Project Requirements Document

Team Name: MAN^2

Project Sponsor: NAVAIR-PIPELINES

Project Name: NOMADS

Members:

Nick Arenberg - narenberg@ucsb.edu (Team Lead)
Nick Mattair - nmattair@ucsb.edu (Scribe)
Andy Ho - andyho@ucsb.edu
Matthew Chen - matthewchen@ucsb.edu
Max Medearis - m_medearis@ucsb.edu

Background

The Problem

Cybersecurity is more important than ever. With the frequency of cyberattacks increasing, it is necessary to ensure that networks that are thought to be secure have no leakage. Even if network transmissions are secure, attacks could be siphoning vital data through faulty software or hardware on an endpoint. However, manually gathering information on every endpoint device on a network is time-consuming and expensive. It wastes valuable security resources. Plus, if a security audit misses a threat, it would either exist for adversaries to exploit or require another expensive audit to root out. In short, human security teams are too slow and unreliable to perform detailed security audits of large networks. Human teams are valuable for analysis and response, so using them for that purpose is highly inefficient. Some automated solutions exist for detecting some of these vulnerabilities. However, existing solutions like Red Seal don't quite meet the specifications required by the Navy.

Why is this important?

The need for a low-cost, quick, and accurate security audit application is clear. In this increasingly online and interconnected world, and with the rise of the Internet of Things, being able to spot and shut down potential security threats is vital. Especially when it comes to matters of security and national defense, like those that would be handled on Navy networks. Instead of dedicating valuable manpower to manually check every endpoint, a software that could automate the process would not only increase security, but allow for the allocation of resources to other, more pressing areas.

Existing Solutions

Red Seal

https://www.redseal.net/#home

Endpoint Detection and Response (EDR) Platforms

https://www.cynet.com/endpoint-protection-and-edr/top-6-edr-tools-compared/
https://www.crowdstrike.com/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/

Palo Alto Networks Cortex XDR

https://www.paloaltonetworks.com/cortex/endpoint-protection

Each of the above options does its own version of endpoint detection and response. Generally, that includes constant monitoring of endpoint devices, alert triage, incident response capabilities, and data analytics to identify threats.

Project Goals

Outcome

With this project, we will create an application that continuously monitors a network to determine the hardware and software installed on each endpoint. This data will be checked against a ledger of known software to locate any irregularities or issues. It will be presented to users in an easily consumed manner. Users will be able to see a network graph and view reports from potential threats with detailed information on each specific endpoint and their flagged activities.

Assumptions

This application will only be run on Windows 10 and Linux machines. It will only be run by system administrators who have a set of credentials that will allow the program to remotely access the machines.

Use Cases/User Stories

1. As a user, I can run a script so that I am able to connect to the different types of devices connected to the network and gather information on them as long as I have credentials for them.
    - Scenario 1: User runs the script with a correct set of credentials.
        - The user will get the option to store the data that was collected after running the script.
    - Scenario 2: User runs the script with an incorrect set of credentials.
        - The user will get an error message.
        - The user will be prompted to try a different set of credentials.

2. As a user, I can download the data dump files so that I can store a log of data from each time I run the main script.
    Github issue: https://github.com/xavierholt/nomads/issues/17
    - Scenario 1: The user downloads the information on all the devices hit upon running the script.
        - After running the script, the user gets prompted to see if they want to store the data from the run that was just made.
        - If yes, then the program logs the data into a file for the user to view later.
    - Scenario 2: The user downloads the information on all the devices that possess possible network insecurities upon running the script.
        - If any insecurities are found after running the script, the program prompts the user to see if they want to have a separate file storing data on the vulnerable systems.
        - If yes, then the program logs the data on the vulnerable systems in a separate log file.

3. As a user, I can run a script so that I can get an inventory of all of the programs installed on a Linux host that I run the script on/have the credentials for.
    Github issue: https://github.com/xavierholt/nomads/issues/11
    - Scenario 1: The user runs the script on a host
        - The user will get the option to store the data that was collected after running the script.
    - Scenario 2: The user runs the script on a network
        - If the user has correct credentials, the script will collect the relevant data, and the user will get the option to store the data
        - If the user provides an incorrect set of credentials:
            - The user will get an error message.
            - The user will be prompted to try a different set of credentials.

4. As a user, I can run a script on a network so that I can receive a Visio diagram summarizing the relevant data about said network.
   Github issue:
   - Scenario 1: The user runs the script on a network, and a visio file is outputted which provides a visual representation of the network structure, along with the hosts on the network
   - Scenario 2: The user runs the script on a network, but lacks credentials for some/all of the hosts. A visio file is generated which leaves out devices it couldn't access.

5. As a user, I can interact with a desktop GUI so that I can generate a visio diagram.
   Github issue: Basic GUI that can run C# visio generation script
   - Scenario 1: The user opens the GUI, has the correct credentials, and is able to generate a visio diagram.
   - Scenario 2: The user opens the GUI, does not have the correct credentials, and is not able to run any of the desired features.

6. As a user, I can interact with a desktop GUI so that I can get an inventory of all the programs installed, and access different devices on the network.
   Github issue: Basic GUI that can run ansible scripts
   - Scenario 1: The user opens the GUI, has the correct credentials, and is able to get an inventory of the network.
   - Scenario 2: The user opens the GUI, does not have the correct credentials, and is not able to run any of the desired features.

7. As a user, I can generate a report so that I can get a summary of .
   Github issue: Report Generator that can visualize data
   - Scenario 1: The user opens the GUI, runs a network scan, and is able to get a pdf report with a summary of all of the data found.
   - Scenario 2: The user opens the GUI and selects previously generated json data with a previously generated Visio diagram and is able to get a pdf report with a summary of the data included.

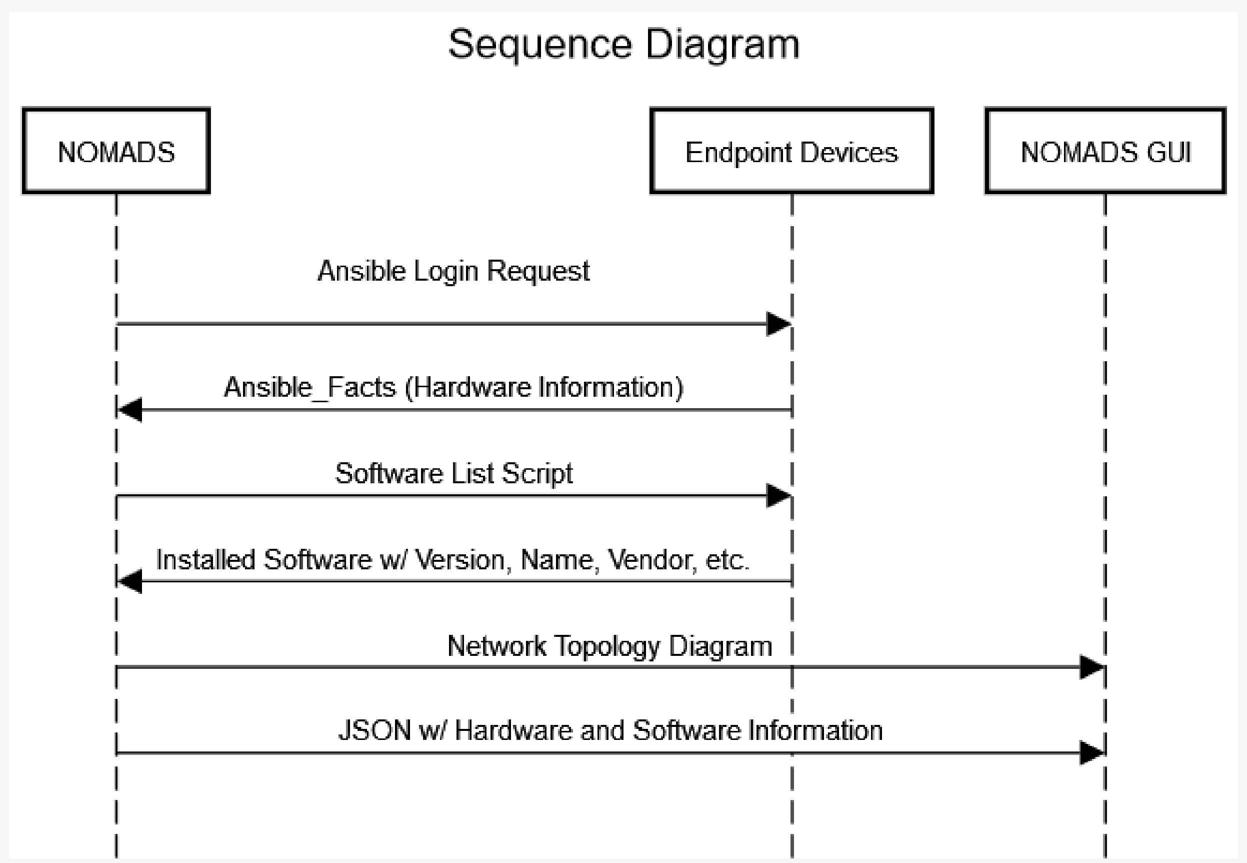
8. As a user, I can run a Powershell script so that I can get an inventory of all of the programs installed on a Windows host that I run the script on/have the credentials for.
   Github issue: https://github.com/xavierholt/nomads/issues/12
   - Scenario 1: The user runs the script on a host
     - The user will get the option to store the data that was collected after running the script.
   - Scenario 2: The user runs the script on a network

- If the user has correct credentials, the script will collect the relevant data, and the user will get the option to store the data
- If the user provides an incorrect set of credentials:
    - The user will get an error message.
    - The user will be prompted to try a different set of credentials.

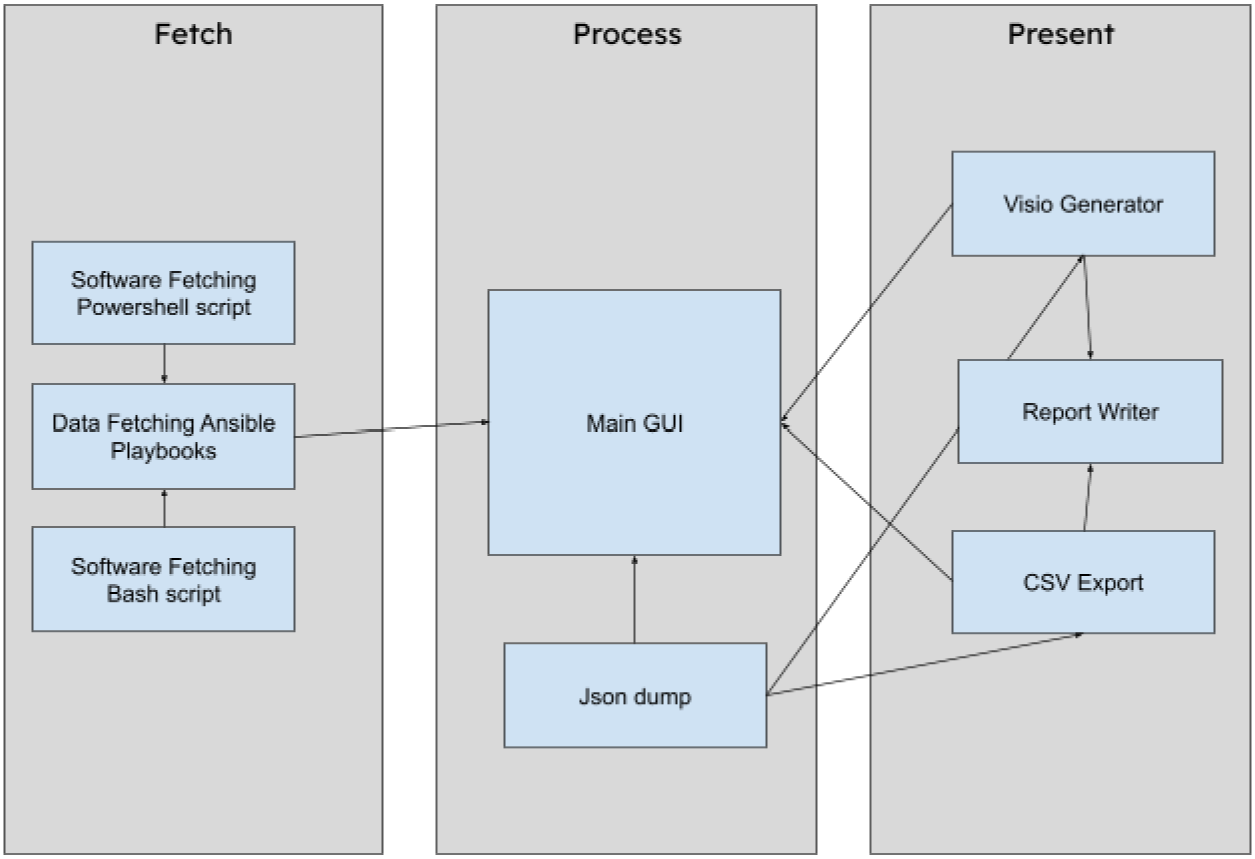
9. As a user without scripting skills, I can interact with a GUI to view the system information so that I can easily assess potential threats.
    - Scenario 1: The user wants to know if any software on their endpoints is out of date.
        i. The user will click a button that generates a log file that shows out of date software.
    - Scenario 2: A manager without programming skills wants to see a network topology diagram.
        i. The manager opens the GUI, presses a button and views the outputted Visio file.

10. As a user, I can transfer the system information to my peers so that anyone who needs the information can easily see it.
    Github issue:
    - Scenario 1: A user’s boss wants to see a log file.
        - The app will generate log files that can be emailed or passed via usb stick by the user.
    - Scenario 2: A user wants to show a network topology diagram at a meeting.
        - The app will generate a Visio file that the user can present at their meeting.

Sequence Diagram


Sequence Diagram
NOMADS
Endpoint Devices
NOMADS GUI
Ansible Login Request
Ansible_Facts (Hardware Information)
Software List Script
Installed Software w/ Version, Name, Vendor, etc.
Network Topology Diagram
JSON w/ Hardware and Software Information

System Architecture Diagram


Fetch
Process
Present
Software Fetching Powershell script
Data Fetching Ansible Playbooks
Software Fetching Bash script
Main GUI
Json dump
Visio Generator
Report Writer
CSV Export

Appendix

A. Implementation Technologies
    a. Ansible/ansible-runner
    b. Bash Scripting
    c. Python
    d. PyQt
    e. Microsoft Visio