# A Python Hacking Tool for Ethical Hackers

Joseph (JT) Cavallaro - jocavall

Department of Cybersecurity, Utica College

## Author Note

Correspondence concerning this article should be addressed to Joseph (JT) Cavallaro, Department of Criminal Justice, Utica College, 1600 Burrstone Road, Utica, New York, United States. Email: jocavall@utica.edu

# Abstract

Cybersecurity has had a major impact on the security of organizations, their classified data, and their confidentiality and integrity. However, there are many people out there that have this unique skill set in computer and network security that use their knowledge and understanding to harm these organizations. Many attacks that occur use malicious code that was created using Coding Languages, which includes Python. Python is an extremely powerful tool if used in the correct way, however, this could also be used for evil. Malicious hackers are responsible for an extreme rise in cyberterrorism (cyber-attacks) around the globe, and these organizations need a way to protect their money, data, and employees. This paper will show how a hacking tool I created could be implemented into an organization's security regimen and how it could benefit the security. As well, other alternatives of security like anti-hacking tools and ethical hackers.

*Keywords:* college teaching, Python, ethical hacking, hacking tools, organizations, cyberterrorism, cyber-attacks, computer security, network security, system security

# A Python Hacking Tool That Could Protect A Company

In the world of Cybersecurity, people with a unique skill set in computer and network security and malicious intents have ‘superpowers.’ They look at organizations, small businesses, and other companies including health care facilities and hospitals, as a treasure trove of classified data and personal gain. “Cyber-attacks are a growing threat for small businesses and the U.S. economy. According to the FBI’s Internet Crime Report, the cost of cybercrimes reached $2.7 billion in 2020 alone” (SBA, 2022). As well, studies from 2017 show that these attacks “cost the global economy approximately $445 billion per year” (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 1). Many of these attackers use malicious scrips to compromise a designated target. The main purpose presented in this paper is to show how Python, a coding language, can be used to help organizations protect from and prevent these attacks. The main purpose of this paper is to show how the Python Coding Language can be used by organizations to protect from internal/external attacks. We will also cover how hackers send out different attacks using Python and the ease of doing so. This paper will meet the goals of the Senior Project. However, it will be set up differently than past student papers.

We will first cover Python, its history, and how it functions. Second, we will cover what hacking and hackers are, the types of hackers, and their methodology. The third is hacking tools, what they are, how they work, and their history. As well, we will look at different types and examples of hacking tools, exploits/vulnerabilities/attacks they take advantage of, and anti-hacking tools. Fourth, we will cover how organizations protect themselves from cyber-attacks, what ethical hackers and ethical hacking are, and the need for ethical hackers/hacking in any organization. As well, we will look at examples of the Python Coding Language being used in attacks.

Fifth, we will look at the hacking tool that I created using Python, the prerequisites needed for the tool to run properly and error-free. As well, the main Python script for my tool. Before the start of this project, this hacking tool had only nine sub-tools. By the beginning of week three, I added three more tools to make it 12. Finally, for the sixth topic, we will cover all 12 sub-tools in my hacking tool, explain how they work/function, then relate them to our purpose by describing how an organization could use that particular sub-tool to protect themselves and improve their security.

## What is the Python Coding Language?

In this section, we will cover the Python Coding Language, its history, its functions, and its inner workings.

### What is Python (Explanation & History):

Python, a short name for the Python Coding Language, is a common high-level and general-purpose programming language. The original creator of this language was “Guido van Rossum in 1991 and developed by Python Software Foundation” (Pramanick, 2022). Python is an “object-oriented, interpreted, and interactive programming language” and is similar to other programming languages such as “Lisp, Tcl, Perl, Ruby, C#, Visual Basic, Visual Fox Pro, Scheme or Java” (Holden, 2018). The main reason for the development of this tool was to emphasize “code readability,” as well; its easy-to-use “syntax allows programmers to express concepts in fewer lines of code” (Pramanick, 2022).

Rossum, the Python Coding Language creator, started working on his project “in the late 1980s” (Pramanick, 2022). He began working on Python’s “application-based work in December of 1989” in the Netherlands, “at Centrum Wiskunde & Informatica (CWI)” (Pramanick, 2022). At first, Rossum worked on Python as just a hobby to keep occupied during Christmas. It was created on the Operating System (OS) “Amoeba” and was “said to have succeeded in ABC Programming Language,” which allowed for the “feature of exception handling” (Pramanick, 2022). Earlier in his career, he was part of the creation of ABC but has seen some issues with it that could be changed. To create Python, Rossum took the “syntax of ABC, and some of its good features.” He fixed any complaints about ABC “completely and had created a good scripting language which had removed all the flaws” (Pramanick, 2022). The Python Coding Language was fully released for public access, download, and use “in 1991” (Pramanick, 2022). The Python Coding Language, when released, was able to use “fewer codes to express the concepts” that users desired when compared with other coding languages such as “Java, C++ & C” (Pramanick, 2022).

The philosophy behind Python was, and still is, very good, as the main objective is to provide its users with “readability and advanced developer productivity” (Pramanick, 2022). In 1991 when Python was released, it was able to offer users “more than enough capability to provide classes with inheritance, several core data types exception handling and functions” (Pramanick, 2022).

The name of the Python Coding Language “came from BBC’s TV Show – ‘Monty Python’s Flying Circus” (Pramanick, 2022). “On 12th July 2018,” Rossum “stepped down from” his position as “Benevolent dictator for life (BDFL)” (Pramanick, 2022). Rossum then worked “for Google” and now “Dropbox” (Pramanick, 2022).

**How Does it work?**

Python functions by using its power alongside straightforward syntax. It works using “modules, classes, exceptions, very high-level dynamic data types, and dynamic typing” (Holden, 2018). The user can call to any of these modules, classes, exceptions, etc., as well third-party modules, to access a vast amount of "system calls and libraries, as well as to various windowing systems" (Holden, 2018). As well it features other modules "written in C or C++ (or other languages, depending on the chosen implementation)" (Holden, 2018). The Python Coding Language allows the user to also use it as an "extension language for applications written in other languages" that require "scripting or automation interfaces" (Holden, 2018).

Python does not convert source code to "machine code," which hardware understands. It converts the code to "byte code (.pyc or .pyo)" (Kothari, 2021). A machine's CPU cannot understand byte code. To run this code, a Python interpreter is needed. The internal working of Python is a little complicated when first starting. However, it only requires the source code takes three steps to "generate an executable code" (Kothari, 2021).

The first step is that the Python "compiler reads a python source code or instruction," where it then confirms the code attempting execution, or "instruction, is well-formatted" and error-free (Kothari, 2021). It does this by checking "the syntax of each line" (Kothari, 2021). If an error occurs, Python will instantly stop the "translation and shows an error message" (Kothari, 2021). The second step starts if the code is error-free. This happens if the Python "instruction or source code is well-formatted," allowing the compiler to continue translation. This converts the source code to "its equivalent form in an intermediate language called 'Byte code'" (Kothari, 2021). The third and final step of the source code generation starts when the byte code finalizes. After this, the code is "sent to the Python Virtual Machine (PVM)," also known as the Python Interpreter, which converts the "byte code" back "into machine-executable code" (Kothari, 2021). This is like a second check of the code, as when an "error occurs during this interpretation, then the conversion is halted with an error message" (Kothari, 2021).

## What are Hackers/Hacking?

Hackers are customarily described as persons with malicious intentions and experts with hardware and software. This, however, is just a label that the media has created about them. Hacking itself is defined as:

“The technique of finding the weak links or loopholes in the computer systems or the networks and exploiting it to gain unauthorized access to data or to change the features of the target computer systems or the networks” (Gupta & Anand, 2017, p. 2).

This describes the changes made in the network, server, or computer hardware to gain unauthorized access/control of a system and locate classified data. The goals of a malicious hacker (black hat) may differ from a white or grey hat hacker and the average user.

### Types of Hackers:

There are three types of hackers in the world of cybersecurity and technology as a whole. The difference between these three types is solely based on their intentions or goals. The first is a White Hat Hacker; this hacker specializes in the security of networks, computers, servers, etc. They also work for organizations and perform legal attacks to protect against malicious hackers. The second is a Black Hat Hacker, the same as the White Hat. However, their goals are purely malicious, and they will perform illegal actions to complete them. The final type of hacker is the Grey Hat Hacker. This type of hacker is the same as the White Hat; however, they will perform illegal actions to protect an organization.

#### *White Hat Hackers:*

The definition of a White Hat Hacker “is a computer security specialist that breaks into and finds loopholes in the protected networks or the computer systems of some organization or company and corrects them to improve the security” (Gupta & Anand, 2017, p. 3). They use their unique skillset alongside their knowledge of computer/network security to protect an organization. They find vulnerabilities within before any Black Hat (malicious) hackers find it and use it to gain unauthorized access. White Hats are “authorized persons in the industry” (Gupta & Anand, 2017, p. 3), that may use the same methods as Black Hats; however, they do so with special permission from the organization they work for. They also do “not tell anybody until it is being fixed” (Gupta & Anand, 2017, p. 3) or until the vulnerability/exploit is completely patched and secure.

***Black Hat Hackers:***

The definition of a Black Hat Hacker is an expert in computer software, hardware, and security that has malicious intents and has the sole purpose “of stealing or damaging their important or secret information, compromising the security of big organizations, shutting down or altering functions of websites and networks” (Gupta & Anand, 2017, p. 3). They do this by violating computer security and its laws for their benefit (See Appendix A). They want to prove their skill set or knowledge and do so by completing all kinds of cybercrimes. These include credit card fraud, identity theft, ransomware attacks, creating a virus and spreading it, etc. As well, “black hat hackers illegally exploit the computer system or network to find vulnerabilities and tell others how to do so” (Gupta & Anand, 2017, p. 3). They do this so that others may also gain from the vulnerabilities and exploits they discovered.

***Grey Hat Hackers:***

These hackers are defined as “a computer hacker or security expert who sometimes violates the laws but does not have any malicious intentions like the black hat hackers” (Gupta & Anand, 2017, p. 3). This term comes from both White Hat and Black Hat Hackers. The White Hats find vulnerabilities in a computer system and networks and doesn’t tell anyone until it is patched. The Black Hats may do the same, but they spread the knowledge of these vulnerabilities to other Black Hats so that they may also benefit. The "grey hat hacker neither illegally exploits it nor tells anybody how to do so" (Gupta & Anand, 2017, p. 3). Grey Hats are seen as the line between White and Black Hat Hackers. They operate to protect a system's security and act maliciously to find and exploit vulnerabilities on a computer system.

## Methodology:

The methodology of all three types of hackers is the same regarding steps taken from start to finish when performing an attack. This is also known as the Cyber Attack or "Kill" Chain model based on a malicious hacker's mindset. There are seven steps to this attack chain: Reconnaissance; Weaponization; Delivery; Exploitation; Installation; Command & Control; Actions & Objectives. (Capano, 2019, p. 2). For this paper, as we want to look at the Ethical Hackers Attack Chain, we will condense these down to five steps. Some steps will be left out as an Ethical Hacker would not take these steps. From start to finish, they are as follows: Reconnaissance; Scanning; Gaining Control; Maintaining Access; Clearing Tracks (Log Clearing) (Gupta & Anand, 2017, p. 4 - 5). When condensing these steps together, they are as follows: Reconnaissance; Scanning – Weaponization (Intrusion); Gaining Control – Delivery, Exploitation; Maintaining Access – Installation, Command & Control; Clearing Tracks (Log Cleaning).

### *Reconnaissance:*

The reconnaissance stage is the "process of collecting information about the target system" (Gupta & Anand, 2017, p. 4). It can also be seen as the steps taken to "research, identify and select targets" (Capano, 2019, Figure 2). This is the process of "planning, observation, and research of and into a prospective target" (Capano, 2019, p. 2). Tools that hackers would use in this stage are social media like “Facebook and LinkedIn, which gather personal information or detailed information about a company” (Capano, 2019, p. 2). By doing this, the hacker can build up a target profile, allowing them to create an effective attack method. This process also includes searching for vulnerabilities or exploits in the target computer system or network. This is done by looking for the targets: Naming Domains; Services; Servers; IP Addresses; Emails; Usernames & Passwords; and Physical Location (Sengupta, 2017). By the end of this stage, the hacker will hold a large amount of data, which, if used correctly, “can construct a promising attack on the target system” (Gupta & Anand, 2017, p. 4).

### *Scanning:*

The hacker will search for open and closed ports in the scanning stage. This is done before finding what applications are in use and their versions. This is done so to search for another way into the target system. During this stage of the attack chain, “information gathered in the reconnaissance phase is used to examine the network” (Gupta & Anand, 2017, p. 4). The tools that would be used could consist of: Network Mapping; Port Scanning; Vulnerability Scanning (SNMP Sweepers, Ping Sweeps, Network Mappers, Vulnerability Scanners). (Sengupta, 2017). As well, “tools like Dialers … are used” (Gupta & Anand, 2017, p. 4). A widespread tool used is Nmap.

In the Black Hat Attack Chain, weaponization would occur in this stage. Weaponization is the pairing of “remote access malware”  and the “exploit into a deliverable payload” (Capano, 2019, Figure 2). This would consist of “an Adobe PDF or Microsoft Office file” (Capano, 2019, Figure 2).

### *Gaining Control:*

In the Gaining Control stage, the data and information gathered in the previous two steps are “used to enter and take control of the target system through the network or physically” (Gupta & Anand, 2017, p. 4). This stage is when the system becomes compromised by the attack. This stage relies on the data gathered from the previous two stages as “at this stage weaponized code can be dropped on servers, and the attacker can obtain sensitive data such as password files, certificates, or even RSA tokens” (Capano, 2019, p. 1). The attacker can now “exploit system vulnerabilities” and use them to their advantage (Capano, 2019, p. 1). They would use attacks like Buffer Overflow; SYN Flood; DDoS Attacks; Phishing; SQL Injection Attacks (Sengupta, 2017, Table 1); Cryptors; Keyloggers; Web and Database exploits (Samtani, Chinn, Chen, & Nunamaker, 2017, Table 1).

In the Black Hat Attack Chain, this is the stage when Delivery and Exploitation take place. Delivery is the sending or ‘dropping off’ of the “weapon to” the attackers “target” by use of “email attachments, websites, or USB drives” (Capano, 2019, Figure 2). The most common attack seen at this stage is “phishing emails,” which “are used against the target that has been designed around the information gathered about the target during the recon stage” (Capano, 2019, p. 1). Other Cyber Attack Chains seen or created have attack vectors that are created solely based on “software penetration methods, such as port scanning or brute force password attacks” (Capano, 2019, p. 1). However, in most Cyber Attacks, “the most effective vector is the compromise of the human asset” (Capano, 2019, p. 1).

Exploitation is when the weaponized code placed by the attacker is “triggered, exploiting vulnerable applications or systems” on the target’s computer system or network (Capano, 2019, Figure 2). Once the attacker can gain access to the computer system or network, “there is almost nowhere the attacker cannot go" (Capano, 2019, p. 1). If the attack has reached this point, "the system is compromised," any stored data is at risk (Capano, 2019, p. 1). If this attack was against an organization, which has a critical infrastructure, any control systems in place have also been compromised. This means malware was dropped into their system and resides in the crucial system. From here, "The attacker can hold data or systems for ransom or wreak havoc on the target system" (Capano, 2019, p. 1).

### ***Maintaining Access:***

The next stage is Maintaining Access. After the hacker has gained entry to the system, "the hacker maintains the access" (Gupta & Anand, 2017, p. 4). This is done to conduct "future attacks" on the target, and to conduct future "changes in the system" (Gupta & Anand, 2017, p. 4). These changes are made to block "any other security personal or any other hacker" from gaining access to the compromised system (Gupta & Anand, 2017, p. 4).

For the Black Hat Attack Chain, Installation and Command & Control reside in this stage. Installation is when "the weaponized code" dropped by the attacker "installs a backdoor on the target system to allow persistent access" (Capano, 2019, Figure 2). Command & Control occurs when the attacker can use "an outside server" to communicate "with weapons delivering hands-on keyboard access inside the target network" (Capano, 2019, Figure 2). Data that was gained from the weaponized exploits, which contains system or network vulnerabilities, allows the attacker to "use stolen credentials to obtain higher permissions to access higher security systems or file areas" (Capano, 2019, p. 2). As the attacker works their way up, they will 'unlock' "more secure and sensitive data because of escalated privileges on the compromised system" (Capano, 2019, p. 2). This means the attacker holds the ability to "access protected systems requiring high-level privileges" (Capano, 2019, p. 2). With this high-level access, they can maneuver around the system, searching systems or interconnected networks and locations where files reside. They would also look for "caches of sensitive data to exploit," which in turn would allow "higher escalation of privileges, acquisition of higher permissions, and greater access to critical systems" (Capano, 2019, p. 2).

***Clearing Tracks (Log Clearing):***

Log Clearing is the "technique of removing any leftover log files or … other types of evidence" left on the compromised system due to the attack (Gupta & Anand, 2017, p. 5). Ethical Hackers have "various tools" to expose "a hacker," which allows for them to "be caught" (Gupta & Anand, 2017, p. 5). An example of this would be penetration testing. This would consist of: Uninstalling Scripts/Applications; Changing Registry Entries; Clearing Logs; Deleting Created Folders; (Undetected ways: Tunneling, Stenography)" (Sengupta, 2017).

For any Black Hat Hacker to complete an attack, they must hide the evidence of an attack. Any method used to retrieve access to classified data leaves a trail. However, "Data compromise methods can alter data to remove evidence of compromise, plant false data trails that lead nowhere in the system, and clear operating logs to foil network forensics" (Capano, 2019, p. 2). At this stage, the attacker's goal is to conceal or delete any evidence of unauthorized access, which is done to deter "attribution and countermeasures" (Capano, 2019, p. 2). Many attacks that have taken place go unnoticed for years and are "most often discovered by accident" (Capano, 2019, p. 2). An example of this would be a "RootKit – … used to hide files, network connections, memory addresses, or registry entries from administrators to detect intended or unintended special privilege accesses to the computer resources" (Sharma, 2010, p. 12).

Performing a Denial-of-Service (DoS or DDoS) Attack is another way for an attacker to avoid detection. This method works by shutting down the system using various methods, or "to destroy data on a mass scale" (Capano, 2019, p. 2). This method is "very effective in obscuring an attack"; however, it "is very destructive" and will result "in the loss of property" (Capano, 2019, p. 2). In an organization with critical infrastructure in place, at "this stage of the attack," the shutting down or destruction of high-level servers of vital importance "can be very disruptive" (Capano, 2019, p. 2). An example of this can be seen "in Ukraine in 2015," when "an electrical grid" was shut down during an attack (Capano, 2019, p. 2).

### *Methodology Wrap-up:*

After reading about the different types of hackers, and the Cyber Attack Chain described from both views, it seems that "some way or some technique of protecting the computer system or the computer networks from the malicious hackers" should be put into place (Gupta & Anand, 2017, p. 5). "The terms "Ethical Hacking" and "Ethical Hackers" came into the industry" to solve this problem, as well why I created my hacking tool (Gupta & Anand, 2017, p. 5).

## What are Hacking Tools?

Black and white hat hackers use hacking tools to assist in attacking a target. "Hacking Tool – A program designed to assist with hacking, or a legitimate utility that can also be used for hacking" (Sharma, 2010, p. 12). Hacking has become more of a tool-driven process to identify the most widespread vulnerabilities, and in most cases now, web applications are at high risk (Pauli, 2013, p. 1). "Hackers are using a multitude of approaches and tools, including ransomware threats, to take over targeted systems" (Aziz, 2017, p. 3), and a lot of times are being seen coded "in multi-platform programming languages such as Java, JavaScript, C based languages, and python" (Aziz, 2017, p. 17).

## What is a Hacking Tool & How They Work:

To define a hacking tool, they are programs, software, or applications that can “crack or break computer and network security measures” (TrendMicro, 2022). Most tools, however, only assist in this endeavor. Depending on what system a hacking tool is created to attack, they all “have different capabilities” (TrendMicro, 2022). “System administrators have been known to use similar tools - if not the same programs - to test security and identify possible avenues for intrusion” (TrendMicro, 2022).

## Types of Hacking Tools:

In today’s age, hacking tools come in a wide variety, with most tools, “there are probably five other tools that can do the same job” (Pauli, 2013, p. 7). However, of this wide variety of tools, a few “recurring tools of the trade used by criminals and security experts” always show up (Sharma, 2010, p. 12). These consist of Security Exploits, Packet Sniffers, Hacking Tools, Root Kits, Social Engineering, Worms, Trojan Horses, and Viruses.

**Security Exploits.** These are coded software or the manual entry of commands that “take advantage of a bug, glitch, or vulnerability” in a target’s computer or network systems (Sharma, 2010, p. 12). This would, in turn, “allow all privilege” to the attacker (Sharma, 2010, p. 12).

**Packet Sniffers.** These are hardware or software that allows the collection or capture of all packets traveling on the target’s network “and decodes it to steal information” (Sharma, 2010, p. 12). This can also be used by an attacker to snoop and trail users on a target network and “collect sensitive info such as passwords” (Sharma, 2010, p. 12). Packet sniffers also allow an attacker to perform debugging protocols on “client/server communications” and “network protocol implementations” (Sharma, 2010, p. 12).

**Hacking Tools.** While also being the category for this list, Hacking tools also have a spot as these are programs “designed to assist with hacking or a legitimate utility that can also be used for hacking” (Sharma, 2010, p. 12). Examples of hacking tools that will also be explained later are: “Nmap, Nessus, Remote Security Scanner, John the Ripper” (Sharma, 2010, p. 12).

**Rootkits.** These are used to “hide files, network connections, memory addresses, or registry entries” on a target system (Sharma, 2010, p. 12). This is another way for an attacker to cover their tracks and evade detection from system administrators. This is done to gain “special privilege access to the computer resources” or collect data on a target (Sharma, 2010, p. 12).

**Social Engineering.** Social Engineering is a wide range of methods known and used by white and black hat hackers to attempt the manipulation or “manipulate people into performing an action or divulging confidential information” (Sharma, 2010, p. 12). This is the process of convincing an end-user to “provide some form of information about a system, often under fake premises” (Sharma, 2010, p. 12). Even with computer and network security being improved over the years, almost all recorded cyber-attacks and security breaches targeting organizations are blamed on “human error,” which indicates that the “employees are the ones most at fault” (Roohparvar, 2021). “The security of the web application has also improved just like the network; the attack surface has again shifted; this time toward attacking web users” (Pauli, 2013, p. 2). The problem with this is that “there is very little that network administrators and web programmers can do to protect web users against these user-on-user attacks” (Pauli, 2013, p. 2). Even with training, mistakes can happen, which could cause the compromisation of an entire organization. This allows black hat hackers to avoid “any safeguards developed in the last 10+ years for networks and web applications” (Pauli, 2013, p. 2).

**Worm.** A worm is highly malicious and most commonly used by black hat hackers. This is because “a computer worm is a self-replicating computer program” (Sharma, 2010, p. 12). It works by using the targeted network to duplicate itself and send the copies “to other computer terminals on the network” and does so “without any user intervention” (Sharma, 2010, p. 12).

**Trojan Horse.** A Trojan Horse is like a worm as it is highly malicious and most commonly used by black hat hackers. It is defined as “a program that installs malicious software while under the guise of doing something else” (Sharma, 2010, p. 12). They are mainly known for downloading and installing “backdoor programs” to a target’s computer system (Sharma, 2010, p. 12). This allows the attacker full access to the targeted system and gives the ability to a vast list of attacks, vulnerabilities, and exploits. The main objectives seen with Trojan Horses are Data Destruction, Remote Access, Spying, Keyloggers, and Phishing. Data Destruction is when the attacker deletes, destroys, erases, or overwrites any or all data on the target machine. The “remote access to the victim's computer” is when the attacker can connect to the network system from anywhere around the world (Sharma, 2010, p. 12). This is most commonly done through FTP protocol ports open on a network. Backdoors allow malicious hackers to spy on users/employees connected to a target’s computer and network systems. This, in turn, allows for the collection of data and reporting of users’ “browsing habits” on the network (Sharma, 2010, p. 12). Keyloggers, when installed on a target system, will enable the attacker to “steal information such as passwords and credit card numbers” (Sharma, 2010, p. 12). Finally, the Trojan Horse will enable attackers to perform social engineering techniques like “Phishing,” which can allow for the collection of “bank or other account details which can be used for criminal activities” (Sharma, 2010, p. 12).

**Virus.** Again, like the worm, a virus is highly malicious computer software and is most commonly used by black hat hackers. It is defined as “a computer program that can copy itself and infect a computer without permission or knowledge of the user” (Sharma, 2010, p. 13). Most viruses can “modify the copies, or the copies may modify themselves, as occurs in a metamorphic virus” (Sharma, 2010, p. 13). There are also cases of viruses designed to “damage the computer by damaging programs, deleting files, or reformatting the hard disk,” which would destroy the compromised system (Sharma, 2010, p. 13). Like the Trojan Horse, some viruses are also created to place backdoors on a target, which again “allow unauthorized access to computers” on the targeted network, system, or server.

### ***Examples of Hacking Tools:***

As stated earlier, a wide range of hacking tools assist attackers, both white and black hats, in performing the Cyber Attack Chain in fake and real-life attacks. In this section, we will cover some of the tools. They are “beginner” tools but are the backbone for all cyber-attacks, and some have been used in my hacking tool. Some examples of these hacking tools are Burp Suite, SQLmap, Zed Attack Proxy (ZAP), … Nmap, Nikto, Nessus, Metasploit, John the Ripper, and netcat (Pauli, 2013, p. 7).

**Burp Suite.** This is a widely and vastly used hacking tool as it is “accepted as the #1 web hacking tool collection,” and as well, “is a must-have for any web hacker” (Pauli, 2013, p. 7). It is used to “automate repetitive testing tasks” and helps organizations protect against attackers (S, H, H, 2022). It was developed alongside PortSwigger Research. According to the Burp Suite home page, with the help of PortSwigger Research, they can “regularly discover new vulnerabilities before criminals can exploit them,” which offers organizations “unrivaled protection against these zero-day threats” (S, H, H, 2022). This tool helps organizations find more vulnerabilities faster, helps maintain security testing routinely, and provides training for employees. However, this tool could be installed on a targets system during an attack and grant an attacker full access to any data.

**Zed Attack Proxy.** ZAP, or Zed Attack Proxy, is similar to Burp Suit; however, it "also includes a free vulnerability scanner that applies to web applications" (Pauli, 2013, p. 7). This tool was developed by OWASP, which aims to expand and "improve the security of software through its community-led open-source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences" (Bennetts, Pereira, Mitchell, 2022). The project is also maintained actively "by a dedicated international team of volunteers" (Bennetts, Pereira, Mitchell, 2022). This tool, like Burp Suite, allows for security task automation like "Quick Start command line, Docker Packaged Scans, GitHub Actions, Automation Framework, API and Daemon mode" (Bennetts, Pereira, Mitchell, 2022).

**Network Scanning & Hacking Tools.** These kinds of tools are a significant part of any cyber-attack. There are many hacking tools created for these purposes. These tools include Nmap, Nessus, Nikto, Metasploit, SQLmap, John the Ripper (JtR), and Social Engineering Toolkit (SET) (Pauli, 2013, p. 7).

***Nmap.*** Is used for network scanning and reconnaissance data collecting. It helps attackers collect data like open ports, domain names, locations, etc. In most cases, Nmap is used for port scanning (Pauli, 2013, p. 7).

***Nessus & Nikto.*** Attackers use the tools Nessus and Nikto for vulnerability scanning, which would be used in the scanning stage of an attack (Pauli, 2013, p. 7).

***Metasploit.*** A tool that holds many exploits and vulnerabilities that could be used against a target. This tool is used in the gaining access stage of an attack (Pauli, 2013, p. 7).

***SQLmap & JtR.*** Assists hackers with SQL injection attacks (Pauli, 2013, p. 7). John the Ripper (JtR) is used for “offline password cracking” (Pauli, 2013, p. 7).

***SET.*** “The Social Engineering Toolkit (SET) for technical, social engineering attacks against web users” (Pauli, 2013, p. 7).

### ***Attacks Hacking Tools Take Advantage of:***

In this section, we will not be covering the tools from above. Still, different attacks and vulnerabilities that hacking tools take advantage of could cause extreme harm to an organization. We will cover SQL Injection and other “Injection” attacks: “Cross-site Scripting (XSS)” attacks, “Cross-site Request Forgery (CSRF)” attacks, “Broken Authentication and Session Management” vulnerabilities, and lastly, “Security Misconfiguration” vulnerabilities (Pauli, 2013, p. 9 - 11).

**SQL injection.** These attacks are also known as “Injection” and are “one of the oldest attacks against web applications, but it’s still the king of the vulnerabilities because it is still widespread” (Pauli, 2013, p. 9). This attack occurs when unauthorized or “untrusted user data” is sent to a target “web application as part of a command or query” (Pauli, 2013, p. 9). This malicious data confuses the target web application and forces it “into executing unintended commands or accessing unauthorized data” (Pauli, 2013, p. 9). This attack works by feeding malicious commands or code to a target web application. This code then executes on the target and can result in data loss and the compromisation of the victim’s system and be “very damaging” to organizations (Pauli, 2013, p. 9). The vulnerabilities that SQL Injection attacks take advantage of can be seen in many different sections of the targets network and web applications. These vulnerabilities allow the attacker “to provide malicious input” (Pauli, 2013, p. 9). Of all SQL Injection attacks, the following list is of the most common attacks and what vulnerability they take advantage of: “Structured query language (SQL) queries; Lightweight directory access protocol (LDAP) queries; XML path language (XPATH) queries; Operating system (OS) command” (Pauli, 2013, p. 9).

**XSS.** Cross-Site Scripting (XSS) Attacks could be compared to a hackers candy store. XSS allows a malicious hacker to choose from many different exploits. This attack “allows attackers to execute scripts in the victim’s browser, which can hijack user sessions, act as a key logger, redirect the user to malicious sites,” and much more (Pauli, 2013, p. 10). This attack happens when a user’s “input is accepted by the application as part of a request and then is used in the output of the response without proper output encoding in place for validation and sanitization.” (Pauli, 2013, p. 10). From here, an attacker can perform numerous malicious methods to further compromise the target system or network. They can “inject malicious script (oftentimes JavaScript, but it also could be VBScript) that is then rendered in the browser of the victim” (Pauli, 2013, p. 10). This allows the scrip to proceed as the “this script is part of the response from the application,” which means the target browser will authorize the scrip to execute (Pauli, 2013, p. 10).

**CSRF.** Also known as Cross-Site Request Forgery, is similar to the process explained before and “reflected XSS” (Pauli, 2013, p. 11). This is because to carry out this attack, the attacker “must coerce the victim to perform some action on the web application” (Pauli, 2013, p. 11). This attack works when the “hacker can send a well-crafted, yet malicious, request to an authenticated user that includes the necessary parameters (variables) to complete a valid application request without the victim (user) ever realizing it” (Pauli, 2013, p. 11).

**Session Management & Broken Authentication.** These are huge vulnerabilities to any organization aware or unaware of their existence. Authentication is defined as “unique identifiers assigned to users after authenticating" (Pauli, 2013, p. 10). Broken Authentication systems have many "vulnerabilities or attacks associated with how these identifiers are used by the web application" (Pauli, 2013, p. 10). Session Management is of enormous importance as well as it is "a key component of hacking the web user" (Pauli, 2013, p. 10). The functions or systems associated with Session Management and Authentication in an organization, most times, are not set up, configured, or "implemented correctly" (Pauli, 2013, p. 10). In turn, this causes malicious hackers unauthorized access, and allows for the compromise or changing of "passwords, keys, session tokens, … password reset, password change, and account recovery" and as well will enable them to take advantage of vulnerabilities "to assume other users' identities" (Pauli, 2013, p. 11).

**Security Misconfiguration.** The final vulnerability that can be taken advantage of is the target's Security Misconfiguration. This vulnerability solely works in the overuse/lack of security being used by the target network and/or web "application stack," which "refers to operating system, web server, and database management systems that run and are accessed by the actual web application code" (Pauli, 2013, p. 11). The risk that comes with this is extremely harmful to any organization, especially when "security hardening practices aren't followed to best protect the web server from unauthorized access" (Pauli, 2013, p. 11). Many vulnerabilities can "plague" a target's computer, network, web application, and security systems in place (Pauli, 2013, p. 11). They are as follows: "Out-of-date or unnecessary software; Unnecessary services enabled: Insecure account policies; Verbose error message" (Pauli, 2013, p. 11).

**Human Error.** Many of the vulnerabilities seen in this section are caused by human error. This is because of the lack of security in place due to being unsuccessfully configured or because an employee and/or user on a computer system was convinced into downloading and installing malware by an attacker. Human error cannot be fixed, even by training, as attackers evolve even faster than the security. (Capano, 2019, p. 41 - 44)

### *Anti-Hacking Tools:*

**Anti-Hacking Tools.** Tools that are created to do the exact opposite of the hacking tools described before. They are "security tools available to protect against various types of hacking methods" (Sharma, 2010, p. 13). These consist of: Firewalls, Intrusion Detection Systems (IDS), Anti-virus, Encryption, Vulnerability Scanners, Port Scanners, Authentication, and Passwords.

**Firewalls.** These programs are extremely common on all different device types. The primary purpose of firewalls is to "prevent outsiders from gaining access to your network" or computer systems (Sharma, 2010, p. 13). They work by using rules that exclude and "sort out wanted and unwanted addresses" (Sharma, 2010, p. 13). The three main types of firewalls are as follows: "Network Address Transition (NAT); Packet filter; and Proxys" (Sharma, 2010, p. 13).

**IDS.** Intrusion Detection Systems (IDS) are used for computer or network system security. The purpose of these programs is to "detect intruders attacking the system," as well as to locate and correct "any unwanted changes in the system" (Sharma, 2010, p. 13). There are two main types of Intrusion Detection Systems: "Network Intrusion Detection Systems; Host-based Intrusion Detection Systems" (Sharma, 2010, p. 13).

**AVS.** Anti-Virus (AV) or Anti-Virus Software (AVS) are programs or applications created to help users protect their systems securely. "Antivirus is software which consists of computer programs that attempt to identify and eliminate computer viruses and other malicious software" (Sharma, 2010, p. 13). The two prominent families of AVS are categorized in "approaches," they are as follows: "Dictionary approach, Suspicious Behavior approach" (Sharma, 2010, p. 13). The problem with this kind of software is that many fake sources exist for these programs. When installed onto the system, the program runs as it was the original but in the background performs malicious actions and compromises the user system. These are the Trojan Horses of the security world and the average daily internet user.

**Encryption.** Just like the AVS, is used by malicious hackers in attacks that cause the complete compromise of a system. “Encryption is the process of transforming information, to make it unreadable to anyone except those possessing special knowledge, referred to as a key” (Sharma, 2010, p. 13). This process turns plain-text, text readable by humans, into “ciphertext,” which is unreadable without knowledge of private and/or public keys (Sharma, 2010, p. 13). “Encryption methods can be divided into two types of algorithms: Symmetric-key algorithm; Asymmetric-key algorithm” (Sharma, 2010, p. 13). These malicious attacks are called Ransomware attacks, which take advantage of “malware and Trojan forms” and block any user attempting to gain access to any sections of their computer and/or network systems (Aziz, 2017, p. 6). This also holds the targets systems at “crypto-ransomware, which only encrypts the user files” (Aziz, 2017, p. 6).

**Vulnerability Scanners.** Programs that are created to “search for and map systems for weaknesses in applications, computers, or networks” (Sharma, 2010, p. 13). White hat hackers could use this to look for vulnerabilities on a system needing to be corrected. However, black hat hackers can also use these programs on a target to find vulnerabilities that could be exploited to gain successful privilege access. Types of vulnerability scanners consist of: “Network Scanners; Web Application Security Scanners; Computer Worms” (Sharma, 2010, p. 13).

**Port Scanners.** Software created for the sole purpose of locating “open ports” on a target network (Sharma, 2010, p. 13). They can gather data like IP Addresses, domain names, physical locations, etc. The most common application used is Nmap, which I have also included in my hacking tool. These scanners are used by white hate hackers to "easily detect most vulnerable ports so that they can be blocked," but black hat hackers could use this to find vulnerabilities on a target network (Sharma, 2010, p. 13).

**Authorization.** This is known as "the process used to decide if person, program, or device X is allowed to have access to data, functionality, or service Y" (Sharma, 2010, p. 13). In simple terms, this is like a card swipe door lock used on rooms or buildings that require special access to enter. If you do not have the means of authentication (swiping the card), you cannot get in. This coincides with the next topic of passwords. "Passwords are the most reliable option for prevention against hacking" (Sharma, 2010, p. 13). "According to a study, even the hackers find it quite challenging to crack passwords" (Sharma, 2010, p. 13). This is a very reliable method of protection if the password is strong. Two extremely common methods can easily crack weak passwords: "Dictionary attacks; Brute force attacks" (Sharma, 2010, p. 13). These attacks use a word bank to test an excessive number of passwords on a target. However, with solid passwords, "you can protect your documents or system from being hacked" (Sharma, 2010, p. 13). Please keep in mind that this is not the only means of protection that should be in place. This section only helps if correct precaution is taken in other security areas.

## How Organizations Defend from Cyber-Attacks

Computer technology, cybersecurity, or just average internet users, have access to two of the greatest yet worst inventions: computers, and the internet. With computer technology, large organizations are able "to conduct their operations with a level of convenience and efficiency like never before" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2). As covered previously, this is because of Black Hat Hackers, people that are masters with computer software, hardware, and security, with malicious intent, "often leverage dangerous cyber tools or assets to conduct destructive cyber-attacks against technologically driven organizations" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2). "These acts of cybercrime lead to huge damages in areas of business, healthcare systems, industry sectors, and other fields" (Aziz, 2017, p. 5). Cyber-attacks are "the deliberate exploitation of computer systems through the use of malicious tools and techniques," and such techniques would include "Ransomware, Zeus Trojans, and Keyloggers," as well "SQL injections, and DDoS" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2). "Ransomware is considered as a high-risk threat, which is designed to hijack the data" (Aziz, 2017, p. 5). The goal of all kinds of cyber-attacks is to be able to "cover the most file extensions and be able to exploit maximum amount of data" (Aziz, 2017, p. 19). This caused the "global economy" to owe around "$445 billion per year" on damages and reparations (Samtani, Chinn, Chen, & Nunamaker, 2017). The targets of these attacks were "health-care organizations like Premera Blue Cross, government entities such as the Office of Personnel Management (OPM), and large retail and consumer companies including Target, Home Depot, Sony, and Xbox Live" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2).

Some companies like "FireEye and Cyveillance" have started providing "Cyber Threat Intelligence (CTI) reports" to their users (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2). These were created to assist organizations in mitigating cyber-attacks. However, these reports are based on previous cyber-attacks, not current (See Table 1 & Figure 1). These reports use data gathered from actual cyber-attacks by using methods like:

"network logs, antivirus logs, honeypots, database access events, system login attempts, and intrusion 1024 SAMTANI, CHINN, CHEN, AND NUNAMAKER defense system/intrusion protection system (IDS/IPS) events" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2 - 3). This system has a lot of errors, so another way for an organization to protect itself from Black Hat

Hackers is to hire White Hat Hackers/Ethical Hackers. Another option would be a Grey Hat Hacker; however, they will take unconventional routes when performing security maintenance. The following section will go over ethical hacking/hackers and their essential part in the industry.

## Ethical Hacking/Hackers:

This section will cover Ethical Hacking, Ethical Hackers, and the Code of Conduct they are required to follow. As well their need in any organization, and how they can help protect them. Finally, we will look over different cases of Python being used in malware or other attacks against organizations and other targets.

### *Code of Conduct, Ethical Hacking & Ethical Hackers:*

Ethical Hacking, AKA: Penetration Testing and White Hat Hacking. This type of hacking is performed “by an individual or a company, which helps find threats and loopholes in the computer system or network’s security of the organization” (Gupta & Anand, 2017, p. 5). Malicious (Black Hat) hackers also use these ethical hacking methods. The most significant difference between them is that White Hats are legal and use their skillset/knowledge to be productive, not malicious purposes and personal gain. Any data or vulnerabilities gathered from Ethical Hacking are used to maintain a company's computer and network security, protect users, and protect from future attacks. It is also essential to “understand the time and place for appropriate and ethical use of the tools and techniques” (Pauli, 2013, p. 1). “It’s all fun and game until the FBI shows up!” (Pauli, 2013, p. 1).

Ethical Hackers, also known as White Hat Hackers, are the personnel performing Ethical Hacking in cybersecurity. They are professionals hired by companies to protect them, as explained in the last paragraph on Ethical Hacking:

"…they are the computer experts who hack the computer system or network earlier and correct or fix all the security issues in the system or network before they are being noticed by the bad hackers who tries to break in or act maliciously" (Gupta & Anand, 2017, p. 5).

There is a strict Code of Conduct that an Ethical Hacker must follow. Staying ethical while hacking seems like a tricky task depending on the organization that hires them: They must single out and establish the integrity of the organization's data (confidentiality and privacy) before any kind of hacking starts. They should not "violate any rule and regulations" (Gupta & Anand, 2017, p. 5) put forth by the organization (the tricky part); Any hacking done by them must stay consistently transparent with the organization. This must be done before and after any hacking (Gupta & Anand, 2017, p.5); The intentions (Goal & Purpose) of any Ethical Hacker must stay extremely clear and cautious. This is because they do not want to cause the destruction of data or harm the organization (Gupta & Anand, 2017, p. 5); They must be inside the organizations' guidelines that hire them and never go past it, or it could destroy data, harm the client, or cause the immediate termination of their job (Gupta & Anand, 2017, p. 5); After completing any hack done to the client, any confidential or classified data found during the attacks must never be disclosed and must remain classified (Gupta & Anand, 2017, p. 5).

### ***The Need Of Ethical Hackers In The Industry:***

Cyber-attacks caused by malicious Black Hat Hackers have costed "the global economy approximately $445 billion per year" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2). Organizations must rely on Ethical Hackers and Cyber Threat Intelligence (CTI) to mitigate or cease malicious cyber-attacks. This also includes "threat intelligence related to computers, networks, and information technology (IT)" (Samtani, Chinn, Chen, & Nunamaker, 2017, p. 2).

The Need for Ethical Hackers in any industry as of 2021 is at an all-time high. The rate at which cybercrime has risen over the years has been alarming. Many attacks that occur are targeting private organizations and government agencies. According to CISOMag, “According to a Ponemon Institute study, the cost of data breaches jumped from $3.86 million to $4.24 million in 2021” (Cisomag, 2021).

Every organization has its own classified data/information, and any of this data could be breached at any time. Anyone with the necessary knowledge and skillset can attack an organization, gain access, and breach their data. Another result of this would be the destruction of data. For any organization to protect itself, they hire White Hat Hackers (Ethical Hackers). The organization proceeds to “…allow them to hack their own systems ethically any find flaws or loopholes in their systems and correct them before any hacker hacks it” (Gupta & Anand, 2017, p. 5). They start by attempting attacks that Black Hats use on the internet, like SQL Injection. However, “Before that, there is need of knowing Linux operating systems and what are their use in performing hacking attacks” (Gupta & Anand, 2017, p. 5).

**Python Coding Language Used in Attacks:**

This section will go over different examples of malware or attacks coded using Python. We will first cover the PWOBot, then Ransomware Attacks, and finally, two examples of Web-Crawlers.

***PWOBot:***

Organizations residing in Europe have seen a Python coded malware attack their companies. Dubbed “The PWOBot Malware,” because it is Python-based, this “malware could easily be ported to different operating systems, says Palo Alto Networks” (Broersma, 2016). Researchers in the field of IT Security “discovered an unusual family of malicious code written entirely in the Python programming language" (Broersma, 2016). This malware was created using a "modular design," which, allows it to choose from "a selection of different attacks" (Broersma, 2016). This selection includes "executing files, logging keystrokes, mining bitcoins using the affected system's CPU resources, executing arbitrary Python code, and communicating with a remote server, according to Palo Alto Networks" (Broersma, 2016).

The organizations that were targeted, as well as "the open internet," have now seen "six" variants of the "PWOBot malware" (Broersma, 2016). There are a total of "12 variants" of this Python-based malware that are "known to exist" (Broersma, 2016). This malware has been seen "dating back at least to the end of 2013" as being involved in malicious attacks on organizations (Broersma, 2016). The targets of many of these attacks were "European organizations, particularly in Poland" (Broersma, 2016). A few years later, in "2015," this list of targets expanded to "a national research institution, a shipping company, a large retailer and an IT organization, as well as a construction company in Denmark and an optical equipment provider in France, Palo Alto said" (Broersma, 2016). The PWOBot was also seen "affecting Microsoft Windows platforms" (Broersma, 2016). Because Python's "code is cross-platform," it can easily be transferred to "Linux and OSX operating systems," which makes this malware "a potentially significant threat" (Broersma, 2016). The creator of this malware clothed the PWOBot "as various Windows utility programs and has been spotted on popular Polish file-sharing site chomikuj.pl, Palo Alto said" (Broersma, 2016).

It is unclear how the malware could get onto the compromised systems. It is speculated that it was "an email-borne phishing attack or via a user download" (Broersma, 2016). It was also noticed that this malware used "the Tor network," an anonymous and encrypted internet browser, to "communicate with remote servers." (Broersma, 2016). This could help other "organizations spot it on their systems" before it's too late and too much damage is done (Broersma, 2016).

"'While (Tor) provides both encryption and anonymity, it also should raise alerts to an organization's network administrators if viewed, as such traffic likely violates said organization's policies,' Palo Alto said " (Broersma, 2016).

***Ransomware:***

As stated earlier, "Ransomware is considered as a high-risk threat, which is designed to hijack the data" (Aziz, 2017, p. 5). This could consist of a locked or encrypted computer system being held at a 'bond' or ransom. The word 'Ransomware' comes from merging the words "ransom and software" (Aziz, 2017). It is defined as "a program that is designed to attack a targeted system with the aim of holding the user as a hostage and restricting users from accessing their devices" (Aziz, 2017, p. 6). As well, Ransomware can be used, in most cases, to "encrypt the user's data, forcing the victim to pay the ransom" (Aziz, 2017, p. 6).

There are many different forms of Ransomware, and they differ in how they "evolved from the malware and trojan codes" (Aziz, 2017, p. 5). The most common cipher algorithms like "AES and RSA" can be used and have been seen in ransomware, used in the infection stages of attack (Aziz, 2017, p. 5). This is done in the infection stage to "produce complex threats" (Aziz, 2017, p. 5). The most common or "practical approach for data encryption uses a python programming language to show the efficiency of those algorithms in real attacks by executing this section on Ubuntu virtual machine" (Aziz, 2017, p. 5).

"Most of the ransomware codes are written in multi-platform programming languages such as Java, JavaScript, C based languages, and python to cover the most file extensions and be able to exploit maximum amount of data. " (Aziz, 2017, p. 19)

Most times, the creator of the Ransomware will take advantage of “malware and Trojan forms” so that it can evade as well compromise (infect) the system being targeted (Aziz, 2017, p. 6). There are “two major types” of Ransomware, the first is “lockers” (Aziz, 2017, p. 6). This hinders any user from gaining access to their entire computer and/or network systems. As well as holding the targets’ systems at ransom. Also known as “crypto-ransomware, which only encrypts the user files” (Aziz, 2017, p. 6). The main targets, seen from previous Ransomware attacks, are organizations, healthcare facilities like hospitals, companies, and endpoint users. They come in many forms which differ in context. A few examples would be “email attachment, compromised websites, advertising, running untrusted program on the machine, sharing networks and communicating with an infected system” (Aziz, 2017, p. 6).

***Web-Crawler:***

A study done, using a Python coded web crawler, was conducted on “the recruitment website of hxxps://www.lagou.com, working place as "Hangzhou" (Ying & Zhang, 2019, p. 81). The target webpage is a “vertical recruitment platform of the internet industry in China,” and the name “Hangzhou is known as a capital of the internet” (Ying & Zhang, 2019, p. 81). The study was done to gain data on “the big data recruitment information in the ocean of Web,” however, the main goal was to create a Python web crawler (Ying & Zhang, 2019, p. 81). This was done by “using the Pandas + Matplotlib to implement data cleansing, data analysis, and data visualization” and working with “Python third-party libraries to develop,” and “use web crawler technology to collect recruitment information rapidly and accurately” (Ying & Zhang, 2019, p. 81).

Another study was done to create a web crawler with three goals similar to the one above. The first goal is to create a web crawler with “numerous anti-crawling countermeasures” to collect malicious hackers’ shared vulnerabilities, exploits, malware, etc. This would happen “on an ongoing basis” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 94). The second goal aims to “employ a state-of-the-art deep learning approach, Long Short-Term Memory (LSTM)” and “Recurrent Neural Network (RNN)” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 94). This is done to automatically classify data found and sort them into “pre-defined categories on the fly” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 94). The third goal is to create “interactive visualizations enabling CTI practitioners and researchers to explore collected exploits for proactive, timely CTI” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 94). The data collected from this study will be used to prove and indicate the problem of “system and network exploits” are being “shared significantly” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 94).

The web crawler created was coded “in the Python programming language” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 96). This was done as Python gives the user many “libraries dedicated to web crawling, HTML parsing, data science, and other important functionalities” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 96). They used the Python library “requests_html,” “BeautifulSoup,” and the library “Keras” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 96). “requests_html” works with HTTP requests needed for the crawler to access the webpage. “BeautifulSoup,” a module used in my hacking tool, is used for “HTML parsing” (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 96). Finally, the library “Keras” is for data classification (Williams, Samtani, Patton, Chen, & IEEE, 2018, p. 96).

## My Hacking Tool

This section of this paper will cover the hacking tool that I created and the needed requirements to run the tool at this stage in the process. This is not a tutorial on using or setting up the hacking tool, software, and other requirements. It will be going in-depth on how the tool works, the kind of attack, and relating it to our purpose. The dependencies for Ubuntu, Python and its modules, third-party modules, the main Python file used to run the tool, and all 12 sub-tools.

This tool has been in the works for a while now, but I have not had much time to work on it, as in creating more tools, debugging the tools, and adding customization/functions to each sub-tool. In total, this tool has undergone a month of work, so there are still bugs. A beginner also made this tool, so it is not highly complex.

To view the code for yourself, or if you would like to test it out, it can be found on my Github: https://github.com/JT-Ca/JT-Ca. As well, I have uploaded the code to my Utica email Google Drive along with the old file paper report on my hacking tool “CYB 339 – Final Project Report – jocavall.docx”:

https://drive.google.com/drive/folders/1Obge_KUuQ4DJuSfVicKp5PRJL_2MJPO7.

**Prerequisites:**

For my hacking tool to work/run correctly and successfully, a few pieces of software are required to be installed onto your PC (Cavallaro, 2021, p. 4). First is VMWare Workstation version 16.x or higher (Cavallaro, 2021, p. 4). It will also require that an Ubuntu 20.04 or higher Linux VM be downloaded, installed, and configured/set up alongside a Metasplotable2 Linux VM (Cavallaro, 2021, p. 5). The link’s to download these can be found in the references section of this paper. Ubuntu will be the primary VM used as my hacking tool was created on it

(Cavallaro, 2021, p. 5). This is standard practice when coding with Python, as seen in Ransomware in High-Risk Environments: “This paper uses AES encryption algorithm which is written in python language on the Ubuntu virtual machine” (Aziz, 2017, p. 10). The Metasploitable2 VM will be used as a target for the tool (Cavallaro, 2021, p. 5). This is because this Operating System (OS) is made to be an easy target for learning how to perform cyber-attacks (Cavallaro, 2021, p. 5).

***VMWare Workstation Pro:***

VMWrokstation is a program used to run Virtual Machines (VMs) (Cavallaro, 2021, p. 4). This allows you to install many different Operating Systems (OSs) onto a single computer (Cavallaro, 2021, p. 4). Think of this as a computer inside of your computer; in other words, think of the movie Inception (Nolan, 2010).

According to the VMWare Glossary, “A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps” (VMWare, 2022). It is also stated that there can be “one or more virtual “guest” machines run on a physical “host” machine” (VMWare, 2022). These will each run their own Operating System and run separately from each other.

So, how do these Virtual Machines work? They allow businesses to run a separate operating system in an application window on the host computer’s desktop (VMWare, 2022). They were created to accommodate “different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment” (VMWare, 2022). This means that they can perform tasks deemed too risky to test on a host environment like security testing, anti-virus testing, etc. With this being said, because the "… virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer" (VMWare, 2022).

This will be the program used to run our two Virtual Machines (VMs), Ubuntu 20.04 and Metasploitable2. These are both Linux VMs; however, the hacking tool will only run-on Ubuntu (Cavallaro, 2021, p. 5).

***Metasploitable2 Linux Virtual Machine:***

Metasploitable2 is a version of Ubuntu Linux virtual machine that is purposely left vulnerable by design for learning or testing security tools and to practice attacking common vulnerabilities (Rapid7, 2022). The second version of this VM holds more exploits and vulnerabilities than the original, which is why it is being used as a target for my hacking tool (Cavallaro, 2021). The default configuration of this VM's network interfaces "are bound to the NAT and Host-only network adapters" (Rapid7, 2022). An extreme note of caution: "The image should never be exposed to a hostile network" (Rapid7, 2022).

This will be the target for a few of the tools in my hacking tool as we do not want to break the law; we will be attacking this VM (Cavallaro, 2021). Article 156 of the New York State Penal Law has many violations if this VM is not used as a target (Penal Law, 2022), see Appendix A.

***Ubuntu 20.04 Linux Virtual Machine:***

This is our primary VM and where we will be installing all the needed dependencies to run my hacking tool (Cavallaro, 2021, p. 6). This is an open-source, "Debian-based Linux distribution" (Tech Target, 2009), or Operating System that offers users "open-source security, ultra-secure appliances, workstations (VM) and desktops, data centre automation, and more (Canonical, 2022). "Ubuntu is the modern, open-source operating system on Linux for the enterprise server, desktop, cloud, and IoT” (Canonical, 2022). It is seen as a starting point “for beginners,” however, it was created mainly “for personal computers (PCs), but it can also be used on servers” (Tech Target, 2009).

***Dependencies for Ubuntu VM:***

My hacking tool runs on the Ubuntu VM; however, it will execute any script without error unless the programs and dependencies are needed (Cavallaro, 2021, p. 6). The list below is all of the required programs needing to be installed: Python 3.10.0; pip – The newest version available; Snort & all its dependencies; Nmap & all its dependencies; Mechanize & all its dependencies (Cavallaro, 2021, p. 6). Pip and Python should be installed first, followed by Snort, Nmap, and Mechanize (Cavallaro, 2021, p. 6). These programs must also have all of their dependencies installed as well (Cavallaro, 2021, p. 6).

***Dependencies for Python:***

The first program on the list is Python (Cavallaro, 2021, p. 6). This coding language is prevalent and is similar to other languages (Cavallaro, 2021, p. 6). For my tool to work, however, you need to install the modules listed below and all dependencies for each Python 3 Module (Cavallaro, 2021, p. 6). Please note that there are some third-party modules on this list: Sys; Os; Time; PyPDF2; PdfFileReader from PyPDF2; Mechanize; Random; randint from random; http.cookiejar; CookieJar from http.cookiejar; DefaultCookiePolicy from http.cookiejar; urllib.request; re; bs4; BeautifulSoup from bs4; time as time_module; scapy[complete]; IPy; IP as IPTEST from IPy; Nmap; Ipaddress (Cavallaro, 2021, p. 6).

**Main File for Hacking Tool – FP.py:**

This Python script is the main file to be run as an exe, but a beginner made this, so it’s a start! Use the command “sudo python3 FP.py” to start the script in the terminal (Cavallaro, 2021,

p. 7 - 19). Keep in mind you must use the “cd” command to change directories; the Python command above will not work unless you are in the correct directory (Cavallaro, 2021, p. 7 - 19).

This program gives a primary user interface, a list of the tools, and how to use the program (Cavallaro, 2021, p. 7 - 19). This tool can call each of the other Python scripts used (Tools #1 - 12), and they can also call it (Cavallaro, 2021, p. 7 - 19). Three of the 12 sub-tools in this tool are located in a different directory, and they can be called as well call this file (Cavallaro, 2021). You will notice that Tools 1, 3, and 4 will have the line ‘sys.path.insert(0, '<tool-name.py')’, which is used to call the Python script in the other directory (Cavallaro, 2021). See Appendix B Figure 1 (Cavallaro, 2021).

## The Sub-Tools

In this section of the paper, we will be going over each of the sub-tools in my hacking too, how they work, if it is for reconnaissance or for malicious attacks, the purpose of the tool, and how it could be used when correlated with the goal of this paper. At the time of writing this, there are currently 12 different sub-tools.

First is the sub-tool PDF Metadata Collector, which grabs metadata from PDF documents. The second sub-tool is the HTML Grabber; this tool allows users to view a target’s web page as raw data without direct connection and anonymity. The third sub-tool is the Cookie Grabber, which grabs five unique cookies from a target webpage. The Link Parser, the fourth sub-tool, uses BeautifulSoup to display all the href links/domains on a target webpage. The fifth sub-tool, the NMap Port Scanner, scans a target IP address to find open ports. Then the sixth sub-tool, Intrusion Detection System (IDS) Tricker, sends fake packets to the IDS to confuse anyone monitoring the network. The seventh sub-tool is the TCP Calculator, which measures the distance between sending and receiving TCP packets. Eighth-place is the SYN Flooder, which sends the user inputted packets to the target network. The ninth sub-tool is the TTL Packet Parser, which monitors live network traffic. The tenth sub-tool is the Anonymous FTP Scanner, which targets FTP anonymous login access. The eleventh sub-tool, Brute Force FTP Credentials, uses a text file (.txt) to attempt many usernames and passwords to access the target FTP servers. The twelfth sub-tool, FTP Server Webpage Search, uses the credentials from the previous sub-tool to search for webpages on the target FTP servers.

## Sub-Tool #1 ~ PDF Metadata Collector - PDFMetadata.py & ANONOPS_The_Press_Release.pdf:

The PDF Metadata Collector finds metadata (creation date, author, etc.) and displays the data found to the user (Cavallaro, 2021, p. 20 - 23). This sub-tool can be used to gather data on a target, also known as reconnaissance (Cavallaro, 2021, p. 20 - 23). This tool is in a different folder than the main Python file, FP.py, and is called by the line ‘sys.path.insert(0, 'FinProj')’ (Cavallaro, 2021, p. 20 - 23).

This sub-tool could help an organization defend from or solve current attacks:

“The Australian National Data Service provides the following definition: Metadata can be applied to anything. It is possible to describe a file on a computer exactly as one would describe a piece of art on a wall, a person on a job, or a place on a map…. Metadata is structured information that describes, explains, locates or otherwise simplifies the retrieval, usage or management of an information resource” (Tech, 2017). See Appendix B Figure 2.

## Sub-Tool #2 ~ HTML Grabber – HTMLGraber.py:

The HTML Grabber uses Mechanize to search the wide web anonymously and grabs the target’s raw webpage data (Cavallaro, 2021, p. 24 - 28). When displayed to the user, it will look like a series of links and code in a string format (Cavallaro, 2021, p. 24 - 28). An attacker could use this to gather reconnaissance on a target and collect domain names and other network data that can be used to find vulnerabilities (Cavallaro, 2021, p. 24 - 28). An organization could do the same as above to protect itself from attacks. This would be because they could find a vulnerability before any malicious hacker does. See Appendix B Figure’s 3 through 5.

### Sub-Tool #3 ~ Cookie Grabber – Ccook.py & hideB.py:

The Cookie Grabber again uses Mechanize and collects five unique cookies from a target webpage; however, if the webpage does not block the connection and allows it (Cavallaro, 2021, p. 29 - 34). An example of this error occurs when using Twitter as a target (Cavallaro, 2021, p. 29 - 34). If this is used too many times and an error of “bad request” occurs, the program quits (Cavallaro, 2021, p. 29 - 34). This tool works by using two different Python scrips, not including the main file FP.py (Cavallaro, 2021, p. 29 - 34). CCook.py is the main file called by FP.py; however, CCook.py also calls the second file hideB.py (Cavallaro, 2021, p. 29 - 34). This second script is used to open the mechanize browser, provide anonymity, change the user's proxy and user agent, and clears and collects the cookies (Cavallaro, 2021, p. 29 - 34). “CCook.py uses hideB.py, opens mechanize, goes to the user inputted page, and prints the output to the user each time a cookie is collected after its cleared” (Cavallaro, 2021, p. 29 - 34). Both files, CCook.py, and hideB.py, are located in a different directory than FP.py and are called by the line ‘sys.path.insert(0, 'FinProj')’ (Cavallaro, 2021, p. 29 - 34). See Appendix B Figure 6.

This sub-tool could also be used for reconnaissance by an attacker to find vulnerabilities for injection attacks. An organization could also use this sub-tool to find obvious vulnerabilities on their system and fix them before any malicious hackers use the exploit to attack.

### Sub-Tool #4 ~ Link Parser – ParL.py & hideB2.py:

The Link Parser gathers any links found on a webpage that are 'href' links, collects them, and prints them to the user (Cavallaro, 2021, p. 35 - 40). To do this, BeautifulSoup4 (bs4) was used (Cavallaro, 2021, p. 35 - 40). There is a Regex option, but this is not set up to work correctly (Cavallaro, 2021, p. 35 - 40). Like the previous sub-tool, ParL.py is called by FP.py and vice versa, as well hideB.py renamed hideB2.py (Cavallaro, 2021, p. 35 - 40). Again, just as the last sub-tool, this uses mechanize (Cavallaro, 2021, p. 35 - 40). See Appendix B Figure 7.

This is again a reconnaissance stage sub-tool used to gather domain names from a target. Both Black and White Hat Hackers could use this to find domain names, and using these names, find vulnerabilities to exploit.

### Sub-Tool #5 ~ Nmap Port Scanner – nmapPortScan.py:

As stated in the name, the Nmap Port Scanner uses Nmap to scan a target's network to find open ports (Cavallaro, 2021, p. 41 - 45). Any available port on a target can contain a vulnerability to exploit for an attack (Cavallaro, 2021, p. 41 - 45). This sub-tool is a reconnaissance stage sub-tool, which can be used to gather data, in this case, open ports (Cavallaro, 2021, p. 41 - 45). The target for this tool was my Metasploitable2 Linux VM, as performing this attack on an actual target could result in breaking the law (Cavallaro, 2021, p. 41 - 45). See Appendix B Figure 8 through 10.

An organization or malicious hacker could use this tool to protect systems/attack a target. Finding open ports on a target could mean a treasure trove of vulnerabilities to exploit. An example of this would be if port 21 were open, then anonymous FTP attacks are possible.

**Sub-Tool #6 ~ IDS Tricker – idsTrick.py:**

The Intrusion Detection Detector (IDS) Tricker sends spoofed packets specifically designed to confuse Network Analysts (White Hat) (Cavallaro, 2021, p. 46 - 54). These packets contain data stating: ‘1234’, ‘LEMME MESS WITH YOU!’, etc (Cavallaro, 2021, p. 46 - 54). The target for this tool was Snort running on my Ubuntu 20.04 VM (Cavallaro, 2021, p. 46 - 54). This sub-tool is malicious as it is attacking a target (Cavallaro, 2021, p. 46 - 54). Knowing this, please be careful, respectful, and ethical with this (Cavallaro, 2021, p. 46 - 54). See Appendix B Figure 11 through 12.

An organization, however, could use this tool for computer and network security training and testing. This could teach a new Network Analyst or cybersecurity student how to spot attacks and teach them what to do in that situation. This could also test how secure an organization's network is.

**Sub-Tool #7 ~ TCP Calculator – TCPCalc.py:**

The TCP Packet Calculator works by targeting the TCP’s three-way handshake ACK (Cavallaro, 2021, p. 55 - 58). It calculates the distance between successful communication between the TCP and ACK packets (Cavallaro, 2021, p. 55 - 58). It calculates the difference between the two sequence numbers and gives the user the following upcoming sequence number (Cavallaro, 2021, p. 55 - 58). This could be used as a reconnaissance tool to see if the port is secure, like a ping test (Cavallaro, 2021, p. 55 - 58). An organization could use this tool to find the most optimal bandwidth usage. “The time is taken to traverse the network between two hosts affects how responsive services are, and it affects the effective bandwidth available to end hosts” (Strowes, 2013). See Appendix B Figure 13.

**Sub-Tool #8 ~ SYN Flooder – SYNFlood.py:**

The SYN Packet Flooder targets the TCP three-way handshake and floods it with packets (Cavallaro, 2021, p. 59 - 67). Using this is extremely malicious as it is a Denial-of-Service Attack (DDoS), which means this sub-tool could cause damage or harm to data during an attack on any target (Cavallaro, 2021, p. 59 - 67). Knowing this, please be careful, respectful, and ethical with this (Cavallaro, 2021, p. 59 - 67). The target for this tool was the Metasploitable 2 VM and Snort (Cavallaro, 2021, p. 59 - 67). See Appendix B Figure 14 through 15.

This sub-tool could be used by an organization but not in security testing but instead security training. It could teach a new IT Security employee how to halt and defend against/from attacks like this. Using this tool for security testing could cause the loss or damage of data and the shutting down of the organization's network.

**Sub-Tool #9 ~ TTL Pkt Parser – ParseTTL.py:**

The TTL Packet Parser sub-tool is buggy as it does not stop unless the user uses the keyboard shortcut ‘Ctrl’ + ‘z’ (Cavallaro, 2021, p. 68 - 74). There have been many attempts to make a stop function but failed in both methods of using count and time (Cavallaro, 2021, p. 68 - 74). This program also froze my VM many times, so be careful when testing this tool (Cavallaro, 2021, p. 68 - 74)! See Appendix B Figure 16 through 17. It works by collecting packets of any user on the network (Cavallaro, 2021, p. 68 - 74). In the case of testing this tool on the Ubuntu VM, opening Firefox will allow the collection of packets (Cavallaro, 2021, p. 68 - 74). If any packet collected is spoofed, it is displayed to the user (Cavallaro, 2021, p. 68 - 74).

This is a reconnaissance stage sub-tool as it can be used to gather and monitor packet flow and gather IP Addresses on a target’s network (Cavallaro, 2021, p. 68 - 74). This sub-tool only works when securely connected to the target’s network, no matter if that connection is wireless or wired (Cavallaro, 2021, p. 68 - 74). An organization could use this to ensure employees are actively working and not slacking off or if there is an internal threat to the company. If these logs are copied and stored, they could be used for later evidence of the attack.

**Sub-Tool #10 ~ Anonymous FTP Scanner – FTPScan.py:**

The Anonymous FTP Scanner sub-tool determines if a target network allows for anonymous FTP login access. It “takes the hostname” given and displays a “Boolean” if the login was successful or not to the user (O’Connor, 2012, p. 57). Using the module ftplib, we can create a function that attempts to make an FTP connection using anonymous usernames and passwords (O’Connor, 2012, p. 57). Please keep in mind that data collected by this tool is used to continue to the next sub-tool. This is more of a reconnaissance stage sub-tool; however, it may be used in the scanning stage. The target used in testing for this tool was the Metasploitable 2 VM. See Appendix B Figure 18.

An organization or malicious hacker could use this to determine any vulnerabilities in the target’s anonymous FTP servers. If the login is successful, there may be many vulnerabilities to exploit; however, there may be none. An example of this would be the mass compromise, “dubbed k985ytv” (O’Connor, 2012, p. 56). “This occurred when attackers used anonymous and stolen FTP credentials to gain access to 22,400 unique domains and 536,000 infected pages (Huang, 2011)” (O’Connor, 2012, p. 56).

After the attackers gained access, they “injected JavaScript to redirect benign pages to a malicious domain in the Ukraine” (O’Connor, 2012, p. 56). When the victims of this compromise connected to the compromised server, the “Ukrainian host exploited victims in order to install a fake antivirus program that stole credit card information from the clients” (O’Connor, 2012, p. 56).

### Sub-Tool #11 ~ Brute Force FTP Credentials – FTPBF.py & userPass.txt:

The Brute Force FTP Credentials sub-tool could be used for malicious reasons, resulting in the compromise of an organization. Again, using the module ftplib, a function was created that would take a user inputted host IP Address, a password file, and display the usernames and passwords that are successful in giving access to the host and ones that are not (O'Connor, 2012, p. 58). The password file is read line by line, and the username and password are separated by using a colon ':' (O'Connor, 2012, p. 56). For this hacking tool, the password file is a flat text file. It takes the user's input data to attempt credentials to the target's server (O'Connor, 2012, p. 58). Again, please keep in mind that data collected by this tool is used to continue to the next sub-tool. See Appendix B Figure 19 through 20.

Like the last sub-tool, this is more of a reconnaissance stage sub-tool; however, it may be used in the scanning or gaining/maintaining access stages. From an organization's view, the function of anonymous FTP login allows "anonymous access," which gives an attacker "one way to enter into systems" (O'Connor, 2012, p. 57). There are many cases where malicious hackers can be "successful with using stolen credentials to gain access to legitimate FTP servers" (O'Connor, 2012, p. 57). FTP friendly programs like "FileZilla, often store passwords in plaintext configuration files (Huang, 2011)" (O'Connor, 2012, p. 57).

A way for an organization to protect itself from malicious attackers is to store them in a different location from the default and encrypt them. "Storing passwords in cleartext in a default location allows custom malware to steal credentials quickly. Security experts have found FTP stealing credentials as recent malware" (O'Connor, 2012, p. 57). To expand on this more, an example seen from "HD Moore even included the get_filezilla_creds.rb script in a recent

Metasploit release allowing users to quickly scan for FTP credentials after exploiting a target" (O'Connor, 2012, p. 57 - 58).

**Sub-Tool #12 ~ FTP Server Webpage Search (Does Not Work) – FTPWP.py:**

Once more, the FTP Server Webpage Search is more of a reconnaissance stage sub-tool; however, it may be used in the scanning or gaining/maintaining access stages. This sub-tool uses the successful username and password found in the last sub-tool to test if a target FTP server allows anonymous web access (O'Connor, 2012, p. 59). To try this, the sub-tool will login using the stolen credentials, list the data held within the FTP server's directories, search the data, and collect default web pages (O'Connor, 2012, p. 59). We created a function that uses a user's input to look at an FTP connection (O'Connor, 2012, p. 59). It then displays a list of the default web pages that it could locate. See Appendix B Figure 21.

This tool, if it worked properly, could be expanded on to create a Python script that adds malicious injections to the FTP webpages found on the targets server (O'Connor, 2012, p. 59). For a malicious hacker to do this, one could "use the Metasploit framework" to "quickly create a malicious server and page hosted" (O'Connor, 2012, p. 60). Exploits like "ms10_002_aurora" and the page hosted (http:// = "h11q://") "h11q://10.10.10.112:8080/exploit" can be used (O'Connor, 2012, p. 60). This exploit happens to be "the very same exploit used during Operation Aurora against Google" (O'Connor, 2012, p. 60). The page hosted "will exploit redirected victims, which will provide a call back to our command-and-control server" (O'Connor, 2012, p. 60). If this sub-tool were to be used in an attack, it would be used in the scanning, gaining control, or maintaining access stage. If combined with Tools #10 and #11, this would be used in the reconnaissance stage along with the others listed above.

An organization could use this sub-tool and the first two FTP sub-tools to search for exposed domain names and other vulnerabilities that may be found and correct the issues. The next step after searching for web pages on the target FTP server, a malicious hacker could perform injection attacks on the domains, which could cause the compromise of the organization. Securing open FTP ports is extremely important for an organization as it could lead to the compromise of the computer and network systems/servers and loss of data security and integrity.

## Conclusion

Cyber-attacks targeting organizations, small businesses, healthcare facilities, etc. have caused an extreme amount of damage to not only the targets, but employees, customers, patients, and even the economy. These attacks do not stop even when security evolves as the malicious black hats still are able to find exploits and vulnerabilities in the systems. This is why ethical hackers, white and grey hats, need to be implemented by all organizations. Included with this, they also need an increase of funds toward employee training on spotting cyber-attacks. The hacking tool I created could help these organizations in this endeavor however to be most effective, all of these remedies must be implemented and updated continuously. Technology never stops evolving, which means these attackers will never stop evolving alongside it.

## References

A. H. Lashkari, B. Li, T. L. Carrier and G. Kaur, "VolMemLyzer: Volatile Memory Analyzer for Malware Classification using Feature Engineering," 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), 2021, pp. 1-8, doi: 10.1109/RDAAPS48126.2021.9452028, from https://ieeexplore.ieee.org/abstract/document/9452028

Akbanov, M., Vassilakis, V. G., & Logothetis, M. D. (2019). Ransomware detection and mitigation using software-defined networking: The case of WannaCry. Computers & Electrical Engineering, 76, 111-121, from https://eprints.whiterose.ac.uk/144194/1/WannaCry_SDN_Akbanov_et_al.pdf

Argaw, S.T., Bempong, NE., Eshaya-Chauvin, B. et al. The state of research on cyberattacks against hospitals and available best practice recommendations: a scoping review. BMC Med Inform Decis Mak 19, 10 (2019), from https://doi.org/10.1186/s12911-018-0724-5

Aziz, Shallaw M., "Ransomware in High-Risk Environments" (2016). Information Technology Capstone Research Project Reports. 1, from https://scholar.valpo.edu/itcrpr/1

Babcock, J. (2016). Mastering predictive analytics with python: exploit the power of data in your business by building advanced predictive modeling applications with python (Ser. Community experience distilled). Packt Publishing. Retrieved January 25, 2022, from https://uclibraryts.on.worldcat.org/search/detail/958874809?queryString=python%20exploits&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Bemardo, D. A. G., & Miroslav, S. (2022). Sqlmap®. Retrieved February 10, 2022, from https://sqlmap.org/

Bennetts, S., Pereira, R., & Mitchell, R. (2022). *Owasp zap*. OWASP. Retrieved February 9, 2022, from https://owasp.org/www-project-zap/ & https://www.zaproxy.org/

Broersma, M. (2016). Python-Based Malware Infects European Companies. EWeek, 1, from https://uclibraryts.on.worldcat.org/search/detail/6033208475?queryString=python%20malware&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Canonical, U. (2022). *Enterprise open source and linux*. Ubuntu. Retrieved February 10, 2022, from https://ubuntu.com/

Capano, D. E. (2019). Understand the cyber-attack lifecycle. Control Engineering, 66(7), 32–33 & 66(11), 41–43, from https://uclibraryts.on.worldcat.org/search/detail/8212949150?queryString=Understand%20the%20cyber-attack%20lifecycle%3A&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Cavallaro, T. J. (2021). CYB 339 – Final Project Report – jocavall.docx [Unpublished Paper]. Cybersecurity Department, Utica University, from https://drive.google.com/drive/folders/1Obge_KUuQ4DJuSfVicKp5PRJL_2MJPO7

Cavallaro, T. J. (2021). CYB 339 – Final Project Code – jocavall. MHT Folder [Unpublished Code]. Cybersecurity Department, Utica University, from https://github.com/JT-Ca/JT-Ca & https://drive.google.com/drive/folders/1Obge_KUuQ4DJuSfVicKp5PRJL_2MJPO7

Cielen, D., Meysman, A. (2016). Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools. United States: Manning, from https://books.google.com/books?hl=en&lr=&id=bTozEAAAQBAJ&oi=fnd&pg=PT14&dq=how+can+python+benefit+a+large+organization&ots=-ilVFog81M&sig=AJuYBcNAtE8Uf4z47WhkSqNwhSY#v=onepage&q=how%20can%20python%20benefit%20a%20large%20organization&f=false

Cisomag, C. I. S. O. M. A. G. (2021, October 18). *How to build a career in ethical hacking in 2021 and Beyond*. CISO MAG | Cyber Security Magazine. Retrieved February 12, 2022, from https://cisomag.eccouncil.org/how-to-build-a-career-in-ethical-hacking-in-2021-and-beyond/

CSIS, -. (2022). *Significant cyber incidents*. Significant Cyber Incidents | Center for Strategic and International Studies. Retrieved February 17, 2022, from https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents

Cusack, G., Michel, O., & Keller, E. (2018, March). Machine learning-based detection of ransomware using SDN. In Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (pp. 1-6), from https://dl.acm.org/doi/pdf/10.1145/3180465.3180467

Duffy, C. (2016). Python: penetration testing for developers: unleash the power of python scripting to execute effective and efficient penetration tests: a course in three modules (Ser. Learning path). Packt Publishing. Retrieved January 25, 2022, from https://uclibraryts.on.worldcat.org/search/detail/962192181?queryString=organization%20hacked%20with%20the%20use%20of%20python&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Gupta, A., & Anand, A. (2017). *Ethical Hacking and Hacking Attacks "International.pdf*. *Academia* (4th ed., Vol. 6). International Journal Of Engineering And Computer Science. Retrieved February 7, 2022, from https://www.academia.edu/38177776/Ethical_Hacking_and_Hacking_Attacks_International_pdf

Holden, S. (2018, September 16). *Page*. Python Wiki. Retrieved February 12, 2022, from https://wiki.python.org/moin/

Jackson, A. (2020, July 14). *Python malware on the rise*. Cyborg Security. Retrieved February 4, 2022, from https://www.cyborgsecurity.com/cyborg_labs/python-malware-on-the-rise/

Khan, Mohd & Khan, Ihtiram Raza. (2017). Malware Detection and Analysis. 8. 1147-1149, from https://www.researchgate.net/profile/Ihtiram-Raza-Khan/publication/342491461_Malware_Detection_and_Analysis/links/5ef6d247458515505507530e2/Malware-Detection-and-Analysis.pdf

Kothari, V. (2021, May 27). *Internal working of python*. Internal working of Python. Retrieved February 12, 2022, from https://www.geeksforgeeks.org/internal-working-of-python/

Lyon, G. (2022). Nmap. Retrieved February 10, 2022, from https://nmap.org/ & https://insecure.org/fyodor/

Micro, T. (2022). *Hacking tools*. Definition of Hacking Tools. Retrieved February 8, 2022, from https://www.trendmicro.com/vinfo/us/security/definition/hacking-tools#:~:text=Hacking%20tools%20are%20programs%20that,have%20been%20designed%20to%20penetrate.

Miller, P., & Bryce, C. (2017). Python digital forensics cookbook : effective python recipes for digital investigations. Packt Publishing. Retrieved February 8, 2022, from https://uclibraryts.on.worldcat.org/search/detail/1007536293?queryString=malware%20created%20with%20python&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false.

Mohit. (2015). Python penetration testing essentials: employ the power of python to get the best out of pen-testing. Packt Publishing. Retrieved January 25, 2022, from https://uclibraryts.on.worldcat.org/search/detail/903957504?queryString=python%20hacking&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Nolan, C. (2010, July 16). Inception. Retrieved February 10, 2022, from https://www.imdb.com/title/tt1375666/

O'Connor, T. (2012). *Violent Python: A cookbook for hackers, forensic analysts, penetration testers, and security engineers*. *BryteWave*. Elsevier Science. Retrieved 2021, from https://uclibraryts.on.worldcat.org/search/detail/900825104?queryString=Violent%20Python%20A%20Cookbook%20for%20Hackers%2C%20Forensic%20Analysts%2C%20Penet ration%20Testers%20and%20Security%20Engineers%20by%20TJ%20O%27Connor&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Pauli, J. J. (2013). The basics of web hacking : tools and techniques to attack the web (Ser. The basics). Syngress, an imprint of Elsevier. Retrieved February 15, 2022, from https://uclibraryts.on.worldcat.org/search?queryString=The%20basics%20of%20web%20hacking%20%3A%20tools%20and%20techniques%20to%20attack%20the%20Web&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

Penal Law, N. Y. S. L. (2022). *New York State law*. Article 156 - NY Penal Law. Retrieved February 10, 2022, from https://ypdcrime.com/penal.law/article156.php

Pramanick, S. (2022, February 11). *History of python*. History of Python. Retrieved February 7, 2022, from https://www.geeksforgeeks.org/history-of-python/

PurpleSec, -. (2021, August 6). *2021 Cyber Security Statistics Trends & Data*. 2021 Cyber Security Statistics The Ultimate List Of Stats, Data & Trends. Retrieved February 17, 2022, from https://purplesec.us/resources/cyber-security-statistics/

Rapid7, M. 2. (2022). *Metasploitable 2 exploitability guide*. Metasploitable 2 Exploitability Guide | Metasploit Documentation. Retrieved February 10, 2022, from https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide

Ripper, J. T. (2022). *John the ripper password cracker*. Openwall - John the Ripper password cracker. Retrieved February 10, 2022, from https://www.openwall.com/john/

Roohparvar, R. (2021, May 23). People - the weakest link in Cybersecurity. People – the Weakest Link in Cybersecurity. Retrieved February 7, 2022, from https://www.infoguardsecurity.com/people-the-weakest-link-in-cybersecurity/

Samtani, S., Chinn, R., Chen, H., & Nunamaker, J. F. (2017). *Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence. EBSCOhost* (4th ed., Vol. 34), Journal of Management Information Systems. Retrieved February 8, 2022, from https://uclibraryts.on.worldcat.org/search?queryString=Exploring%20Emerging%20Hacker%20Assets%20and%20Key%20Hackers%20for%20Proactive%20Cyber%20Threat%20Intelligence&clusterResults=true&stickyFacetsChecked=true&groupVariantRecords=false

SBA, -. (2022). Stay safe from cybersecurity threats. Retrieved February 16, 2022, from https://www.sba.gov/business-guide/manage-your-business/stay-safe-cybersecurity-threats

Sengupta, S. (2021, December 17). *【5 phases of ethical hacking】* What Are The Five Steps of Ethical Hacking? Retrieved February 10, 2022, from https://crashtest-security.com/five-steps-of-ethical-hacking/

Sharma, A. (2010). An Overview of Hacking. Trinity Journal of Management, IT & Media, 1(1), 11-14, from https://acspublisher.com/journals/tjmitm/archive-issues/2010-toc/an-overview-of-hacking/

S, M., H, M., & H, J. (2022). *BURP suite - application security testing software*. PortSwigger. Retrieved February 9, 2022, from https://portswigger.net/about & https://portswigger.net/burp & https://portswigger.net/burp/pro

Software Foundation, P. (2022). *Welcome to Python.org*. Python.org. Retrieved February 12, 2022, from https://www.python.org/ & https://www.python.org/doc/

Strowes, S. D. (2013, October 28). *Passively measuring TCP round-Trip Times - Volume 11, Issue 8*. Passively Measuring TCP Round-trip Times - ACM Queue. Retrieved February 10, 2022, from https://queue.acm.org/detail.cfm?id=2539132

Tech Target, T. T. (2009, June 15). *What is ubuntu? - definition from whatis.com*. SearchDataCenter - Definition - Ubuntu. Retrieved February 10, 2022, from https://searchdatacenter.techtarget.com/definition/Ubuntu

Tech, T. (2017, November 30). *Behind the data: Investigating metadata*. Investigating Visual Media - Behind the Data: Investigating Metadata. Retrieved February 10, 2022, from https://exposingtheinvisible.org/en/guides/behind-the-data-metadata-investigations/

Tenable, N. (2022, February 14). *Nessus product family*. Tenable® - Nessus. Retrieved February 10, 2022, from https://www.tenable.com/products/nessus

TrustedSec, T. (2019, September 27). *THE SOCIAL-ENGINEER TOOLKIT (SET)*. Open-Source Tools. Retrieved February 10, 2022, from https://www.trustedsec.com/tools/the-social-engineer-toolkit-set/

Tsukerman, E. (2019). Machine Learning for Cybersecurity Cookbook: Over 80 recipes on how to implement machine learning algorithms for building security systems using Python. Packt Publishing Ltd, from https://books.google.com/books?hl=en&lr=&id=iFPADwAAQBAJ&oi=fnd&pg=PP1&dq

=Machine+Learning+for+Cybersecurity+Cookbook:+Over+80+recipes+on+how+to+implement+machine+learning+algorithms+for+building+security+systems+using+Python&ots=Aiw9KSPzQ_&sig=Ic4O1Ke96INz1iwOhgUSoquiVoc#v=onepage&q=Machine%20Learning%20for%20Cybersecurity%20Cookbook%3A%20Over%2080%20recipes%20on%20how%20to%20implement%20machine%20learning%20algorithms%20for%20building%20security%20systems%20using%20Python&f=false

VMWare, V. (2022, January 27). *What is a virtual machine?: Vmware glossary*. VMware - What is a virtual machine? Retrieved February 6, 2022, from https://www.vmware.com/topics/glossary/content/virtual-machine.html

Williams, R., Samtani, S., Patton, M., Chen, H., & 2018 IEEE International Conference on Intelligence and Security Informatics (ISI) Miami, FL, USA 2018 Nov. 9 - 2018 Nov. 11. (2018). 2018 ieee international conference on intelligence and security informatics (isi). In Incremental hacker forum exploit collection and classification for proactive cyber threat intelligence: an exploratory study (pp. 94–99). essay, IEEE, from https://doi.org/10.1109/ISI.2018.8587336 & https://www.researchgate.net/publication/329952002_Incremental_Hacker_Forum_Exploit_Collection_and_Classification_for_Proactive_Cyber_Threat_Intelligence_An_Exploratory_Study

Ying, F., & Zhang, Z. (2019). Data visualization analysis of big data recruitment positions in hangzhou based on python. Review of Computer Engineering Studies, 6(4), 81–86, from https://doi.org/10.18280/rces.060403

**Table 1.**

**Table 1. Malicious Source Code Topics in ExploitIN Forum**

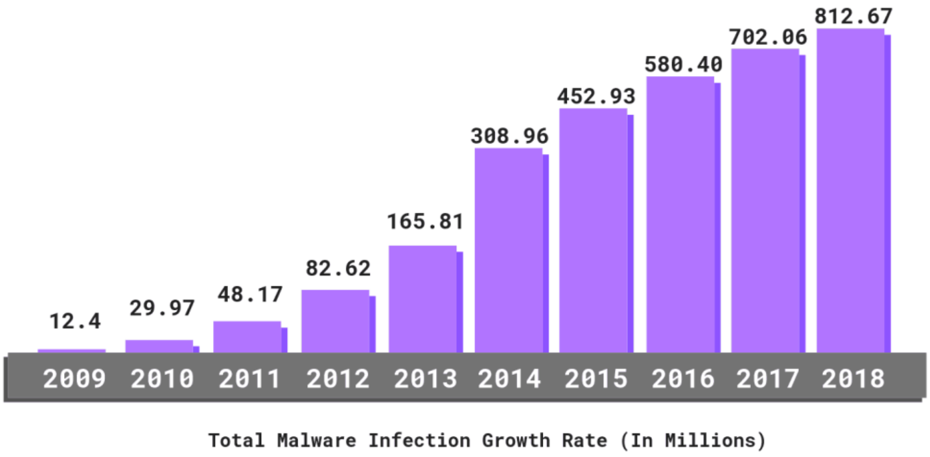
| **Source code topic label** | **The primary language of implementation** | **% of topics in ExploitIN** | **Keywords in the topic(s)** |
|---|---|---|---|
| **Shellcode Exploits** | C/C++ | 6.00 | Buffer, int, shellcode, overflow, argv, stack, xff, x40, exploit, vulnerable. |
| **SQL Injections** | SQL | 3.00 | Select, SQL, table, error, database, userse, injection, query, null, union, request |
| **Network Binders** | C/C++ | 1.50 | Bind, router, network, utility, information, interface, scanner, command, tool, cisco |

| | | | |
|---|---|---|---|
| **Password Cracking** | Python | 1.50 | Password, login, proxy, user, pass, username, get, type, log, auth, set. |
| **Spam Services** | Java | 1.50 | Spam, optimization, site, traffic, engines, website, URL, page, registration, visitors |
| **Banking Rootkits** | Java | 1.50 | Rootkit, bank, malicious, security, network, malware, banks, infected, tools |
| **Crypters** | Java | 1.50 | Crypt, decrypt, encrypt, encrypted, file, install, kriptor, keys. |
| **Total** | — | 16.50 | — |

*Note: Table located on page 18 (Originally labeled Table 6.) (Samtani, Chinn, Chen, & Nunamaker, 2017)*

**Figure 1. (Purplesec, 2021)**

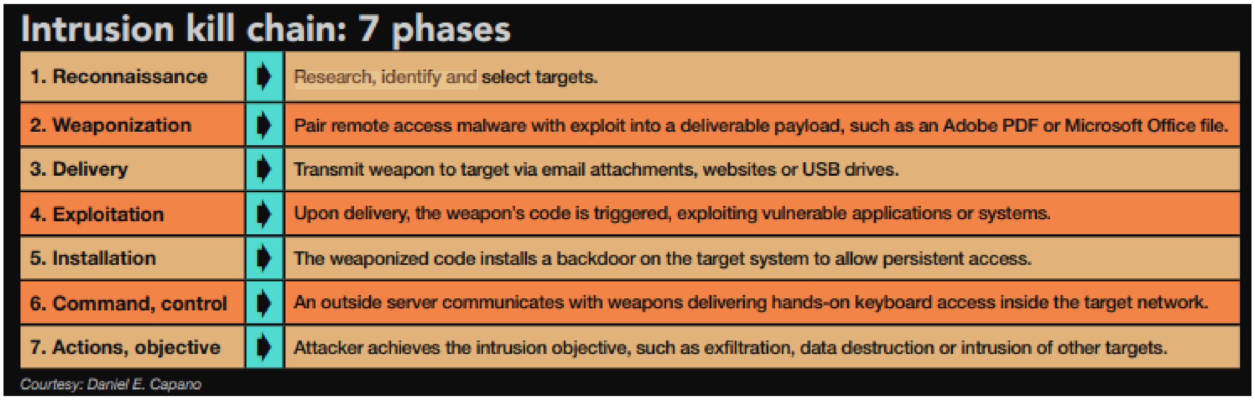
*Graph showing the increase in Malware Infections from 2009 - 2018*



*Note: This graph is the Growth Rate in Millions*

**Figure 2. (Gupta & Anand, 2017, p. 2)**

**Table of the seven stages of a Black Hat's Cyber Attack/Kill Chain**



Intrusion kill chain: 7 phases

| | |
|---|---|
| 1. Reconnaissance | Research, identify and select targets. |
| 2. Weaponization | Pair remote access malware with exploit into a deliverable payload, such as an Adobe PDF or Microsoft Office file. |
| 3. Delivery | Transmit weapon to target via email attachments, websites or USB drives. |
| 4. Exploitation | Upon delivery, the weapon's code is triggered, exploiting vulnerable applications or systems. |
| 5. Installation | The weaponized code installs a backdoor on the target system to allow persistent access. |
| 6. Command, control | An outside server communicates with weapons delivering hands-on keyboard access inside the target network. |
| 7. Actions, objective | Attacker achieves the intrusion objective, such as exfiltration, data destruction or intrusion of other targets. |

Courtesy: Daniel E. Capano

*Note: This table shows malicious attack stages.*

# Appendix A

## In-Text Citations

## In-text Citations

Offenses Involving Computers; Definition of Terms (Penal Law, 2022)

| Section | Offense | Class |
|---|---|---|
| 156.00 | Offenses involving computers; definition of terms. | |
| 156.05 | Unauthorized use of a computer. | A MISD |
| 156.10 | Computer trespass. | E FELONY |
| 156.20 | Computer tampering in the fourth degree. | A MISD |
| 156.25 | Computer tampering in the third degree. | E FELONY |
| 156.26 | Computer tampering in the second degree. | D FELONY |
| 156.27 | Computer tampering in the first degree. | C FELONY |

| Section | Offense | Class |
|---|---|---|
| 156.29 | Unlawful duplication of computer related material in the second degree. | B MISD |
| 156.30 | Unlawful duplication of computer related material in the first degree. | E FELONY |
| 156.35 | Criminal possession of computer related material. | E FELONY |
| 156.50 | Offenses involving computers; defenses. | |

S 156.00 Offenses involving computers; definition of terms.

The following definitions are applicable to this chapter except where different meanings are expressly specified:

1. "Computer" means a device or group of devices which, by manipulation of electronic, magnetic, optical or electrochemical impulses, pursuant to a computer program, can automatically perform arithmetic, logical, storage or retrieval operations with or on computer data, and includes any connected or directly related device, equipment or facility which enables such computer to store, retrieve or

communicate to or from a person, another computer or another device the results of computer operations, computer programs or computer data.

2. "**Computer program**" is property and means an ordered set of data representing coded instructions or statements that, when executed by computer, cause the computer to process data or direct the computer to perform one or more computer operations or both and may be in any form, including magnetic storage media, punched cards, or stored internally in the memory of the computer.

3. "**Computer data**" is property and means a representation of information, knowledge, facts, concepts or instructions which are being processed, or have been processed in a computer and may be in any form, including magnetic storage media, punched cards, or stored internally in the memory of the computer.

4. "**Computer service**" means any and all services provided by or through the facilities of any computer communication system allowing the input, output, examination, or transfer, of computer data or computer programs from one computer to another.

5. "**Computer material**" is property and means any computer data or

computer program which:

(a) contains records of the medical history or medical treatment of an identified or readily identifiable individual or individuals. This term shall not apply to the gaining access to or duplication solely of the medical history or medical treatment records of a person by that person or by another specifically authorized by the person whose records are gained access to or duplicated; or

(b) contains records maintained by the state or any political subdivision thereof or any governmental instrumentality within the state which contains any information concerning a person, as defined in subdivision seven of section 10.00 of this chapter, which because of name, number, symbol, mark or other identifier, can be used to identify the person and which is otherwise prohibited by law from being disclosed. This term shall not apply to the gaining access to or duplication solely of records of a person by that person or by another specifically authorized by the person whose records are gained access to or duplicated; or

(c) is not and is not intended to be available to anyone other than the person or persons rightfully in possession thereof or selected persons having access thereto with his, her or their consent and which accords or may accord such rightful possessors an advantage over competitors or other persons who do not have knowledge or the benefit thereof.

6. "**Computer network**" means the interconnection of hardwire or wireless communication lines with a computer through remote terminals, or a complex consisting of two or more interconnected computers.

7. "**Access**" means to instruct, communicate with, store data in, retrieve from, or otherwise make use of any resources of a computer, physically, directly or by electronic means.

8. "**Without authorization**" means to use or to access a computer, computer service or computer network without the permission of the owner or lessor or someone licensed or privileged by the owner or lessor where such person knew that his or her use or access was without permission or after actual notice to such person that such use or access was without permission. It shall also mean the access of a computer service by a person without permission where such person knew that such access was

without permission or after actual notice to such person, that such access was without permission.

Proof that such person used or accessed a computer, computer service or computer network through the knowing use of a set of instructions, code or computer program that bypasses, defrauds or otherwise circumvents a security measure installed or used with the user's authorization on the computer, computer service or computer network shall be presumptive evidence that such person used or accessed such computer, computer service or computer network without authorization.

9. "**Felony**" as used in this article means any felony defined in the laws of this state or any offense defined in the laws of any other jurisdiction for which a sentence to a term of imprisonment in excess of one year is authorized in this state.

S 156.05 **Unauthorized use of a computer.**

A person is guilty of unauthorized use of a computer when he or she knowingly uses, causes to be used, or accesses a computer, computer service, or computer network without authorization.

Unauthorized use of a computer is a class A misdemeanor.

S 156.10 **Computer trespass.**

A person is guilty of computer trespass when he or she knowingly uses, causes to be used, or accesses a computer, computer service, or computer network without authorization and:

1. he or she does so with an intent to commit or attempt to commit or further the commission of any felony; or

2. he or she thereby knowingly gains access to computer material.

Computer trespass is a class E felony.

S 156.20 **Computer tampering in the fourth degree.**

A person is guilty of computer tampering in the fourth degree when he or she uses, causes to be used, or accesses a computer, computer service, or computer network without authorization and he or she intentionally alters in any manner or destroys computer data or a computer program of another person.

Computer tampering in the fourth degree is a class A misdemeanor.

S 156.25 **Computer tampering in the third degree.**

A person is guilty of computer tampering in the third degree when he commits the crime of computer tampering in the fourth degree and:

1. he does so with an intent to commit or attempt to commit or further the commission of any felony; or

2. he has been previously convicted of any crime under this article or subdivision eleven of section 165.15 of this chapter; or

3. he intentionally alters in any manner or destroys computer material; or

4. he intentionally alters in any manner or destroys computer data or a computer program so as to cause damages in an aggregate amount exceeding one thousand dollars.

Computer tampering in the third degree is a class E felony.

S 156.26 **Computer tampering in the second degree.**

A person is guilty of computer tampering in the second degree when he or she commits the crime of computer tampering in the fourth degree and he or she intentionally alters in any manner or destroys:

1. computer data or a computer program so as to cause damages in an aggregate amount exceeding three thousand dollars; or

2. computer material that contains records of the medical history or medical treatment of an identified or readily identifiable individual or individuals and as a result of such alteration or destruction, such individual or individuals suffer serious physical injury, and he or she is aware of and consciously disregards a substantial and unjustifiable risk that such serious physical injury may occur.

Computer tampering in the second degree is a class D felony.

S 156.27 **Computer tampering in the first degree.**

A person is guilty of computer tampering in the first degree when he commits the crime of computer tampering in the fourth degree and he intentionally alters in any manner or destroys computer data or a computer program so as to cause damages in an aggregate amount exceeding fifty thousand dollars.

Computer tampering in the first degree is a class C felony.

S 156.29 **Unlawful duplication of computer related material in the second degree.**

A person is guilty of unlawful duplication of computer related material in the second degree when having no right to do so, he or she copies, reproduces or duplicates in any manner computer material that contains records of the medical history or medical treatment of an identified or readily identifiable individual or individuals with an intent to commit or further the commission of any crime under this chapter.

Unlawful duplication of computer related material in the second degree is a class B misdemeanor.

S 156.30 **Unlawful duplication of computer related material in the first degree.**

A person is guilty of unlawful duplication of computer related in the first degree material when having no right to do so, he or she copies, reproduces or duplicates in any manner:

1. any computer data or computer program and thereby intentionally and wrongfully deprives or appropriates from an owner thereof an economic

value or benefit in excess of two thousand five hundred dollars; or

2. any computer data or computer program with an intent to commit or attempt to commit or further the commission of any felony.

Unlawful duplication of computer related material in the first degree is a class E felony.

S 156.35 **Criminal possession of computer related material.**

A person is guilty of criminal possession of computer related material when having no right to do so, he knowingly possesses, in any form, any copy, reproduction or duplicate of any computer data or computer program which was copied, reproduced or duplicated in violation of section 156.30 of this article, with intent to benefit himself or a person other than an owner thereof.

Criminal possession of computer related material is a class E felony.

S 156.50 **Offenses involving computers; defenses.**

In any prosecution:

1. under section 156.05 or 156.10 of this article, it shall be a

defense that the defendant had reasonable grounds to believe that he had authorization to use the computer;

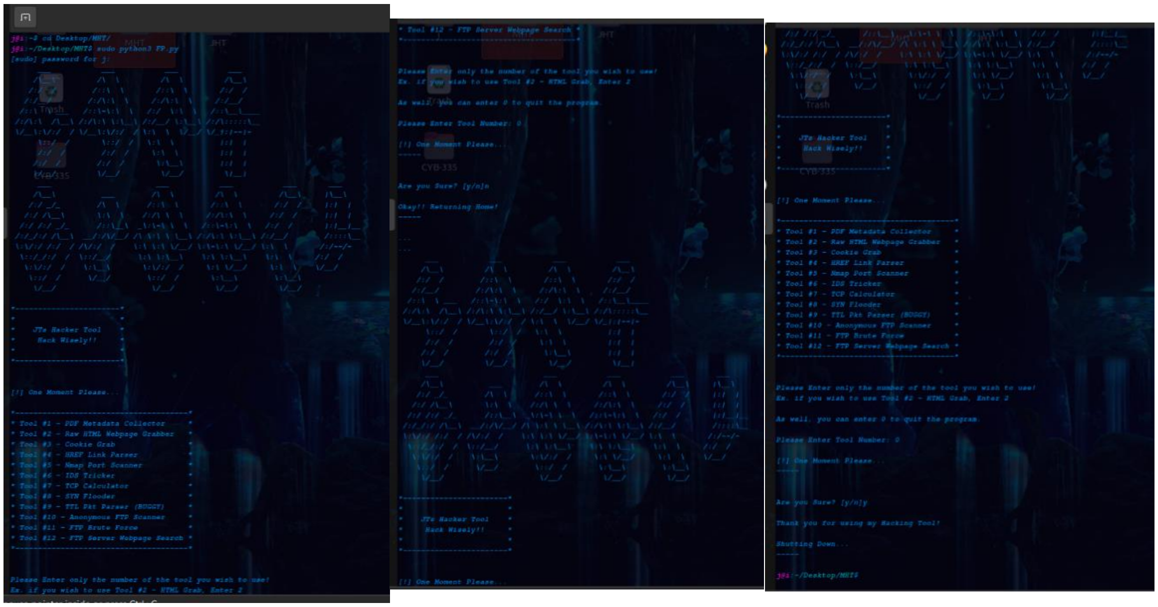
2. under section 156.20, 156.25, 156.26 or 156.27 of this article it shall be a defense that the defendant had reasonable grounds to believe that he had the right to alter in any manner or destroy the computer data or the computer program;

3. under section 156.29 or 156.30 of this article it shall be a defense that the defendant had reasonable grounds to believe that he had the right to copy, reproduce or duplicate in any manner the computer data or the computer program.

# Appendix B

## Referenced Figures

Main File – FP.py



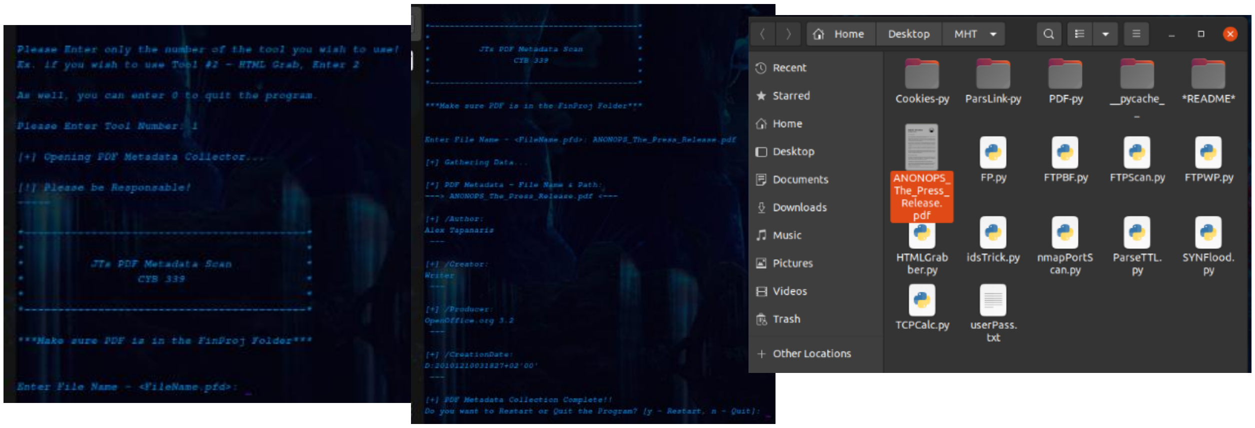
```
j@i:~$ cd Desktop/MHT/
j@i:~/Desktop/MHT$ sudo python3 FP.py
[sudo] password for j:

*-------------------------*
*                         *
*    JTs Hacker Tool      *
*     Hack Wisely!!       *
*                         *
*-------------------------*

[!] One Moment Please...

*----------------------------------------*
* Tool #1 - PDF Metadata Collector       *
* Tool #2 - Raw HTML Webpage Grabber     *
* Tool #3 - Cookie Grab                  *
* Tool #4 - HREF Link Parser             *
* Tool #5 - Nmap Port Scanner            *
* Tool #6 - IDS Tricker                  *
* Tool #7 - TCP Calculator               *
* Tool #8 - SYN Flooder                  *
* Tool #9 - TTL Pkt Parser (BUGGY)       *
* Tool #10 - Anonymous FTP Scanner       *
* Tool #11 - FTP Brute Force             *
* Tool #12 - FTP Server Webpage Search   *
*----------------------------------------*

Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2
```

```
* Tool #12 - FTP Server Webpage Search *
*--------------------------------------*

Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 0

[!] One Moment Please...
-----

Are you Sure? [y/n]n

Okay!! Returning Home!
-----

...
...

*-------------------------*
*                         *
*    JTs Hacker Tool      *
*     Hack Wisely!!       *
*                         *
*-------------------------*

[!] One Moment Please...
```

```
*-------------------------*
*                         *
*    JTs Hacker Tool      *
*     Hack Wisely!!       *
*                         *
*-------------------------*

[!] One Moment Please...

*----------------------------------------*
* Tool #1 - PDF Metadata Collector       *
* Tool #2 - Raw HTML Webpage Grabber     *
* Tool #3 - Cookie Grab                  *
* Tool #4 - HREF Link Parser             *
* Tool #5 - Nmap Port Scanner            *
* Tool #6 - IDS Tricker                  *
* Tool #7 - TCP Calculator               *
* Tool #8 - SYN Flooder                  *
* Tool #9 - TTL Pkt Parser (BUGGY)       *
* Tool #10 - Anonymous FTP Scanner       *
* Tool #11 - FTP Brute Force             *
* Tool #12 - FTP Server Webpage Search   *
*----------------------------------------*

Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 0

[!] One Moment Please...
-----

Are you Sure? [y/n]y

Thank you for using my Hacking Tool!

Shutting Down...
-----

j@i:~/Desktop/MHT$
```

*Figure 1: Shows Main File in use*

Tool #1 – PDF Metadata Collector

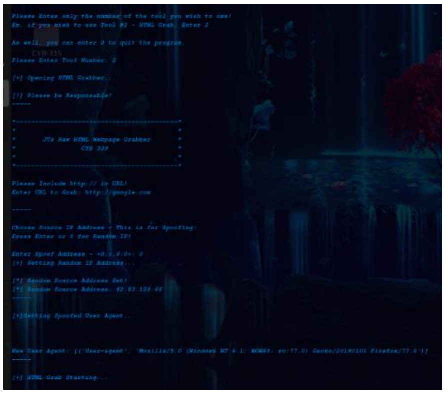


```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 1

[+] Opening PDF Metadata Collector...

[!] Please be Responsable!
-----

*------------------------------------------*
*                                          *
*          JTs PDF Metadata Scan           *
*                CYB 339                   *
*                                          *
*------------------------------------------*

***Make sure PDF is in the FinProj Folder***

Enter File Name - <FileName.pfd>:
```

```
*------------------------------------------*
*                                          *
*          JTs PDF Metadata Scan           *
*                CYB 339                   *
*                                          *
*------------------------------------------*

***Make sure PDF is in the FinProj Folder***

Enter File Name - <FileName.pfd>: ANONOPS_The_Press_Release.pdf

[+] Gathering Data...

[*] PDF Metadata - File Name & Path:
---> ANONOPS_The_Press_Release.pdf <---

[+] /Author:
Alex Tapanaris
 ---

[+] /Creator:
Writer
 ---

[+] /Producer:
OpenOffice.org 3.2
 ---

[+] /CreationDate:
D:20101210031827+02'00'
 ---

[+] PDF Metadata Collection Complete!!
Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

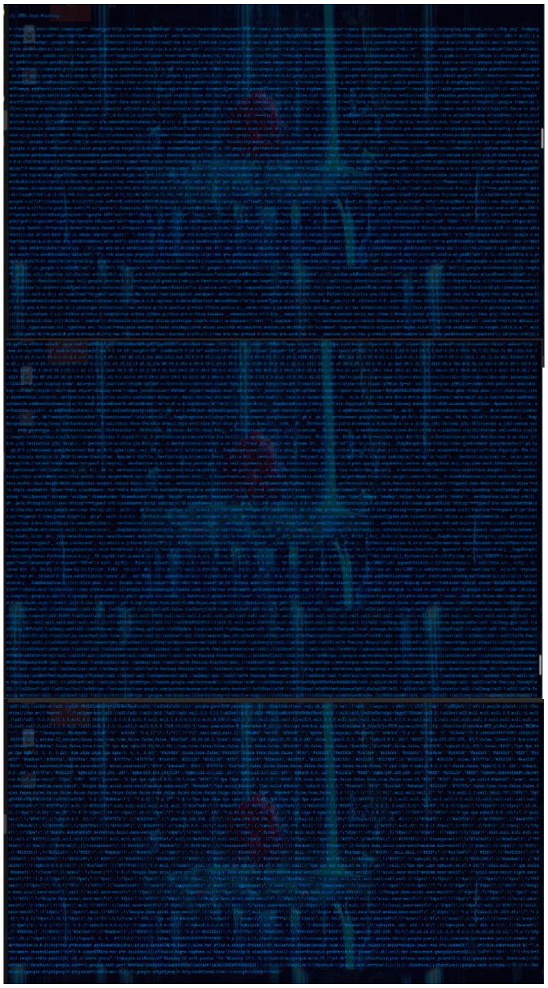
*Figure 2: Shows PDF Metadata Collector in use*

## Tool #2 – Raw HTML Webpage Grabber



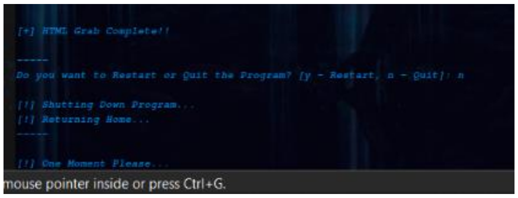
```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 2

[+] Opening HTML Grabber...

[!] Please be Responsable!
-----

*------------------------------------------*
*                                          *
*        JTs Raw HTML Webpage Grabber      *
*                 CYB 333                  *
*                                          *
*------------------------------------------*

Please Include http:// in URL!
Enter URL to Grab: http://google.com

-----

Choose Source IP Address - This is for Spoofing:
Press Enter or 0 for Random IP!

Enter Spoof Address - <0.0.0.0>: 0
[+] Setting Random IP Address...

[*] Random Source Address Set!
[*] Random Source Address: 42.83.128.46
-----

[+]Setting Spoofed User Agent...



New User Agent: [('User-agent', 'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:77.0) Gecko/20190101 Firefox/77.0')]
-----

[+] HTML Grab Starting...
```

*Figure 3: Shows Raw HTML Webpage Grabber*



```
[+] HTML Grab Starting...

[illegible]
```

*Figure 4: Shows Output of Raw HTML Webpage Grabber*

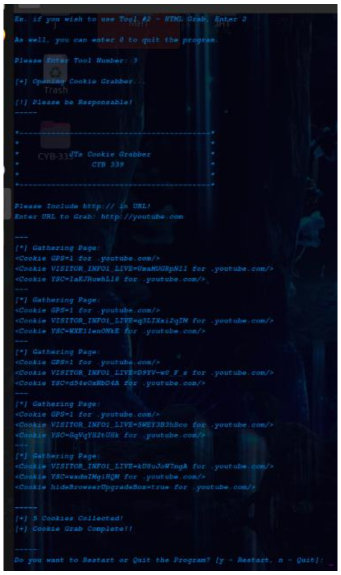


```
[+] HTML Grab Complete!!

-----
Do you want to Restart or Quit the Program? [y - Restart, n - Quit]: n

[!] Shutting Down Program...
[!] Returning Home...
-----


[!] One Moment Please...
```

mouse pointer inside or press Ctrl+G.

*Figure 5: Quitting Raw HTML Webpage Grabber*

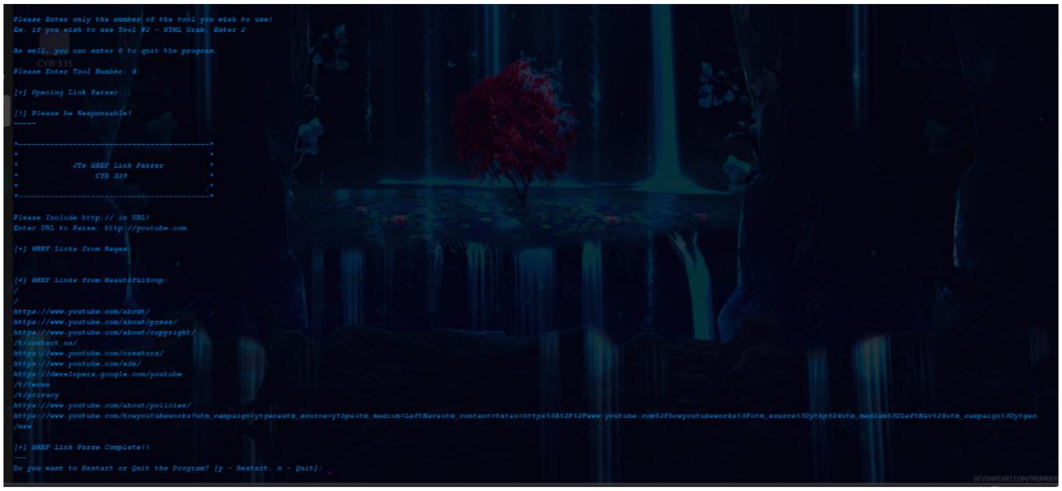
Tool #3 – Cookie Grabber



```
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 3

[+] Opening Cookie Grabber...

[!] Please be Responsable!
-----

*-------------------------------------------*
*                                           *
*            JTs Cookie Grabber             *
*                 CYB 339                   *
*                                           *
*-------------------------------------------*

Please Include http:// in URL!
Enter URL to Grab: http://youtube.com

---
[*] Gathering Page:
<Cookie GPS=1 for .youtube.com/>
<Cookie VISITOR_INFO1_LIVE=UmaMUGRpNII for .youtube.com/>
<Cookie YSC=IaKJRuwhLl8 for .youtube.com/>
---
[*] Gathering Page:
<Cookie GPS=1 for .youtube.com/>
<Cookie VISITOR_INFO1_LIVE=q3LIXxi2qIM for .youtube.com/>
<Cookie YSC=WXE11enONkE for .youtube.com/>
---
[*] Gathering Page:
<Cookie GPS=1 for .youtube.com/>
<Cookie VISITOR_INFO1_LIVE=D9YV-w0_F_s for .youtube.com/>
<Cookie YSC=d54eOxNbD4A for .youtube.com/>
---
[*] Gathering Page:
<Cookie GPS=1 for .youtube.com/>
<Cookie VISITOR_INFO1_LIVE=5WEY3B3hBco for .youtube.com/>
<Cookie YSC=GqVqYH2tUHk for .youtube.com/>
---
[*] Gathering Page:
<Cookie VISITOR_INFO1_LIVE=kU8uJoW7mgA for .youtube.com/>
<Cookie YSC=wxdmIMgiHQM for .youtube.com/>
<Cookie hideBrowserUpgradeBox=true for .youtube.com/>

-----
[+] 5 Cookies Collected!
[+] Cookie Grab Complete!!

-----
Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

*Figure 6: Shows Cookie Grabber in use*

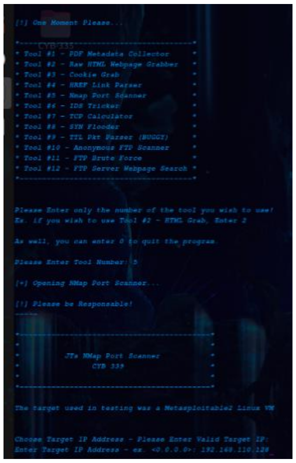
Tool #4 – HREF Link Parser



```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 4

[+] Opening Link Parser...

[!] Please be Responsable!
-----

*-------------------------------------------*
*                                           *
*            JTs HREF Link Parser           *
*                 CYB 339                   *
*                                           *
*-------------------------------------------*

Please Include http:// in URL!
Enter URL to Parse: http://youtube.com

[+] HREF Links from Regex:


[+] HREF Links from BeautifulSoup:
/
/
https://www.youtube.com/about/
https://www.youtube.com/about/press/
https://www.youtube.com/about/copyright/
/t/contact_us/
https://www.youtube.com/creators/
https://www.youtube.com/ads/
https://developers.google.com/youtube
/t/terms
/t/privacy
https://www.youtube.com/about/policies/
https://www.youtube.com/howyoutubeworks?utm_campaign=ytgen&utm_source=ythp&utm_medium=LeftNav&utm_content=txt&u=https%3A%2F%2Fwww.youtube.com%2Fhowyoutubeworks%3Futm_source%3Dythp%26utm_medium%3DLeftNav%26utm_campaign%3Dytgen
/new

[+] HREF Link Parse Complete!!
---
Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

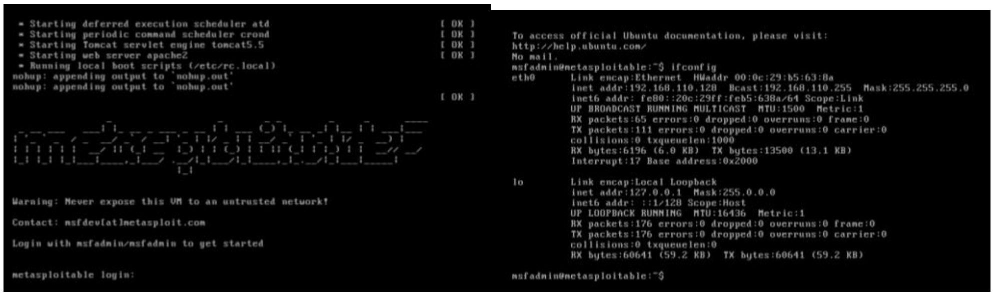
*Figure 7: Shows HREF Link Parser in use*

Tool #5 – Nmap Port Scanner



```
[!] One Moment Please...

*-----------------------------------------*
* Tool #1 - PDF Metadata Collector        *
* Tool #2 - Raw HTML Webpage Grabber      *
* Tool #3 - Cookie Grab                   *
* Tool #4 - HREF Link Parser              *
* Tool #5 - Nmap Port Scanner             *
* Tool #6 - IDS Tricker                   *
* Tool #7 - TCP Calculator                *
* Tool #8 - SYN Flooder                   *
* Tool #9 - TTL Pkt Parser (BUGGY)        *
* Tool #10 - Anonymous FTP Scanner        *
* Tool #11 - FTP Brute Force              *
* Tool #12 - FTP Server Webpage Search    *
*-----------------------------------------*


Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 5

[+] Opening NMap Port Scanner...

[!] Please be Responsable!
-----

*--------------------------------------------*
*                                            *
*           JTs NMap Port Scanner            *
*                 CYB 339                    *
*                                            *
*--------------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128
```

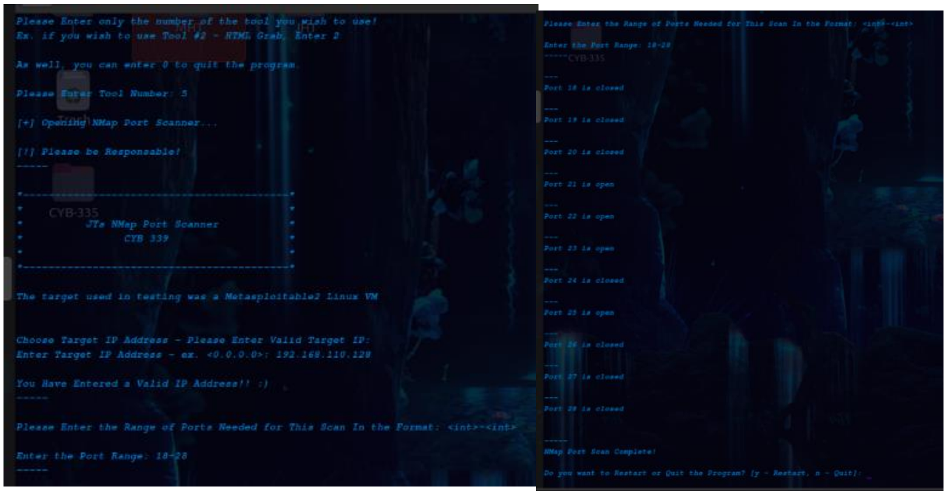
*Figure 8: Starting Nmap Port Scanner*



```
 * Starting deferred execution scheduler atd                          [ OK ]
 * Starting periodic command scheduler crond                          [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                           [ OK ]
 * Starting web server apache2                                        [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                      [ OK ]


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login:
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b5:63:8a
          inet addr:192.168.110.128  Bcast:192.168.110.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb5:638a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6196 (6.0 KB)  TX bytes:13500 (13.1 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:60641 (59.2 KB)  TX bytes:60641 (59.2 KB)

msfadmin@metasploitable:~$
```

*Figure 9: Metasploit2 Linux VM & Data Needed*

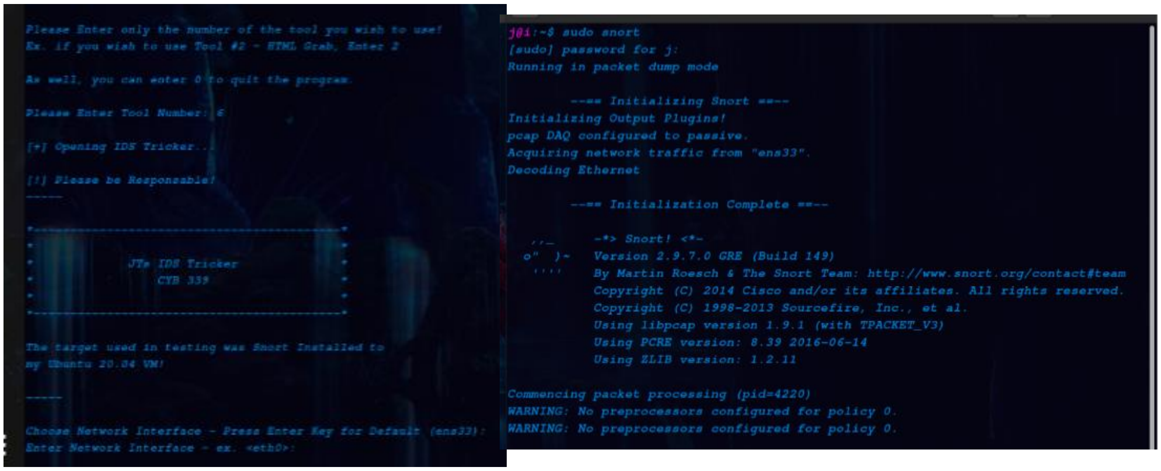


```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 5

[+] Opening NMap Port Scanner...

[!] Please be Responsable!
-----

*--------------------------------------------*
*                                            *
*           JTs NMap Port Scanner            *
*                 CYB 339                    *
*                                            *
*--------------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
-----

Please Enter the Range of Ports Needed for This Scan In the Format: <int>-<int>

Enter the Port Range: 18-28
-----
```

```
Please Enter the Range of Ports Needed for This Scan In the Format: <int>-<int>

Enter the Port Range: 18-28
-----

---
Port 18 is closed

---
Port 19 is closed

---
Port 20 is closed

---
Port 21 is open

---
Port 22 is open

---
Port 23 is open

---
Port 24 is closed

---
Port 25 is open

---
Port 26 is closed

---
Port 27 is closed

---
Port 28 is closed


------
NMap Port Scan Complete!

Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

*Figure 10: Shows Nmap Port Scanner in use*

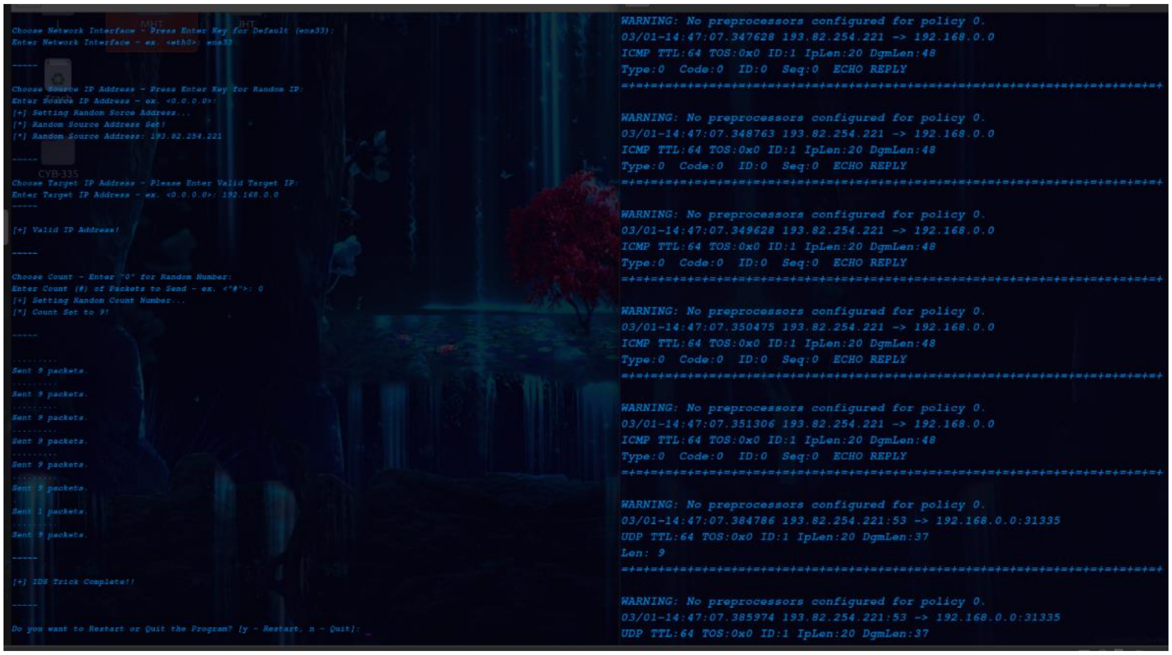
## Tool #6 – IDS Tricker



```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 6

[+] Opening IDS Tricker...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*             JTs IDS Tricker             *
*                 CYB 339                 *
*                                         *
*-----------------------------------------*

The target used in testing was Snort Installed to
my Ubuntu 20.04 VM!

-----

Choose Network Interface - Press Enter Key for Default (ens33):
Enter Network Interface - ex. <eth0>:
```

```
j@i:~$ sudo snort
[sudo] password for j:
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=4220)
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
```

*Figure 11: Shows IDS Tricker in use & Opening Target - Snort*

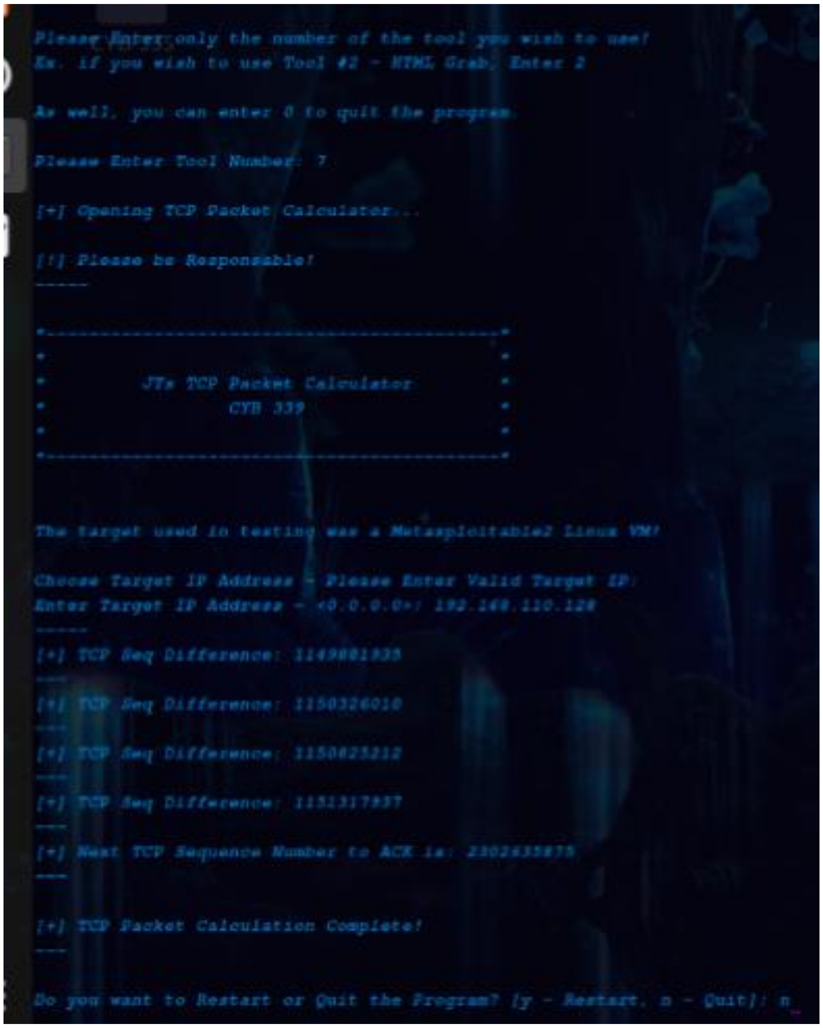


```
Choose Network Interface - Press Enter Key for Default (ens33):
Enter Network Interface - ex. <eth0>: ens33

-----

Choose Source IP Address - Press Enter Key for Random IP:
Enter Source IP Address - ex. <0.0.0.0>:
[+] Setting Random Sorce Address...
[*] Random Source Address Set!
[*] Random Source Address: 193.82.254.221

-----

Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.0.0
-----

[+] Valid IP Address!

-----

Choose Count - Enter "0" for Random Number:
Enter Count (#) of Packets to Send - ex. <"#">: 0
[+] Setting Random Count Number...
[*] Count Set to 9!

-----

.........
Sent 9 packets.
.........
Sent 9 packets.
.........
Sent 9 packets.
.........
Sent 9 packets.
.........
Sent 9 packets.
.........
Sent 9 packets.
.
Sent 1 packets.
.........
Sent 9 packets.

-----

[+] IDS Trick Complete!!

-----

Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

```
WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.347628 193.82.254.221 -> 192.168.0.0
ICMP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:48
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.348763 193.82.254.221 -> 192.168.0.0
ICMP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:48
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.349628 193.82.254.221 -> 192.168.0.0
ICMP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:48
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.350475 193.82.254.221 -> 192.168.0.0
ICMP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:48
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.351306 193.82.254.221 -> 192.168.0.0
ICMP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:48
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.384786 193.82.254.221:53 -> 192.168.0.0:31335
UDP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:37
Len: 9
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:47:07.385974 193.82.254.221:53 -> 192.168.0.0:31335
UDP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:37
```

*Figure 12: Shows IDS Tricker & Snort Output*

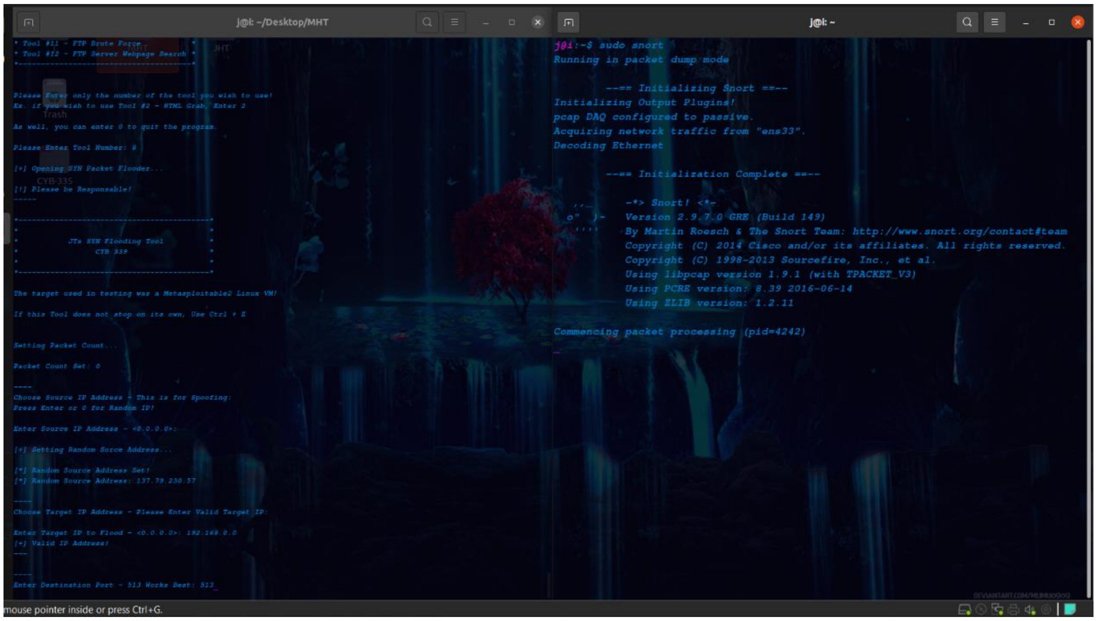
## Tool #7 – TCP Calculator



```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 7

[+] Opening TCP Packet Calculator...

[!] Please be Responsable!
------

*---------------------------------------------*
*                                             *
*          JTs TCP Packet Calculator          *
*                  CYB 339                    *
*                                             *
*---------------------------------------------*


The target used in testing was a Metasploitable2 Linux VM!

Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - <0.0.0.0>: 192.168.110.128
------
[+] TCP Seq Difference: 1149881935
----
[+] TCP Seq Difference: 1150326010
----
[+] TCP Seq Difference: 1150825212
----
[+] TCP Seq Difference: 1151317937
----
[+] Next TCP Sequence Number to ACK is: 2302635875
----

[+] TCP Packet Calculation Complete!
----

Do you want to Restart or Quit the Program? [y - Restart, n - Quit]: n
```

*Figure 13: Shows TCP Calculator in use*

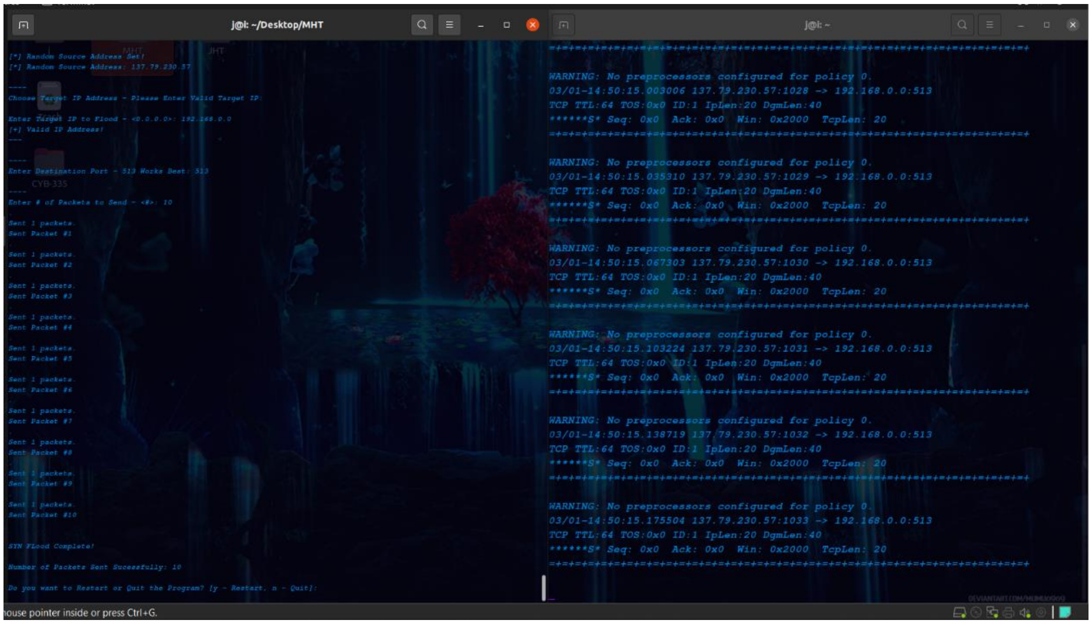
## Tool #8 – SYN Flooder



```
* Tool #11 - FTP Brute Force              *
* Tool #12 - FTP Server Webpage Search    *
*-----------------------------------------*


Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 8

[+] Opening SYN Packet Flooder...

[!] Please be Responsable!
-----

*-------------------------------------------*
*                                           *
*           JTs SYN Flooding Tool           *
*                 CYB 339                   *
*                                           *
*-------------------------------------------*

The target used in testing was a Metasploitable2 Linux VM!

If this Tool does not stop on its own, Use Ctrl + Z


Setting Packet Count...

Packet Count Set: 0

----
Choose Source IP Address - This is for Spoofing:
Press Enter or 0 for Random IP!

Enter Source IP Address - <0.0.0.0>:

[+] Setting Random Sorce Address...

[*] Random Source Address Set!
[*] Random Source Address: 137.79.230.57

----
Choose Target IP Address - Please Enter Valid Target IP:

Enter Target IP to Flood - <0.0.0.0>: 192.168.0.0
[+] Valid IP Address!
---


----
Enter Destination Port - 513 Works Best: 513
```

```
j@i:~$ sudo snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=4242)
```

*Figure 14: Shows SYN Flooder & Opening Target - Snort*

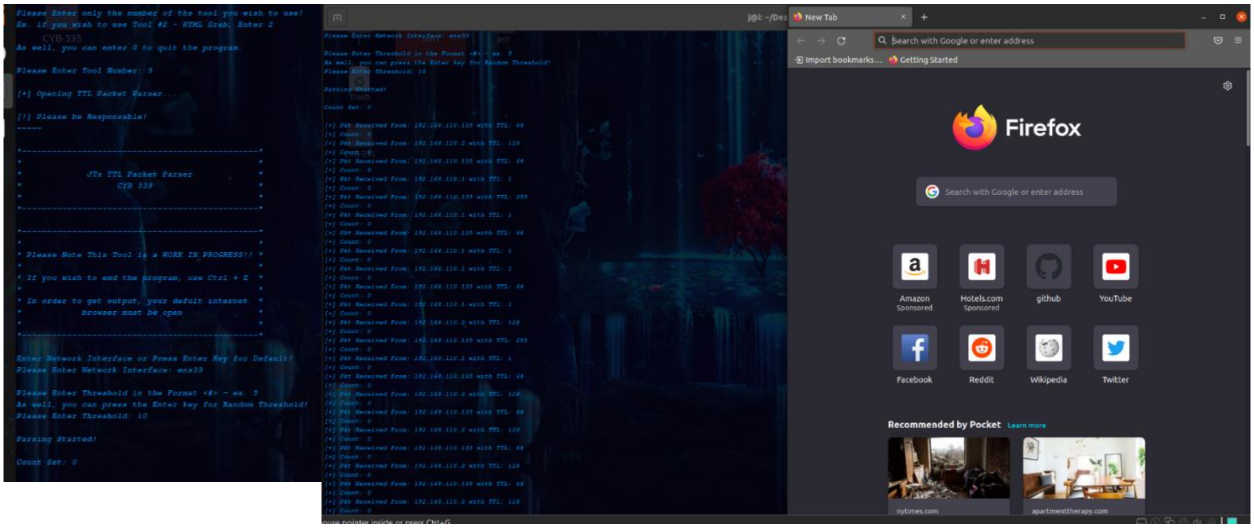


```
[*] Random Source Address Set!
[*] Random Source Address: 137.79.230.57

----
Choose Target IP Address - Please Enter Valid Target IP:

Enter Target IP to Flood - <0.0.0.0>: 192.168.0.0
[+] Valid IP Address!
---

----
Enter Destination Port - 513 Works Best: 513

----
Enter # of Packets to Send - <#>: 10
.
Sent 1 packets.
Sent Packet #1
.
Sent 1 packets.
Sent Packet #2
.
Sent 1 packets.
Sent Packet #3
.
Sent 1 packets.
Sent Packet #4
.
Sent 1 packets.
Sent Packet #5
.
Sent 1 packets.
Sent Packet #6
.
Sent 1 packets.
Sent Packet #7
.
Sent 1 packets.
Sent Packet #8
.
Sent 1 packets.
Sent Packet #9
.
Sent 1 packets.
Sent Packet #10


SYN FLood Complete!

Number of Packets Sent Sucessfully: 10

Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:50:15.003006 137.79.230.57:1028 -> 192.168.0.0:513
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0  Ack: 0x0  Win: 0x2000  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:50:15.035310 137.79.230.57:1029 -> 192.168.0.0:513
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0  Ack: 0x0  Win: 0x2000  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:50:15.067303 137.79.230.57:1030 -> 192.168.0.0:513
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0  Ack: 0x0  Win: 0x2000  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:50:15.103224 137.79.230.57:1031 -> 192.168.0.0:513
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0  Ack: 0x0  Win: 0x2000  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:50:15.138719 137.79.230.57:1032 -> 192.168.0.0:513
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0  Ack: 0x0  Win: 0x2000  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
03/01-14:50:15.175504 137.79.230.57:1033 -> 192.168.0.0:513
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0  Ack: 0x0  Win: 0x2000  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

*Figure 15: SYN Flooder & Snort Outputs*

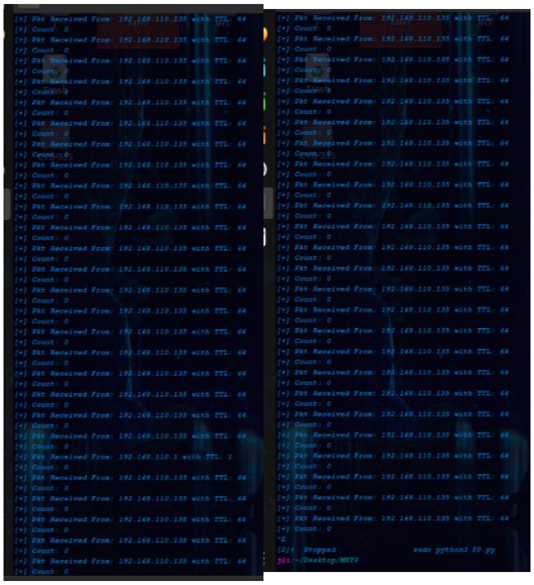
## Tool #9 – TTL Pkt Parser (BUGGY)



```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 9

[+] Opening TTL Packet Parser...

[!] Please be Responsable!
-----

*-----------------------------------------------------*
*                                                     *
*                JTs TTL Packet Parser                *
*                       CYB 339                       *
*                                                     *
*-----------------------------------------------------*

*-----------------------------------------------------*
*                                                     *
* Please Note This Tool is a WORK IN PROGRESS!!       *
*                                                     *
* If you wish to end the program, use Ctrl + Z        *
*                                                     *
* In order to get output, your defult internet        *
*           browser must be open                      *
*                                                     *
*-----------------------------------------------------*

Enter Network Interface or Press Enter Key for Default!
Please Enter Network Interface: ens33

Please Enter Threshold in the Format <#> - ex. 5
As well, you can press the Enter key for Random Threshold!
Please Enter Threshold: 10

Parsing Started!

Count Set: 0
```

```
Please Enter Network Interface: ens33

Please Enter Threshold in the Format <#> - ex. 5
As well, you can press the Enter key for Random Threshold!
Please Enter Threshold: 10

Parsing Started!

Count Set: 0

[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.2 with TTL: 128
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 255
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.2 with TTL: 128
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 255
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.2 with TTL: 128
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.2 with TTL: 128
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.2 with TTL: 128
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.2 with TTL: 128
[+] Count: 0
```

*Figure 16: Shows Opening & Output of TTL Packet Parser Before & After Opening FireFox*

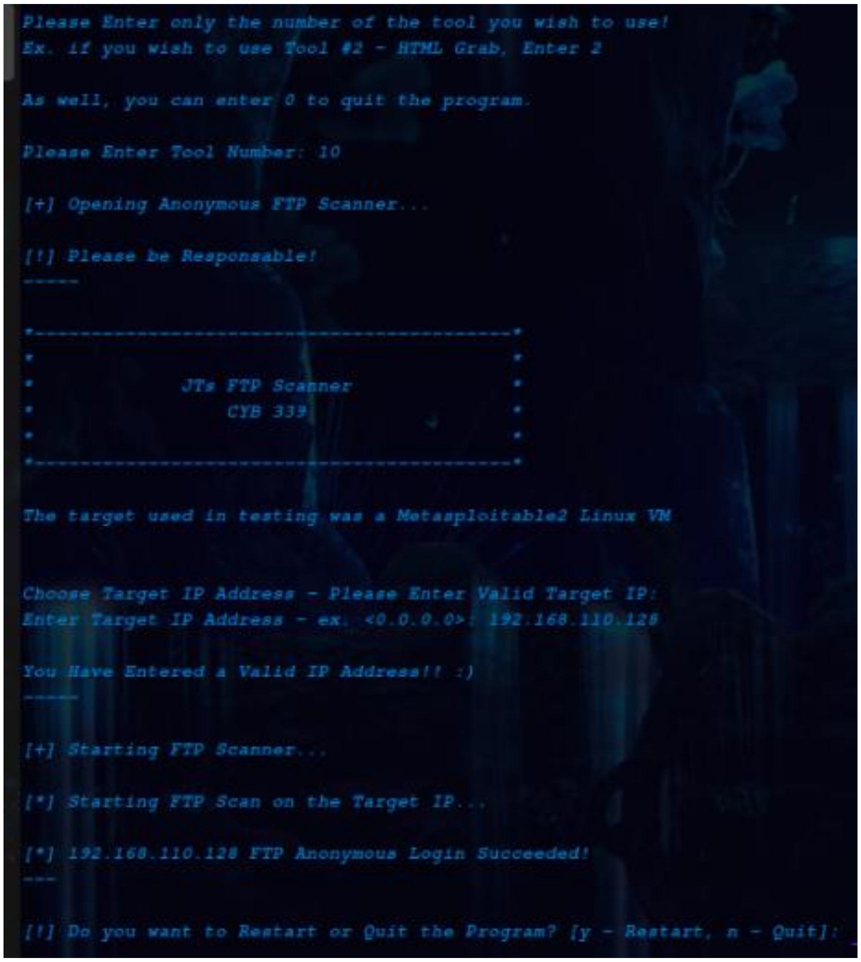


```
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.1 with TTL: 1
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
```

```
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
[+] Pkt Received From: 192.168.110.135 with TTL: 64
[+] Count: 0
^Z
[2]+  Stopped                 sudo python3 TP.py
j@i:~/Desktop/MHT$
```

*Figure 17: Shows Output of TTL Packet Parser After Opening FireFox*

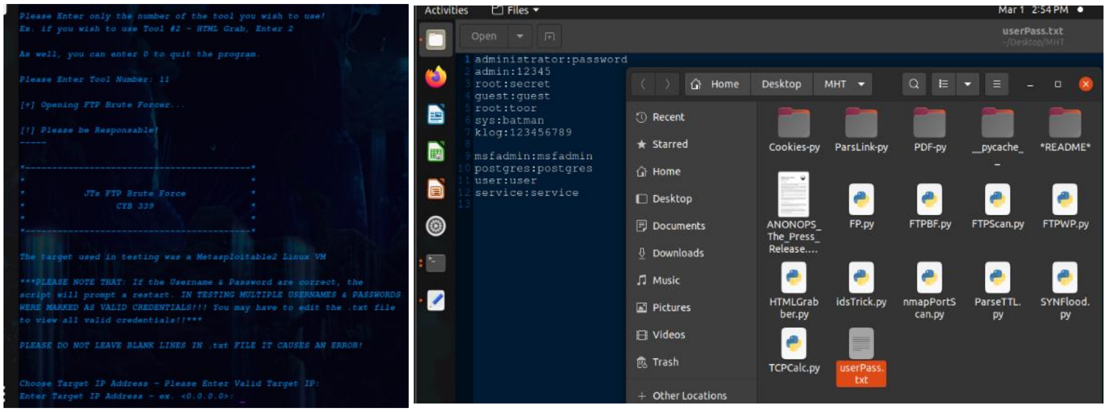
Tool #10 – Anonymous FTP Scanner



```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 10

[+] Opening Anonymous FTP Scanner...

[!] Please be Responsable!
------

*------------------------------------------*
*                                          *
*             JTs FTP Scanner              *
*                 CYB 339                  *
*                                          *
*------------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
------

[+] Starting FTP Scanner...

[*] Starting FTP Scan on the Target IP...

[*] 192.168.110.128 FTP Anonymous Login Succeeded!
---

[!] Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

*Figure 18: Shows Anonymous FTP Scanner in use*

## Tool #11 – FTP Brute Force

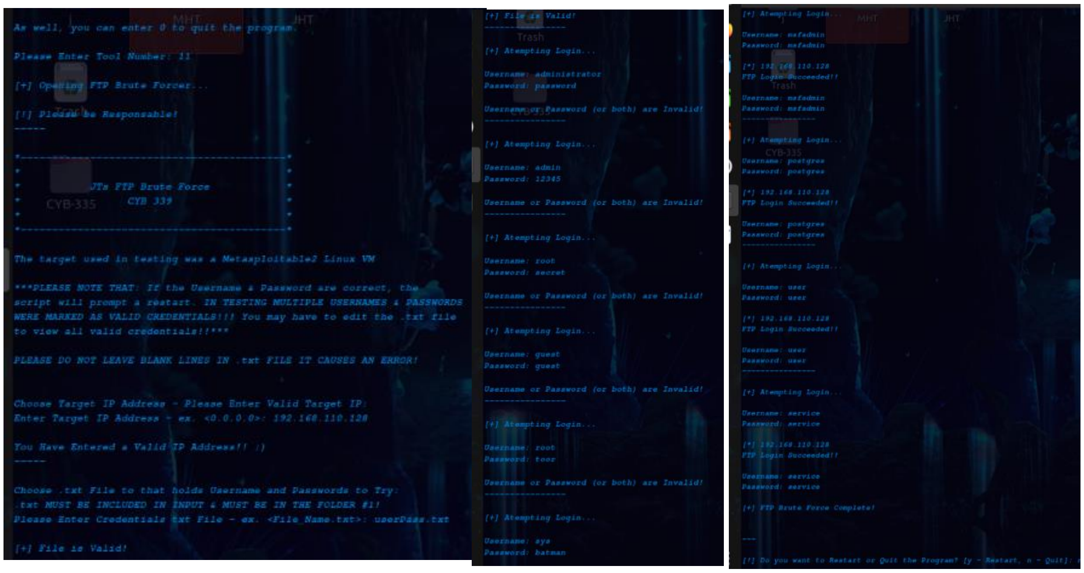


```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 11

[+] Opening FTP Brute Forcer...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*          JTs FTP Brute Force            *
*               CYB 339                   *
*                                         *
*-----------------------------------------*

The target used in testing was a Metasploitable2 Linux VM

***PLEASE NOTE THAT: If the Username & Password are correct, the
script will prompt a restart. IN TESTING MULTIPLE USERNAMES & PASSWORDS
WERE MARKED AS VALID CREDENTIALS!!! You may have to edit the .txt file
to view all valid credentials!!***

PLEASE DO NOT LEAVE BLANK LINES IN .txt FILE IT CAUSES AN ERROR!


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: _
```

userPass.txt
~/Desktop/MHT

```
administrator:password
admin:12345
root:secret
guest:guest
root:toor
sys:batman
klog:123456789

msfadmin:msfadmin
postgres:postgres
user:user
service:service
```

*Figure 19: Opening FTP Brute Force & .txt file & It's Contents*



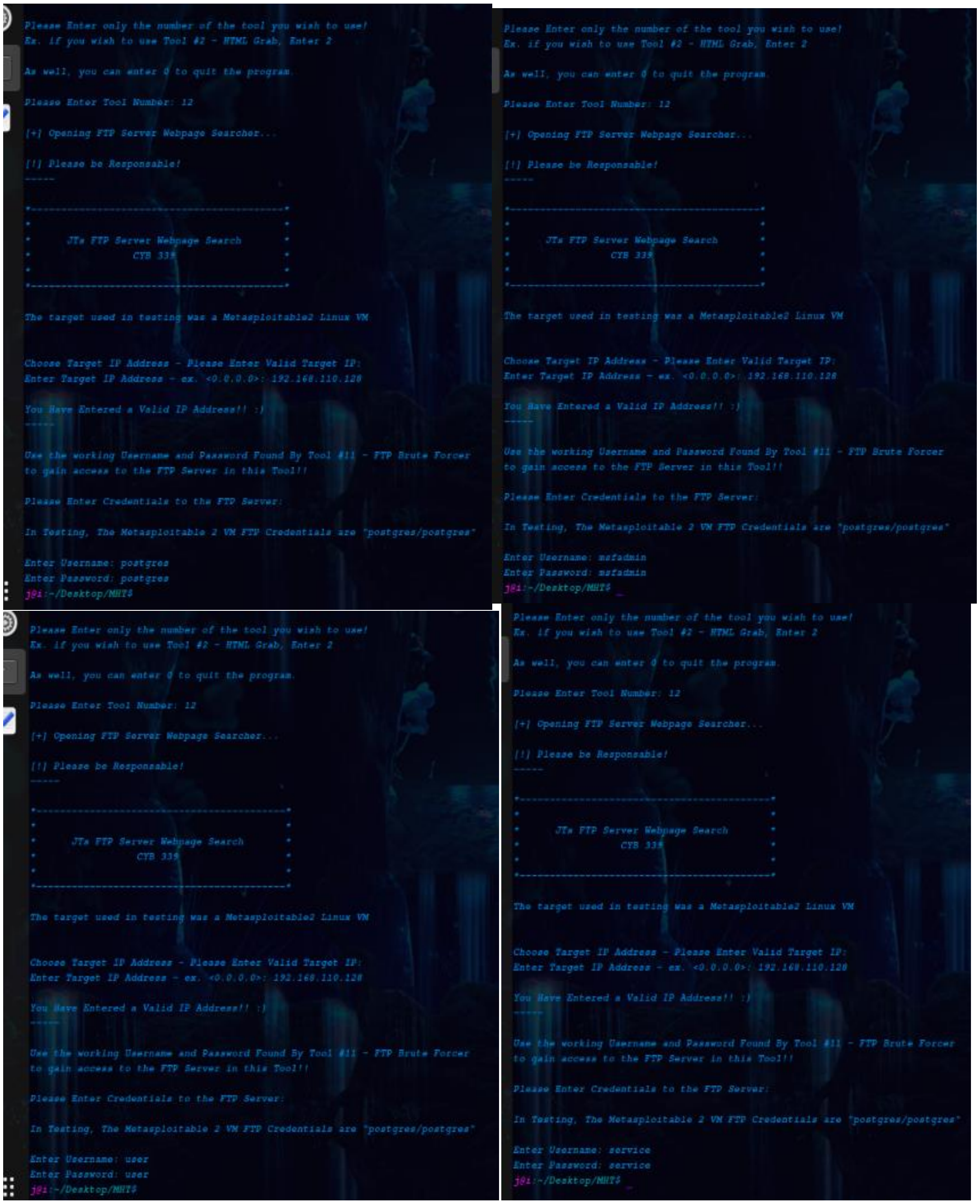
```
As well, you can enter 0 to quit the program.

Please Enter Tool Number: 11

[+] Opening FTP Brute Forcer...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*          JTs FTP Brute Force            *
*               CYB 339                   *
*                                         *
*-----------------------------------------*

The target used in testing was a Metasploitable2 Linux VM

***PLEASE NOTE THAT: If the Username & Password are correct, the
script will prompt a restart. IN TESTING MULTIPLE USERNAMES & PASSWORDS
WERE MARKED AS VALID CREDENTIALS!!! You may have to edit the .txt file
to view all valid credentials!!***

PLEASE DO NOT LEAVE BLANK LINES IN .txt FILE IT CAUSES AN ERROR!


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
-----

Choose .txt File to that holds Username and Passwords to Try:
.txt MUST BE INCLUDED IN INPUT & MUST BE IN THE FOLDER #1!
Please Enter Credentials txt File - ex. <File Name.txt>: userPass.txt

[+] File is Valid!
```

```
[+] File is Valid!
------------------

[+] Atempting Login...

Username: administrator
Password: password

Username or Password (or both) are Invalid!
------------------

[+] Atempting Login...

Username: admin
Password: 12345

Username or Password (or both) are Invalid!
------------------

[+] Atempting Login...

Username: root
Password: secret

Username or Password (or both) are Invalid!
------------------

[+] Atempting Login...

Username: guest
Password: guest

Username or Password (or both) are Invalid!
------------------

[+] Atempting Login...

Username: root
Password: toor

Username or Password (or both) are Invalid!
------------------

[+] Atempting Login...

Username: sys
Password: batman
```

```
[+] Atempting Login...

Username: msfadmin
Password: msfadmin

[*] 192.168.110.128
FTP Login Succeeded!!

Username: msfadmin
Password: msfadmin
------------------

[+] Atempting Login...

Username: postgres
Password: postgres

[*] 192.168.110.128
FTP Login Succeeded!!

Username: postgres
Password: postgres
------------------

[+] Atempting Login...

Username: user
Password: user

[*] 192.168.110.128
FTP Login Succeeded!!

Username: user
Password: user
------------------

[+] Atempting Login...

Username: service
Password: service

[*] 192.168.110.128
FTP Login Succeeded!!

Username: service
Password: service

[+] FTP Brute Force Complete!


---

[!] Do you want to Restart or Quit the Program? [y - Restart, n - Quit]:
```

*Figure 20: Shows FTP Brute Force in use*

## Tool #12 – FTP Server Webpage Search (Does Not Work)



```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 12

[+] Opening FTP Server Webpage Searcher...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*      JTs FTP Server Webpage Search      *
*                CYB 339                  *
*                                         *
*-----------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
-----

Use the working Username and Password Found By Tool #11 - FTP Brute Forcer
to gain access to the FTP Server in this Tool!!

Please Enter Credentials to the FTP Server:

In Testing, The Metasploitable 2 VM FTP Credentials are "postgres/postgres"

Enter Username: postgres
Enter Password: postgres
j@i:~/Desktop/MHT$
```

```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 12

[+] Opening FTP Server Webpage Searcher...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*      JTs FTP Server Webpage Search      *
*                CYB 339                  *
*                                         *
*-----------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
-----

Use the working Username and Password Found By Tool #11 - FTP Brute Forcer
to gain access to the FTP Server in this Tool!!

Please Enter Credentials to the FTP Server:

In Testing, The Metasploitable 2 VM FTP Credentials are "postgres/postgres"

Enter Username: msfadmin
Enter Password: msfadmin
j@i:~/Desktop/MHT$ _
```

```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 12

[+] Opening FTP Server Webpage Searcher...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*      JTs FTP Server Webpage Search      *
*                CYB 339                  *
*                                         *
*-----------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
-----

Use the working Username and Password Found By Tool #11 - FTP Brute Forcer
to gain access to the FTP Server in this Tool!!

Please Enter Credentials to the FTP Server:

In Testing, The Metasploitable 2 VM FTP Credentials are "postgres/postgres"

Enter Username: user
Enter Password: user
j@i:~/Desktop/MHT$
```

```
Please Enter only the number of the tool you wish to use!
Ex. if you wish to use Tool #2 - HTML Grab, Enter 2

As well, you can enter 0 to quit the program.

Please Enter Tool Number: 12

[+] Opening FTP Server Webpage Searcher...

[!] Please be Responsable!
-----

*-----------------------------------------*
*                                         *
*      JTs FTP Server Webpage Search      *
*                CYB 339                  *
*                                         *
*-----------------------------------------*

The target used in testing was a Metasploitable2 Linux VM


Choose Target IP Address - Please Enter Valid Target IP:
Enter Target IP Address - ex. <0.0.0.0>: 192.168.110.128

You Have Entered a Valid IP Address!! :)
-----

Use the working Username and Password Found By Tool #11 - FTP Brute Forcer
to gain access to the FTP Server in this Tool!!

Please Enter Credentials to the FTP Server:

In Testing, The Metasploitable 2 VM FTP Credentials are "postgres/postgres"

Enter Username: service
Enter Password: service
j@i:~/Desktop/MHT$ _
```

*Figure 21: Shows All Four Attempts of FTP Server Webpage Search*