

# Project Requirements Document

**Team Name:** MAN^2

**Project Sponsor:** NAVAIR-PIPELINES

**Project Name:** NOMADS

## Members:

Nick Arenberg - [narenberg@ucsb.edu](mailto:narenberg@ucsb.edu) (Team Lead)

Nick Mattair - [nmattair@ucsb.edu](mailto:nmattair@ucsb.edu) (Scribe)

Andy Ho - [andyho@ucsb.edu](mailto:andyho@ucsb.edu)

Matthew Chen - [matthewchen@ucsb.edu](mailto:matthewchen@ucsb.edu)

Max Medearis - [m\\_medearis@ucsb.edu](mailto: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. None of these solutions cover all of the needs that NOMADS does since NAVAIR requires network visualization into a specific format, JSON dumps of data, and other specific requirements.

## Project Goals

---

### Outcome

With this project, we will create an application that audits a network to determine the hardware and software installed on each endpoint and generate an interactive network graph. This data will be checked against a ledger of known software to locate any irregularities or issues. It will be presented to users in both the aforementioned interactive graph and a static format. Users will be able to deep dive into the network graph, looking at individual machines, and view reports from potential threats with detailed information on each specific endpoint and their flagged software or hardware installation(s).

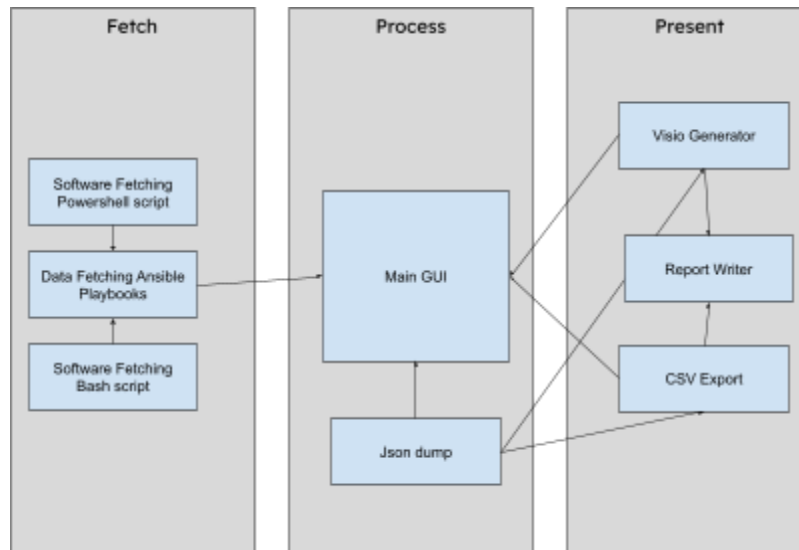
### Assumptions

---

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.

## System Architecture Diagram

---



## User Interaction and Design

---

Any user of NOMADS will first see a GUI that allows them to upload a list of credentials for specific hosts if they wish to. If they choose not to, then hosts that cannot be accessed without credentials will not be able to be searched. Once that is done, the user will be able to click a button to scan the network that the current host is connected to, and be presented with:

1. A button to download a csv detailing the hosts on the network and software on those hosts.
2. A button to download a visio file that visually represents the csv data as a network diagram.
3. An option to display the csv as a table in the GUI.
4. An option to visualize the data as a network diagram in the GUI.

## 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 that gathers a list of possible hosts from the arp cache of the current computer.
  - Scenario 1: The user finds no new hosts that they do already know about
    - The user is given a list of hosts generated from the arp cache.
  - Scenario 2: The user finds more hosts that they did not know about before
    - The user is notified that there are more possible hosts and they are given a list of all the potential hosts found
4. As a user, I can run a script that determines the type of host an unknown host is.

- Scenario 1: The user runs the script on a Windows host
    - When the user receives the report, it will include that the host is a Windows system
  
  - Scenario 2: The user runs the script on a Linux host
    - When the user receives the report, it will include that the host is a Linux system
5. As a user, I can run a script that gathers a list of possible hosts from active connections of the current computer.
- Scenario 1: The user finds no new hosts that they do already know about
    - The user is given a list of hosts generated from the netstat output
  - Scenario 2: The user finds more possible hosts that they did not know about before
    - The user is notified that there are more possible hosts via active connections and they are given a list of all the potential hosts found through the netstat output
6. As a user, I can use an invasive scan to try and find other potential hosts through arp scan.
- Scenario 1: The user attempts to run a script that collects a list of hosts via an arp scan and proceeds knowing the risk.
    - The user is warned that this method could notify system administrators about the potential suspicious activity.
    - After acknowledging this, the script will make an attempt to generate a list of hosts based on the arp scan output
  - Scenario 2: The user attempts to run a script that collects a list of hosts via an arp scan and does not proceed knowing the risk.
    - The user is prompted that the script did not initiate the scan and returns the user back to the main gui.
7. 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.
8. 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: [Basic GUI that can run C# visio generation script](#)
- 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.
9. 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.
10. 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.
11. As a user, I can generate a report so that I can get a summary of the network info.
- 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.

12. 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.

13. 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.
  - 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.
  - The manager opens the GUI, presses a button and views the outputted Visio file.

14. 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.

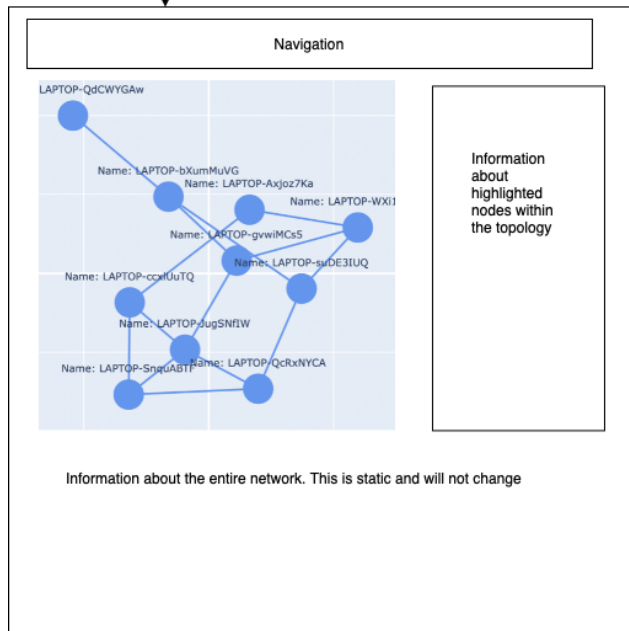
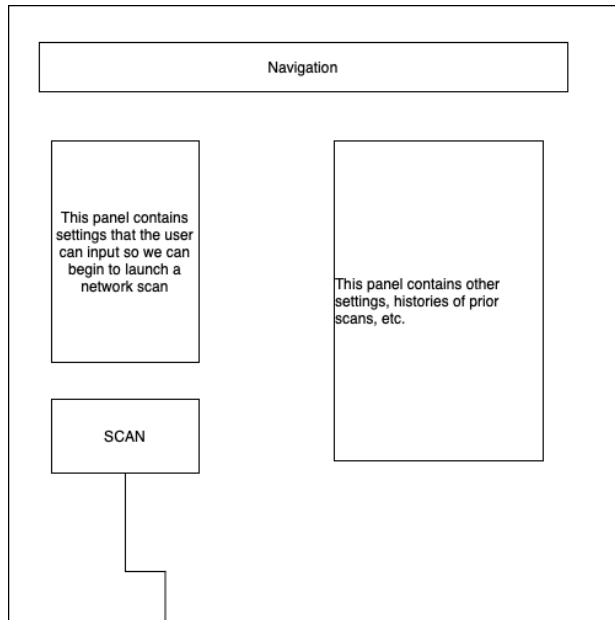
15. As a user, I can upload a list of approved software, and receive a report detailing which hosts on a network have software installed that is not on the list.

- Scenario 1: There is no unapproved software on a host
  - The script will not generate an unapproved software file
- Scenario 2: There is unapproved software on a host

- The script will generate a report listing the software and the host it is installed on
16. As a user, I can upload a list of approved software, and receive a report detailing which hosts on a network have approved software installed that is out of date/not the latest version
- Scenario 1: There is no out of date software on a host
    - The script will not generate an out of date software file
  - Scenario 2: There is out of date software on a host
    - The script will generate a report listing the out of date software and the host it is installed on
17. As a user, I can save the information generated by the app to my computer so I can access it later.
- Scenario: I want to email the software list to a colleague.
    - I follow the directions in the app to save the json dump, then email it to my colleague.
18. As a user, I can convert the generated network graph to a .vsdx file to be opened in Microsoft Visio so that it will not require a change in systems.
- Scenario: I want to send the network graph to my superior but he only has Microsoft Visio on his computer to view diagrams.
    - The app has an option to convert the graph, so I click the button and send the file.
19. As a user, I can access previously generated json dumps for view in GUI without technical knowledge
- Scenario: I want to see a graph that I generated last week.
    - The app has an option open a json file, and will, by default, allow the user to select json files from within the default json dump folder
20. As a user, I can generate a spreadsheet of the data discovered by the scripts to be opened in Microsoft Excel
- Scenario: I want to view the hardware and software of devices on a network in a spreadsheet for data analysis
    - The app has an option to export data from the json dumps to csv format



# UI Mock Diagram



The first diagram shows the home page, where the user can augment the settings related to the network scan. The second diagram shows the page the appears as a result of a scan, where the user can view the network topology, other information, and download visio document representing this topology.

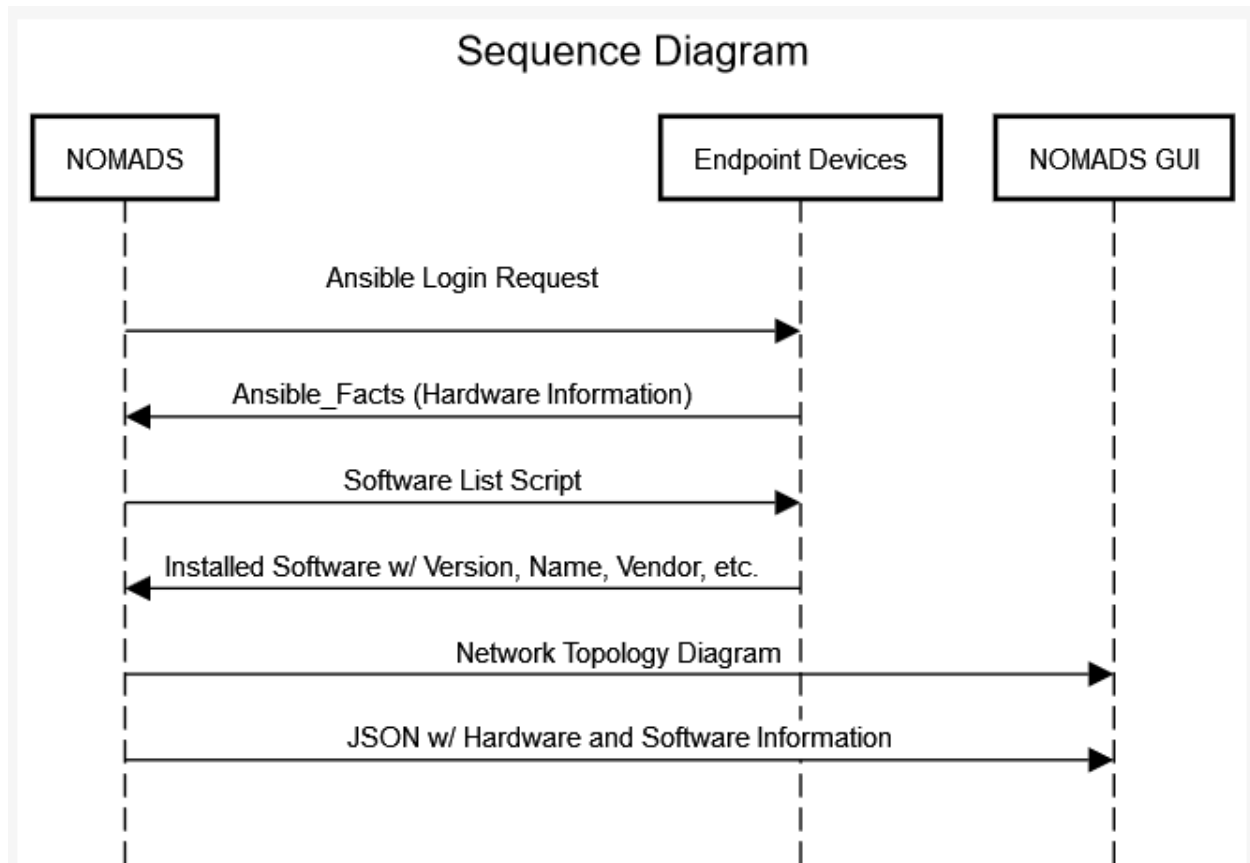
## Machine Profile Object Structure

---

```
{
  "Hardware": {
    "name": String,
    "ip_address": String,
    "virtual": Boolean,
    "manufacturer": String,
    "model_number": String,
    "serial_number": String,
    "os": String,
    "location": String
  },
  "Software": [
    {
      "name": String,
      "vendor": String,
      "version": String,
      "purpose": String,
      "software_type": String
    },
    {
      "name": String,
      "vendor": String,
      "version": String,
      "purpose": String,
      "software_type": String
    }
  ]
}
```

## Sequence Diagram

---



## Appendix

---

### Implementation Technologies:

- A. Fetch:
  - a. Ansible/ansible-runner
  - b. Bash Scripting
  - c. Powershell Scripting
  - d. Windows Remote Management
- B. Process:
  - a. Python
- C. Present
  - a. Flask

- b. Microsoft Visio
- c. Pyvis