Grid-SIEM

Trent Bickford, Ella Cook, Daniel Ocampo, Westin Chamberlain

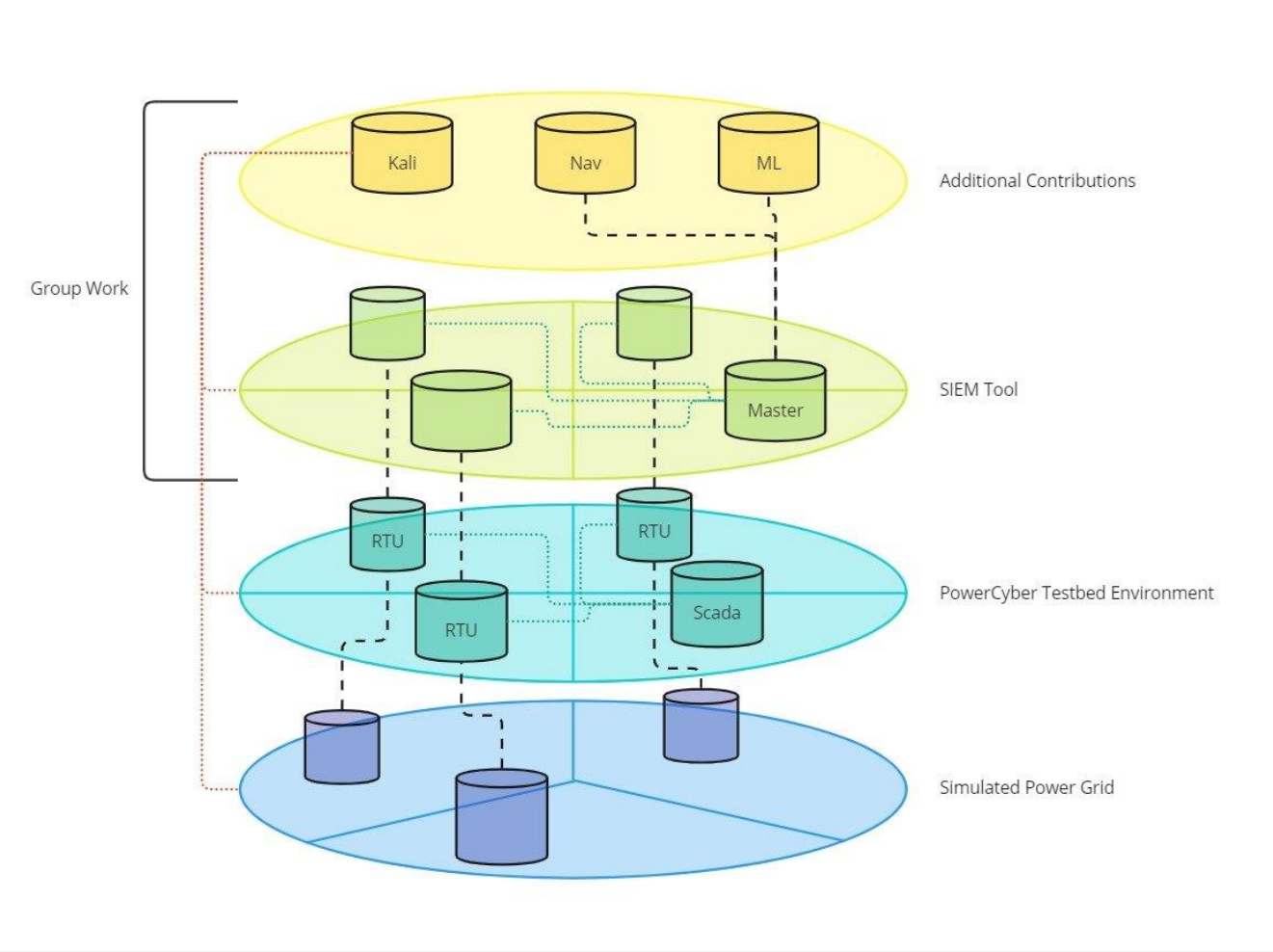
Sdmay24-29

Introduction

- The goal of our project is to secure a power grid by utilizing a Security Information and Event Management (SIEM) system supplemented by a machine learning algorithm.
  - SIEM: Security Onion
  - Machine Learning: Custom
- It is meant to be utilized by a security team, monitored as frequently as possible.
- Important because it helps secure the power grid

Detailed Design

- Four Layers
  - We work on two
- SIEM Tool layer
  - Security Onion
    - Kibana, Zeek, Suricata
- Kali Linux machine
  - Caldera, Custom Scripts
- Machine Learning
  - Supplement SIEM, detect malicious activity
- Attack Navigator
  - Assists in keeping track of attacks
  - Helps identify attacks



Security Onion Work Progress

- 80%

- Done:
  - Set up the Manager Node
  - Set up the Sensor Nodes
  - Review traffic coming into the system
  - Set up the additional tools in Security Onion
- Tasks:
  - Fine tune the alerting and rules in Security Onion
  - Automatically forwarding the logs to the machine learning component

Kali Work Progress

- 70%

- Done:
  - Set up the Kali box with tools that we will need
  - Established a PowerShell connection to the other devices
  - Used cybersecurity attacks against the system
- Tasks:
  - Find more network-based attacks to be picked up by the system

Machine Learning Work Progress

- 70%

- Done:
  - Decide on the machine learning framework to use
  - Decide on the algorithm that machine learning will use
  - Create a script to intake logs and classify them
- Tasks:
  - Ingesting the logs directly from Security Onion
  - Training the model using our data

Attack Navigator Work Progress

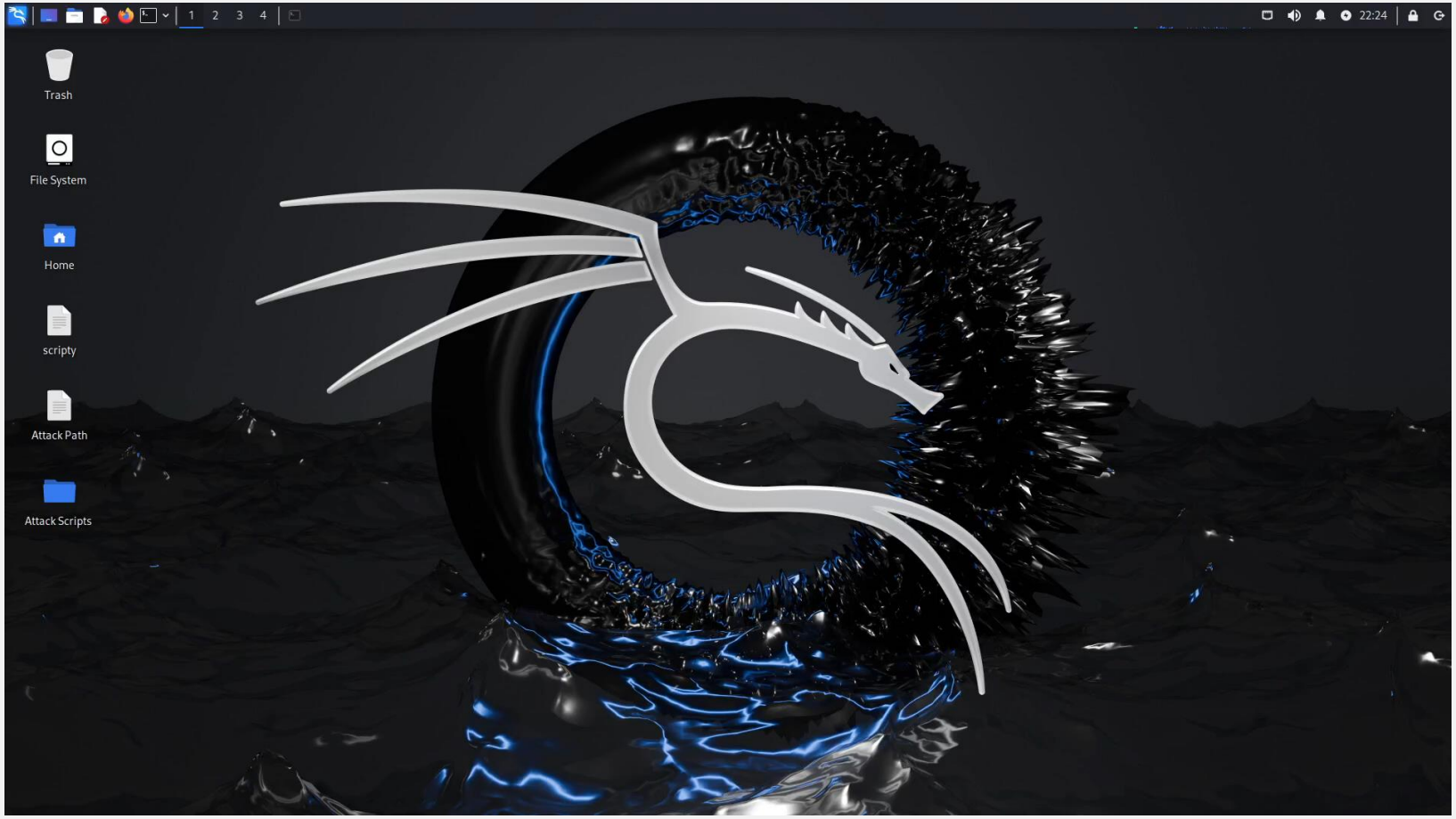
- 70%

- Done:
  - Reviewed the existing Attack Navigator tool
  - Explored the Playbook tool as an option to assist Navigator
  - Created a test python script for demonstration

- Tasks:
  - Integrate the Attack Navigator with a machine learning component
  - Ensure that the Playbook measures effectively counteract attack simulations.

Security Onion Demonstration

Security Onion Demonstration



Machine Learning Demonstration

- Displays classification report for both algorithms
  - Random forest is meant to classify
    - Accuracy was 100% in this case with single test log – overfitted case, this will change as the script is modified & tested
  - Isolation forest is meant to identify anomalies
    - Accuracy of 16% indicates anomalies are rare
    - Expected result most of the time – normal data points whether malicious or non-malicious will be the case most of the time
    - Zero-day attacks will be rare

```
/home/ubuntu/Desktop/tmpzeek1/logssensor1/logs/2024-02-17
Random Forest Results:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         7
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00         3
           3       1.00      1.00      1.00        30

    accuracy                           1.00        43
   macro avg       1.00      1.00      1.00        43
weighted avg       1.00      1.00      1.00        43

Accuracy: 1.0

Isolation Forest Results:
/home/ubuntu/anaconda3/lib/python3.10/site-packages/sklearn/metrics/_
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(resul
/home/ubuntu/anaconda3/lib/python3.10/site-packages/sklearn/metrics/_
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(resul
/home/ubuntu/anaconda3/lib/python3.10/site-packages/sklearn/metrics/_
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(resul
              precision    recall  f1-score   support

           0       0.17      1.00      0.29         7
           1       0.00      0.00      0.00         3
           2       0.00      0.00      0.00         3
           3       0.00      0.00      0.00        30

    accuracy                           0.16        43
   macro avg       0.04      0.25      0.07        43
weighted avg       0.03      0.16      0.05        43

Accuracy: 0.16279069767441862
(base) ubuntu@ubuntu-vm-master-120:~/Desktop$
```

Challenges & Solutions

- Security Onion
  - Challenges
    - Architecture problems – most recent version of SO needed 2 NICs
    - Network problems – manager & sensor nodes couldn't communicate
  - Solutions
    - Remade VMs to accommodate 2 NICs
    - Adjusted firewall rules to accommodate allow proper communication between nodes
- Caldera
  - Challenges
    - Issues with compatibility
    - 32-bit OS, but Caldera requires 64-bit
  - Solutions
    - Looking into updating Windows versions on VMs

Challenges & Solutions

- Machine Learning
  - Challenges
    - Log formatting – zipped logs
    - Ingesting multiple logs at a time while still providing an output within a reasonable timeframe
  - Solutions
    - Using multithreading to unzip logs and consider ML as a secondary process that is read in smaller chunks
- Attack Navigator
  - Challenges
    - Introduced to the project beginning of second semester – might run out of time to fully implement
  - Solutions
    - Provide a proof of concept for future students to continue working on

Conclusion

- Establish an effective defense perimeter to protect the Power Cyber architecture. Several layers of tools and frameworks work in sync to provide these capabilities as shown in the detailed design slide.
- Make use of the free an open-source tool Security Onion to develop strong capabilities. This was achieved by strategically implementing an assortment of manager and sensor nodes to forward valuable information to a master node.
- Evaluate the effectiveness of our defense strategy by implementing various tools such as Mitre Caldera or Metasploit.
- Implement existing defense frameworks such as Mitre Navigator to close gaps in our defense strategy.
- Deploy our own machine learning tools to learn from traffic data logs.
- We encountered several challenges and roadblocks while working on this project, including software compatibility and technical issues.