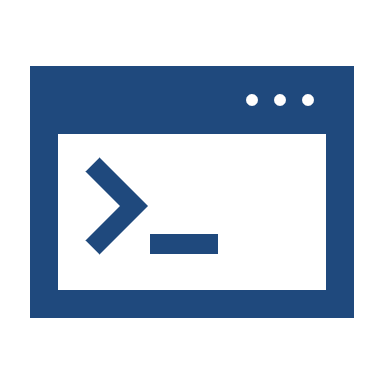


Scan\_and\_Grab

**https://github.com/98cstillm/Scan\_and\_Grab**

**Christal Stillman**



# Project Goals & Problem Solving

|  |  |
| --- | --- |
|  |  |
| Automation | **Goal:**  Create a tool that can provide cybersecurity professionals with the ability to automate & improve efficiency.  **Why?**  There are many vulnerabilities that can threaten a network, therefore it is important that administrators and cybersecurity professionals alike are able to intervene and remediate issues in real time. This is where automated tools like Scan\_and\_Grab come into play to improve the time it takes to perform audits by performing automated port scans on targeted machines. |
| Security | **Goal:**  Create a tool that can assist with providing confidentiality, integrity, and accessibility on a node or network.  **Why?**  While port scanning is not a great form of security on its own, it is important to run one like Scan\_and\_Grab that can identify what ports are open on a node. This collection of data plays a part in making sure that only the appropriate ports are open so as not to risk the unauthorized release of confidential information which could risk the integrity of data as well as how available that data is to authorized users. |
| Identification | **Goal:**  Create a tool that can help identify potential vulnerabilities and actions on a node or network.  **Why?**  Identifying where vulnerabilities stem from is the first step to managing security on a node, and when it comes to ports they can open unexpected doors for attackers and hackers alike. But utilizing a port scanner like Scan\_and\_Grab administrators are able to identify ports that invite unauthorized users and close them if need be. |

# \*Project Plan & Timeline Below\*

# Project Plan & Timeline

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Research** | Closed Book | **Build** | Monitor | **Test** | Test Dummy |  |

**Due Date (all components):** October 25, 2020

|  |  |  |
| --- | --- | --- |
|  | Due Date | Objective Description |
| **□** | Week 1  10/3/20 | Research common network security needs within the cybersecurity infrastructure & the ways they can be alleviated by security automation tools. |
| **□** | Week 1  10/3/20 | Pick a tool that will provide automation, security, identification, and documentation to users, (Port scanner) |
| **□** | Week 1  10/3/20 | Begin research on the tool and identify common features and results that are created by that tool & think of ways to capture that in mine. |
| **□** | Week 2  10/10/20 | Research to gain a better understanding of how socket connections work between the host and the target when port scanning. |
| **□** | Week 2  10/10/20 | Research methods used to create the actual scanning portion of a port scanner. |
| **□** | Week 2  10/10/20 | Consider what environments, software, and python modules will be necessary for a port scanner. Decide what system will be my target. |
| **□** | Week 3  10/17/20 | Create script for socket connection using the socket module & necessary functions. Be sure to carefully add comments (#) to describe each part of code). |
| **□** | Week 3  10/17/20 | Create opportunities in script to receive input from user regarding desired target IP Address and port range. (Add a for and except loop to handle invalid formats possibly?) Be sure to carefully add comments (#) to describe each part of code). |
| **□** | Week 3  10/17/20 | Use VM Ware Workstation to create a Metasploitable 2 machine that will serve as the demonstrative target for this tool. Metasploitable also has a list of known open ports, so this will act as a control to know that the scanner is identifying the correct ports. |
| **□** | Week 4  10/23/20 | Begin testing the port scanner’s ability to connect to the target. Trouble shoot if needed and document every change. |
| **□** | Week 4  10/23/20 | Create GitHub repository & README file to house script information, project plan & dependencies. |
| **□** | Week 4  10/23/20 | Film demonstrative video (use screen cast) & compile all parts to turn in! |