Midterm

October 16, 2019

Enterprise network security.

You will use the tools learned in class to complete the following exercises. For certain exercises, you will need to write and/or complete a separate source file. For others, you will need to do some analysis and write up your observations.

Your final submission on Canvas should include:

- Any source files written or changed.
- A single write-up addressing all exercises that require longform responses.

Please zip the relevant files up as a zip archive and submit on Canvas.

Kubernetes

Write a Deployment and Service yaml file for a message queuing monitoring application: RabbitMQ You can run the service in docker by: `docker run -d --hostname my-rabbit --name some-rabbit rabbitmq:3-management` Docker image for rabbitMQ is `rabbitmq:3-management`. You know that the chatbot application requests a minimum of 150Mi RAM and 0.05 CPU, and maximum of 300 Mi of RAM and 0.125 CPU. The application is exposed on default port 15672 on the cluster and you want to access it on Port 32500 from your host machine. Write a deployment for this application with atleast 5 instances of it and the given minimum and maximum resources. Write a corresponding service file for the same exposing the service on the desired port on the host.

Networking:

As a sysadmin you notice an interesting pcap file (interesting.pcap). Open the pcap and find out what is happening in the pcap at a high level. Write no more than a one paragraph.

SELinux

You work at a bank and are trying to deploy a simple SELinux policy to manage bank customers' access to their bank balances. Below is a template for the Type Enforcement file.

Assume the following:

- `user_home_t` is the domain of an unprivileged bank user like you or me.
- The bank runs an administrative daemon in the domain `bank_admin_t`.
- Bank balances are text files, and should have type `bank_balance_t`.
- Users can have multiple accounts with balances in separate text files.
- A user's account balances will be stored together in one directory of type `bank_balance_t` associated with that user.

Please fill in the blanks to meet the requirements described below:

- Bank users should have read-only acccess to their balance.
- The bank admins should have access to see balances in a directory as well as read, write, delete and create user balances.
- The bank admins should also be able to create new `bank_balance_t` folders for new bank customers.
- Don't worry about preventing one user from reading a different user's balance.

```
policy_module(bank, 1.0.0)

type bank_exec_t;
type bank_admin_t;
type bank_balance_t;
init_daemon_domain(bank_admin_t, bank_exec_t)
files_type(bank_balance_t)

require {
    type bank_admin_t;
    type bank_balance_t;
    type user_home_t;
    class file { create execute open read write getattr };
    class dir { add_name search write create remove_name getattr };
}

type_transition ____________ bank_balance_t:file ____________;

allow user_home_t bank_balance_t:dir { ______________________________ getattr };
allow ____________ ____________:________ { ______________________________ getattr };
allow bank_admin_t bank_balance_t:dir { ______________________________ getattr };
allow ____________ ____________:________ { create open read write getattr };
```

Monitoring

You are the admin of a system. Your system was attacked and a system capture file called `suspicious.scap` of the attack was recorded. Figure out where the attacker was on the server, and what attack occurred. Use osqueryi to find the user that writes to `/var/www/`, because this user was responsible for starting the server. NOTE: The system setup is the exact same as lab 2 part 3b. The capture files can be read with sysdig-inspect.