CYSE Master Project Idea

Service providers, those internet use enablers, need to muster the responsibility to handle the most basic actions in establishing healthy cyber hygiene for home users. There is already a complicated atmosphere and difficult path forward for internet access given the societal reliance on the internet and cost of the home user's access. Security, or basic cyber hygiene, needs to be addressed thoughtfully so as to not end up in a similar costly situation for the home user. I believe there is hope for home users to develop more awareness and care more deeply about their own online behavior as well as their online security. However, the foundation must be paved by the organizations and groups with the most control within the existing ecosystem. The goal requires a hefty lift from service providers in order to implement the necessary technology. This could also include Government incentivizing the cooperation of these providers.

Requirements:

- Easy to implement home security solution (centralized alert dashboard)
- Proposal of ISP/related providers to adapt and include with all service tiers
- Suggested use of LCD panel/device near router/networking gear for comprehensive, graphical alert dashboard
- Demonstration of alerting system and dashboard conception as well as explanations for value in protecting home users generally
- Outline potential avenues for IR (incident response) in the event of alert on LCD dashboard called in by home users
- Future direction for server provided security features to include at no additional cost for the greater value of national and civil security

Consider circumstantial questions. For example, the user adds a new device - whats the process for the new device to aggregate and communicate to the SIEM Dashboard.

Updated Brainstorm 2/8/2023-2/9/2023

- Complete k3s Homelab + Security Features and Central Dashboard
- Documentation and demonstration evidence (screenshots and videos)

- Write-up manuscript explaining work completed and practicality of implementing in all homes
- Future direction for possibilities ISP's can include free security features + dashboard in all offerings
- Remote manageability/maintenance service lifecycle
- Failover availability, utilizing Kubernetes allows for automatic failover
- Small footprint physical equipment with lightweight software operation
- Practicality with cost benefit evaluation (estimated; call for detailed financial and cyber health* cost-risk analysis)

**cyber health can be primary in direct negative impact to the organizational network or secondary, tertiary, and so on by providing no positive force against an adversarial advantage*

Part I:

Includes the synopsis, documentation, and conceptual reasoning for a home k3s security monitoring solution.

Part II:

Includes an argument for ISPs baseline security offering including a custom dashboard with service alerting capability. Question whether service should be a small additional fee. Role of government in incentivizing solution from ISPs. Responsibility of ISPs to provide basic rights to security for home internet users. I should be sure to include the concept of "secure by design" that is currently gaining traction within the tech and cyber related spaces. This originates from DHS (if I am not mistaken) and has been advocated for heavily by CISA. My idea is to bring this to the ISP as a responsibility they hold.

Potential Spec/SBOM:

- 4x RPi running Alpine/Fedora [cluster]
- k3s
- Longhorn
- Prometheus
- Wazuh (SIEM)
- Grafana
- 1 or more IoT (RPi) running balenaOS
- Wazuh/Network Vulnerability Scanning Agent
- Communicating data to RPi cluster (ISP)

Step-by-Step Phased Project:

- Install OS RPi's

- Enable remote access such as with cockpit or ssh to each RPi
- Ensure completely updated/upgrade systems
- Install k3s on master
- Install k3s on all workers and link to master
- Install Longhorn for storage
- Install Prometheus for monitoring
- Install wazuh SIEM
- Install Grafana for visualizing wazuh data
- Install balenaOS to IoT (RPi)
- Install Wazuh communicating agent

ChatGPT Brainstorm Session 2/11/2023

Developing a Kubernetes project that includes a Security Information and Event Management (SIEM) solution and additional security features can be a complex undertaking, and requires expertise in both Kubernetes and cybersecurity. Here's a high-level overview of the steps you can take to accomplish this:

1. Plan and design your project: Start by defining your goals and requirements for the project, including the security features that you want to include. Determine what type of SIEM solution you want to use and how it will integrate with your Kubernetes cluster.
2. Set up a Kubernetes cluster: Create a Kubernetes cluster on a cloud provider or on-premises infrastructure. Ensure that the cluster is properly configured and secured, following best practices for security and privacy.
3. Install and configure the SIEM solution: Install the SIEM solution of your choice on the Kubernetes cluster, following the vendor's instructions. Configure the SIEM to collect security-related data from the cluster and other relevant sources, such as network devices and firewalls.
4. Implement additional security features: Depending on your requirements, you may want to implement additional security features, such as network segmentation, role-based access control, and encryption. Consider using Kubernetes extensions and add-ons to simplify the implementation of these features.
5. Test and validate the solution: Test the SIEM solution and additional security features to ensure that they're working as expected. Validate the solution by simulating potential security threats and checking the SIEM's response.

6. Monitor and maintain the solution: Regularly monitor the SIEM and additional security features to ensure that they continue to function as expected. Stay informed of new security threats and vulnerabilities, and update the solution as needed to stay ahead of potential threats.

This is just a high-level overview of the steps involved in developing a Kubernetes project that includes a SIEM and additional security features. Depending on the complexity of the project and your experience with Kubernetes and cybersecurity, you may need additional support from a team of experts or external consultants.

PROJECT EXPANSION IDEA

If I can manage more time to sink into this project after the semester I would like to incorporate a "hacker" layer of my rack setup. This would include hacking devices that attack the "home" layer. The "ISP" layer should be able to automatically detect these attacks, alert, and resolve (remotely and on-site as needed). In addition, I would love to video log the project and experience overall. As demonstrated with the current photos and screenshots this will not be to perfect detail. This is more a snapshot overview of various steps and the completed project demonstration.

Day 1 (2/16/2023):

One of the great benefits to Linux and open source technology generally, is the specificity of products. For instance, "balenaOS is an operating system optimized for running Docker containers on embedded devices", and this is an operating system I will use for this project. It is perfect since a goal is to have both a small physical footprint as well as low energy requirements. In order to start my project and ensure the remaining steps go smoothly I will spend a good amount of time and focus into flashing, initializing, and setting up remote access to the Pi with balenaOS. The device running balenaOS will be the endpoint communicating back to the ISP. This will include a lightweight agent for metric collection on the network. This information will be sent out to the ISP where it will be visualized and monitored on a central dashboard. In addition, a personal web console can be accessed by the home user in the event they prefer to view their own graphical representation of their own network hygiene.

The installation got a little complicated. I realized I only needed a single device for balenaOS, essentially the IoT relay device for security data to the cluster. The cluster must be running a different OS as well as k3s and Grafana to monitor the SIEM data. I chose Fedora CoreOS. It requires some additional steps to provision for deployment on a Raspberry Pi 4. Alas, I am spending a little extra time getting all the proper OS installations completed.

It is 9:30pm and my first coreOS installation failed after many provisional steps. I will put this method on the back burner and use Ubuntu server for this project. It is known to be an easier solution to use and I have seen the most tutorial work including this OS as choice.

Day 2 (2/17/2023):
Today I have my head on straight in regards to a path forward. However, I am still patiently taking each step as it comes. Most of today was spent initializing the k3s cluster. I had fresh Ubuntu server instances on each Pi. I needed to make sure to do my due diligence with checking security or necessary updates, changing root access and other passwords, as well as linking up the k3s master with each worker node. In addition, I spent some time setting up Cockpit for ease of accessing and configuring the services needed to get the cluster up and running. In the future I will have a Rancher access from the web.

OKAY... so I keep running into a problem. The issue is every time I try to add a worker node it gets hung up connecting to the master server. I am really stumped since I have messed with firewalls etc and can not get it straight. Now, I am considering an Ansible approach to automate the deployment from a pre-developed playbook on Github.

In order to run the ansible playbook I needed to set passwordless SSH remote access to the Pi's. This is also more secure in comparison to the use of a traditional password authentication.

By the end of the day I was able to have a complete k3s cluster with 1 server and 3 worker nodes. In addition, I was able to get kubernetes-dashboard installed and accessible.

Day 3 (2/18/2023)
It is Saturday so I did not want to spend entirely too much time tinkering with my cluster as it is easy to do. However, I wanted to check that it is still running and complete at least one more install. It has all been stable for the past 16+hours and I was able to complete the setup of Prometheus and Grafana. I utilized an operator called kube-prometheus-stack. This is an open-source solution provided to us by the

Prometheus Monitoring Community. Now I am able to monitor different aspects of the cluster from a Grafana graphical representation. Mainly this includes computing resource data among some other basic information.

I believe my next step will be to configure wazuh so that the RPi3 running balenaOS acts as a wazuh agent and all of the data is visualized in the Grafana instance.

Day 4 (2/19/2023)
No significant work completed to report today.
Day 5 (2/20/2023)
(2/27/2023)
I think it is time to write out a summary on the technologies that will certainly be used in the lab, thus far.

First and foremost, the core infrastructure including hardware devices and OS.
Raspberry Pi 4 - This is an extremely popular and widely supported SBC (single-board computer) that offers a lot of performance at a very low price. Granted, these are currently (Feb. 2023) extremely difficult to procure and mine were collected a few years back for conducting various home lab practices.

Ubuntu 22.10 Server - Likewise the Pi, one significant reason this OS was chosen is due to the widespread support and documentation (among Linux distributions). Honestly, I would prefer as light and secure an OS as possible (such as with Alpine).. However, there would be additional steps for a lot of installation which would inevitably prolong the time to complete this lab practice. In a project with less time constraints I would opt to use Alpine.
Lightweight Kubernetes (k3s)

that's when it all went wrong. i quickly began discovering the depths of everything involved in Kubernetes (especially from a security perspective). alas, when exercising the brain as much as when learning kubernetes hands on there is no shortage in stuff to write about — *and that's a win win*.
my ultimate decision has been to both simplify and intensify the specificity of the project. the goal is to ensure continuation upon my previous efforts working on researching and configuring my cluster as well as increase the value gain of specific knowledge. my intention is to become more "specialized" in

kubernetes security and particularly focus on security policy management and enforcement. my written component will focus almost entirely on the technologies leveraged and the pressure for adopting these security practices early in a DevOps lifecycle. hopefully i will one day pick this original idea back up. at minimum, to provide a written argument for the isp/open-source/government collaborative effort to protect our citizens' home networks.