MAN^2

Navy Operational Machine Auditing Development Suite (NOMADS)

Nick Arenberg

Nick Mattair

Andy Ho

Max Medearis

Matthew Chen

Mentors: Mike Cloud, Kevin Burk

Project Background

The Problem

- With the frequency of cyber attacks increasing, it is necessary to ensure that networks that are thought to be secure have no leakage.
- Manually gathering information on every endpoint device on a network is time-consuming and expensive. It wastes valuable security resources.
- Human security teams are too slow and unreliable to perform detailed security audits of large networks.

Issues with Existing Solutions

- There are many tools that provide their own version of endpoint detection and response.
- 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.

An automated, low-cost, low-footprint network auditor and visualizer.

Project Goals (FP2)

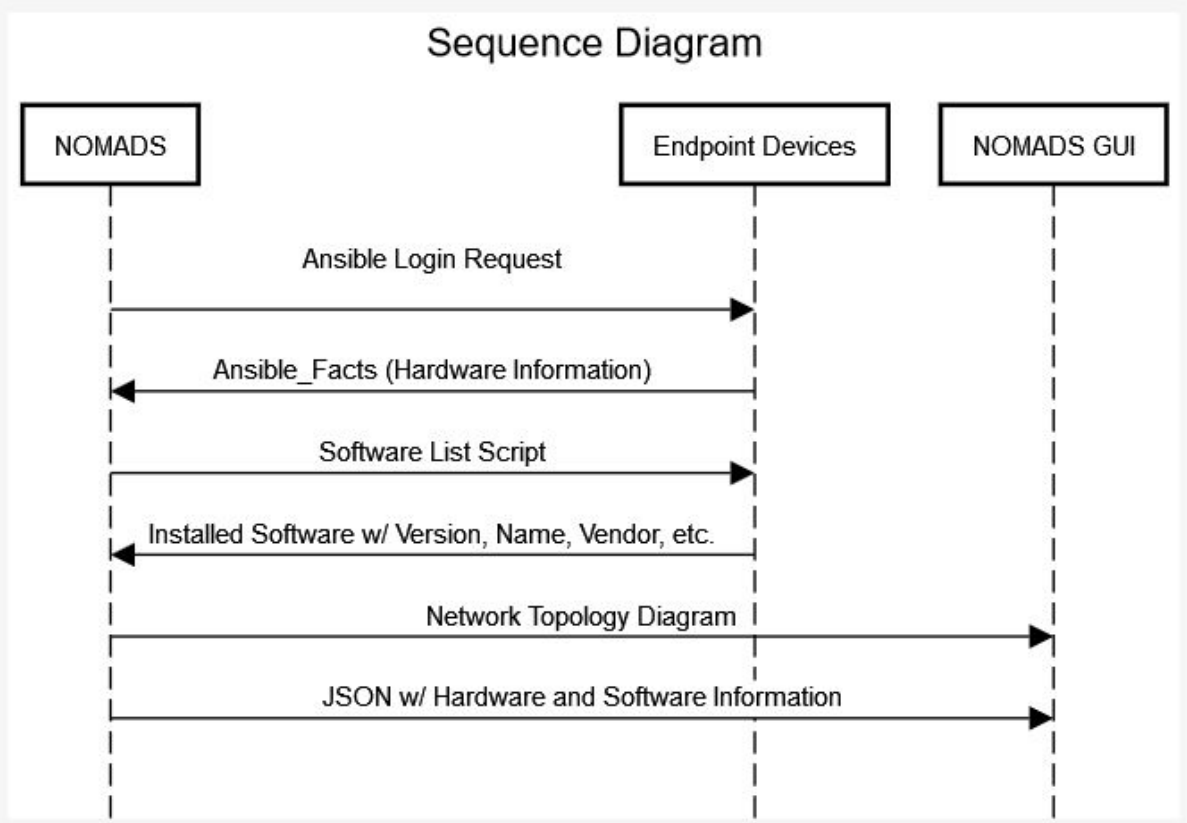
Fetch, Process

- NOMADS audits 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.
- Data is then stored into json data dumps and parsed into human readable spreadsheets

Present

- The data gathered during the audit will be used to generate an interactive network graph, which details individual machines, and creates reports on potential threats with detailed information on each specific endpoint and their flagged software or hardware installation(s).
- Users will also be able to view a topological Visio document that is programmatically generated from the data.

How it works


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

Project Technologies

Fetch, Process

- Ansible/ansible-runner
- Bash Scripting
- Powershell Scripting
- Windows Remote Management
- Python

Present

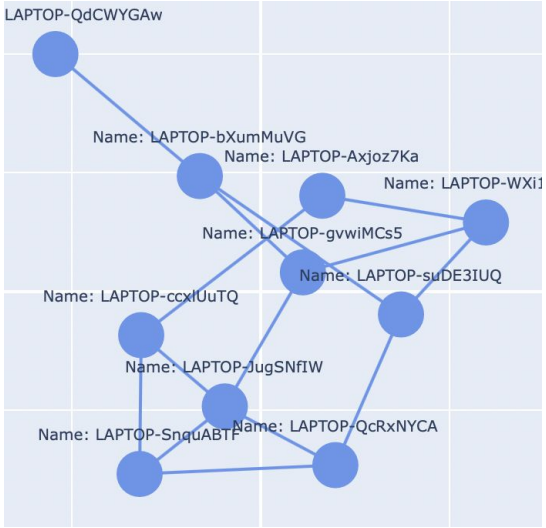
- Flask
- Microsoft Visio
- Pyvis

Project Outputs

JSON Data

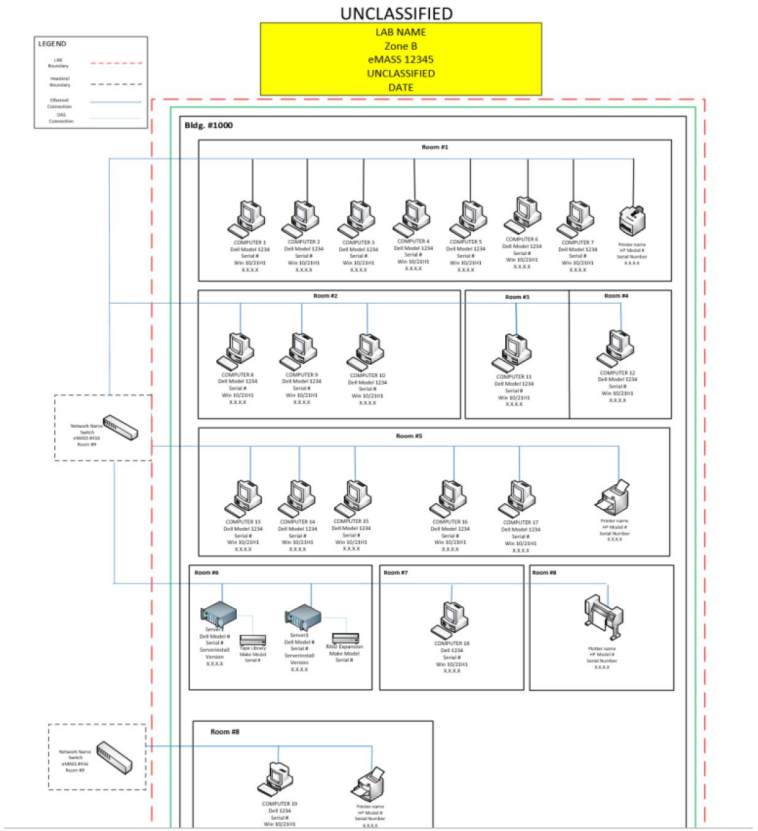
```
{
    "Hardware": {
      "name": String,
      "ip_address": String,
      "virtual": Boolean,
      "manufacturer": String,
      "model_number": String,
      "serial_number": String,
      "os": String,
      "location": String
    },
    "Software": [
      {
        "name": String,
       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

Network Graph



Next Steps

- Augmenting the Scanner
- Improving Visualization
- File Utilities
- UI Development



Special Thanks

- Mike Cloud
- Kevin Burk
- Angelique Zamarron
- Douglas Bradley