EzSec Tool

Introduction

Our project, the EzSec Tool aims to bridge the gap between entry level engineers and high level, senior engineers. Due to an increased reliance on technology as the global economy adopts a more digitalized approach, the market for quality cybersecurity engineers continues to grow. However, it is becoming increasingly apparent that it is not easy to find talented individuals to fill those roles. Many industries are plagued with the increased threat of a cyber-incident due to lacking the correct professionals in their cybersecurity roles. A recent World Economic Forum report showed that hospitals, payment processors, and electricity providers are amongst some of the sectors that are most affected. The World Economic Forum provides three key areas where the cybersecurity field can improve to begin addressing the issue. First, they maintain that there is a misperception in the industry that cybersecurity is a highly technical sector that requires extensive experience in the IT or engineering field. While it may be true for many positions, many of the entry level roles in cybersecurity can be picked up on-the-job with the correct training and support. Next, the talent pool must be expanded by clearly defining roles and what they will entail to encourage candidates to apply confidently. Other initiatives to widen the pool include recruiting from other industries and backgrounds, targeting women and other underrepresented groups, and prioritizing capabilities over certifications for positions where that is possible. Lastly, WE Forum believes in retaining cybersecurity talent. While this issue is not specific to cybersecurity, it remains a critical piece of a business' success. Retaining talented employees is crucial, as they can be developed into roles that the business needs. For example, a junior security engineer can cut their teeth working through the different ranks and roles in a company before becoming a lead senior security engineer. This engineer will likely be more effective and efficient than hiring someone from outside the organization, onboarding them, and training them. Of course, there is always a case for hiring external candidates, but retaining talent remains a highly important aspect of closing the skill gap.

Our tool, EzSec, aims to address these issues within the cybersecurity sector by providing entry level engineers with a simple, one stop shop, tool for basic cybersecurity tasks like port scanning, capturing and analyzing network traffic, and other tasks. Using EzSec, these engineers can gain exposure to the tools that senior engineers are using to probe a network and perform penetration testing. EzSec can also help from a workflow perspective, as senior engineers will be able to assign basic penetration testing tasks to junior engineers to relieve some of the workload from them. We believe that EzSec addresses many of the issues that were brought up by the WE Forum report. A tool that simplifies and streamlines a seemingly complicated task of network discovery and reconnaissance can help address the misperception of cybersecurity being a highly technical field only. Moreover, our tool can help recruiters expand their talent search pool by making it more accessible to junior engineers or candidates that do not yet have work experience in the cybersecurity field. Companies can confidently provide new hires with our tool to help with their daily duties. The EzSec tool can help inexperienced engineers gain valuable experience using penetration testing tools like Nmap, Scapy, and Hashcat that they may not have otherwise gotten due to the sometimes-complicated nature of these programs. Lastly, we believe that our tool will also help address the issue of retaining talent due to engineers being able to gain experience quickly, hopefully allowing them to earn promotions and climb the ladder into more senior positions.

Throughout this report we will review our methodology for creating the EzSec tool, how we implemented the penetration testing tools, and how we designed the program to be as simple as possible for the end user to use. We will also be exploring the tool and going through a simulated run to demonstrate how it works and comparing it to running the pen-testing tools manually. Lastly, we will be wrapping up our report with our findings and concluding it with areas to improve upon as well as other features that can be added to further the usability of our tool.

Methodology

Our methodology for the EzSec tool was to take commonly used penetration testing tools and combining them in one simple GUI with many of the advanced options stripped away so new users are not overwhelmed. In our tool we implemented NMap, Hashcat, Sublist3r, and Scapy. These tools were chosen because they are powerful discovery and reconnaissance tools which are two critical steps during a penetration test.

NMap is arguably the most important tool during a penetration test as it allows the attacker to discover hosts on a network, which ports are open, and what services are running on those ports. NMap works by probing hosts on a network and analyzing the responses. It provides critical information to the user and will usually be one of the first steps in a penetration test. NMap has many different options, including scanning specific ports, scanning speed, scanning all ports, operating system detection, and various output options. The options summary for NMap can be overwhelming for users new to NMap. To combat this, we have chosen a general set of options that will fit the majority of network discovery and scanning needs. While removing the options can possibly lower the effectiveness of the NMap scan, we believe that the NMap scan performed by EzSec will cover all of the bases that a junior engineer will need when performing a scan. Using our tool, an engineer can perform a NMap scan on a single endpoint or on an entire network range. The results are sorted and stored in a text file so they can be later analyzed or passed onto a senior engineer so they can perform the next steps.

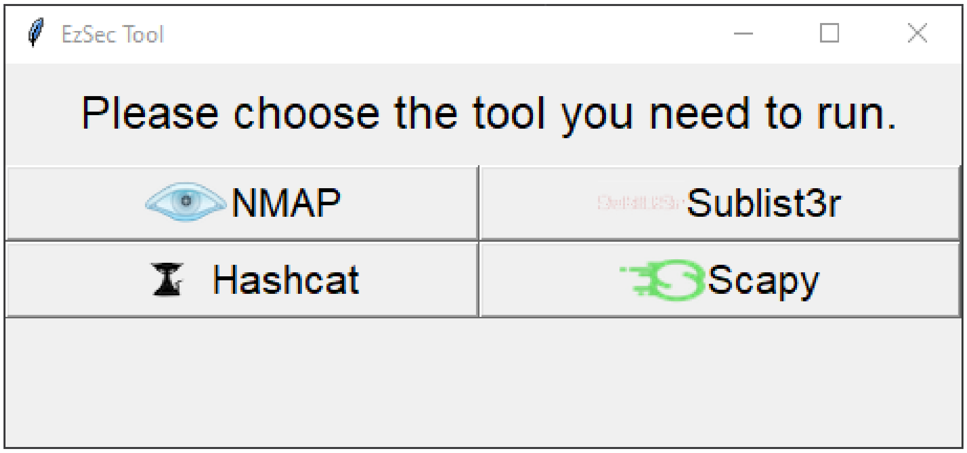
Our next tool is Hashcat. Hashcat is an advanced tool that is used to crack hashes, frequently password hashes. Hashcat can be used to perform brute force and dictionary attacks to break a hash and recover what was initially hashed. Finding hashes during a penetration test is not uncommon and being able to “reverse” the hashing is incredibly powerful, especially if it uncovers a password that can lead to further penetration of the network. Being an advanced tool, Hashcat can be complicated to set up and run successfully. We have simplified the command line into a simple graphical user interface and stripped out uncommon hashes from the options, leaving only hashes that are commonly used. Our thinking is that if it does not fit the criteria listed in our program, a senior engineer will likely have to analyze further to decipher the hash and find the original message. In our program, the Hashcat function will take in a dictionary file as well as the hash, and compute until a match is found or the dictionary is exhausted. The results can then be passed along to senior engineers so they can make a determination on what the next steps will be.

Next, we will look at the Sublist3r integration. Sublist3r is another reconnaissance tool that allows engineers to perform subdomain enumeration. Sublist3r uses a variety of techniques to find subdomains of websites to help find possible vulnerabilities or attack vectors during a penetration test. For example, a website may have the cpanel login page exposed to the internet, allowing attackers to focus their efforts and perform various different attacks against it, like brute force, SQL injection, or password spraying. Websites may sometimes have sensitive information exposed to the internet without the owner knowing, possibly because they are buried behind obscure subdomains or pages that are not frequented. We decided to include Sublist3r to allow junior engineers to also have a tool to use for web reconnaissance and discovery, as it can also be a very important part of a penetration test. Sublist3r will take in a domain and begin scanning for any possible subdomains. Once the scan is completed, the output will be generated and written to a file for review. The report may be provided to a senior engineer who can then make a decision on the next steps in the penetration test.

Our last tool in the suite is Scapy. Scapy is a Python library that allows network package manipulation, forging, capture, and analysis. Scapy is a very powerful tool that can be used in a variety of different ways and will almost certainly be used during a penetration test. In line with the rest of the programs we have integrated into EzSec, we have focused on the capture and analysis portion of Scapy. Our integration will allow for analysis of previously captured PCAP files. Network capture can be an especially useful tool in network discovery and reconnaissance as it can lead to possible new attack vectors or more information on the network as a whole. Packet capture and analysis can sometimes be time consuming, and we believe it is a perfect task for a junior engineer to perform to allow them to gain knowledge of network scanning, learning how a network behaves, and what penetration testers are looking for in a scan. In turn, this will free up senior engineers to perform other, more high-level tasks or to write their final findings report to present to their client.

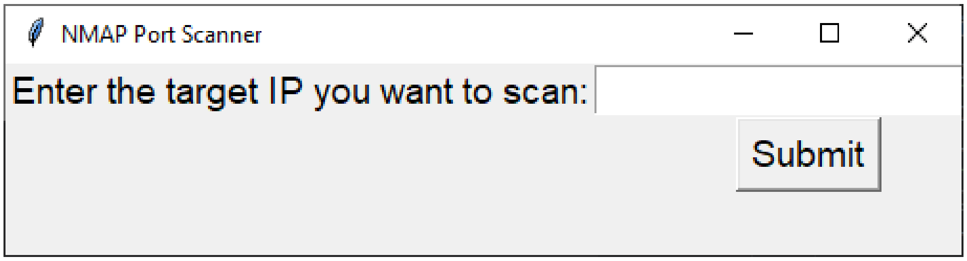
Tech Demo

The EzSec tool has a very basic GUI to allow new engineers to run the tools any confusion. When running the program, the user will be prompted to choose between the four tools provided.



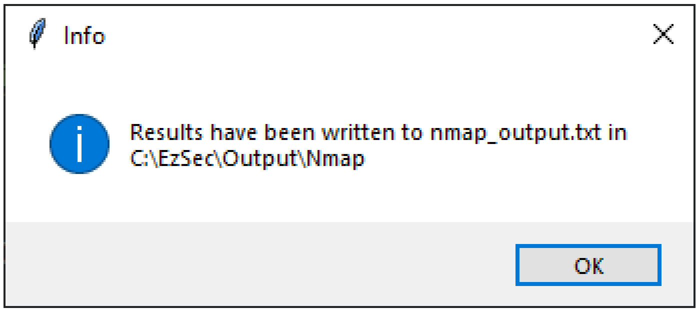
Please choose the tool you need to run.

Upon choosing an option the technician will then be prompted for some user input to get the tool started. Below please see each tool in succession beginning with NMAP. After selecting the NMAP button the technician will be prompted with the below screen:



Enter the target IP you want to scan:

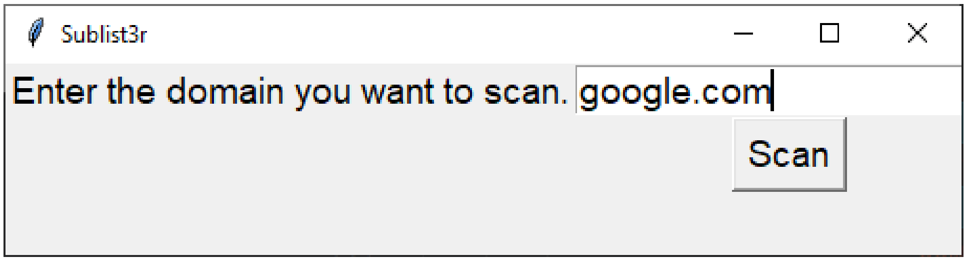
This screen can take user input in the form of a standalone IP address or a CIDR notation network range. After clicking the submit button it will direct the output to the text file stored in the folder: C:\EzSec\Output\Nmap. This will also prompt an information box notifying the user of the location the output is saved:



Results have been written to nmap_output.txt in C:\EzSec\Output\Nmap

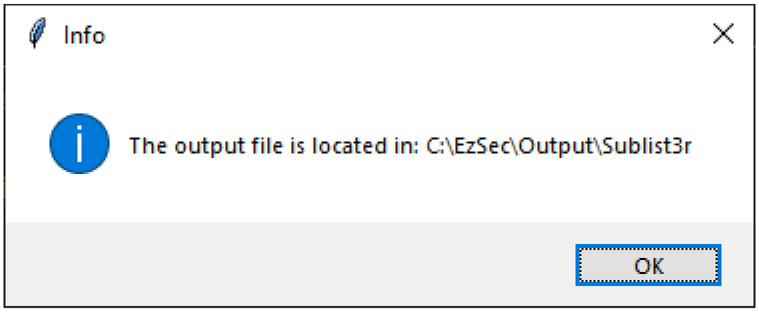
The results are displayed in a text file and can be reviewed or provided to a senior engineer.

Next is the Sublist3r tool. When clicking on the button the user will be prompted with another window to input the domain name to scan:



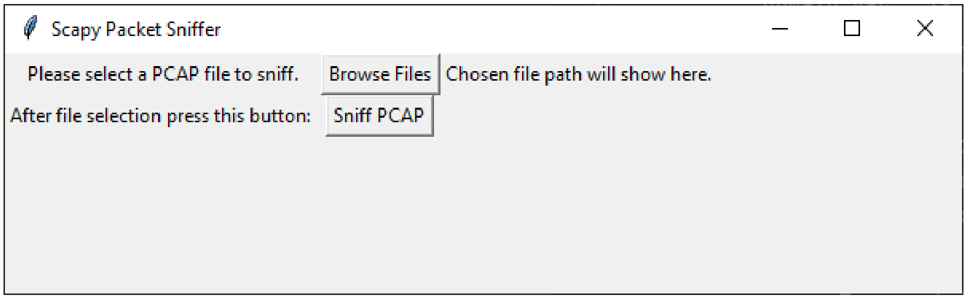
Enter the domain you want to scan. google.com

All the flags are handled within the code. The technician is only required to input the domain name. When this scan button is run the EzSec tool will use sublist3r to check the subdomains for the domain provided. Once the scan is complete the user will be prompted with a message stating where the scan is saved.



The output file is located in: C:\EzSec\Output\Sublist3r

The final tool in the stack is the Scapy tool. Scapy allows the technician to upload a .pcap file and the tool will iterate through the file, saving the results to a text file.

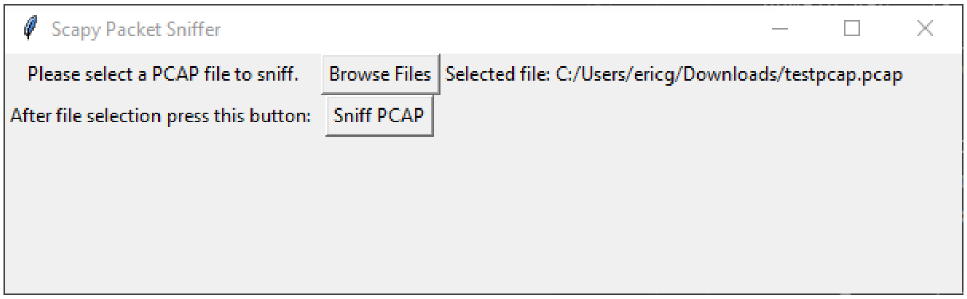


Please select a PCAP file to sniff.

Chosen file path will show here.

After file selection press this button:

After selecting the file, the chosen file path will be displayed:

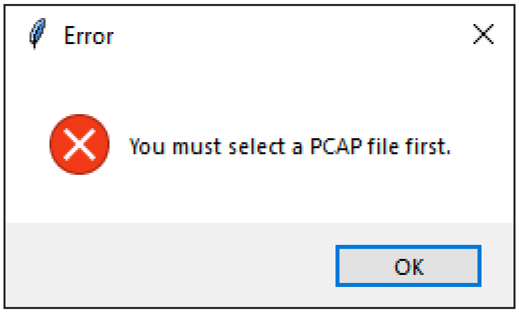


Please select a PCAP file to sniff.

Selected file: C:/Users/ericg/Downloads/testpcap.pcap

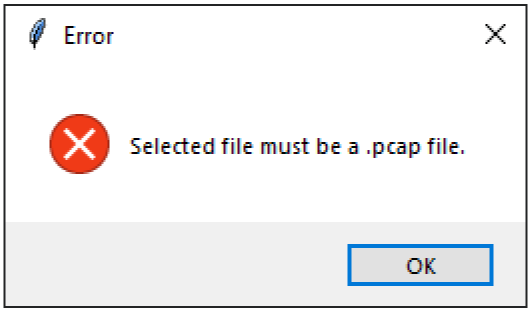
After file selection press this button:

There are also two forms of error checking. The first displays if the user tries to sniff a pcap file before choosing a file.



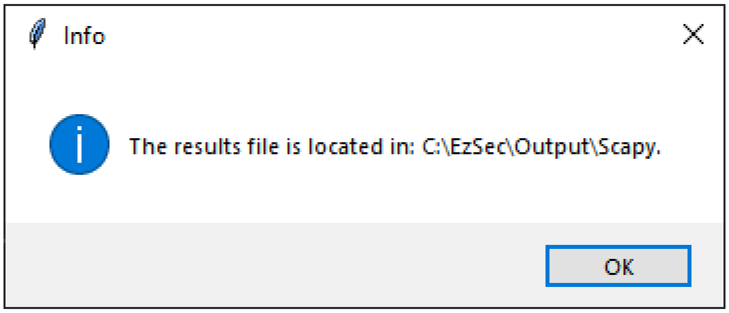
You must select a PCAP file first.

The second error displays if the technician chooses a file that is not .pcap. After clicking on the "Sniff Pcap” button the error will be shown.



Selected file must be a .pcap file.

The results of this will also be exported to a text file. An info popup will display showing where the output is stored:



The results file is located in: C:\EzSec\Output\Scapy.

As you can see, the output is stored in the same path but a different folder for each tool. If these folders do not exist the EzSec tool will create them.

Areas to Improve

There are a few areas where we would like to improve the EzSec tool. One of these is adding live scanning to the Scapy implementation. At the moment, the tool only allows the user to open and review PCAP files that were already previously recorded. However, Scapy has the ability to do live scanning of the network with some user input. This would be beneficial to add because then technicians can scan what is needed immediately without having to first create a PCAP file using another network analyzer, like WireShark. Another area for improvement would be implementing the tool Metasploit. We had originally planned to implement Metasploit, however, to use the Python integration Metasploit you need an RPC server that is running Metasploit. The Python integration would then communicate with it to retrieve the information. Given that we were not able to set up a Metasploit server, we were not able to get this set up. However, this is a tool that is used extensively in penetration testing and would be a great addition to the EzSec suite. The NMap tool can also be improved. While we have full functionality of NMap, there is currently no input validation to confirm if the IP address provided by the user is valid. We were able to get input validation working, however, it then caused issues with CIDR notation and hostname entries. We decided to remove the IP address input validation to instead keep CIDR scanning capabilities. NMap will give an error and not run if the entry is not valid, but there is no user facing error message currently.

In general, we believe the EzSec tool can be made to be more robust. The best way to improve the tool is to build upon the tools that are included and implement more of the features each tool has. This will allow the tool to be more comprehensive and give junior engineers the chance to explore the integrated tools more fully gaining a better understanding of how these tools are used in the cybersecurity field. Given the extra capabilities of the included tools, the EzSec Tool will be able to be used by more experienced users and junior engineers alike. The EzSec tool is built in Python and was done to the best of our ability. However, cleaning up code and making more efficient code is always something that we strive for. The program runs and responds well, however there are likely improvements that could be made to the code to make it faster and easier to understand. Lastly, the EzSec user interface is very simple and intuitive, but we believe it can be spruced up to be more visually appealing as well as possibly more useful.

References

Joshi, A., Doyle, S., & Perucica, N. (2023, May 2). *Here's how to address the global cybersecurity skills gap*. World Economic Forum. https://www.weforum.org/agenda/2023/05/the-cybersecurity-skills-gap-is-a-real-threat-heres-how-to-address-it/

Security Staff. (2023, September 5). *71% of organizations are impacted by cybersecurity skills shortage | Security Magazine*. Www.securitymagazine.com. https://www.securitymagazine.com/articles/99865-71-of-organizations-are-impacted-by-cybersecurity-skills-shortage

nmap Project. (N.D.). Nmap – Free Security Scanner for Network Exploration & Security Audits. https://nmap.org/book/man.html

Hashcat. (N.D.). Hashcat - Advanced Password Recovery. https://hashcat.net/hashcat/

Ahmed Aboul-Ela. (2020, April 6). Sublist3r. https://github.com/aboul3la/Sublist3r

Scapy Project. (N.D.). Scapy (Version 2). https://scapy.net/