student is not able to effectively complete these tasks as designed, the skill level can be adjusted by the instructor by providing the student with the C&C Client. This activity would reduce the difficulty of the lab by allowing the student to generate the interaction with the malware on the victim computer and thereby know what type of behavior is expected to be seen. With this additional knowledge, the student would be

expected to move forward more easily in the process of analyzing the malware and more readily achieve the goal and objective.

#### 4.4 Additional Technical Background

More detailed command and control functionality is illustrated in the following additional technical background of the malware. As a recap, the list of functions is broad and includes many different ways to attack the victim's computer, assess the nearby network devices and to observe and interact with the operator of the victim computer.

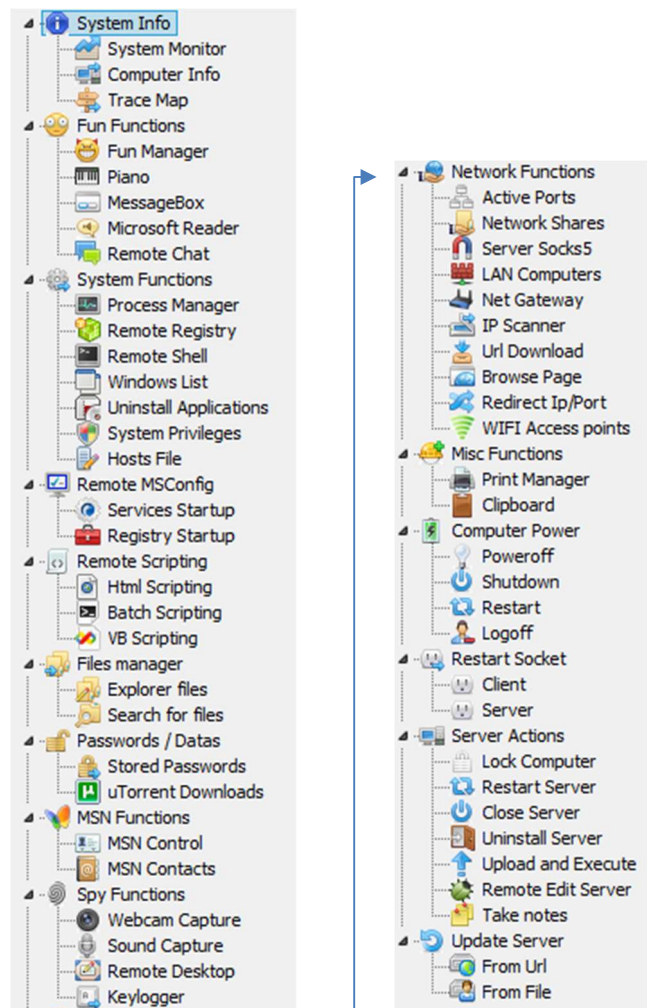


Figure 23 - List of Functions

The System Info functions provide insight into the victim's computer details including potentially Personal Identifiable Information (PII) as shown in Figure 24.

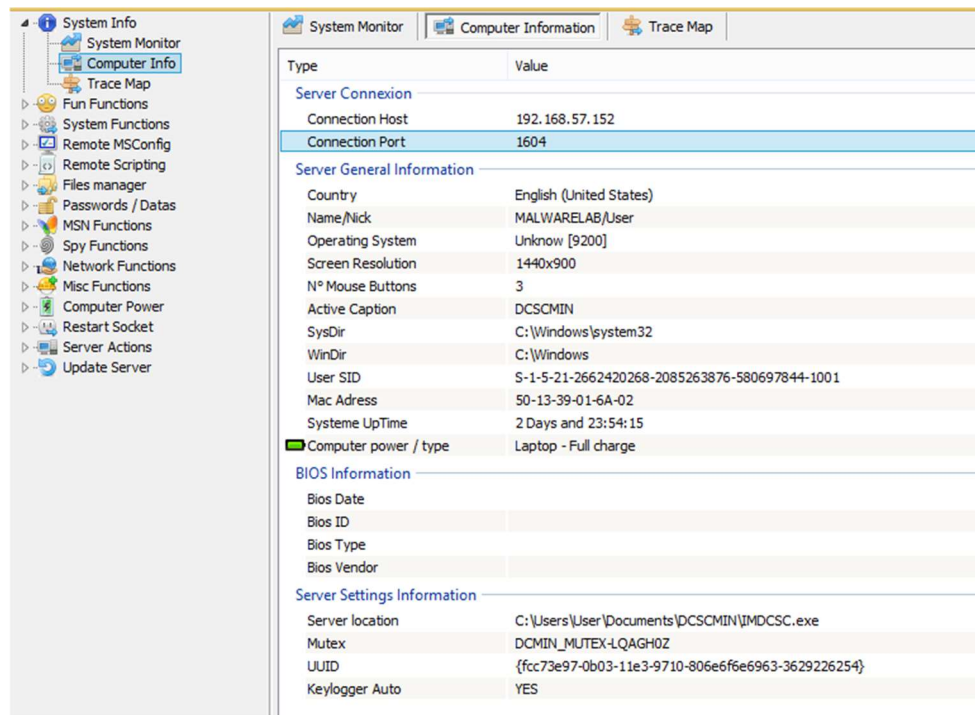


Figure 24

The Fun Functions provide some interesting options grouped into 4 areas: Fun Manager, Piano, Message Box, Microsoft Reader and Remote Chat. The Fun Manager (Figure 25) includes easy ways to hide the desktop, clock, task icons, task manager and even open and close the CD door (which may not be quite as relevant today, but some computers do still ship with CD drives). The MessageBox function provides the victim with a message box designed by the attacker (Figure 26). The Remote Chat function provides the attacker with a two-way real-time chat capability to interact directly with the operator of the victim computer (Figure 27).



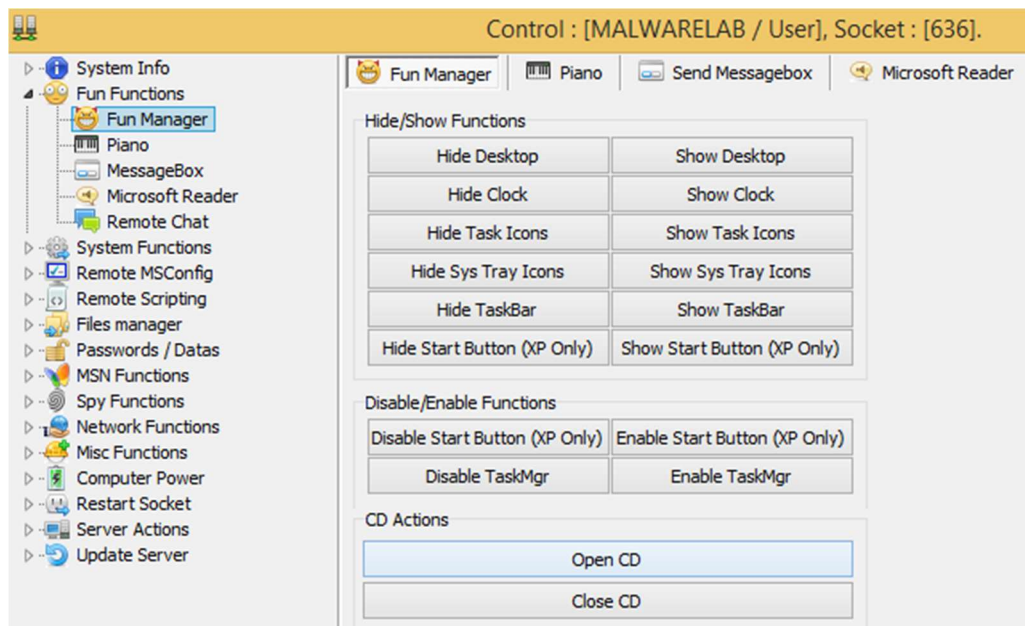


Figure 25 - Fun Manager

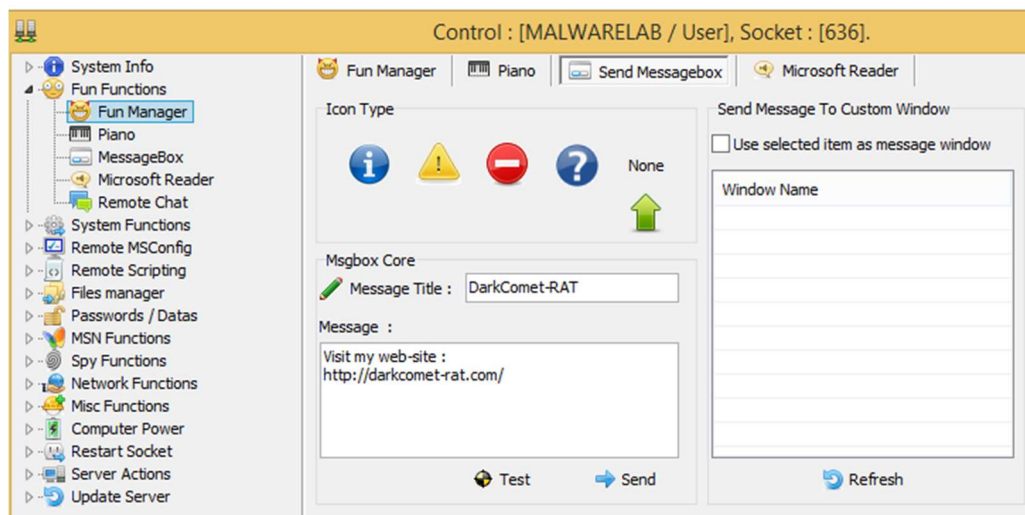


Figure 26 - Fun Manager Send Messagebox



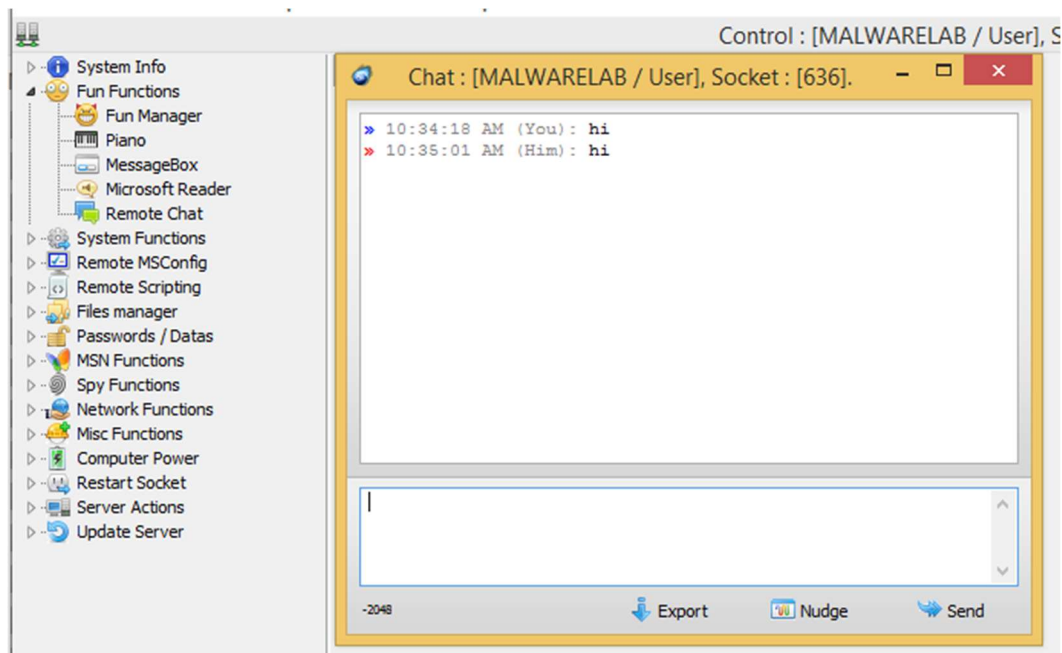


Figure 27 - Fun Manager Chat

The System Functions include Process Manager (Figure 28), Remote Registry (Figure 29), Remote Shell (Figure 30), Windows List, Uninstall Applications (Figure 31), System Privileges, and Hosts File (Figure 32).

Process Name	Process Path	N° T...	PID	User / Domain	Pare...	Size	Priority
[System Process]	ACCESS DENIED (x64)	1	0	-/-	0	0.00 ...	-
csrss.exe	ACCESS DENIED (x64)	8	332	-/-	324	0.00 ...	Hight
csrss.exe	ACCESS DENIED (x64)	9	396	-/-	388	0.00 ...	Hight
dwm.exe	ACCESS DENIED (x64)	8	692	-/-	432	0.00 ...	Hight
explorer.exe	C:\Windows\Explorer.EXE	73	304	User\MalwareLab	228	98.0...	Normal
IMDSC.exe	C:\Users\User\Documents\PCSCMIN\IMDSC.exe	8	2668	User\MalwareLab	304	17.3...	Normal
lsass.exe	ACCESS DENIED (x64)	5	500	-/-	404	0.00 ...	-
SearchIndexer.exe	ACCESS DENIED (x64)	13	1496	-/-	492	0.00 ...	Normal
services.exe	ACCESS DENIED (x64)	1	492	-/-	404	0.00 ...	-
smss.exe	ACCESS DENIED (x64)	2	260	-/-	4	0.00 ...	-
spoolsv.exe	ACCESS DENIED (x64)	8	1124	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	10	1732	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	14	1024	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	17	864	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	18	800	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	21	1164	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	3	1512	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	34	840	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	5	588	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	8	556	-/-	492	0.00 ...	Normal
svchost.exe	ACCESS DENIED (x64)	9	928	-/-	492	0.00 ...	Normal
System	ACCESS DENIED (x64)	78	4	-/-	0	0.00 ...	Normal
SystemSettings.exe	C:\Windows\ImmersiveControlPanel\SystemSettings.exe	17	2804	User\MalwareLab	556	27.0...	Normal
taskhost.exe	C:\Windows\system32\taskhost.exe	9	1996	User\MalwareLab	840	4.23 ...	Normal
VBoxService.exe	ACCESS DENIED (x64)	10	708	-/-	492	0.00 ...	Normal
VBoxTray.exe	C:\Windows\System32\VBoxTray.exe	10	1948	User\MalwareLab	304	2.09 ...	Normal
wininit.exe	ACCESS DENIED (x64)	1	404	-/-	324	0.00 ...	Hight
winlogon.exe	ACCESS DENIED (x64)	2	432	-/-	388	0.00 ...	Hight
wmpnetwk.exe	ACCESS DENIED (x64)	8	2216	-/-	492	0.00 ...	Normal

Figure 28 - Process Manager

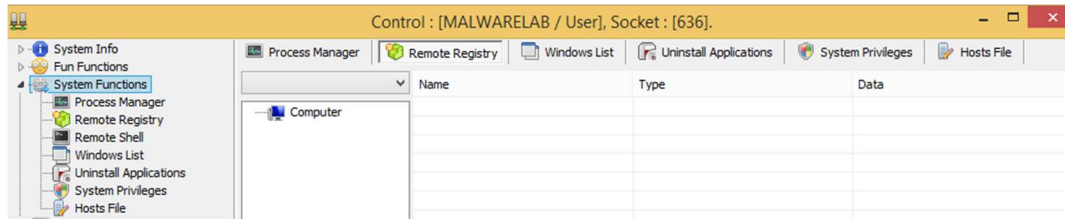


Figure 29 - Remote Registry

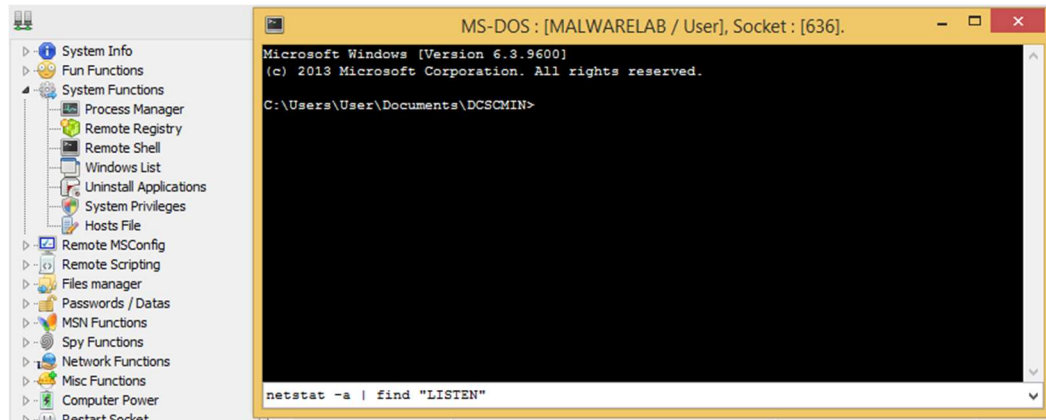


Figure 30 - Remote Shell

N° of app : 23						
Key Name	Display Name	Version	Path	Publisher	Uninstall String	
7-Zip	7-Zip 18.01	18.01	C:\Program Files\7-Zip\	Igor Pavlov	C:\Program Files\7-	
AddressBook	-	-	-	-	-	
Connection Manager	-	-	-	-	-	
DirectDrawEx	-	-	-	-	-	
DXM_Runtime	-	-	-	-	-	
Fontcore	-	-	-	-	-	
IE40	-	-	-	-	-	
IE4Data	-	-	-	-	-	
IESBAKEX	-	-	-	-	-	
IEData	-	-	-	-	-	
MobileOptionPack	-	-	-	-	-	
Mozilla Firefox 58.0.2 (x86 en-US)	Mozilla Firefox 58.0.2 (x86 en-US)	58.0.2	C:\Program Files\Mozila ...	Mozilla	"C:\Program Files\...	
MozillaMaintenanceService	Mozilla Maintenance Service	58.0.2	-	Mozilla	"C:\Program Files\...	
MPlayer2	-	-	-	-	-	
Oracle VM VirtualBox Guest Additions	Oracle VM VirtualBox Guest Additions 5.2.7	5.2.7.0	-	Oracle Corpora...	C:\Program Files\O...	
SchedulingAgent	-	-	-	-	-	
USBPcap	USBPcap 1.1.0.0-g794bf26-5	1.1.0.0-g794bf26-5	-	-	"C:\Program Files\U...	
WIC	-	-	-	-	-	
WinPcapInst	WinPcap 4.1.3	4.1.0.2980	-	Riverbed Tech...	C:\Program Files\W...	
Wireshark	Wireshark 2.2.12 (32-bit)	2.2.12	C:\Program Files\Wiresh...	The Wireshark ...	"C:\Program Files\...	
{35083883-40fa-423c-ae73-2aff7e...	Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.40649	12.0.40649.5	-	Microsoft Corp...	"C:\ProgramData\I...	
{A8589745-51BC-3963-64E9-201C...	Microsoft Visual C++ 2013 x86 Additional Runtime - 12.0.40649	12.0.40649	-	Microsoft Corp...	MsExec.exe /X {A8...	
{DEA7F8E3-67B9-3C3C-945B-7F8C...	Microsoft Visual C++ 2013 x86 Minimum Runtime - 12.0.40649	12.0.40649	-	Microsoft Corp...	MsExec.exe /X {DE...	

Figure 31 - Uninstall Applications

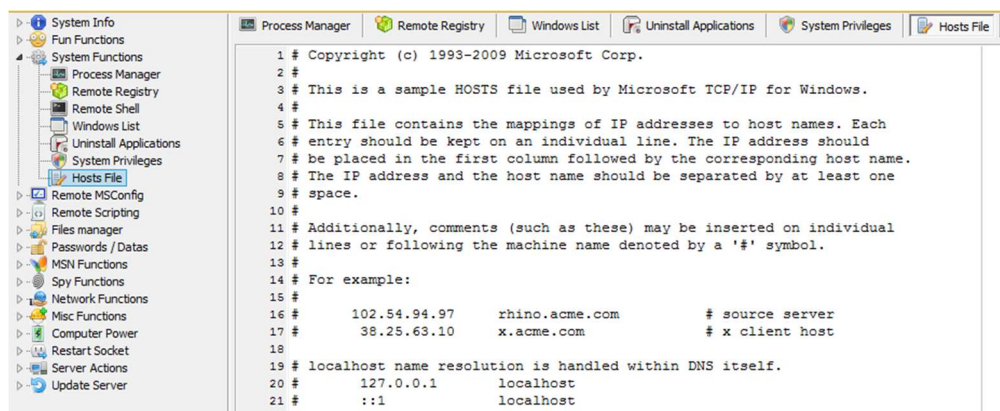


Figure 32 - Hosts File

Together, these System Functions provide the attacker with granular controls to add software, remove software, add registry entries, and modify the system as desired.

The Remote MSConfig functions provide a list of Services (Figure 33) and Registry (Figure 34) items which run at startup. One of the methods for persistence is captured in Figure 34 as the malware added a registry key to run again at startup.

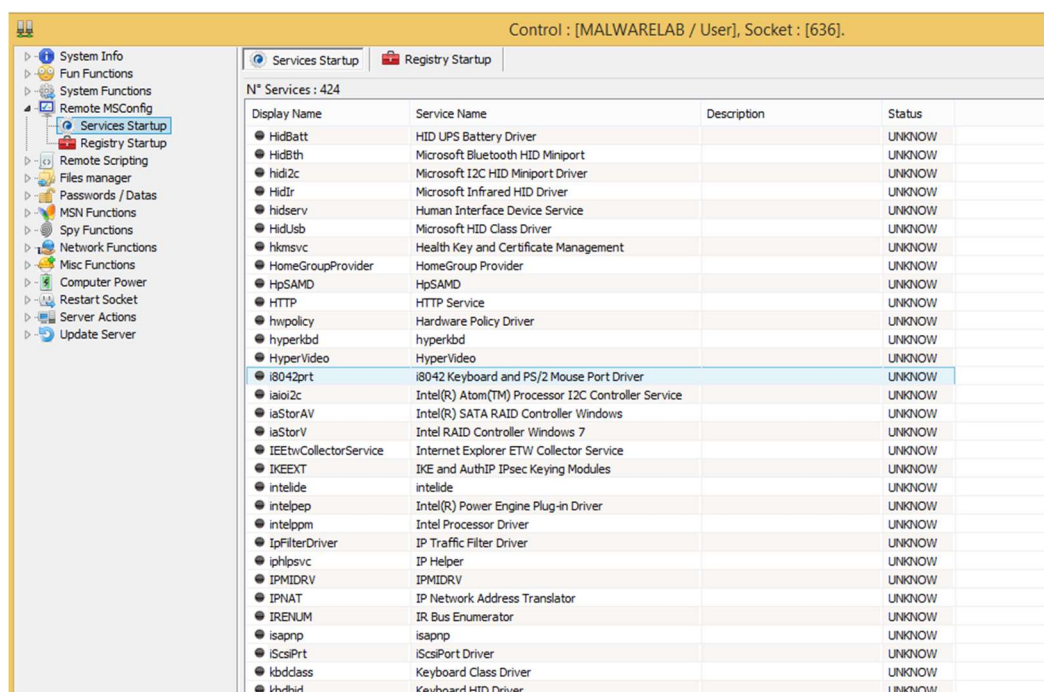


Figure 33 - Services Startup

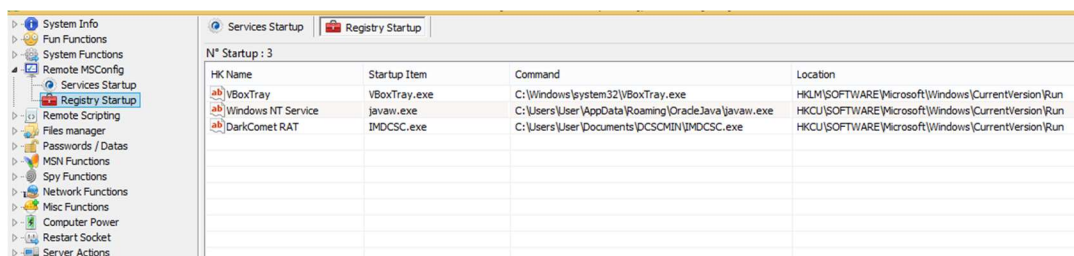


Figure 34- Registry Startup

Remote Scripting (Figure 35) provides HTML, Batch Scripting and Visual Basic Scripting options for the attacker.

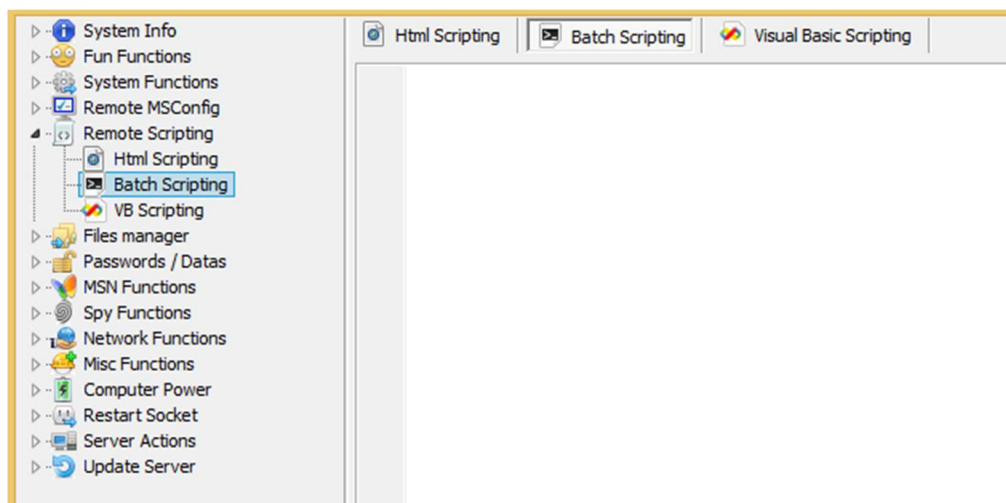
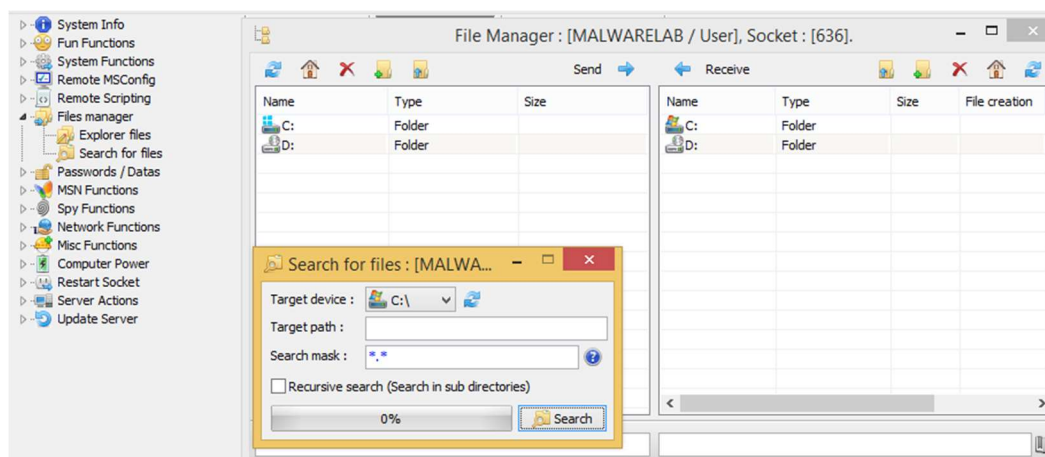


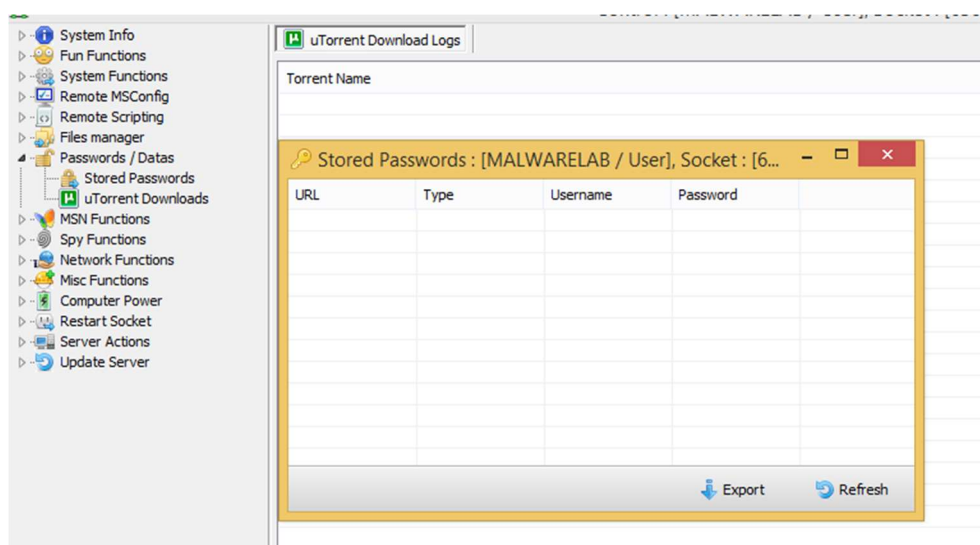
Figure 35 - Remote Scripting

The Files Manager (Figure 36) has 2 options: Explorer files and Search for files. The Explorer files option provides a window to browse through files both on the remote victim computer as well as the attacker's computer and to drag and drop files between the systems. It also provides similar functionality to create folders, as well as delete folders and files.



*Figure 36 - Files Manager*

The functions for Passwords / Datas (Figure 37) provides Stored Passwords and uTorrent Download logs.



*Figure 37 - Passwords*

MSN Functions (Figure 38) have become somewhat outdated, but you can imagine the impact this could have on the victim as the attacker impersonates the victim to external parties on MSN. As the attacker has full access to the computer, the capabilities of MSN are placed here for convenience, but the attacker can impersonate the victim through any applications via remote desktop, remote shell, etc.



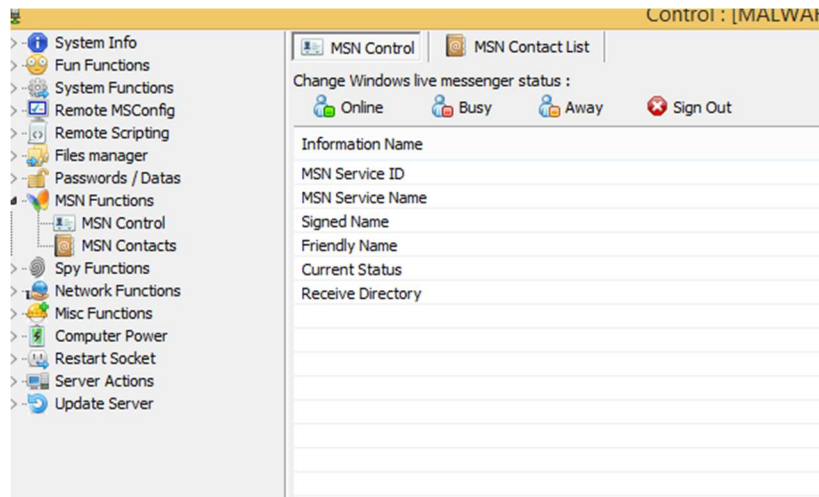


Figure 38 - MSN Functions

The Spy Functions (Figures 39 & 40) provide the most intrusive and direct observations of the victim using the Webcam, Microphone, Remote Desktop, and Keylogger. These options allow the attacker to have access to the victim's camera and microphone which enables the attacker to record video and sound from the victim's computer. This video and sound can be used to frighten and intimidate the victim as well as enable the attacker to extort, blackmail, or otherwise exploit the victim.

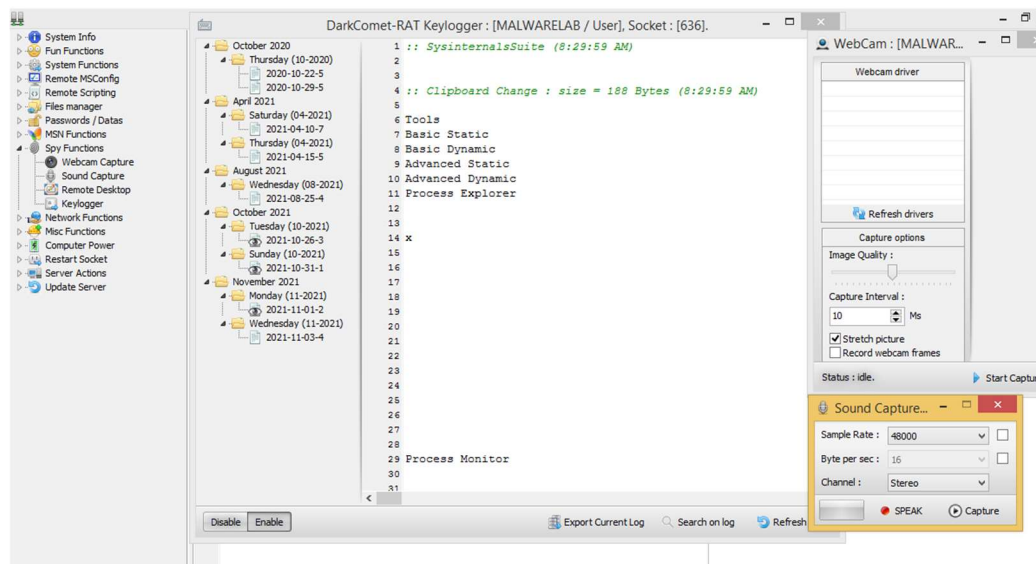
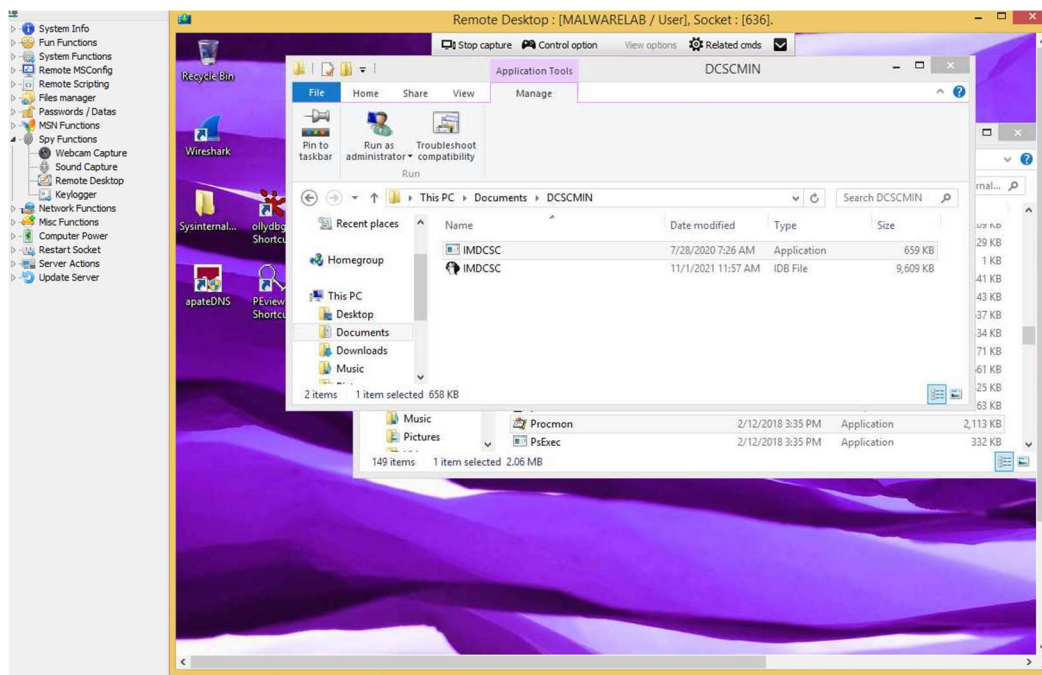


Figure 39 - Spy Functions - WebCam, Microphone, Keylogger



*Figure 40 - Spy Functions - Remote Desktop*

The next group of functions, Network functions, provide insight to the attacker of the victim's network. This can provide possible additional target systems to attack as well as provide more information to the devices and systems which are on the same network as the victim. Lateral attacks can be orchestrated directly from the victim computer or through alternative attack methods.

The Active Ports (Figure 41) provides the victim's active network ports.

Name	PID	Protocol	Local IP	Local Port	Remote IP	Remote Port	Status
svchost.exe	588	TCP	0.0.0.0	135	0.0.0.0	0	LISTENING
System	4	TCP	0.0.0.0	445	0.0.0.0	0	LISTENING
wmpnetwk.exe	2216	TCP	0.0.0.0	554	0.0.0.0	0	LISTENING
System	4	TCP	0.0.0.0	2869	0.0.0.0	0	LISTENING
System	4	TCP	0.0.0.0	10243	0.0.0.0	0	LISTENING
wininit.exe	404	TCP	0.0.0.0	49152	0.0.0.0	0	LISTENING
svchost.exe	800	TCP	0.0.0.0	49153	0.0.0.0	0	LISTENING
svchost.exe	840	TCP	0.0.0.0	49154	0.0.0.0	0	LISTENING
spoolsv.exe	1124	TCP	0.0.0.0	49155	0.0.0.0	0	LISTENING
services.exe	492	TCP	0.0.0.0	49156	0.0.0.0	0	LISTENING
svchost.exe	1512	TCP	0.0.0.0	49157	0.0.0.0	0	LISTENING
lsass.exe	500	TCP	0.0.0.0	49158	0.0.0.0	0	LISTENING
System	4	TCP	192.168.57.151	139	0.0.0.0	0	LISTENING
lsmcs.exe	2668	TCP	192.168.57.151	60438	192.168.57.152	1604	ESTABLISHED
lsmcs.exe	2668	TCP	192.168.57.151	60445	192.168.57.152	1604	ESTABLISHED
lsmcs.exe	2668	TCP	192.168.57.151	60446	192.168.57.152	1604	ESTABLISHED
svchost.exe	588	TCP	~::~:	135	~::~:	0	LISTENING
System	4	TCP	~::~:	445	~::~:	0	LISTENING
wmpnetwk.exe	2216	TCP	~::~:	554	~::~:	0	LISTENING
System	4	TCP	~::~:	2869	~::~:	0	LISTENING
System	4	TCP	~::~:	10243	~::~:	0	LISTENING
wininit.exe	404	TCP	~::~:	49152	~::~:	0	LISTENING
svchost.exe	800	TCP	~::~:	49153	~::~:	0	LISTENING
svchost.exe	840	TCP	~::~:	49154	~::~:	0	LISTENING
spoolsv.exe	1124	TCP	~::~:	49155	~::~:	0	LISTENING
services.exe	492	TCP	~::~:	49156	~::~:	0	LISTENING
svchost.exe	1512	TCP	~::~:	49157	~::~:	0	LISTENING
lsass.exe	500	TCP	~::~:	49158	~::~:	0	LISTENING
svchost.exe	864	UDP	0.0.0.0	123	*	*	
svchost.exe	840	UDP	0.0.0.0	500	*	*	
svchost.exe	840	UDP	0.0.0.0	4500	*	*	
wmpnetwk.exe	2216	UDP	0.0.0.0	5004	*	*	
wmpnetwk.exe	2216	UDP	0.0.0.0	5005	*	*	

Figure 41 - Network Active Ports

Network Shares (Figure 42) provides a list of shares which are being provided to the network from the victim's computer. This could provide an attack vector to replace a file or upload a new file to a network share which could be downloaded or run on another computer.

Name	Path	Type	Permission	Max users	Current Users	Comment	Password	Reserved
A	C:\	2147483648	0	4294967295	0	R	--/--	0
C	C:\	2147483648	0	4294967295	0	D	--/--	0
I	--/--	2147483651	0	4294967295	0	R	--/--	0
U	C:\	0	0	4294967295	0	--/--	--/--	100

Figure 42 - Network Shares

The Network Function, LAN Computers, provides a list of networked IP addresses on the local area network, which provides a list of potential devices to attack (Figure 43).



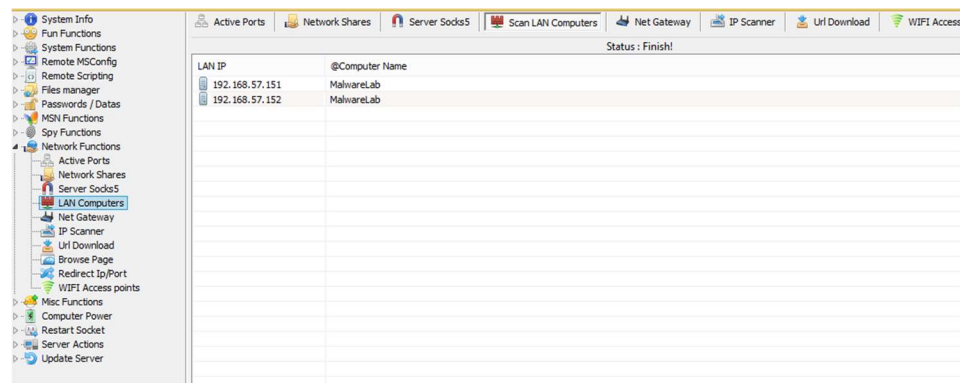


Figure 43 - LAN Computers

The Print Manager and Clipboard functions are listed in the Misc Functions (Figure 44). These tools provide access to the clipboard on the victim computer to see what has been copied to the clipboard as well as create (Write in) new clipboard values as well as clear the remote clipboard.

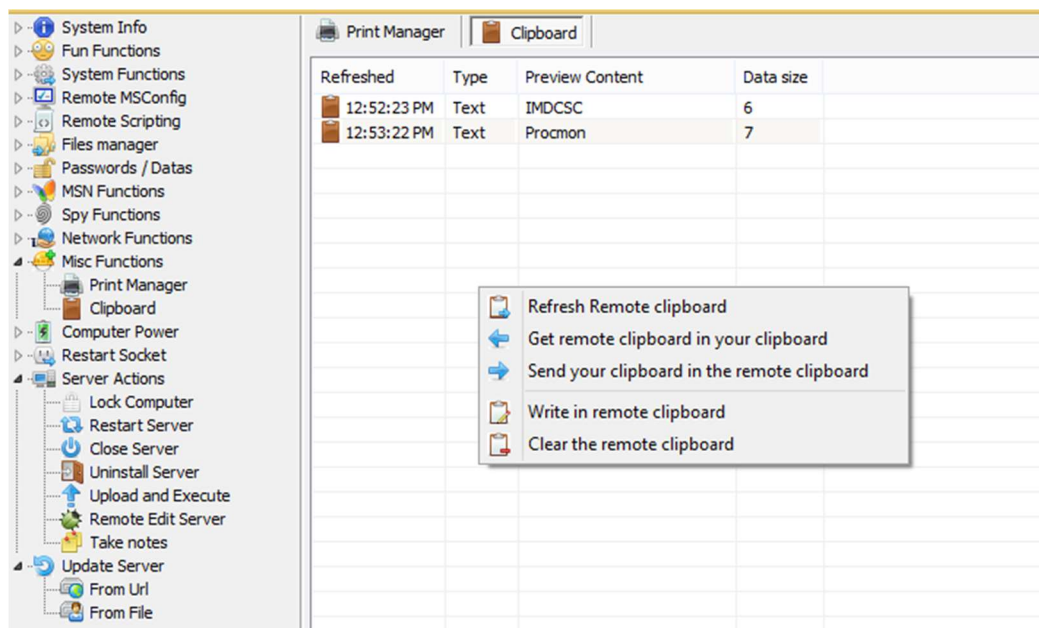
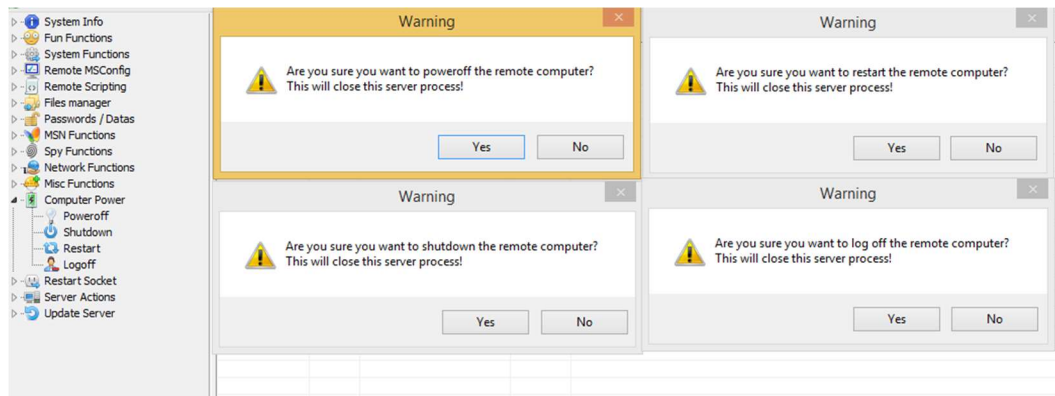


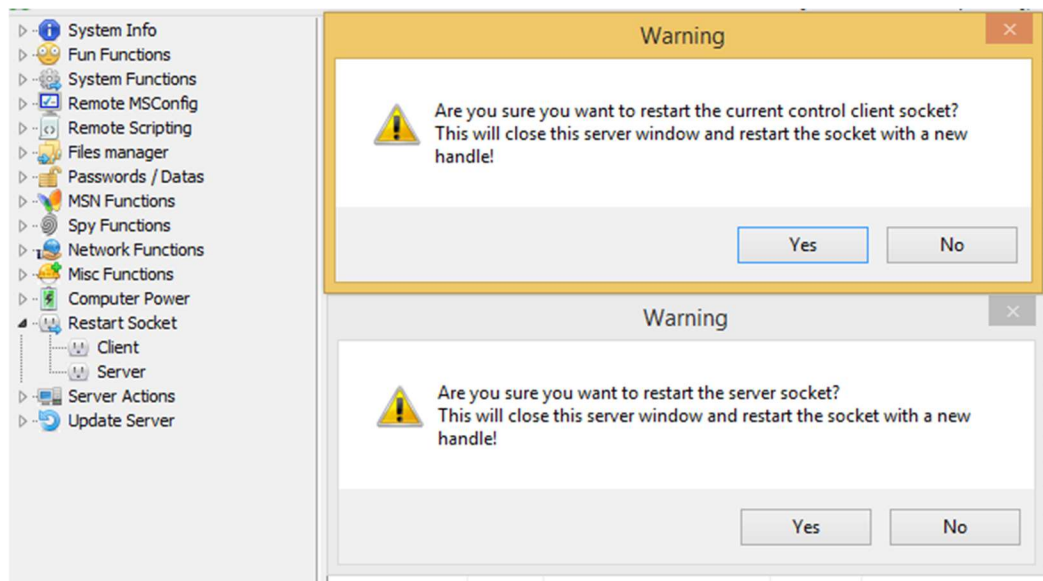
Figure 44 - Misc Functions - Print Manager and Clipboard

The Computer Power functions (Figure 45) include Poweroff, Shutdown, Restart and Logoff.



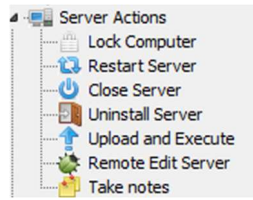
*Figure 45 - Computer Power*

Both the Client and Server have Restart Socket functions which warn the attacker that this action will create a new handle (Figure 46).



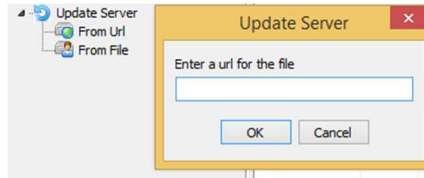
*Figure 46 - Restart Socket (Client and Server)*

The Server Actions functions each perform functions associated to the Windows Server Capabilities including Lock Computer, Restart Server, Close Server, Uninstall Server, Upload and Execute, Remote Edit Server and Take Notes.



*Figure 47 - Server Actions and Update Server*

The Update Server Actions provide both a URL option and a File option (Figure 48).



*Figure 48 - Update Server*

Some more nefarious functions include Dynamic Denial of Service (DDoS) commands: Http flood, Syn flood and UDP flood, as shown in Figure 49. Other research has shown that these functions at different points were not properly functional [19], but nonetheless, their appearance questions the motivations of the developer [21]. Further, the malware has been used in infamous cases, like the Syrian Government's use in 2012 [10][11] which demonstrate that even if the design was with good intent, it was used maliciously.

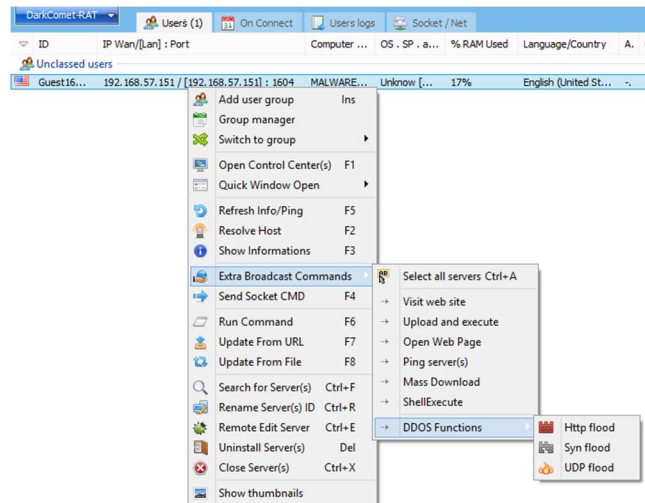


Figure 49 - DDoS Functions

## Chapter 5: Conclusion

### 5.1 Summary of results and alignment to Goals/Objectives

This research paper is intended to help develop and improve the skills which are needed to perform effective cyber threat hunting and further increase the capabilities of our workforce while leveraging free software and tools to help reduce the financial investment required to learn and practice these techniques. To achieve three goals:

- Tailored for Different Skill Levels,
- Accessible/Free, and
- Safely Sustainable/ Extensible/ Customizable,

the labs have established objectives which incorporate Bloom's Taxonomy, the Threat Hunting Skill Set, and the use of Questions and Manuals to encourage the student's investigation. The labs utilize a malware which demonstrates many different capabilities and provides opportunities for students of many different skill levels. From these efforts,

students can continue to practice skills and develop new capabilities in Cyber Threat Hunting.

## **5.2 Possible future work**

As this work is further developed, the automation of variable changes, additional lab buildouts and extending new free tools and software as they become available can help enhance and keep the labs relevant to the always changing landscape of cyber threat hunting.

## References

- [1] Wei, J. Chu, B. Cranford-Wesley, D. & Brown, J. "A Laboratory for Hands-on Cyber Threat Hunting Education." Journal of The Colloquium for Information Systems Security Education, Volume 7, No. 1
- [2] "Cyber Security Skills Roadmap." SANS Institute. <https://www.sans.org/curricula/incident-response-and-threat-hunting>. Last Accessed Dec 12, 2021.
- [3] Du, W. "SEED Project." Seed Labs. <https://seedsecuritylabs.org/>. Last Accessed Dec 12, 2021.
- [4] Hungenburg, T. & Eckert, M. "Requirements." INetSim: Internet Services Simulation Suite. <http://www.inetsim.org/requirements.html>. Last Accessed Dec 12, 2021.
- [5] "The Remote Access Trojan (RAT), a Legacy Product at a Mass Market Price." LOGPOINT. <https://www.secbi.com/the-remote-access-trojan-rat-a-legacy-product-at-a-mass-market-price/>
- [6] Doffman, Z. "Chinese Hackers 'Weaponize' Coronavirus Data For New Cyber Attack: Here's What They Did." FORBES. <https://www.forbes.com/sites/zakdoffman/2020/03/12/chinese-hackers-weaponized-coronavirus-data-to-launch-this-new-cyber-attack/#135714713861>. (2020) Last Accessed Dec 12, 2021.
- [7] Shamir, U. "The 7 'Most Common' RATS In Use Today." Informa. <https://www.darkreading.com/perimeter/the-7-most-common-rats-in-use-today-/a/d-id/1321965>. August 18, 2015. Last Accessed Dec 12, 2021.
- [8] Zeltser, L. Free Malware Sample Sources for Researchers. Lenny Zeltser Content Feed. <https://zeltser.com/malware-sample-sources/>. Last Accessed Dec 12, 2021.
- [9] DarkComet. (2021, November 10). In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=DarkComet&oldid=1054485014>. Last Accessed Dec 12, 2021.
- [10] "Spy code creator kills project after Syrian abuse." BBC News <https://www.bbc.com/news/technology-18783064>. Last Accessed Dec 12, 2021.
- [11] McMillan, R. "How the Boy Next Door Accidentally Built a Syrian Spy Tool." (2012, July 11). Wired. <https://www.wired.com/2012/07/dark-comet-syrian-spy-tool/>. Last Accessed Dec 12, 2021.
- [12] Farinholt, B. Rezaeirad, M. Pearce, P. Dharmdasani, H. Yin, H. Le Blondk, S. McCoy, D. & Levchenko, K. "To Catch a Ratter: Monitoring the Behavior of Amateur DarkComet RAT Operators in the Wild." (2017, May). *IEEE Security and Privacy (2005)*. [Online] Available: <http://damonmccoy.com/papers/rat-sp17.pdf>. Last Accessed Dec 12, 2021.
- [13] Lee, R. M. & Bianco, D. "Generating Hypotheses for Successful Threat Hunting," SANS Institute InfoSec Reading Room. (2016, August 15). [Online] Available: <https://www.sans.org/white-papers/37172/>. Last Accessed Dec 12, 2021.
- [14] Poremba, S. "Security Report: Finding the Right Balance of Automation and Human Interaction ." (2017, March 31). <https://www.hpe.com/us/en/newsroom/blog-post/2017/03/security-report-finding-the-right-balance-of-automation-and-human-interaction.html>. Last Accessed Dec 12, 2021.
- [15] Greenberg, A. "Beyond Kaseya: Everyday IT Tools Can Offer 'God Mode' for Hackers." (2021, July 12). Wired. <https://www.wired.com/story/it-management-tools-hacking-jamf-kaseya/>. Last Accessed Dec 12, 2021.
- [16] Arghire, I. "FBI Issues Alert on Use of Chinese Tax Software" (2020, July 27). SECURITYWEEK NETWORK. <https://www.securityweek.com/fbi-issues-alert-use-chinese-tax-software>. Last Accessed Dec 12, 2021.

- [17] "A 'Worst Nightmare' Cyberattack: The Untold Story Of The SolarWinds Hack." (2021, April 16). NPR. <https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack>. Last Accessed Dec 12, 2021.
- [18] Kujawa, A. "You dirty RAT! Part 1: DarkComet." (2012, June 9). Malwarebytes LABS. <https://blog.malwarebytes.com/threat-analysis/2012/06/you-dirty-rat-part-1-darkcomet>. Last Accessed Dec 12, 2021.
- [19] Edwards, J. "It's not the end of the world: DarkComet misses by a mile." (2012, March 13). Paper-The Essentials of Security Technology. [https://paper.seebug.org/papers/APT/APT\\_CyberCriminal\\_Campagin/2012/Crypto-DarkComet-Report.pdf](https://paper.seebug.org/papers/APT/APT_CyberCriminal_Campagin/2012/Crypto-DarkComet-Report.pdf). Last Accessed Dec 12, 2021.
- [20] Farinholt, B. R. "Understanding the Remote Access Trojan malware ecosystem through the lens of the infamous DarkComet RAT." (2019). UC San Diego. ProQuest ID: Farinholt\_ucsd\_0033D\_18531. Merritt ID: ark:/13030/m5kh5ppz. Retrieved from <https://escholarship.org/uc/item/3vv544n5>. Last Accessed Dec 12, 2021.
- [21] Kujawa, A. "You dirty RAT! Part 2: BlackShades Net." (2012, June 15). Malwarebytes LABS. <https://blog.malwarebytes.com/threat-analysis/2012/06/you-dirty-rat-part-2-blackshades-net/>. Last Accessed Dec 12, 2021.
- [22] Denbow, S. & Hertz, J. "Pest Control: Taming the Rats." Technical report, 2012. [Online] Available: <https://www.steptoocyberblog.com/files/2012/11/PEST-CONTROL1.pdf>. Last Accessed Dec 12, 2021.
- [23] Farinholt, B. Rezaeirad, M. McCoy, D. & Levchenko, K. (2020, April 20-24). "Dark Matter: Uncovering the DarkComet RAT Ecosystem." In WWW '20. <https://par.nsf.gov/servlets/purl/10176434>. Last Accessed Dec 12, 2021.
- [24] Lord, N. "What is Threat Hunting? The Emerging Focus in Threat Detection." (2018, September 11). Digital Guardian. <https://digitalguardian.com/blog/what-threat-hunting-emerging-focus-threat-detection>. Last Accessed Dec 12, 2021.
- [25] "Threat Hunting: Fad or Essential Cyber Security Tactic?" (2021, August 10). <https://www.infocyte.com/blog/2016/6/17/threat-hunting-fad-or-essential-cyber-security-tactic/>. Last Accessed Dec 12, 2021.
- [26] "WRITING MEASURABLE COURSE OBJECTIVES." The Center for Teaching and Learning. <https://teaching.charlotte.edu/teaching-guides/course-design/writing-measurable-course-objectives>. Last Accessed Dec 12, 2021.
- [27] Bloom, B.S. (1956) Taxonomy of Educational Objectives, Handbook: The Cognitive Domain. David McKay, New York.
- [28] Sikorski, M., & Honig, A. (2012). Practical malware analysis: The hands-on guide to dissecting malicious software. San Francisco: No Starch Press.
- [29] Free Software Foundation. (2021, November 20). In Wikipedia. [https://en.wikipedia.org/w/index.php?title=Free\\_Software\\_Foundation&oldid=1056171564](https://en.wikipedia.org/w/index.php?title=Free_Software_Foundation&oldid=1056171564). Last Accessed Dec 12, 2021.
- [30] Open Source Initiative. (2021, December 3). In Wikipedia. [https://en.wikipedia.org/w/index.php?title=Open\\_Source\\_Initiative&oldid=1058384737](https://en.wikipedia.org/w/index.php?title=Open_Source_Initiative&oldid=1058384737). Last Accessed Dec 12, 2021.