MALWARE ANALYSIS SKILLS TAUGHT IN UNIVERSITY COURSES

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

By

SWETHA GORUGANTU
B.Tech., Jawaharlal Nehru Technological University, 2013

2018
Wright State University

WRIGHT STATE UNIVERSITY
THE GRADUATE SCHOOL

April 24, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY SWETHA GORUGANTU ENTITLED MALWARE ANALYSIS SKILLS TAUGHT IN UNIVERSITY COURSES BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE.

Michelle A. Cheatham, Ph.D.
Thesis Director

Mateen M. Rizki, Ph.D.
Chair, Department of Computer Science and Engineering

Committee on
Final Examination

Michelle A. Cheatham, Ph.D.

Yong Pei, Ph.D.

Mateen M. Rizki, Ph.D.

Barry Milligan, Ph.D.
Interim Dean of the Graduate School

ABSTRACT

Gorugantu, Swetha. M.S., Department of Computer Science and Engineering, Wright State University, 2018. Malware Analysis Skills Taught in University Courses.

Career opportunities for malware analysts are growing at a fast pace due to the evolving nature of cyber threats as well as the necessity to counter them. However, employers are often unable to hire analysts fast though due to a lack of the required skillset. Hence, the primary purpose of the thesis is to conduct a gap analysis between the binary analysis skills taught in universities with those that the recruiters are looking for. Malware can be analyzed using three main types of tools and techniques: high-level profiling, static analysis, and dynamic analysis. These methods provide detailed information about the functionality and behavior of the binary executable. To determine the relevant courses taught in universities, three different set of universities were used which consisted of the NSA accredited colleges, top universities in computer science, and top cybersecurity colleges across the world. Based on the analysis, it can be observed that there are few universities that offer cybersecurity programs, among which very few offer a course in malware analysis. To shortlist the skills necessary for career opportunities in the field of malware analysis, a list of job descriptions from three employment-related social networking sites: LinkedIn, Indeed, and Glassdoor was collected. From the inventory of job postings, it can be noticed that most of the openings require experience with malware and reverse engineering tools. The dataset of university courses was compared and paralleled with the dataset of job descriptions using three analysis methods: LDAviz tool,

a word cloud generator, and a pie chart model. Based on the study, it can be concluded that though there are very few universities that teach cyber security analysis as part of their curriculum, they are exceptionally doing well in meeting the current needs of the industry. The only exception is a lack of coverage of topics like threat analysis, incident response, and computer forensics. However, it would be highly beneficial if all the schools could expand their programs and offerings in the field of Malware analysis so that young talent could easily fill these roles.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my thesis advisor Dr. Michelle A. Cheatham for her valuable guidance and insight not only during the course of this project, but also throughout my Master's degree. She was always available whenever I had a question and steered me towards the right direction. I truly enjoyed working with her and am gratefully indebted for her unwavering support.

I would also like to thank Dr. Mateen M. Rizki and Dr. Yong Pei for readily accepting to serve as my thesis committee members. My special thanks to Dr. Sikorski, his book "Practical Malware Analysis" has been the primary source of knowledge throughout my research work.

My deepest acknowledgement belongs to my parents, friends and family for their unconditional faith, love and support in achieving my dreams. This accomplishment would not have been possible without them.

1. Introduction

1.1. Malware

Malware is the abbreviation of malicious software. It is a piece of software that is used to gain access to or damage a user's data, computer or network. More generally, a piece of software that does any operation without the user's permission is considered to be malware. For example, malware may create a backdoor to access user's data. Once installed, malware can greatly reduce a computer's performance, and delete or change the contents of files.

Malware is able to spread through many ways. Prior to the invention of the Internet, malware used to travel through floppy disks. It used to get attached to the floppy disk and then replicate once inserted in to a computer. But now, it gets downloaded within seconds through the Internet. Malware usually gets installed by tricking the users to click a link or any operational buttons on a website. Users often cannot differentiate between trustworthy ads against damaging ones. The process of incorporating malware in online advertisements is called malvertising (an abbreviation of malware advertising). Users tend to click on ads or any links that say "download" expecting to download the advertised software, which will also result in downloading the malware on to the device. Malware might also be present in e-mails as an attachment or a link. The unexpected download of malicious code or a virus to the machine is called a drive-by download.

1.2. Malware Analysis

Malware analysis is important for stopping the spread of malware. Malware analysis is the process of examining a malware to determine how it operates, how to identify it, and the possible methods of eliminating it. Malware analysis provides an understanding of what kinds of malware a piece of software contains, how it damages the computer or network, how it gains access to the user's data, how to detect it and how to measure its damage.

One way to detect malware is through the use of signatures. A signature is a pattern of actions or a string of code that is created based on a known piece of malware. There are two types of signatures, host-based signatures and network signatures. Host based signatures or indicators are effective in detecting malicious code because they help in finding out the operations of the malware on the system instead of its characteristics. For example, a hacker might changes an unimportant line of code, like a comment, so that it will have a different checksum. That makes checksums a weak signature. On the other hand, the hacker cannot change the effect of the malware on the host system without affecting its functionality. That makes host-based signatures stronger. Host-based signatures include things like the files that are created or changed by the malware and the modifications it makes to the registry, if any. These are things the malware analyst's attempts to discover when they analyze the code. Network signatures are used in identify the malware by examining the network traffic. They are more effective when created with the help of malware analysis. For example, we can use firewalls to know the websites which the malware has access to and include details such as IP addresses, port numbers in

the network signatures. Additionally, signatures are also helpful in counting malware based on common properties of the malicious code. After obtaining the signatures, we need to find out how the malware operates which is done by the malware analysis techniques that will be discussed in Chapter 2.

1.3. Importance of Malware Analysis

With millions of malicious codes on the Internet, and the number being encountered increasing day-by-day, malware analysis is gaining importance for detecting and preventing cyber-crime. There were on average 47.4 attempted attacks per computer during the first half of 2017 threatening online security [1]. Cyber-attacks have become equally prominent along with tablets, mobile phones and other smart devices. The number of malicious codes being downloaded is growing exponentially, so it is critical to incorporate a defensive anti-malware strategy to protect information. Moreover, the number of new malware consistently increased in the last few years. The below table shows the number of new malware specimen (in millions) for ten years from 2007 to 2017. A new malware specimen emerged every 4.2 seconds in 2017 [1].

Year	Number of new malware specimen (count in millions)
2007	0.13
2008	0.89
2009	1.58
2010	2.09
2011	2.57
2012	2.64
2013	3.38
2014	5.99
2015	5.14
2016	6.83
2017	9.78

Table 1: Number of new malware appeared in a time period of ten years [1]

Hence, the cybersecurity strategy should include malware detection controls along with preventive measures. As mentioned previously, malware analysts provide the information needed to develop these controls and measures.

It is also essential to have knowledge about digital crimes from an economic point of view. There is an increasing security threat to organizations, industries, banking and private sectors. Economic damage of about ten billion dollars per year is caused by malware [1].

Stopping the spread of malware is an important part of reducing this economic damage. Additionally, it is important for companies to limit the economic damage of malware on their customers when possible. To do this, malware analysts need to determine if any user accounts were accessed or changed, if passwords were stolen, where they were sent, and anything else that can be used to warn the customers who were impacted.

Malware analysis is also an important part of incident response. Incident response is an organized process to handle and manage a cyber-attack, along with the recovery costs. It includes various components such as preparation of technical investigation, resource identification, containment, eradication etc. Since, malware is the cause for many computer or IT incidents, malware analysis plays a crucial role in incident response. Understanding how malware affects a system can reduce infections in an organization or a network, thus reducing the overall cost. Hence, performing malware analysis is necessary to protect the security of a network.

1.4. Skills Needed by Malware Analysts in today's World

According to the Bureau of Labor Statistics in 2016, about 100,000 jobs are available for Information Security analysts and the expected change in employment in the next ten years is 28 percent (compared to the average of 7 percent), which counts to about 28,400 jobs.

Skilled malware analysts are in high demand in today's world due to the lack of malware analytical skills in today's generation. A person who is capable of analyzing a piece of malware and gets a deep understanding of what and how it does is called a malware analyst.

A good knowledge of many aspects of computer science, such as networking, programming, operating systems, compilers, computer architecture and malware in general is required to become a malware analyst. However, the present generation lacks such malware analytical skills not all of these courses are a part of most of the college curriculums. University courses specifically on malware analysis are even rarer because it is a relatively new field within computer science.

This thesis will focus on answering the following question: are universities meeting the needs of employers when it comes to hiring malware analysts, and if not, are there other places for individuals wishing to go into this field to learn the required skills? To answer this question, Chapter 2 of the thesis provides background on the malware analysis process, Chapter 3 explores the skills relevant to malware analysis that are taught in university courses and online learning courses, Chapter 4 reviews the skills that employers that hire malware analysts are looking for, and Chapter 5 concludes with an analysis of how well universities and online course providers are meeting the needs of employers in this area.

2. Malware Analysis Tools and Techniques

The goal of this chapter is to understand the basic skills, tools and techniques commonly used in malware analysis. Based on the time, skills and tools required to perform an analysis, there are three techniques of analyzing malware: high level profiling, static code analysis, and dynamic code analysis. To get a complete idea of a malware executable, these techniques require different tools, which will be discussed further.

2.1. High Level Profiling

High level profiling is the process of analyzing the malware by observing the characteristics of the malware and its impact on the system by running it instead of looking at the code. This process does not provide much information about the functionality of the malware, as it does not involve executing the code line by line. Hence, it does not require as much detailed technical knowledge as code analysis.

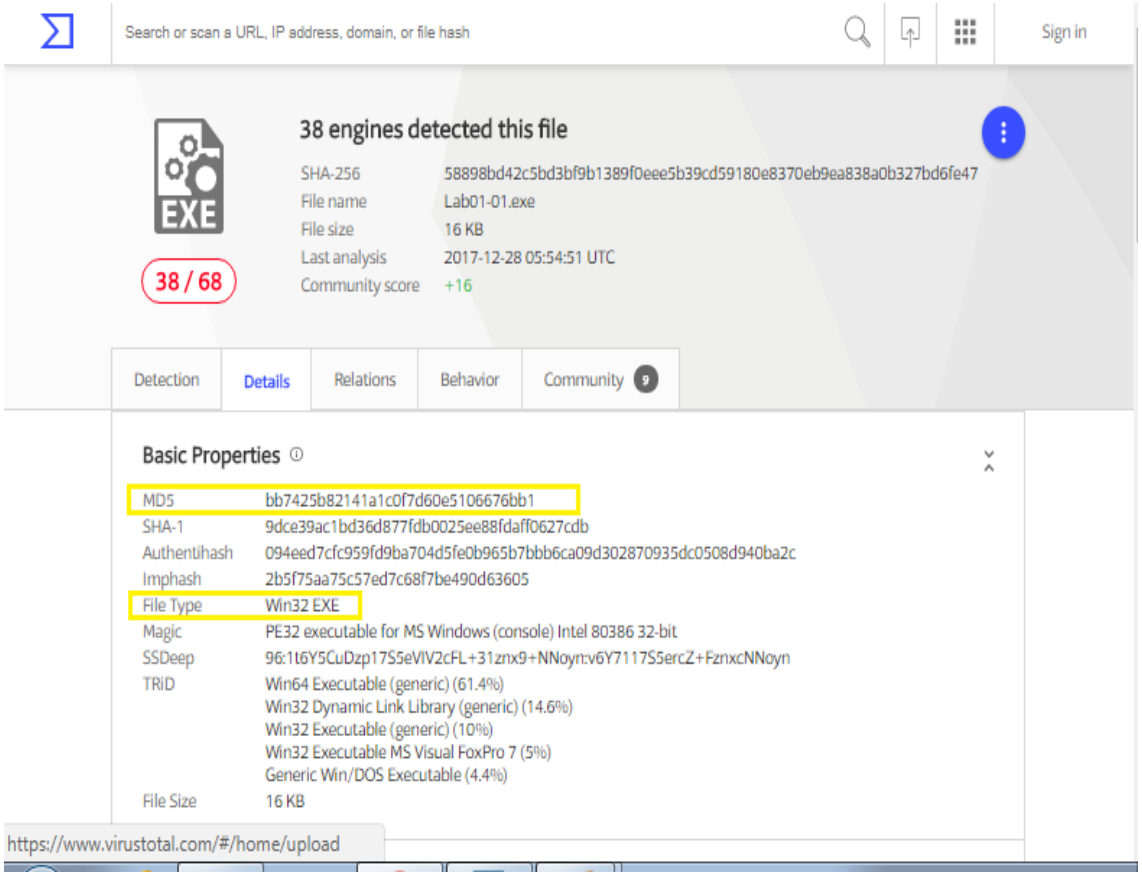

Figure 1: Antivirus program to find the basic file details

2.1.1. Antivirus Scanning

The most obvious thing to do when trying to learn if a particular piece of code is malware and what it does is to use an antivirus program. Antivirus programs compare a file to a database of known malware. If the file matches one in the database, it is considered as malware. This comparison is most often done by comparing specific bits of code with identical pieces of known suspicious code or file signatures. An example of a file signature is a hash. A hash value is a unique alphanumeric string value that is generated by a

computer algorithm based on the contents of a file. Hashes can be used by anti-virus tools to screen for malware and prevent it from being downloaded or executed. There are several antivirus programs that use different signatures and heuristics such as VirusTotal, Malwarebytes, Avast and so on. However, detection of malware through hashing is not so effective because if the malware developer changes the code even slightly, it will have a different hash and therefore avoid detection by the anti-virus system. Because of this, it is desirable to have anti-virus systems that use behavioral and pattern-matching analysis in addition to hash values.

VirusTotal (www.virustotal.com) is an example of an application that aggregates the output of many antivirus tools. It generates a report that shows the total number of antivirus engines that considered the uploaded file as malicious. For example, the file loaded in the below screenshot is regarded by 38 antivirus engines as malicious.

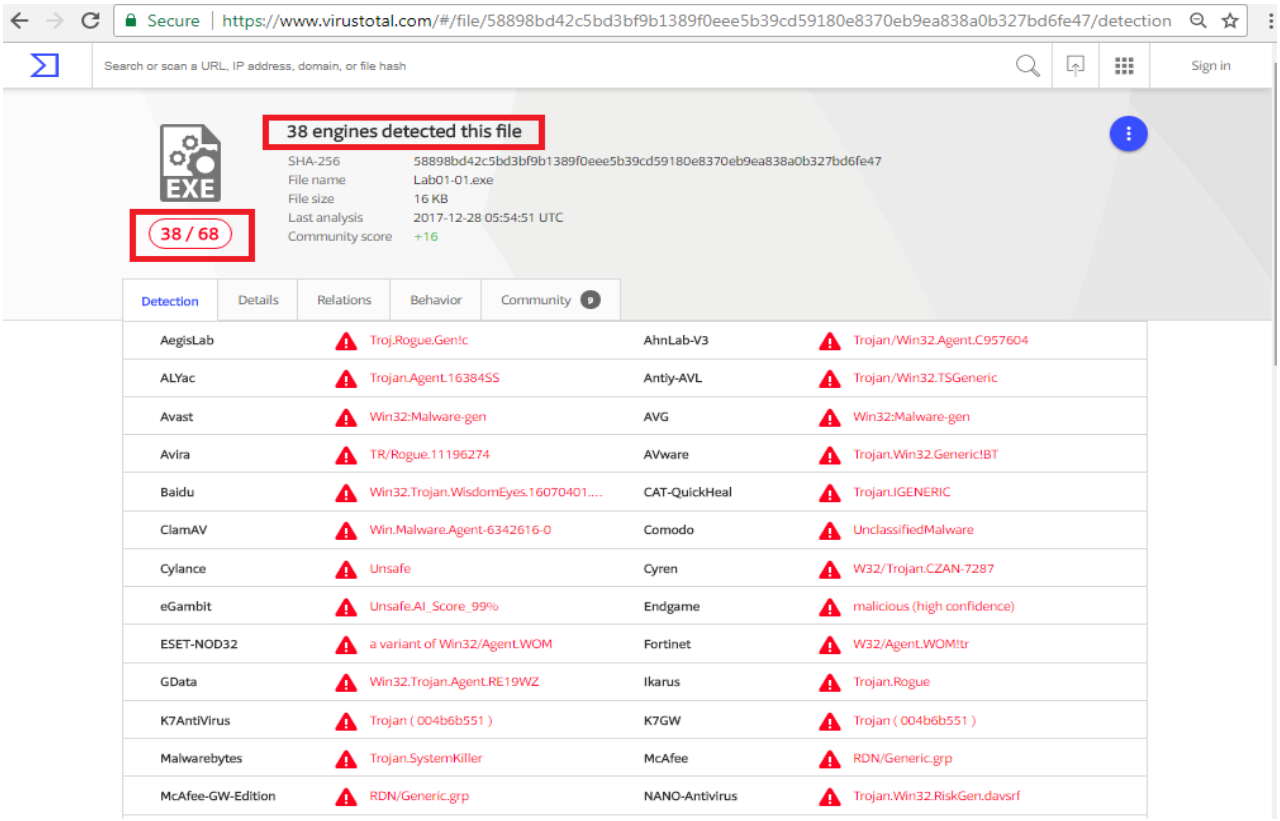

Figure 2: VirusTotal report detecting the file as malware

2.1.2. Dependency Analysis

Not all malware has already been identified and included in antivirus program databases. In these cases, more analysis is needed to learn what the executable is doing. One thing that can be very useful is to know the operating system functions and libraries an executable uses. For example, if an application that is supposed to be a simple calculator makes use of many network-related capabilities, it might contain malware. Tools such as Dependency Walker list the dependencies of the executable file in the form of a recursive tree. They also help in viewing the list of functions that are imported and exported by a module. When

we open a .dll file in Dependency walker, the view will be as follows:

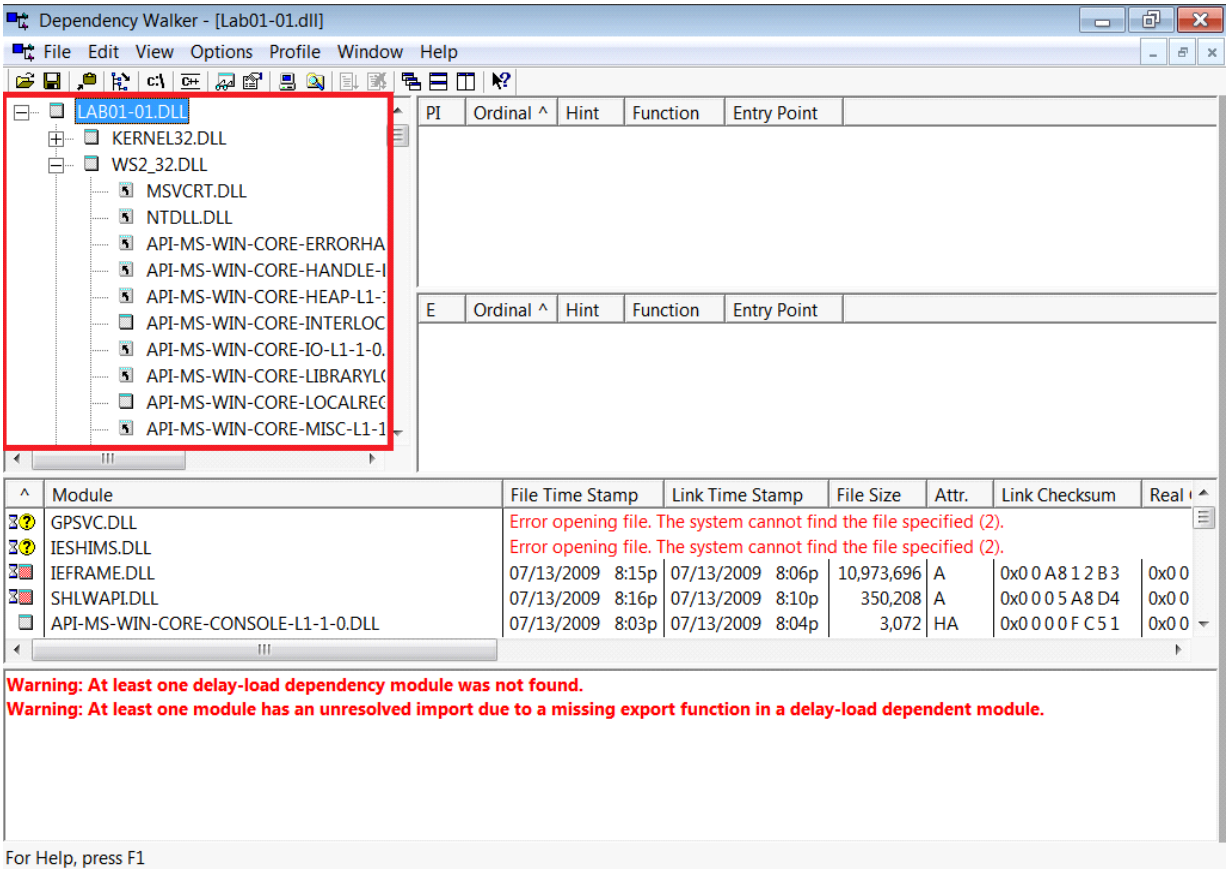

Figure 3: Dependency walker listing the dependencies of the executable

2.1.3. Process Monitoring

Another approach to high level profiling is to run the executable and observe its effect on the system or a network. This type of profiling can be risky because running the malware can spread it to other machines on the network or allow it to violate the system's security, such as sending confidential user information. Hence, we need to follow precaution mechanisms while executing the malware such as using a virtual machine to study malware

safely, using sandboxes to run untrusted programs, etc.

Running the malware executable code helps to find out the processes it spawns. This is important because the malware may create a process on the user's system that does something bad like log the keystrokes. Tools such as ProcMon allow an analyst to capture process information. ProcMon is an abbreviation for Process Monitor. As the name suggests, it monitors all the processes or system calls a piece of malware spawns or invokes when it is run. It can be downloaded from the link: https://docs.microsoft.com/en-us/sysinternals/downloads/procmon.

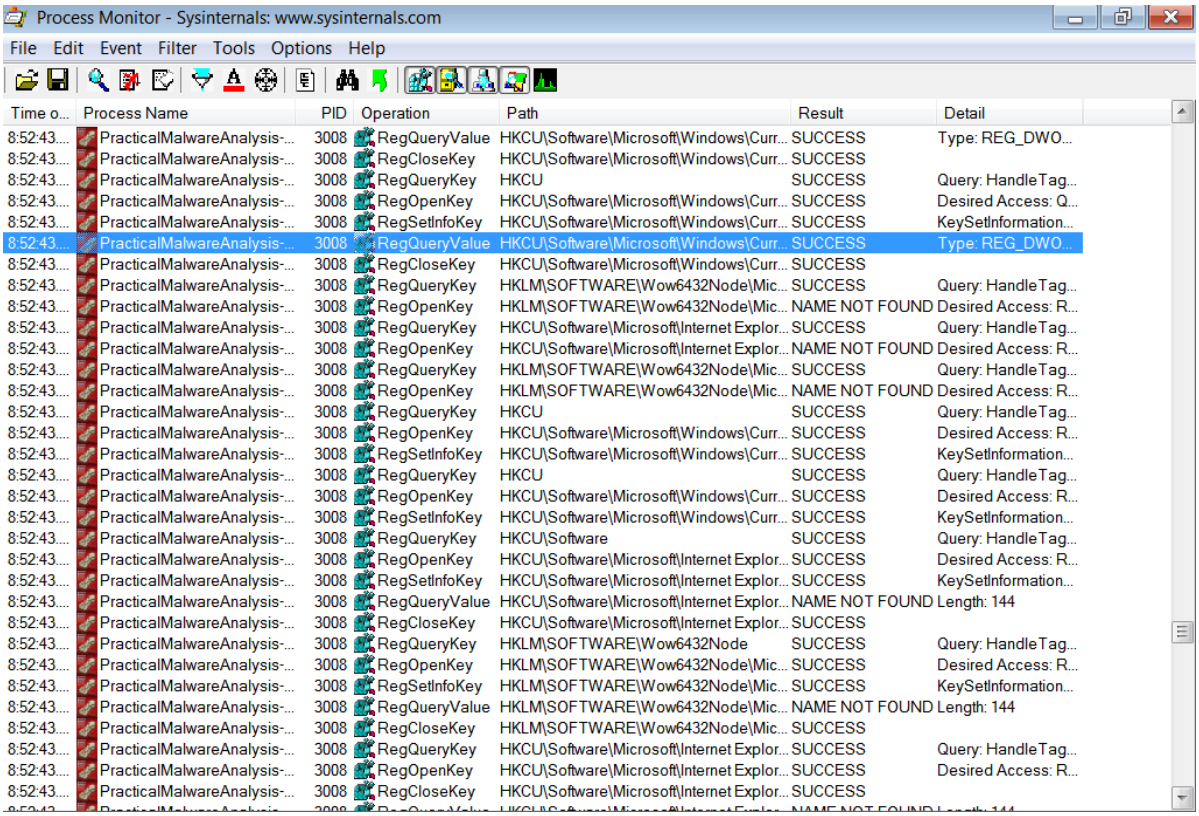

Figure 4: List of ProcMon events while running the malware

Process explorer is a similar tool. It provides a deep understanding of the currently running process. It lists all the file handles and DLL's that are run and loaded in the form of a tree structure for each process. It can be downloaded from: https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer. This is how all the active processes are displayed in Process Explorer.

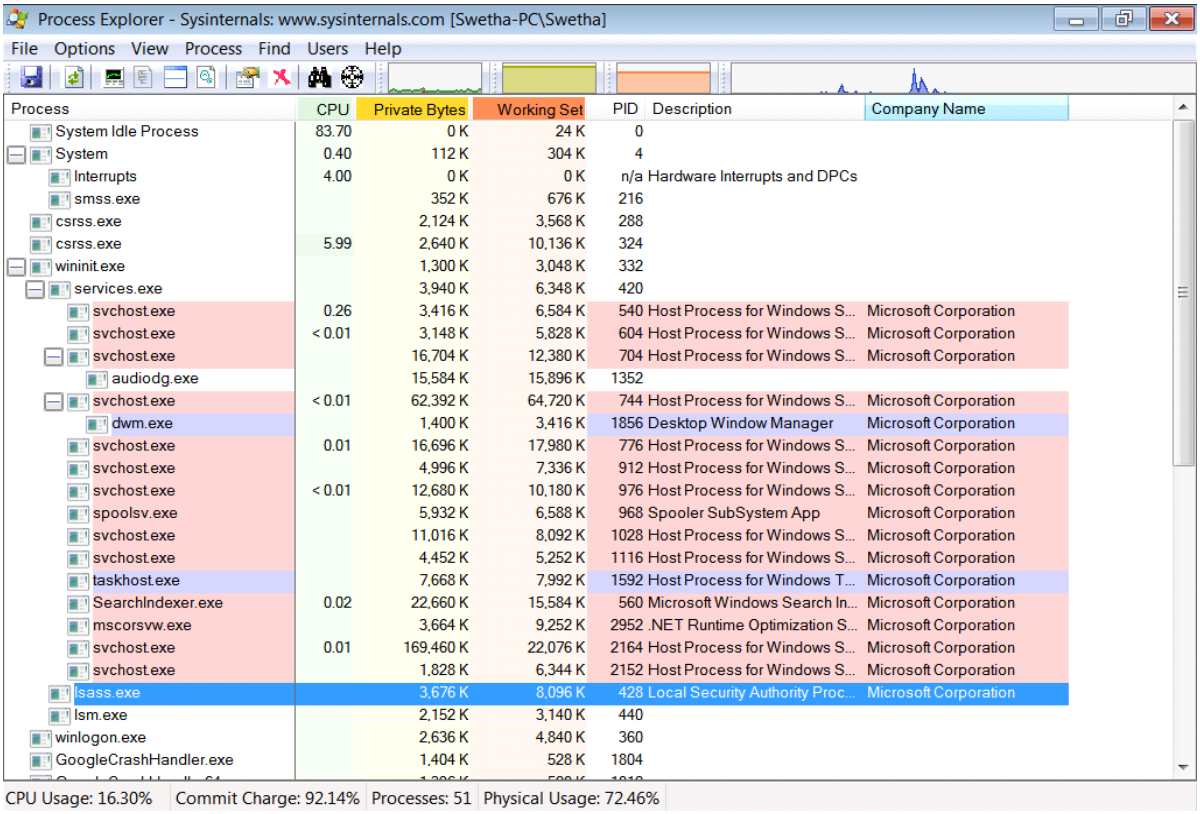

Figure 5: Process explorer displaying the list of processes

2.1.4. Registry Monitoring

Malware can create entries or modify the existing registry values in order to make it difficult to detect or to or to launch itself or integrate with an existing process. Hence, it is

important to verify for any modifications in the registry after running the malware with tools such as RegShot. This tool allows the analyst to take registry snapshots, which help in identifying if there are any modifications made to the registry after running the malware. It can be downloaded from: https://sourceforge.net/projects/regshot/. This is the sample result by RegShot when it compares the snapshots generated before and after running the malware.

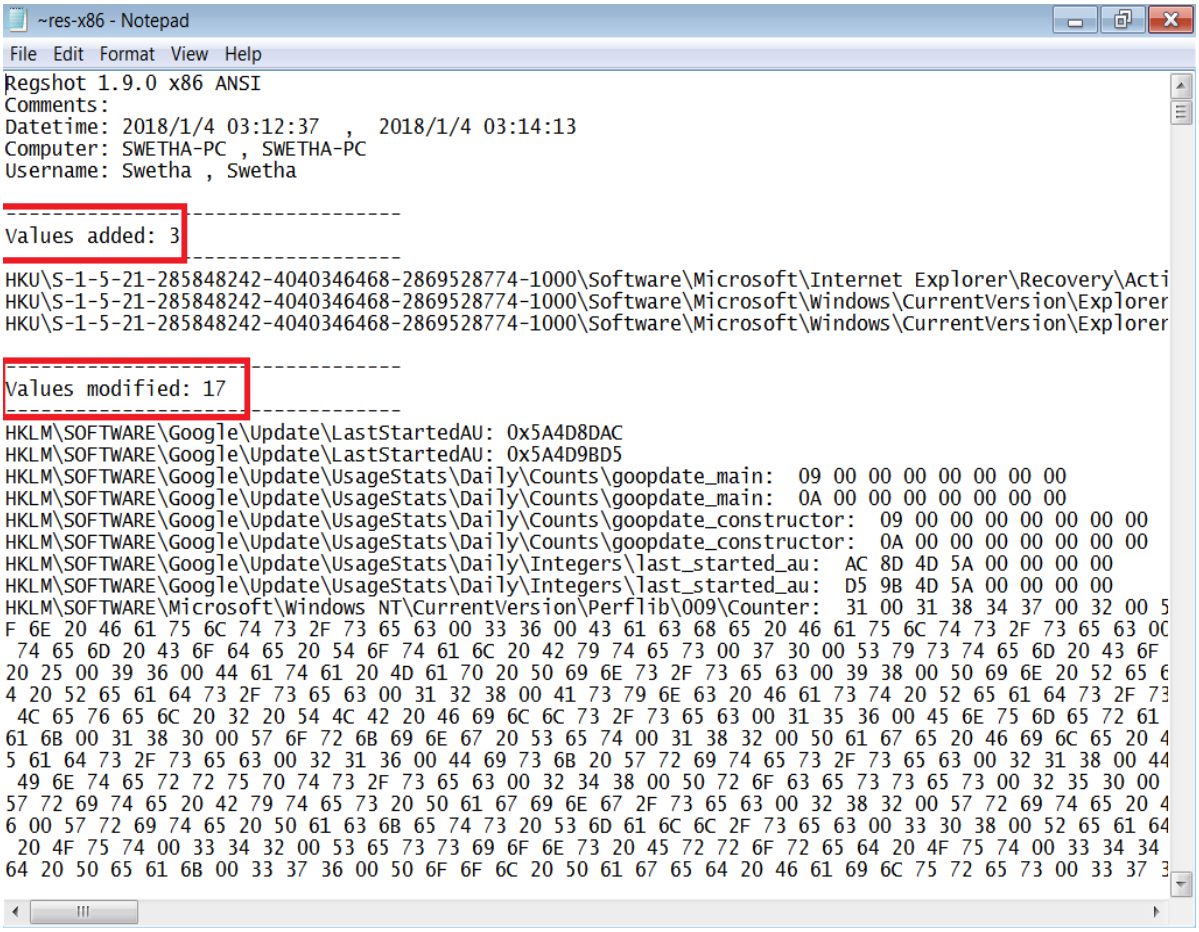

Figure 6: RegShot showing the registry values

2.1.5. Analyzing Network Traffic

It is also important to know what websites or other network resources an executable tries to connect with. For example, a piece of malware may be connecting to a server to transmit sensitive user data or to receive further instructions. However, to prevent malware from spreading, most analysis is done on a computer that is isolated from the rest of the network. One important way to observe the run-time characteristics of the malware is to fake the network it is executed in and prevent it from realizing that it is not actually connected to the Internet. This helps in monitoring the network traffic and obtaining the network indicators such as signatures, IP addresses and domain names that the malware uses. Tools such as ApdateDNS and INetSim can be used for analyzing malware in a network. ApdateDNS is a tool that serves as a DNS server and controls the DNS responses. When the malware generates DNS requests to a user-specified IP address on UDP port 53 on local machine, ApdateDNS spoofs DNS responses to those DNS requests by listening on the UDP port. The below screenshot shows ApdateDNS responding to teredo.ipv6.microsoft.com

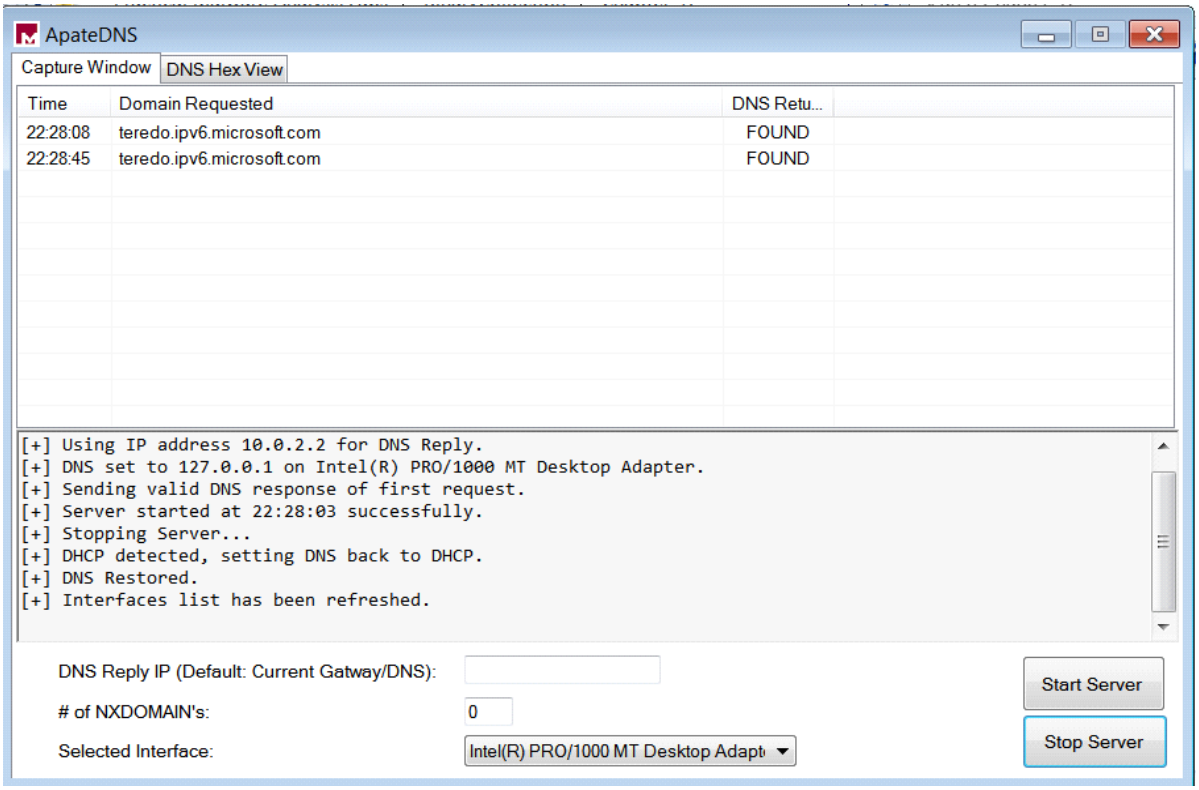

Figure 7: Response of Apdate DNS

INetSim is similar to ApdateDNS, but it fakes a wider variety of services. INetSim is a Linux based tool that simulates most of the internet services used commonly, such as FTP, HTTP, HTTPS, SMTP, DNS etc.

Once the malware analyst has simulated a network environment, they must explore the interaction of the malware with that network, such as what resources it is connecting to and what information it is sending. Tools such as Netcat and Wireshark are useful for this. Netcat is a TCP/IP tool used to read and write to network connections. Due to the features it consists of such as port forwarding, tunneling, port scanning and proxying etc. it is also

known as TCP/IP Swiss Army Knife. Wireshark is a packet analyzer that is used to know the details of the network traffic in a minute level. Wireshark includes packet-stream analysis, filtering, visualization and a deep insight in the analysis of individual packets.

High level profiling can only give an overview of what the malware does. For more information, the code will need to be analyzed in more detail, using either static or dynamic code analysis.

2.2. Static Code Analysis

Static code analysis consists of examining the malware by viewing the assembly code. The advantage of this type of analysis is that we do not need to execute the malware or work around any anti-debugging techniques.

In static code analysis, an executable is loaded into a disassembler and reverse engineering of the malware is performed. The disassembler converts machine-executable code to assembly language source code. In order to know what the program does, we observe the program instructions that are executed by the CPU. This process requires a deep knowledge of assembly language code constructs, windows operating system concepts and disassembly.

The primary tool used for disassembly by most reverse engineers or malware analysts is Interactive Disassembler Professional (IDA Pro). IDA Pro disassembles the malware binary and shows us the entire disassembly in assembly language using two modes: graph mode and text mode. This section will provide an illustration of using IDA Pro to do a

reverse engineering task. In this task, there is an executable file that prompts for a username and a serial number. By default, the executable will say “Error: not a correct Serial.” The goal is to instead get it to say “Correct: Good Work.” To begin, the analyst would open the file in IDA Pro. This results in an assembly code listing.

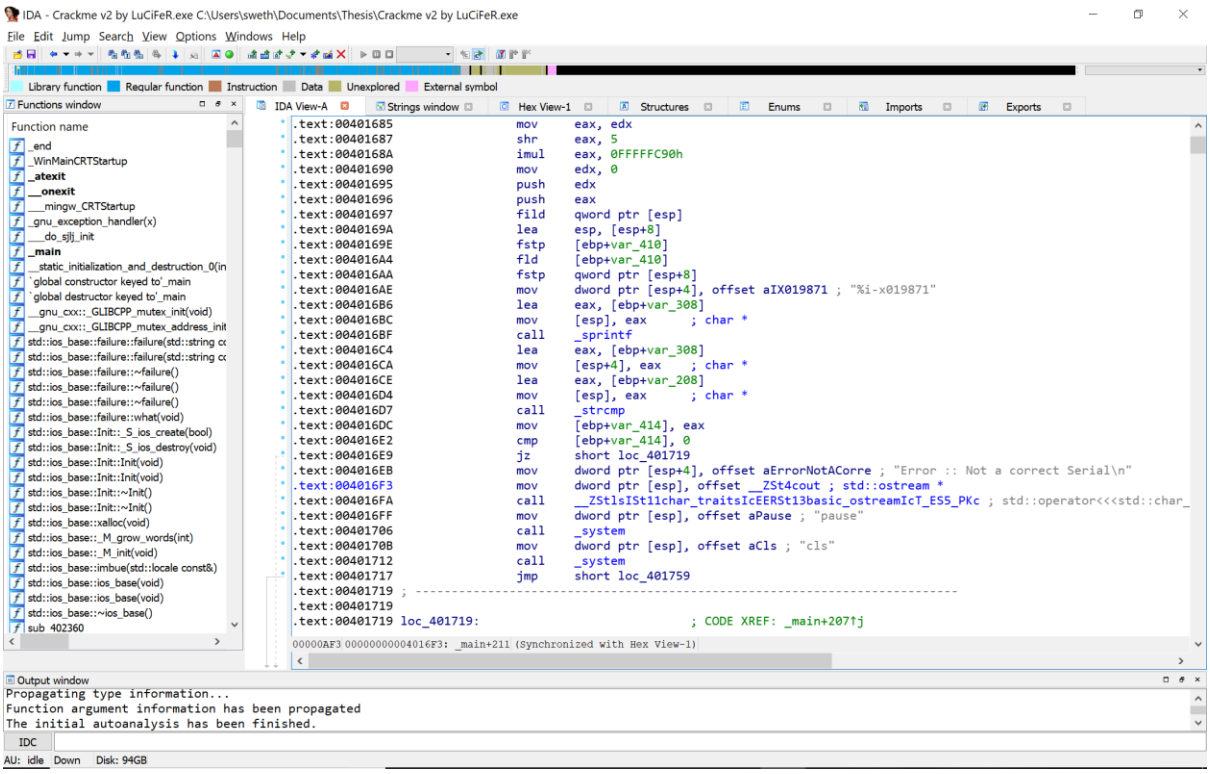

Figure 8: Cracking a file with IDA Pro

The easiest way to proceed is then to use IDA Pro’s strings view to find the string is actually displayed (Error: Not a correct Serial) and the one that is desired (Correct: Good Work). The references to these strings from the code can then easily be followed by clicking on the cross-references for each string.

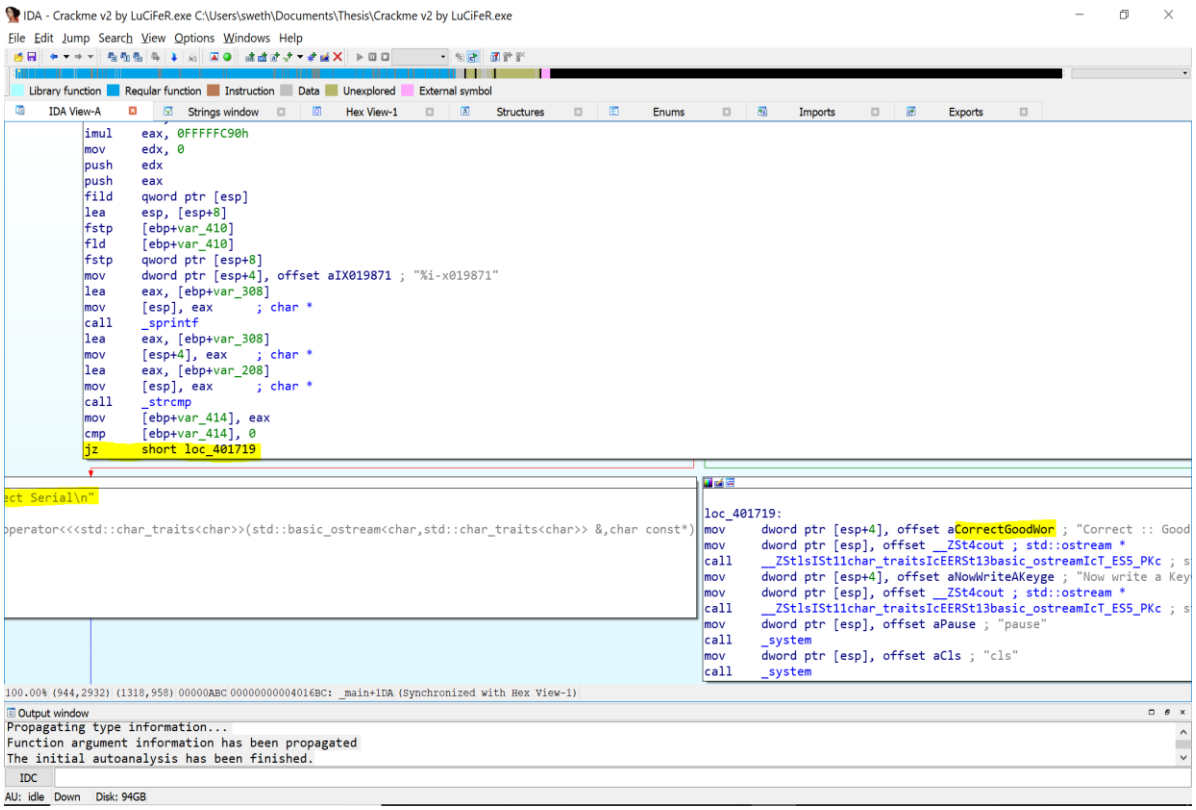

Figure 9: Control flow of IDA Pro

It is clear that the two strings both follow a “jump if zero” (JZ) statement. If the jump is taken, the green arrow to the right is followed and the string referenced is “Correct: Good Work.” If the jump is not taken, the red arrow to the left is followed and the string referenced is “Error: Not a correct Serial.” The goal is to make sure that the jump is always taken. This can be accomplished by changing the JZ command to JMP (always jump). In IDA Pro, this is done by selecting Edit→Patch Program→Assemble and making the change.

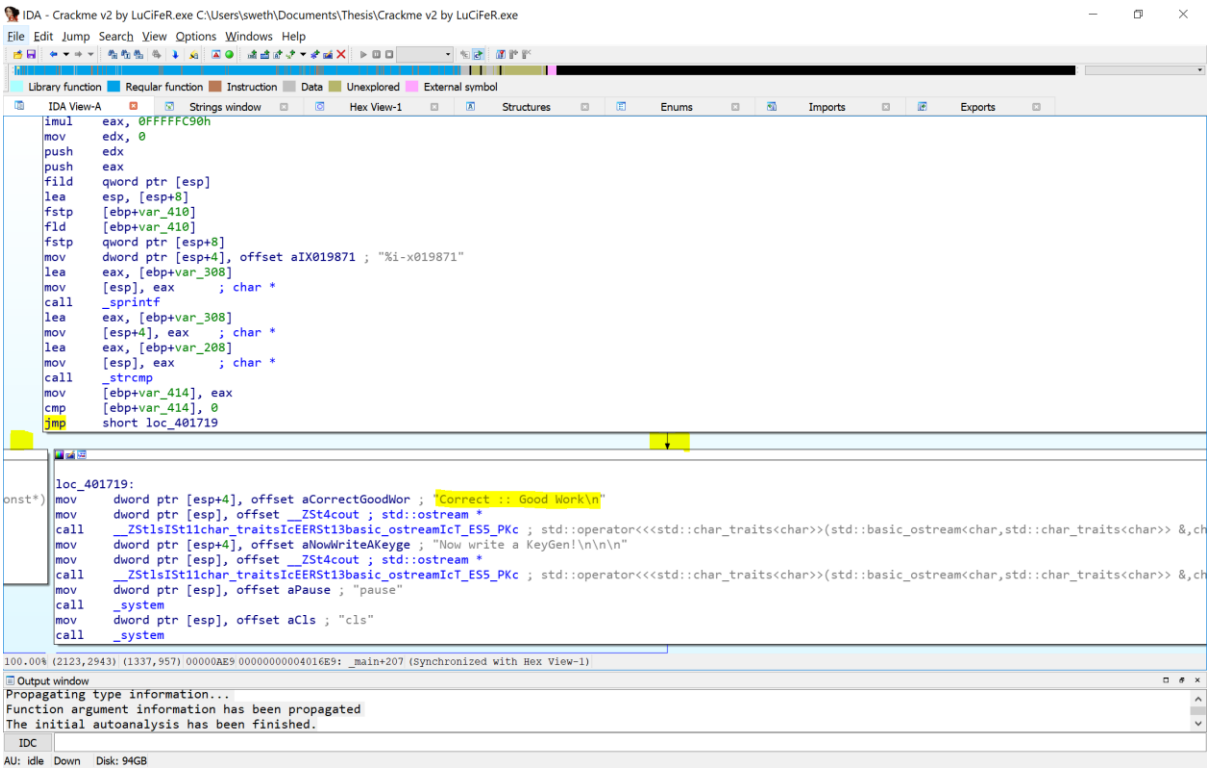

Figure 10: Making changes to the flow of IDA Pro

The executable will now always display “Correct: Good Work” for any credentials that the user enters. As shown in this example, we can analyze malware behavior through statically analyzing the disassembled code and strings. We can modify also the assembly code to change the file’s behavior. However, things are not always this easy. A malware writer might implement mechanisms to hide or protect the malware and make its key functionality hard to find. These are called obfuscations. One very common obfuscation is called packing. In a packed program, the code is encrypted or compressed so that looking at it in IDA Pro does not reveal any useful information. When packed executables run, a small

section called a stub decrypts or decompresses the rest of the program and runs it. Because static analysis does not involve running the code directly, the code must be unpacked before it can be analyzed in this way. Fortunately, there are unpackers available for many common packing techniques. Tools such as PEid can help an analyst to identify which packer has been used so that it can be unpacked. In the screenshot shown below, the file is identified as FSG packed.

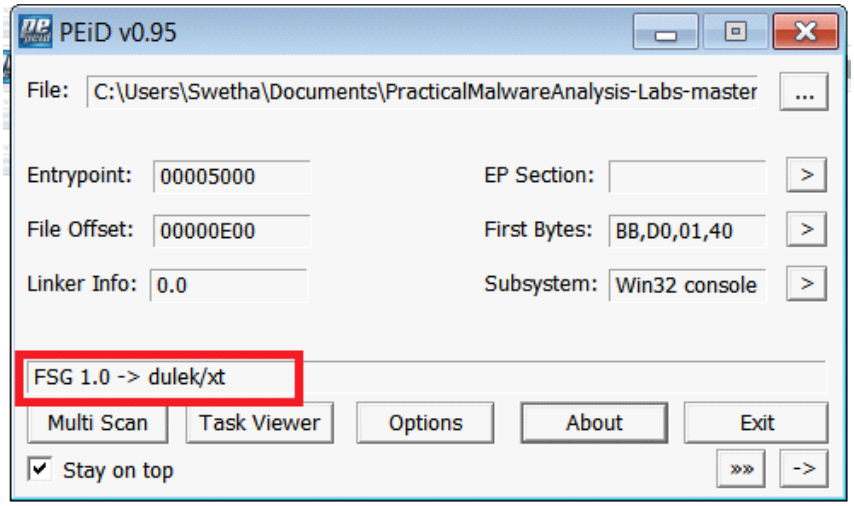

Figure 11: PEid indicating the file as FSG packed

Static code analysis is a safe way to explore a piece of malware since it does not put the system or network at risk, but it is very difficult because in an average program there are many assembly instructions and many of these might be unrelated to the malicious part of the code. Additionally, the code may be packed or obfuscated. If static analysis leads to a dead end, dynamic code analysis may be appropriate.

2.3. Dynamic Code Analysis

In dynamic code analysis, we further analyze the malware by using a debugger such as OllyDbg or WinDbg. Debugging involves executing code line by line and observing its effects. This helps to observe the malware behavior and its impact on the host system by running the malware.

One of the most common tools used for debugging is OllyDbg. This section describes the use of OllyDbg to perform the reverse engineering task as described in the previous section.

Once we load a program executable to OllyDbg, we will be able to view four windows:

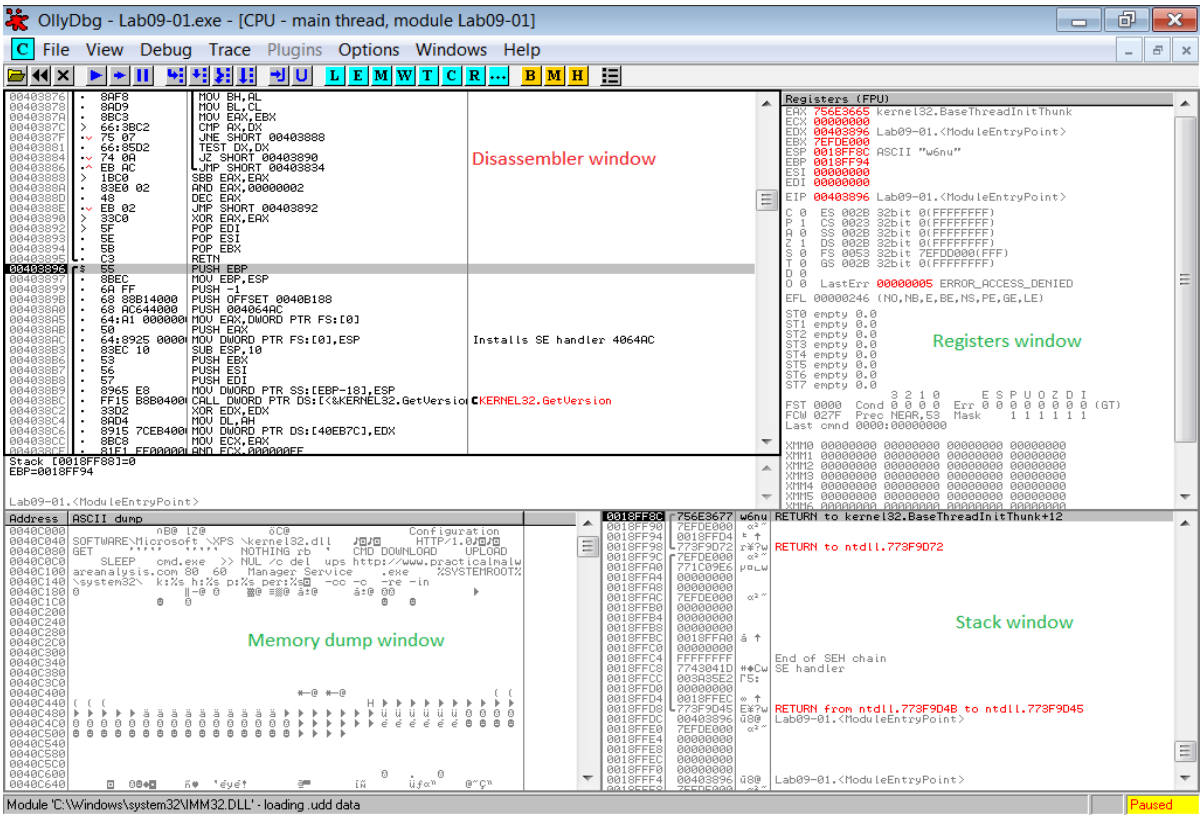

Figure 12: Basic layout of OllyDbg

The disassembly window (upper left) displays the disassembled code of the program and the next instruction to be executed is highlighted. The registers window (upper right) displays the current state of the registers. For the thread being debugged, the stack window (lower right) displays the current state of the stack for that thread. Finally, the memory dump window (lower left) displays the memory dump of the debugged process.

OllyDbg also allows the analyst to set breakpoints in the code. Breakpoints allow us to pause the execution of the program at a point and then debug the program or other parameters of the breakpoint such as strings, memory chunks etc. For this RE task, we can use the string search functionality to again find the key strings we are interested in: Error: Not a correct Serial and Correct: Good Work. This is done by right clicking and selecting Search for → All referenced text strings. The list of strings is displayed in a separate window:

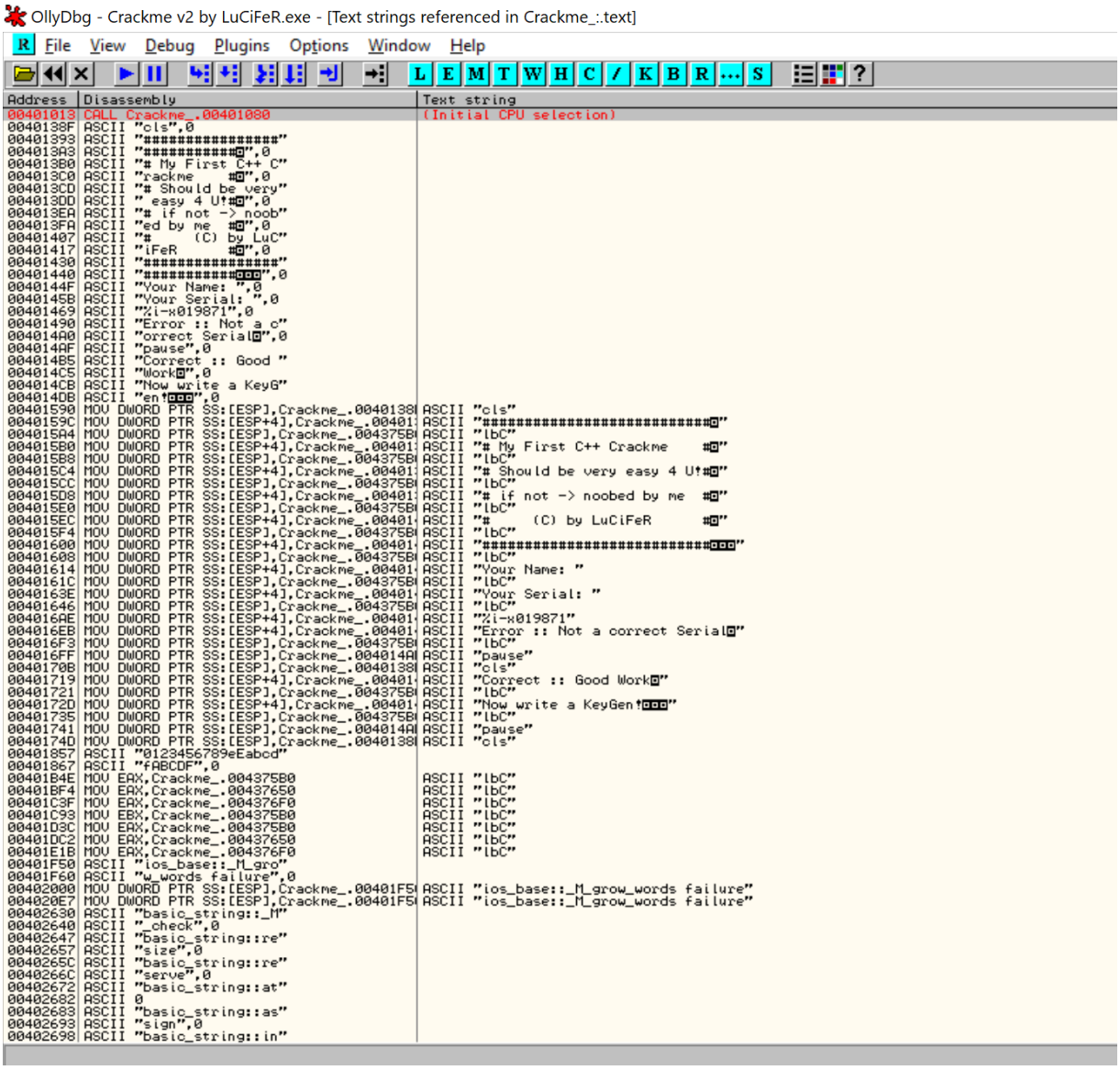

Figure 13: Strings section of the crack me file

Double-clicking on a string will take us to the place in the disassembly window that refers to it. We then set a breakpoint shortly before the place in the code where both the correct and error strings are referenced and instruct OllyDbg to execute the program up to that

point. We then single step through the code line by line and observe its behavior. When we get to the JZ statement just before the string references, OllyDbg displays a message about which direction the execution will go (in this case, to the error string). To avoid this, we need to force the jump to always be taken. Right-clicking on the JZ command in the disassembly window allows us to type in an alternative command, in this case JMP.

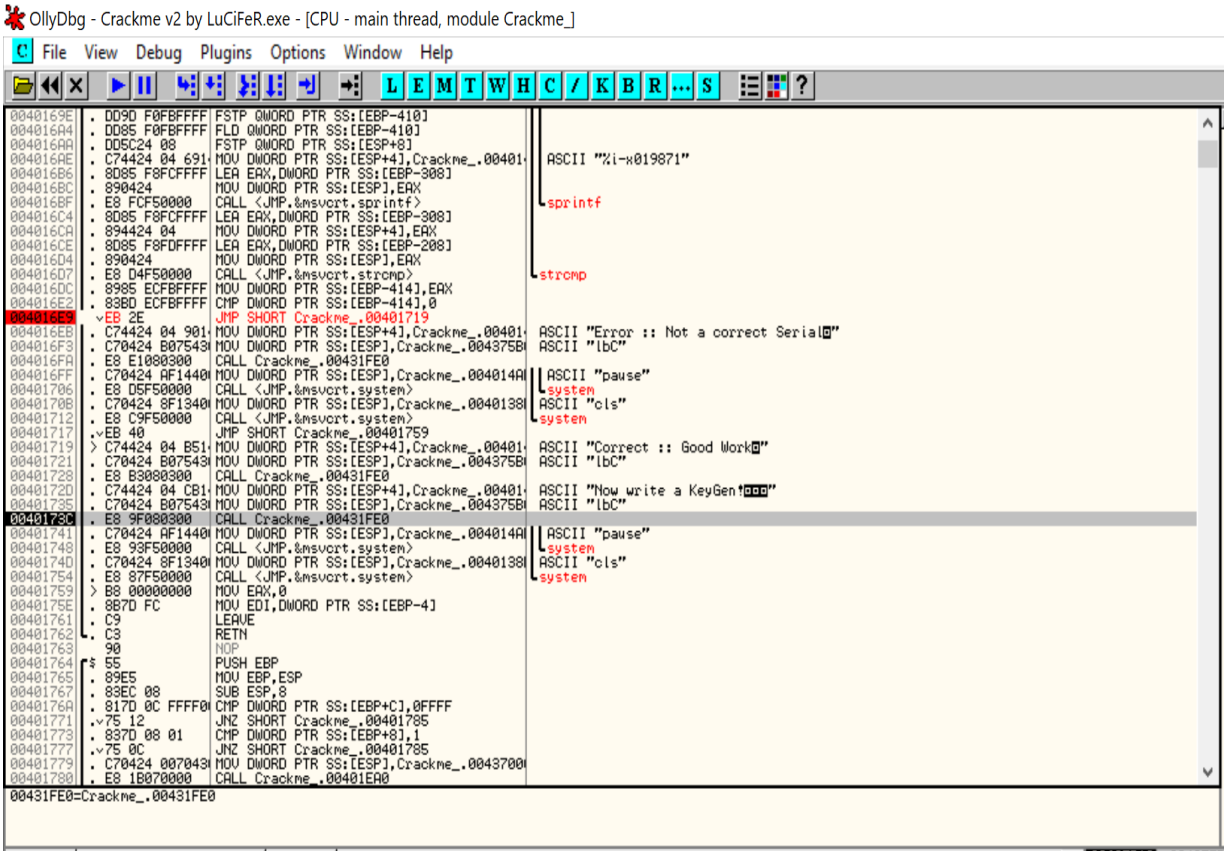

Figure 14: Inserting a breakpoint in OllyDbg

Then we can allow the program to continue executing, and the correct message is displayed.

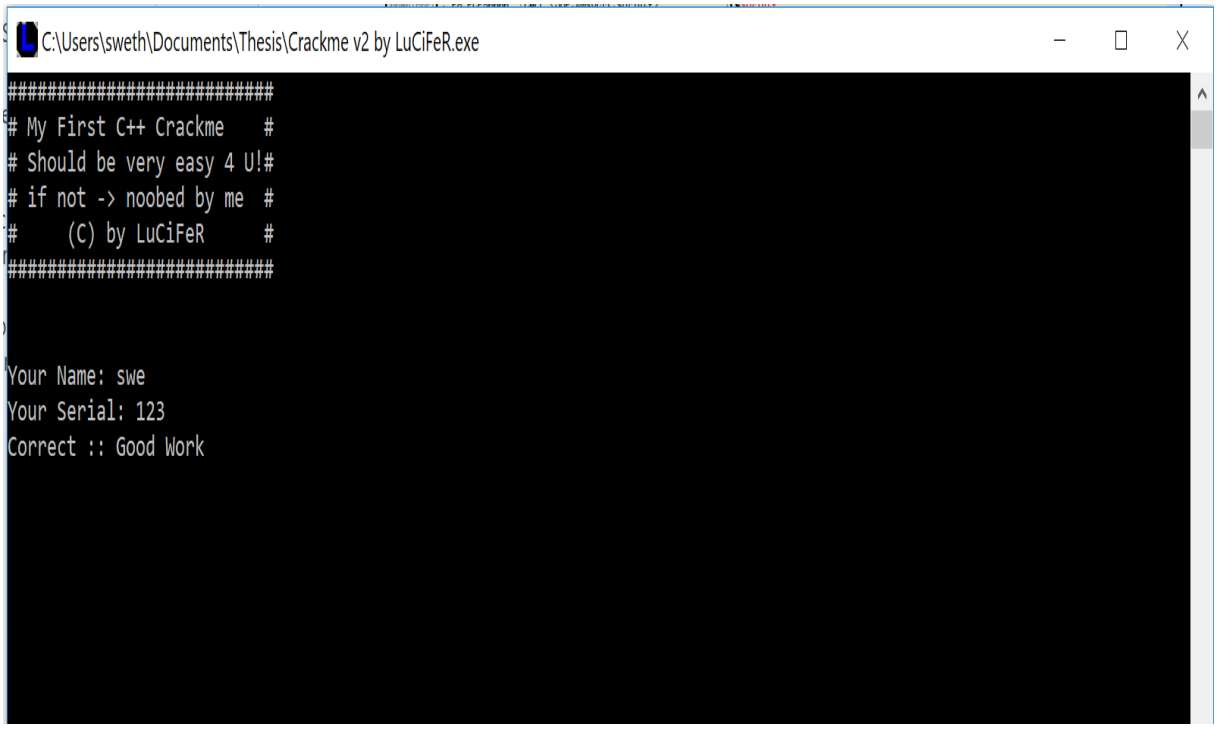

Figure 15: Final screenshot after cracking the executable

As this section shows, OllyDbg helps malware analysts to locate routines from libraries, recognize strings, API calls, and procedures and trace registers all of which provides important ways for the analyst to determine what an executable is doing.

3. Binary Analysis Skills Taught in Universities

The goal of Chapter 3 is to determine which of the malware analytical skills introduced in the previous chapter are taught in universities. The chapter begins with a discussion about the methodology for selecting the schools to be analyzed and the process for finding course descriptions and syllabi relevant to binary analysis from each school. The chapter then discusses the analysis performed on these materials in order to get an overview of universities' approach to this topic.

3.1. Sample Selection

The first step was to establish a shortlist of universities to analyze. The goal of this effort was to establish a good variety of universities. To do this, three different samples were used. The first was a list of National Centers of Academic Excellence in Cyber Operations that consists of 19 universities in the USA. The second sample consisted of the top universities in Computer Science as ranked by the US News & World Report. The third sample consisted of the top universities in cybersecurity across the world, as identified by the Data Insider.

3.1.1. NSA Accredited Centers of Academic Excellence in Cyber Operations

The National Security Agency (NSA) and the Department of Homeland Security (DHS)

work together to recognize and grant designations to online or campus-based colleges and research institutes. They created a program called the National Centers of Academic Excellence in Cyber Defense (CAE-CD) program, which offers three designations:

- Center of Academic Excellence in Cyber Defense Education (CAE-CDE)
- Center of Academic Excellence in Cyber Defense Two-Year Education (CAE-2Y)
- Center of Academic Excellence in Research (CAE-R)

The first set of universities shortlisted for our study consists of the universities recognized as Center of Academic Excellence in Cyber Defense Education (CAE-CDE). This designation is given to universities that offer undergraduate and graduate cyber defense related programs. To receive the CAE designation, colleges must make sure that their degree programs are closely related to specific cybersecurity-related knowledge units and are validated by experts in the field. The institutions must also ensure that the programs they offer include core knowledge units (KUs) on certain topics of study specific to the CAE designations, as presented in [2]:

Center of Academic Excellence in Cyber Defense Two-Year Education (CAE-2Y) programs:

- Basic Data Analysis
- Basic Scripting or Introductory Programming
- Cyber Defense

- Cyber Threats
- Fundamental Security Design Principles
- IA Fundamentals
- Intro to Cryptography
- IT Systems Components
- Networking Concepts
- Policy, Legal, Ethics, and Compliance
- System Administration

Center of Academic Excellence in Cyber Defense (CAE-CDE) programs:

Includes the KUs of CAE-2Y programs, in addition to the following:

- Databases
- Network Defense
- Networking Technology and Protocols
- Operating Systems Concepts
- Probability and Statistics
- Programming

None of these knowledge units are obviously directly related to binary analysis. However, the CAE-CDE program is the strongest set of cyber security university curriculum guidelines in the US currently, and it is likely that some universities with this designation

are offering courses on binary analysis in addition to the content required by the NSA. Below is a list of the universities and the programs they offer which are currently regarded by the NSA as Centers of Academic Excellence in Cyber Operations, as listed in [3]:

- **Air Force Institute of Technology** (Ohio)

  M.S. in Cyber Operations
- **Auburn University** (Alabama)

  B.S or M.S. in Software Engineering / B.S. or M.S. in Computer Science / B.S. of Wireless Engineering (Software Option), with Cyber Operations Certificate
- **Carnegie Mellon University** (Pennsylvania)

  M.S. in Information Security, Specialization in Cyber Operations
- **Dakota State University** (South Dakota)

  B.S. in Cyber Operations
- **Mississippi State University** (Mississippi)

  M.S. in Computer Science with Cyber Operations Certificate
- **Naval Postgraduate School** (California)

  M.S. in Computer Science, Cyber Systems and Operations Specialization
- **Northeastern University** (Massachusetts)

  B.S. in Computer Science, Concentration in Cyber Operations / B.S. in Cyber Security, Concentration in Cyber Operations

- **New York University Tandon School of Engineering** (New York)

  M.S. in Cybersecurity, Cyber Operations Specialization

- **Texas A&M University** (Texas)

  B.S. in Computer Science, Minor in Cybersecurity, Letter of designation in Cyber Operations / B.S. in Computer Engineering, Minor in Cybersecurity, Letter of designation in Cyber Operations

- **Towson University** (Maryland)

  B.S. in Computer Science with a Track in Computer Security

- **United States Air Force Academy** (Colorado)

  B.S. in Computer and Network Security

- **United States Military Academy at West Point** (New York)

  B.S. in Computer Science, Cyber Operations Track

- **University of Cincinnati** (Ohio)

  M.S. of Computer Science / M.S. of Computer Engineering, Graduate Certificate of Proficiency in Cyber Operations

- **University of Nebraska Omaha** (Nebraska)

  B.S. in Cybersecurity, Special Track in Cyber Operations

- **University of New Orleans** (Louisiana)

  B.S., M.S. or Ph.D. in Computer Science with a Specialization in Cyber Operations

- **University of Texas at Dallas** (Texas)

  M.S. or Ph.D. in Computer Science with a Certification in Cyber Operations
- **University of Texas at El Paso** (Texas)

  B.S. in Computer Science Secure Cyber-Systems (SCS) Track
- **University of Tulsa** (Oklahoma)

  B.S. / M.S. / Ph.D. in Computer Science, Specialization in Cyber Operations (a.k.a. Tulsa Cyber Corps Program)
- **Virginia Polytechnic Institute and State University** (Virginia)

  B.S. in Computer Engineering, Minor in Cybersecurity, Certificate in Cyber Operations

3.1.2. Top CS Universities According to US News & World Report

The second sample was drawn from the top universities in computer science according to the US News & World Report rankings. The methodology used by US News & World Report for ranking colleges consists of researching factors that in their view affect the quality of an education. It also includes statistical data concerning both qualitative and quantitative measures that were proposed by educational experts as reliable indicators of academic quality.

When selecting universities for this study from the list of Best Graduate Computer Science Programs by US News & World Report [4], preference was given to universities that had

their course catalog and course syllabi published online. The chosen universities are:

- Carnegie Mellon University – Pittsburgh, PA
- Cornell University – Ithaca, NY
- University of Washington – Seattle, WA
- Georgia Institute of Technology – Atlanta, GA
- University of Texas at Austin – Austin, TX
- University of Wisconsin at Madison – Madison, WI
- University of California at Los Angeles – Los Angeles, CA
- Columbia University – New York, NY
- University of Pennsylvania – Philadelphia, PA
- Purdue University at West Lafayette – West Lafayette, IN
- Rice University – Houston, TX
- Wright State University – Dayton, OH

3.1.3. Cybersecurity Colleges around the World

In order to provide a comparison of binary analysis instruction in American universities versus those in other nations, the third sample consisted of top cyber security colleges outside the USA as determined by [5]. Again, preference was given to universities that published more detail about their courses online. The chosen set is:

- Abertay University Dundee, Scotland

- De Montfort University, U.K
- Eindhoven University of Technology**,** Eindhoven, the Netherlands
- ETH Zurich, Zurich, Switzerland
- Tallinn University of Technology, Tallinn, Estonia

3.2. Data Collection

After selecting the universities to analyze, the next step was to collect information about the courses at these universities related to binary or malware analysis. In particular, the course topics of interest for this study are:

1. Introduction to malware
2. Static binary analysis
3. Dynamic binary analysis

So, for each of the universities, the first step was to determine if there is any cybersecurity degree program offered by the institution. If the university did not have any cybersecurity degree program, the computer science degree programs were identified. Once the degree programs were found, the required and elective courses for those degrees were located and a search of the course catalog or the university website was performed to collect the relevant course descriptions. Finally, when possible a full search of the university's course catalog for the term "malware" was conducted in order to avoid missing any courses related to malware analysis.

Once the relevant courses were identified and the associated course descriptions were collected, an effort was made to locate the course syllabuses. This was done by first checking the college website for the course syllabus. If it was unavailable, a Google search for the syllabus using the course number was conducted. In cases where this was still unsuccessful, an email was sent to the Cybersecurity or CS Program Director or to a professor who had taught the course recently to request the syllabus. In fact, this email was sent to all of the universities in the sample even when a syllabus was found through other mechanisms, in order to ensure that it was correct and current, and that no information was missed.

In most cases the course catalog search yielded only part of the desired information, i.e. though related courses were identified, often only the course overviews and not the course syllabuses were available. For a few universities, even the course descriptor was unavailable. Hence, the next step was to send an email to the university program director or the faculty. This procedure provided the course syllabuses for most of the colleges that responded to the email. However, in a few of the responses, the respondent provided the procedure to navigate me through the college website to search the required course. This procedure was followed both for those universities and for others, who had not responded to the email, in the hopes that their websites were similarly structured.

The table below summarizes statistics related to the data collection process, grouped by university type. By looking at the table, it is evident that, as expected, the majority of NSA accredited colleges have a binary or malware analysis course available in their curriculum.

However, it is concerning that only two of the colleges from the list of top universities in the USA and only one college among the international universities have this type of course.

University type	**Mail received**	**Course available**	**Course overview available**	**Course syllabus available**
1. NSA Accredited colleges	2/19	11/19	9/19	3/19
2. Best universities in Computer science	6/12	2/12	2/12	2/12
3. International universities	3/5	1/5	1/5	0/5

Table 2: Response rate of universities

Of particular note during this data collection process, I was fortunate to receive an email from the author of the reference book that I used for my thesis: "Practical Malware Analysis." Dr. Sikorski is a cybersecurity professor at Columbia University. His book is used by over 20 universities worldwide and there is also a university course (https://github.com/RPISEC/Malware) developed and run using the book to teach malware analytical skills.

3.3. Data Analysis

Once the course overviews and syllabuses were collected, they were analyzed in order to identify major recurring themes.

3.3.1. Analysis of Course Overviews Using LDAviz

A tool called Latent Dirichlet Allocation Visualization (LDAviz) was used to group the course overviews into clusters that had similar topics. LDA is an interactive statistical model that is used to interpret topics using a collection of texts. The idea of LDA is that each document is considered to be a mixture of topics, where each topic is distinguished by words or terms. For the documents in the collection to be analyzed, each occurrence of the word is counted. This term frequency count is compared to the inverse document frequency count, which is used to find the number of times a word occurs in the entire collection of texts [6]. Hence, a directory with all the course descriptions in it was created and the LDAviz tool was launched through an R Language script to analyze the documents in this directory. Below are the screenshots of the data analysis using LDAviz. The frequency of each word within a particular group is shown by the red bars, while the blue bars show the frequency within all of the documents. For example, the word “analysis” appears 10 times in group 1 documents out of 25 times in all the documents. Hence, hovering over different groups lists the top words in terms of importance to those groups.

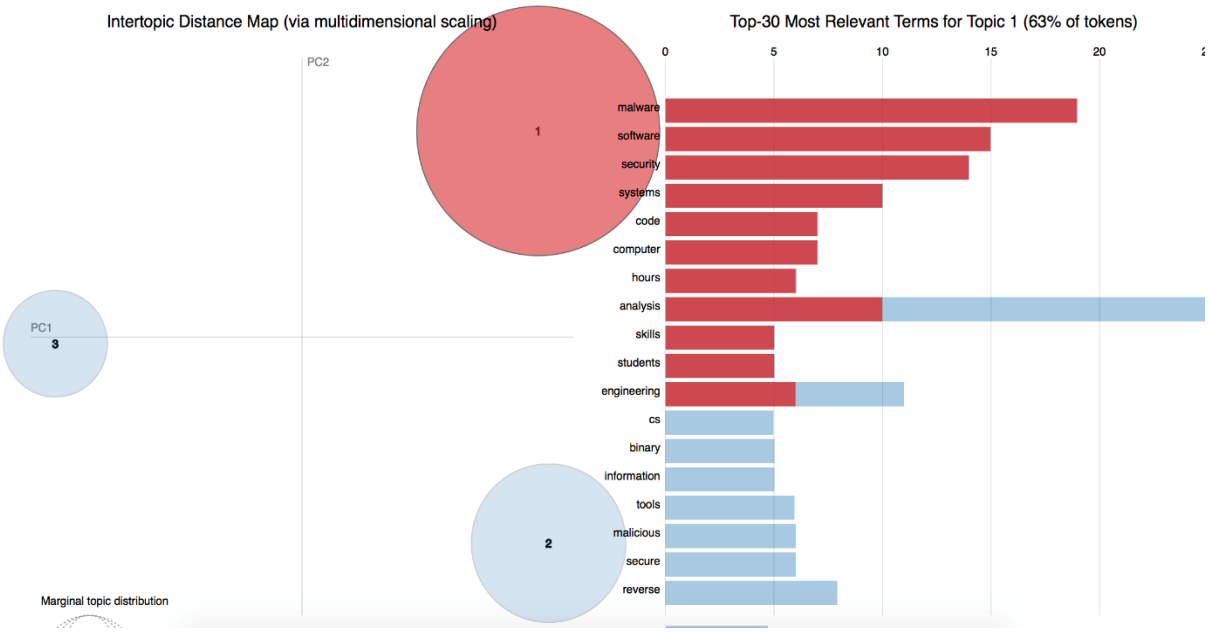

Figure 16: Group 1 of LDAviz data for course descriptions

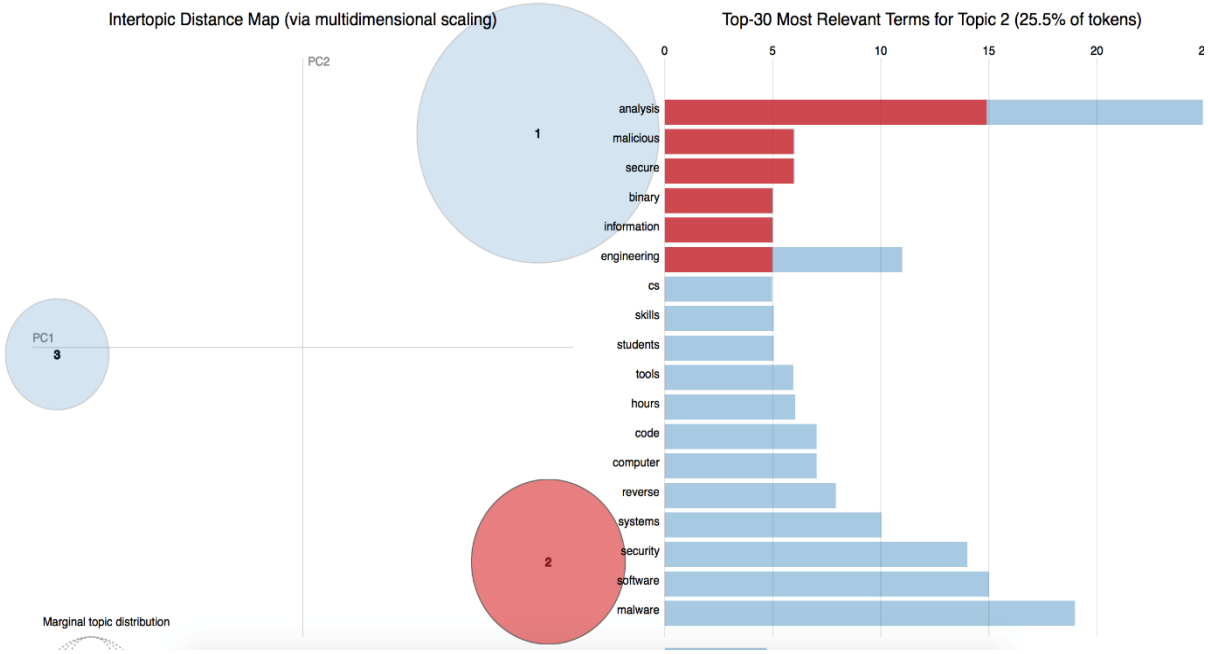

Figure 17: Group 2 of LDAviz data for course descriptions

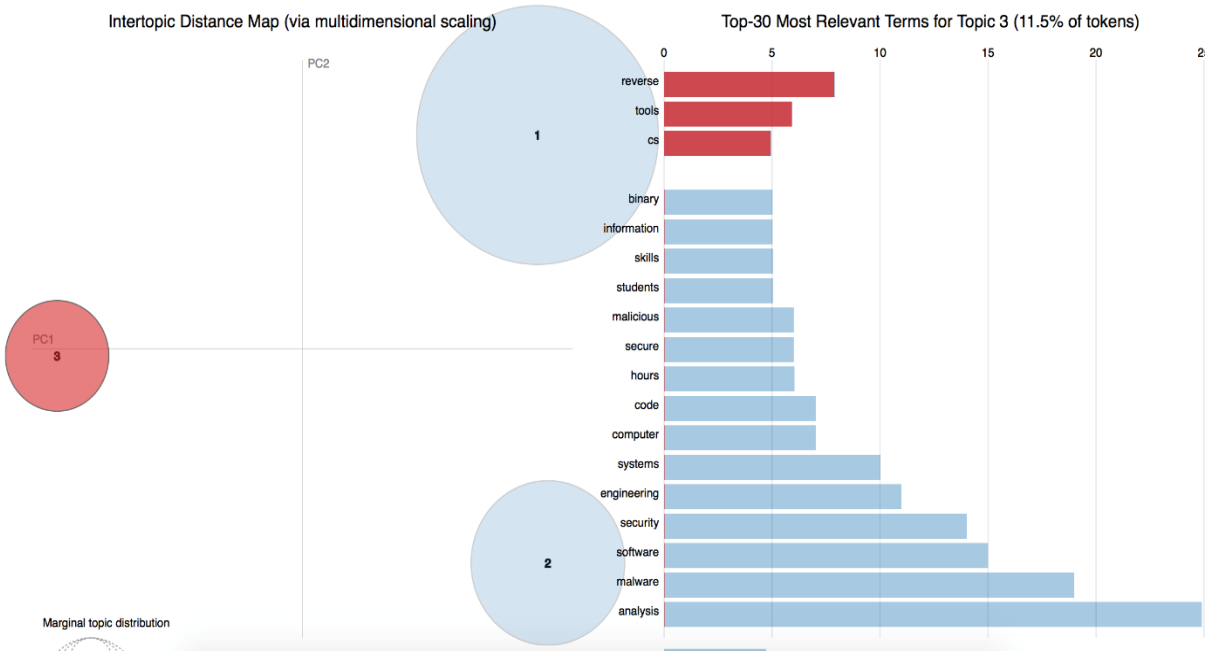

Figure 18: Group 3 of LDAviz data for course descriptions

From the above screenshots, we can infer that Group 1 is focused on analysis as a part of general courses on software engineering and systems security. Group 2 consists of subjects that involve binary analysis, security from malicious code, and information engineering, and Group 3 contains courses that include reverse engineering tools. These screenshots will be compared to those in Chapter 4, which analyze the skills employers are interested in based on job postings. If there were little or no similarity between the datasets, it would imply that universities are not training students in the skills necessary to succeed in today's malware analysis industry.

3.3.2. Analysis of Course Overviews Using a Word Cloud

The second tool used for analyzing the course overviews was a word cloud generator. This is an online tool that generates a pattern of clouds for a given text based on the algorithm (d3-cloud) for placing words without overlap. This is the link for the tool: https://www.jasondavies.com/wordcloud/. The course descriptions were copied and pasted into this website as text input, which resulted in the word cloud shown below:

Figure 19: Word cloud for course overviews

The word cloud generator of course overviews implies that the major parts of the courses are focused on malware analysis. This is unsurprising since these were the keywords used

to identify relevant courses. The next topic being emphasized highly includes software security, followed by reverse engineering tools. This analysis confirms the general conclusions gathered from LDAViz.

3.3.3. Manual Inventory of Course Syllabuses

For the set of universities for which a course syllabus was available, the goal was to identify and categorize the specific topics being covered in the university courses. The table below shows the list of malware analysis topics being covered in the course syllabi along with the universities that teach those topics.

Malware analysis topics	**Universities teaching the courses**
Introduction	Air Force Institute of Technology, University of Cincinnati, Columbia University, Wright State University
Assembly	Air Force Institute of Technology, University of Cincinnati, Columbia University
Binary-level C data structure and Construct analysis	Air Force Institute of Technology, University of Cincinnati, Columbia University, Wright State University
Static analysis	Carnegie Mellon University, University of Cincinnati, Columbia University, Wright State University
Memory management	Air Force Institute of Technology, University of Cincinnati
Dynamic Analysis	University of Cincinnati, Columbia University, Wright State University
Avoiding detection	University of Cincinnati, Columbia University, Wright State University

Behavioral Analysis	University of Cincinnati, Columbia University, Wright State University
Operating Systems	Air Force Institute of Technology, University of Cincinnati
File Formats	Air Force Institute of Technology, University of Cincinnati
Incident Response	Air Force Institute of Technology, Columbia University, Wright State University

Table 3: List of universities and course topics they teach

The above table illustrates that the universities that have provided the course syllabi are teaching many of the binary analysis skills discussed in Chapter 2. It is interesting that some additional topics such as incident response are also covered, which give a better perspective of the importance of malware analysis and the context in which it is used.

The pie chart below shows the percentage of time courses spend on each of the malware analysis topics listed above. From the pie chart, it can be inferred that though colleges teach important topics related to binary analysis, a significant portion of the courses consist of introductory topics such as the course introduction, assembly code, and binary-level C data structure and construct analysis. This is a potential point of concern because it means that less time is available to cover more critical topics such as static analysis, dynamic analysis and avoiding detection.

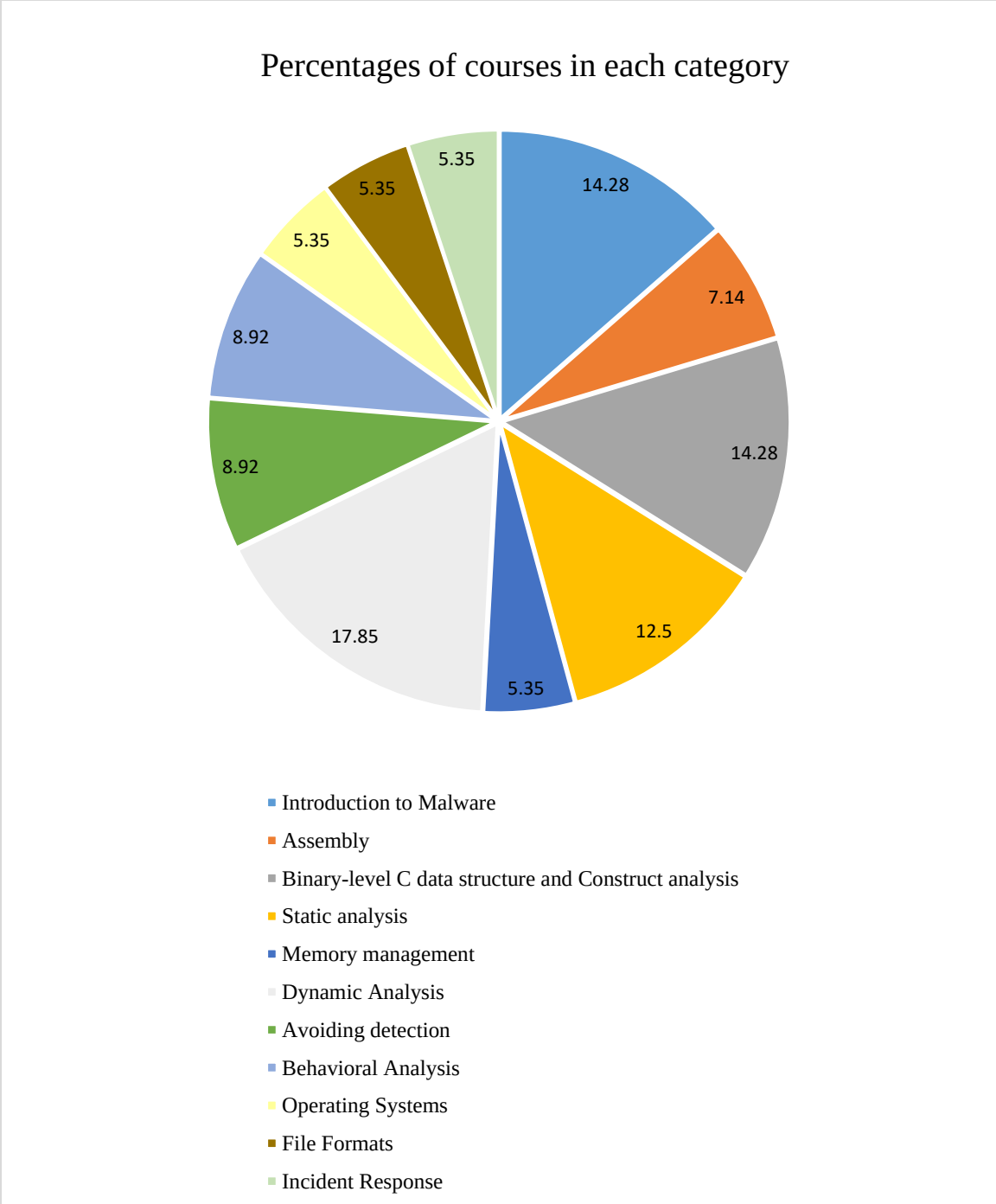

Figure 20: Pie-chart model depicting the percentage of course topics

4. Binary Analysis Skills Required in Industry

The goal of Chapter 4 is to determine the required skills necessary for malware analysts working in industry. The chapter begins with the procedure involved for finding the desired skills for job openings in the field of malware analysis. The chapter then discusses the analysis performed on the collected data to get an overview of the necessary binary skills in today's corporate world.

4.1. Data Collection

The initial step in data collection was to explore the range of job titles in the field of cyber security. As stated by the SANS Institute's Cybersecurity Professional Trends survey, most of the job positions in this field were titled things like Security/IT Director or Manager, CISO/CSO, Systems Administrator, Cybersecurity Analyst, Cybersecurity Engineer or Cybersecurity Architect, Auditor, Systems Engineer or Integrator, Network Architect or Engineer, or Forensics Investigator. Also, based on their survey cybersecurity professionals mostly work in fields such as Information Technology, Government, Banking and Finance, and Consulting Services [7]. However, according to the Information Systems Audit and Control Association (ISACA), the most in-demand job among the cybersecurity roles is that of a Malware or Security Analyst [8]. Since the main focus of the thesis is also malware analysis, a search was made for job postings titled Malware Analyst, Cybersecurity Analyst and Malware Reverse Engineer in the relevant fields they

predominantly work for. This information search was conducted in employment-related social networking sites such as LinkedIn, Indeed and Glassdoor. The job listings on these sites contain a list of the responsibilities and the required and desired skills for a job opening. These skills indicate the qualifications that the companies desire the job applicants to possess. Ten job postings relevant to malware analysis were randomly chosen from each of the three employment-related search engines for further analysis.

4.2. Data Analysis

The most important measure in conducting a gap analysis between the malware analysis skills taught in universities with what the employees are looking for is to compare and contrast the data from the job postings with that of the data analyzed from Chapter 3. Before doing this, it is vital to analyze the information collected from the job postings. To parallel the data examined in Chapter 4 with that of Chapter 3, it is sensible to use a similar analysis method as that of Chapter 3 so that the evaluation will generate similar kind of results, which will make the procedure for finding any inconsistencies much easier. Therefore, in this chapter each method used on the data from universities is employed here on the job posting data. The results are presented and compared to those from the Chapter 3 analysis.

4.2.1. Analysis of Job Descriptions using LDAviz

The notes made for job postings from the employment related networks were again fed to LDAviz for data analysis. There were about 30 job postings (about 10 from each website), which were given as input in the form of .txt files to LDAviz. Below are the screenshots of LDAviz for these job descriptions.

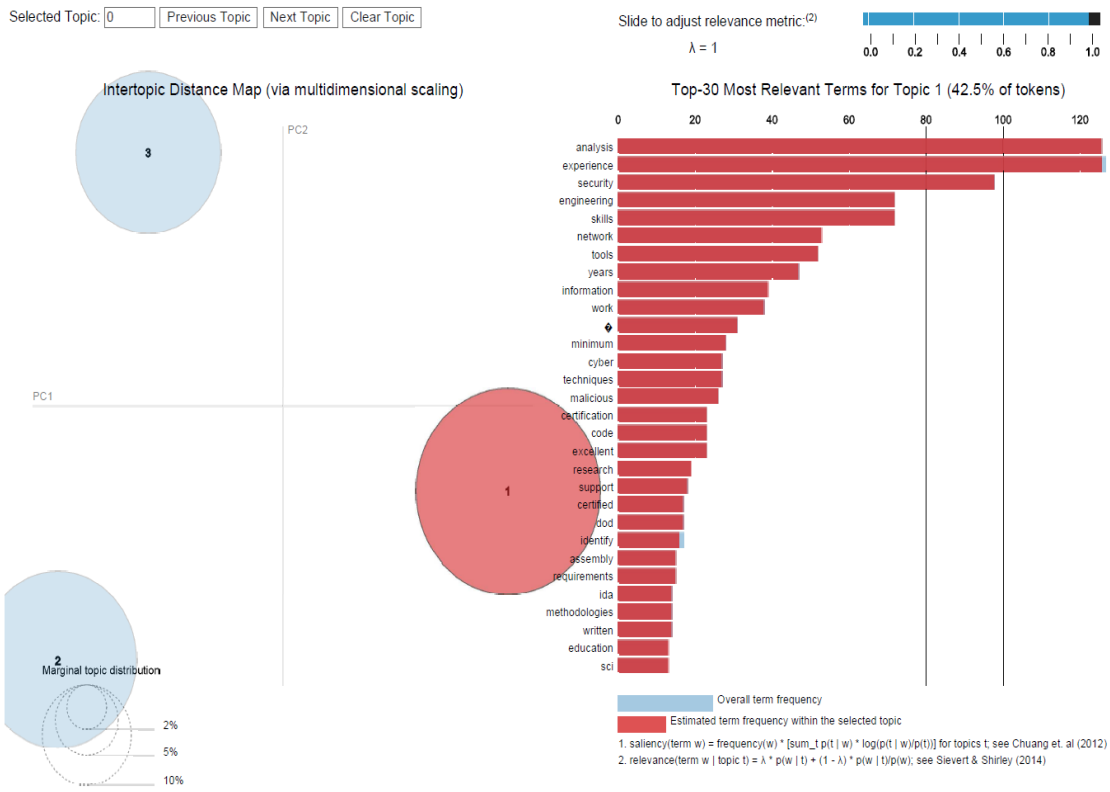

Figure 21: LDAviz data for cluster 1

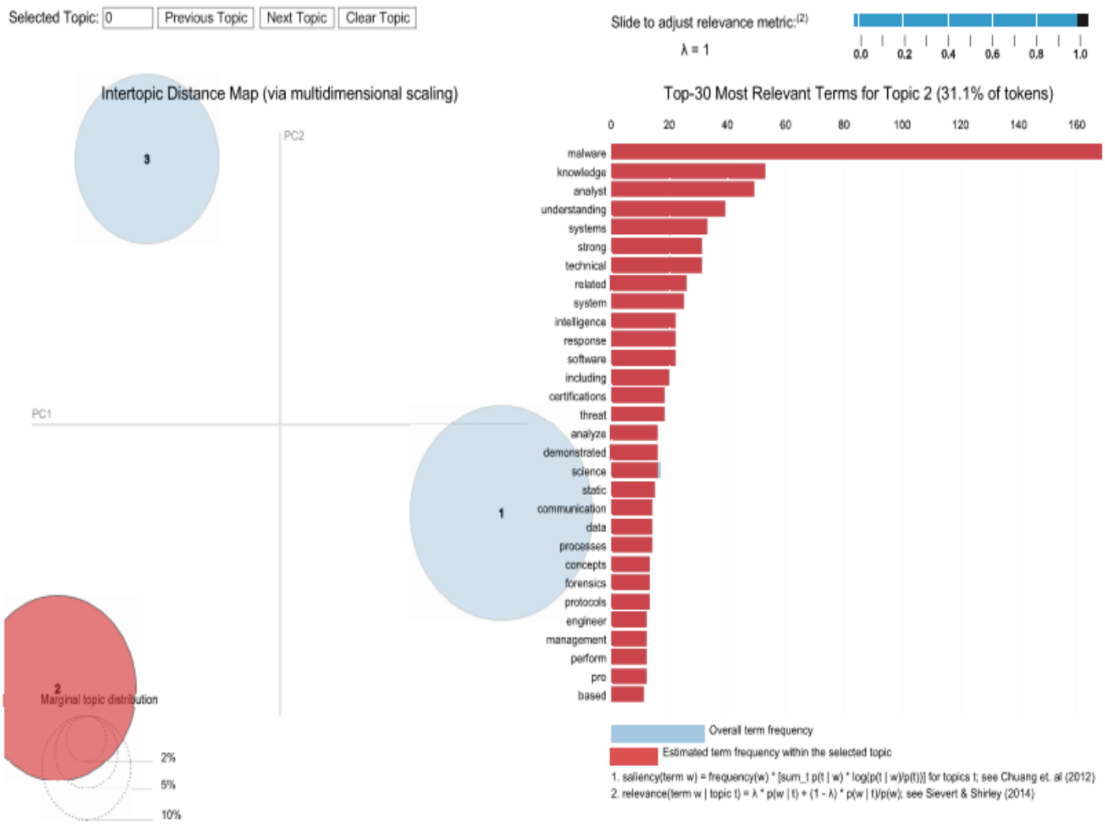

Figure 22: LDAviz data for cluster 2

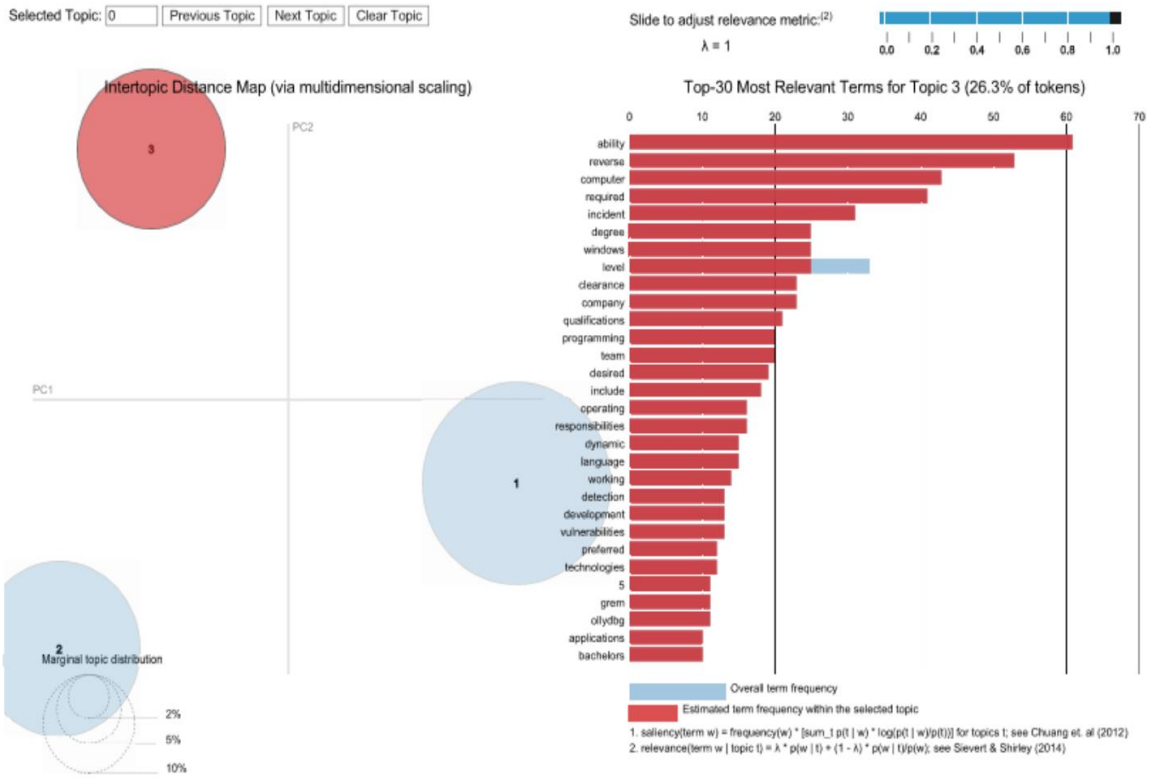

Figure 23: LDAviz data for cluster 3

It can be interpreted that Group 1 focuses on more senior-level cybersecurity jobs that involve both network and software protection. Hence, the preferred candidates have previous work experience with analysis tools. Jobs in Group 2 require qualifications such as risk analysis as well as incident response. Group 3 consists of jobs requiring more traditional malware analysis skills such as ability to perform standard reverse engineering tasks.

4.2.2. Comparison with LDAviz Analysis of Course Descriptions

Since both LDAviz datasets have three clusters, it is easy to check the similarities or differences between them. Recall from chapter 3 that for the universities, Group 1 was focused on binary or malware analysis as a part of general courses on software engineering and systems security, while Group 2 consisted of subjects which involve binary analysis, malicious code, and information engineering and Group 3 contained courses that include reverse engineering tools. The first group in both cases is similar because they both focus on software and network security. Likewise, Group 2 in both chapters are similar because they concentrate more on binary analysis techniques and methodologies for avoiding cyber threats. Also, Group 3 in both of the datasets looks alike as they focus mainly on malware analysis as part of a more general set of skills related to cyber security. Since the clusters in both the datasets could be aligned in this way, we have a preliminary result that universities are teaching the necessary skills required in the industry.

4.2.3. Analysis of Job Postings Using a Word Cloud

Job descriptions from the employment related search engines were then copied and pasted as text input for generating a word cloud. This basically takes all the words and builds a cloud representing the frequency of words in the text.

Figure 24: Word cloud generator for job descriptions

The word cloud implies that the most wanted skills that the employers are looking for include experience with malware analysis skills and reverse engineering tools. However, it is interesting to notice other skills such as knowledge of programming languages like C, C++ and Java, static analysis, assembly language skills for architectures such as x86, tools like Security Information and Event Management (SIEM), Interactive Disassembler (IDA) and certifications such as Global Information Assurance Certification (GIAC) and Certified Ethical Hacker (CEH).

4.2.4. Comparison with Word Cloud Analysis of Course Overviews

By looking at the word clouds of course overviews and job descriptions, it can be noticed that both the course topics and the job posting concentrate mainly on malware analysis and reverse engineering tools. The next topics being emphasized include system security, knowledge of software development and static analysis. Therefore, it can be concluded that malware analysis and reverse engineering tools are the topics of primary focus for both universities and the employers. This is further support for the hypothesis that universities are meeting the general needs of employers for malware analysts.

4.2.5. Manual Inventory of Job Descriptions

The initial step in collecting the information for Chapter 4 included manually extracting the responsibilities, required and desired skills the employers look for while recruiting for each job. From this a huge list was made which consists of about 345 skills. Only technical skills were included – things such as “strong communication skills” or “ability to work in a dynamic environment” were left off the list. This list was then manually grouped into categories. When possible, these categories were given the same labels as those used the course topics in Chapter 3 to maintain consistency. In some cases a category that was present in a majority of the job postings had no similar category in Chapter 3. After segregating the list, the number of skills for each topic were counted and compared to the overall list of topics.

Skills required by employers	**No. of skills in job postings**
Experience with malware	81
Experience with reverse engineering tools	32
Assembly	18
Static analysis	22
Dynamic analysis	28
Avoiding detection	22
Behavioral analysis	5
Operating systems	21
Incident response	26
Computer Forensics	10
Disassembly	11
Threat Intelligence	12

Table 4: List of skills required by employers

The table shows that the primary skill that the job posters are looking for is experience with malware, which includes developing, refining and performing malicious analysis at an enterprise level. The next skill being emphasized is the ability to reverse engineer binaries to determine the nature of vulnerabilities using tools like IDAPro, OllyDbg, Windbg and REMnux.

To visualize the relative importance of the different skills to employers, a pie chart was created based on the categorization provided in the table. This was then compared to the pie chart in chapter 3.

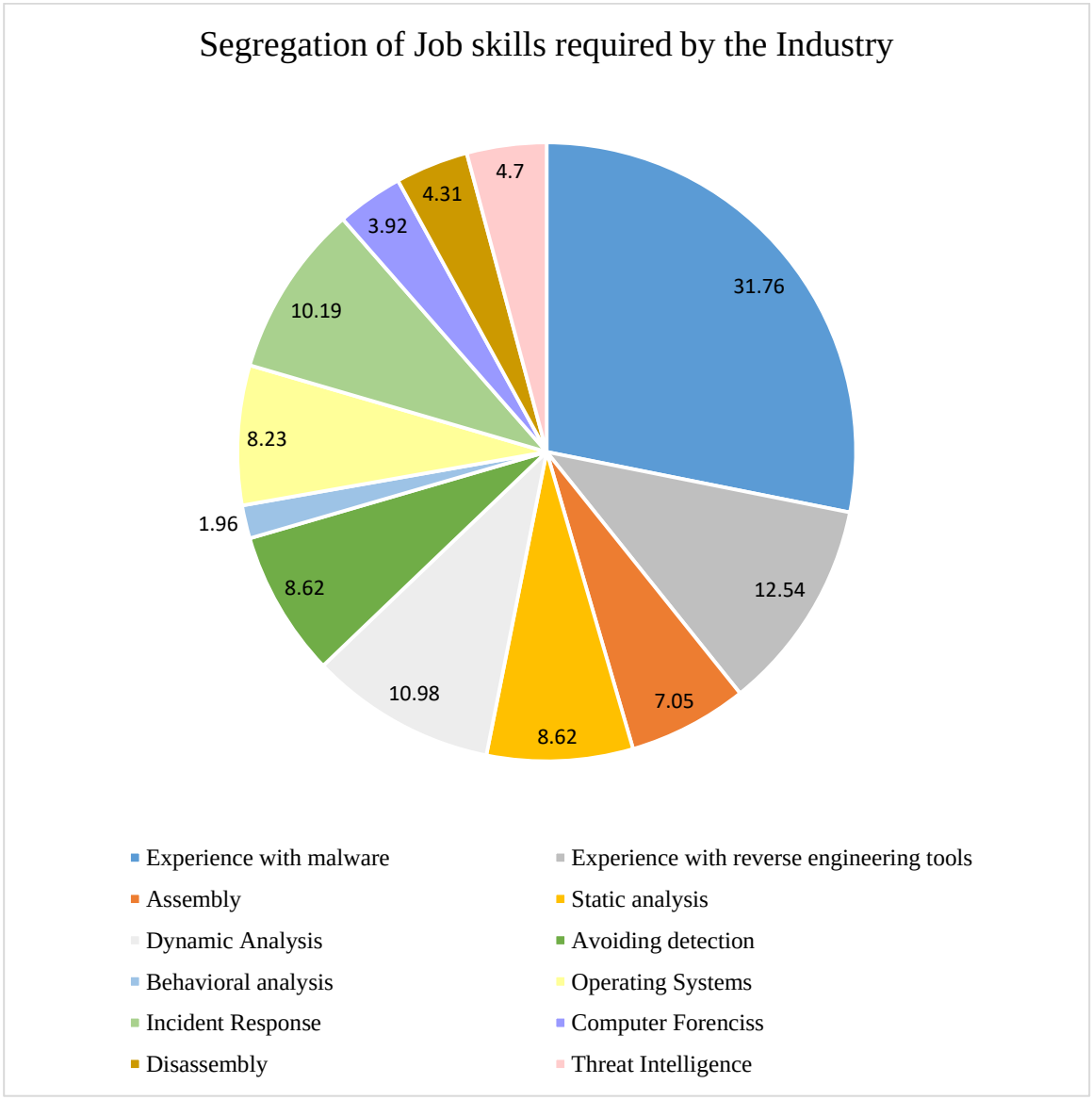

Figure 25: Pie-chart model depicting the categorization of job skills

It is unsurprising that as observed in the previous analysis tools, the most required skills for securing a job as a malware analyst include experience with malware and reverse engineering tools.

4.2.6. Comparison with the Pie Chart Model of Course Topics

In both universities and companies, priority is placed on general experience with malware (reflected in universities in multiple topics, beginning with the course introduction). However, behavioral analysis is a major topic being taught in university courses whereas it is not a skill that the employers are mainly looking for. Conversely, companies are very focused on reverse engineering tools. Universities do not often focus directly on the tools, but they generally cover them in the process of teaching students about different aspects of malware analysis, like dynamic, static, and behavioral analysis. It is likely that companies do actually value these topics a lot, but that they are not specifically listed in job postings but are implied by the naming of specific tools. In order to explore this possibility, a future study on this topic should get IRB approval to survey hiring managers in this field specifically about their importance in the hiring decision. Another difference is that companies place a higher priority on skills related to threat analysis, incidence response and computer forensics. Incident response is given very little time in the university courses analyzed in chapter, and threat analysis is not addressed at all. This is a gap that universities should potentially consider addressing.

5. Conclusion

With the increasing amount of cybercrime, there is a high demand for cyber security jobs and so is the need to fill them. Security analyst is one of the most in-demand jobs in the field of cybersecurity. However, there are 40,000 such jobs that go unfilled every year in the U.S.A [8]. Being exposed to a large number of cyber-attacks is a threat to the user's data, and exposes the dearth of specialized malware analysis skills. Students graduating from the top computer science colleges are taught few or no courses in cybersecurity. Therefore, this thesis conducts a gap analysis between the binary analysis skills that employers are looking for with those that are being taught in universities.

Before conducting the analysis it is vital to know the various skills that malware analysts require in today's corporate world. The techniques and tools used for analyzing malware can be divided into three categories: high-level profiling, static analysis and dynamic analysis. High-level profiling is the process of examining the malware's general characteristics and its effects by running the binary executable or the malicious code. Tools used in this technique include antivirus scanning, dependency analysis, process monitor and Regshot. They also include tools for analyzing network traffic such as ApdateDNS and INetSim. Static analysis is the process of observing the malware by viewing the code instead of running the executable. This process uses tools like IDA Pro and de-obfuscation techniques such as PEid. Dynamic analysis involve debugging tools like OllyDbg or WinDbg which give a clear idea about malware functioning by running the code.

To get an overview of the universities' approach to this topic, three different samples were

used. The first list consisted of 19 universities in the USA that belong to the list of NSA accredited colleges. The second set consisted of the top 12 ranked colleges in computer science according to US News & World Report, and the third sample contains the five top cybersecurity universities across the world ranked by Data Insider. It is exceptional that eleven of nineteen universities teach subjects required in the industry in the first set. However, it is very concerning that only three of fifteen universities have the courses relevant to malware analysis in the second and third samples. The course data collected from the three samples was analyzed in three different ways: LDAviz and a word cloud generator were used for analyzing course overviews, and a manual inventory along with a pie chart model was used for analysis of course syllabuses.

For finding the skillset desired for the malware analysis jobs in industry a search was made for related job postings on three job boards: LinkedIn, Indeed and Glassdoor. The skills required by each job opening were noted and analyzed using the same methods that were used to analyze the courses to maintain consistency.

To conduct a gap analysis of the malware analysis skills taught in university courses with that of the ones required in the industry, it is necessary to compare and contrast the analysis methods of courses as well as job descriptions. Through the analysis of LDAviz and Word Cloud, it can be noticed that the universities are generally covering the topics important for the malware analysis job such as exposure to malicious code and reverse engineering tools. The main exception is a lack of coverage of threat analysis, incident response, and computer forensics.

Finally, it can be summarized that there are only a few universities that offer courses relevant to malware analysis. But, it is interesting that the course topics of the institutions that do offer these subjects match closely with what the employers are looking for. However, there are many schools which do not have any courses related to cybersecurity. To meet the growing expansion of cybersecurity analysis jobs as well as prevent the current professionals to struggle during the hiring process due to a lack of the required skillset, it is highly recommended for colleges to include cybersecurity degree programs and offer courses related to malware analysis as part of their curriculum.

Finally, it is noted that future work can be done to further the survey presented in this thesis in order to affirm these conclusions and do more analysis. For example, the needs of employers in the field of cybersecurity can be more fully evaluated by using a survey, which could not be done here due to the lack of time to achieve the required Internal Review Board approval. Also, the research can be extended to a larger number of universities in USA as well as across the world.

Bibliography

[1] Benzmüller, Ralf. “Malware numbers of the first half of 2017.” G DATA, 07/24/2017, https://www.gdatasoftware.com/blog/2017/07/29905-malware-zahlen-des-ersten-halbjahrs-2017. Accessed 20 December 2017.

[2] “State-by-State Guide to Schools that Hold DHS and NSA CAE-CD Designations.” Cyber security masters degree.org, https://www.cybersecuritymastersdegree.org/dhs-and-nsa-cae-cd-designated-schools-by-state/. Accessed 11 February 2018.

[3] “List of Centers of Academic Excellence in Cyber Operations.” https://www.nsa.gov/resources/educators/centers-academic-excellence/cyber-operations/centers.shtml. Accessed 11 February 2018.

[4] “Best Graduate Computer Science Programs.” U.S. News & World Report, https://www.usnews.com/best-graduate-schools/top-science-schools/computer-science-rankings. Accessed 11 February 2018

[5] “Cybersecurity Higher Education: The Top Cybersecurity Colleges And Degrees.” Data Insider, 02/09/2018, https://digitalguardian.com/blog/cybersecurity-higher-education-top-cybersecurity-colleges-and-degrees. Accessed 11 February 2018.

[6] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet allocation." Journal of machine Learning research, 3 Jan 2003: 993-1022.

[7] “The Future of Cybersecurity Jobs.” Monster Career Advice, www.monster.com/career-advice/article/future-of-cybersecurity-jobs. Accessed 03 April 2018.

[8] Kauflin, Jeff. “The Fast-Growing Job With A Huge Skills Gap: Cyber Security.” Forbes, Forbes Magazine, 17 Mar. 2017, www.forbes.com/sites/jeffkauflin/2017/03/16/the-fast-growing-job-with-a-huge-skills-gap-cyber-security/#4426c1135163. Accessed 03 April 2018.

[9] Sikorski, Michael, and Andrew Honig. Practical malware analysis: the hands-on guide to dissecting malicious software. No starch press, 2012.