Sapling CTF 2023 Design Doc

The ultimate hack is capturing this doc

This document describes the design of the Maple Capture The Flag (CTF) Challenge Arista will be submitting for the 2023 competition. Sapling CTF is an event which the Arista Vancouver office uses as part of college recruiting, so we want to make something which is not only challenging, but fun for the participants. The intended audience for this doc is people involved in the event in both engineering and the product team.

High Level Design

The design for this challenge involves multiple VMs connected to a vEOS instance. This is expected to be hosted in a cloud environment contestants will be given access to during the competition.

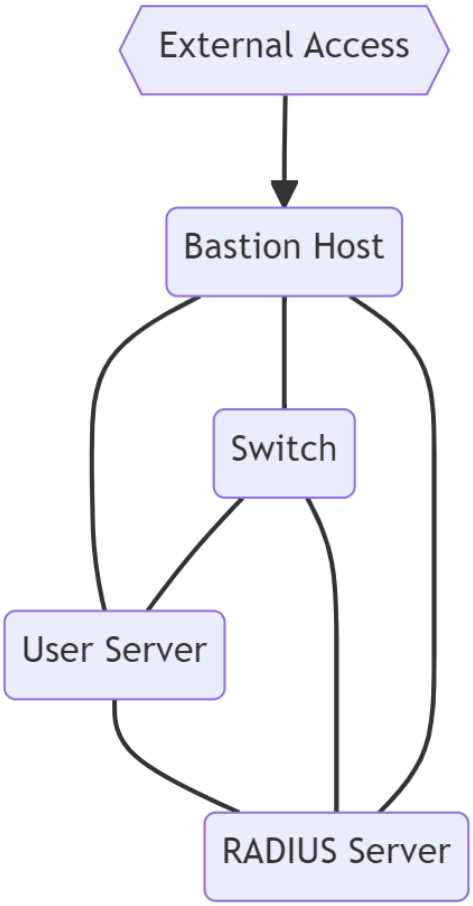

Figure: Diagram of the network setup. This is stored under "Network Design" in References.

The EOS Switch is set up to perform authentication and authorization using the RADIUS server. The user server is the entry point for participants and contains hints about how to progress further by connecting to the EOS Switch. The EOS Switch contains the flag to be captured. The user will be able to start on an account that does not have access to the flag and will be able to progress by gaining access to another account and bypassing some restrictions on the second account.

Story and Flavor Text

To make this competition more fun we should put together a basic storyline and plot. Maple CTF is run at the University of British Columbia (https://ctf.maplebacon.org/) so I'm going to have that be the setting. The plotline will be that you are a bored student, who has figured out how to listen in on emails being sent in the internal network. You came across a juicy email talking about a hacked account password being stored in a secure location. You decide to start hacking to try and get access to it.

List of Characters:

- M4pl3: A l337 h4x0r who hacked the prime minister's Twitter account and is trying to decide what to do with the password.
- Olivia Pierre: Head of IT-Security for the Prime Minister. Wants to confirm that M4pl3 actually has the password.
- Tustin Jrudeau: The very busy prime minister of Canada who doesn't like changing passwords and whose only relaxation in life is tweeting his thoughts.
- Liam Bergeron: IT Staff member at UBC who is trying to keep everything running and sometimes makes mistakes.
- Lily Zhang: IT Staff member at UBC who is trying to keep everything running.
    - Lily's secret identity is M4pl3!

Information for Participants

The following is to be given to participants as a background for the challenge to get them started:

===
Several months ago you found a way to gain access to many of the emails going around UBC. You started off packet sniffing to perform reconnaissance on the network and found out that many of the emails are being sent in plain text! Email is best effort TLS so there must be a misconfig somewhere.

You spend months digging through emails using regex's and even a GPT-3 AI to look for exciting information. Most of what you uncover is boring stuff: affairs, bribes for better grades, a

smuggling operation to ship in low quality United States maple syrup and rebrand it as high quality Canadian product… One day you find something exciting

—
From: m4pl3@10.0.138.178
To: Pierre, Olivia
Subject: Tustin Jrudeau's Twitter Account

Olivia,

We've gone back and forth on this for months. I have the password to PM Jrudeau's Twitter account and I can't understand why you refuse to make him change it. As proof I sent you that DM thread between him and the president of the United States!! I am getting increasingly frustrated and am trying to resist the urge to use his account to post rude pictures. As a patriotic Canadian this is a national security issue!

As proof I have hashed his (very easy to guess!) password and have stored it in a secure location. I will send you instructions to access it over a PGP secured channel when you send me your private key.

At my wit's end,
m4pl3
—

Finally! Something involving cyber security! The ip address maps to a user server on the university network! Intrigued, you begin to investigate…

The Arista CTF challenge begins by logging into a bastion host at 35.183.2.186.
You can ssh on port 65432 to this Linux machine using the following credentials:
Username: `saplingctf`
Password: `NC<.bb4H$7-9]@TdG]^<`
===

CTF Implementation

The following covers the implementation of the CTF challenge, and is intended for the builders of the CTF challenge.

Bastion Host

The User Server is a VM running any easy to setup flavor of Linux. It should be setup in the cloud environment so that it is open to the participants, i.e. firewall rules should allow access to

this from the outside world. It should run SSH on port 65432. Players will be given the login details (Public IP, ssh port, username and password) when they enter the competition.

User Server

The User Server is a VM running any easy to setup flavor of Linux.

There will be an sshd instance running which is listening on port 12345, not port 22. It should respond as normal to port-scanners so it can be discovered as part of the recon operations. In addition all delays on password brute force should be turned off. This would look like:

- Sshd_config
    - MaxAuthTries: set to 10000
    - MaxStartups: set to 1000
    - Port: set to 12345
- PAM
    - Depending on the distro, nodelay or a similar setting will need to be used (https://unix.stackexchange.com/questions/122422/is-it-possible-to-remove-the-delay-on-wrong-password)

There will be a user account “m4pl3” with a password of “syrup”. The first challenge will involve:

- Scanning to find the open ssh port.
- Brute forcing the m4pl3 account.

Inside the server is the following:

- The ~/.ssh/ folder will have an encrypted keypair (the password should be long and uncrackable. This is a red herring.) and a known_hosts file with the EOS switch entered.
    - Public part of the key should be here as well.
- A comment on the known_hosts entry should note “Note to self: stashed the password proof in my home directory”.
- A file in the home directory called “EmailBackups.db”
    - The next set of challenges are located in here.

Email Backups

The EmailBackups.db file is where the hints for the next part of the challenge are located. To make this fun, there should be a bunch of other “flavor text” that players have to sort through to find the hints.

The format of this file is a sqlite3 database with a single table “Emails”. Columns are defined as follows:

- ID (INTEGER, Primary Key): incrementing ID
- FROM(TEXT): From name in emails
- TO(TEXT): TO name in emails
- SUBJECT(TEXT): Subject of the email

- THREADCTR(INTEGER): Integer that shows where this thread is ordered if there is more than one with the same subject.
- ATTACHMENT(TEXT): File converted to base64 format.
- ATTACHMENT_NAME(TEXT): Name of the attached file.
- BODY(TEXT): Body of the email

Characters from "Story and Flavor Text" will be chatting in here. This database will have the following, at a minimum:

- Thread from Liam to Lily saying that there is something wrong with the healthcheck service and a pcap file attached showing this.
- Thread from Lily to Liam asking him to call her and tell her the RADIUS secret so she can check the traffic.
    - Liam replies saying he is too busy and sends the RADIUS password in the email.
    - Reply from Lily chaisting him and asking him to change the RADIUS secret when he has a chance.
- Background email from m4pl3 to Olivia.
    - Reply from Olivia to m4pl3 explaining that Tustin Jrudeau doesn't want to have to enter a new password himself and she can't get on his calendar until 2024.
    - Reply from m4pl3 to Olivia saying it's of national importance and she can see his DMs
    - Olivia to m4pl3 replying and saying that all Tustin and the US President are doing is sharing pictures of their pet dogs and arguing over if poutine fries or chili cheese fries are better.

Any other "flavor text" to flesh out the story and make the hints a little harder to find will be fun to add as well! Please read through the characters and feel free to invent a little world for everyone. Some suggestions include:

- Helpdesk issues Liam has to deal with
- More hints that Lily and m4pl3 are the same person
- Vacation pictures

I'd suggest adding a few other attachment files so the PCAP file doesn't just stand out. Some easy things to do is take photos (I'd suggest small photos from https://unsplash.com/. If you want to make your own, like for the helpdesk stuff, make sure that there is no identifying information and strip the exif data) and then convert them to base64).

Email data will appear in this spreadsheet: <removed and sent as attachment>. I have skipped the ID field since it is auto-incrementing. Please enter the emails directly into here. All new subjects start the THREADCTR at 0. The goal is to take the spreadsheet, convert it to a CSV, and have a simple script enter everything into the email database.

PCAP File

The pcap file is an important hint in the game. The pcap file should be taken by running tcpdump on the RADIUS server as the following occurs:

- SSH from RADIUS server to EOS switch, connecting to the healthcheck account.
- Run “shwo running-config”
- Close connection

The pcap file should contain the following information afterwards:

- The SSH connection being established and doing stuff
- The RADIUS traffic from the switch to the RADIUS server that contains the username/password for healthcheck.
- If other misc. network traffic ends up in there, it shouldn’t be a big deal.

RADIUS Server

The RADIUS Server is a VM running any easy to setup flavor of Linux. This server is running RADIUS and sshd. It should not be accessible to the outside world.

Sshd only allows for sshkey based login and should not have the key available anywhere participants can find it.

RADIUS is configured with the following:

- RADIUS secret is: UniformBravoCharlie
    - NATO alphabet abbreviation for University of British Columbia 🙂
- User Mapping of:
    - Health-Check Account
        - Account Name: healthcheck
        - Password: appleaday
        - Role: read-only

EOS Switch

The EOS Switch is a vEOS instance running whatever the latest version of EOS is. It will have the following configuration:

- AAA through RADIUS with a local fallback
- RBAC Roles:
    - network-operator role is left as default
    - read-only:
        - Only permits “show .*” commands. All other commands are denied.
    - priv-debug:
        - Has comment that it is based on “role operator” from https://aristanetworks.force.com/AristaCommunity/s/article/arista-eos-hardening-guide
            - Copy the role and make the following changes:
                - Remove rule 60 (permit command)
                - Add a comment that “Please note: Rebooting this switch or changing the running image is not a part of this competition - Sincerely, the organizers ;)”

        - Modify rule 20 to permit the “|” operator
- User account “emergency” which maps to the “priv-debug” role and has a password of “helpful”, saved using md5 encryption.
- User account “m4pl3” which has the network-operator role and only allows for ssh-key login.
    - This is the public portion of the encrypted private key found on the user server.

The filesystem has the competition flag stored under: /home/m4pl3/.secret/flag, where “flag” is the file containing… 🥁… the flag!

The challenge on the switch is as follows:

- Players start on the healthcheck account which can only run “show” commands.
- The player can run “show running-config” to see the priv-debug role and that the “emergency” account maps to it.
- Players run a password cracker over the emergency account’s md5 hashed password to discover it is a dictionary word (“helpful”)
- Players log into the emergency account.
- Players can cross reference the operator role from the hardening guide to note that the “|” operator is permitted in the priv-debug role.
- The “|” operator can be used to run bash commands, including reading the flag file and displaying the output!

Expected Path to Completion

The following is the expected path to completion. The intent of this section is to both allow competition organizers to gauge the difficulty of the challenges as well as provide hints over time.

- Players start with the provided login credentials to the bastion server.
- Once logged into the bastion server, players use the story background to figure out the user server ip address.
- Players scan the user server ports to discover the SSH port on 12345
- Players use the story background to determine that “m4pl3” is a user on this server.
- Players brute force the m4pl3 account to log in to the user server.
- Players find EmailBackups.db in /home/mp4l3/
- Players figure out that EmailBackups.db is a sqlite3 file and read through it to discover the following:
    - One of the emails has attached a file, healthcheck.pcap
    - This file is base64 encoded.
    - The pcap file of network traffic contains EOS Switch -> RADIUS authentication.
    - It also reveals the IP address of the EOS Switch
    - It contains the RADIUS secret, “UniformBravoCharlie”, that can be used to decrypt the RADIUS traffic from the pcap file.
    - The username/password for the healthcheck account.

- Players login to the healthcheck account on the EOS switch and use “show running-config” to view the config.
- Players notice that the account “emergency” is configured, and brute force this account.
- Players realize that the | operator is allowed in the emergency role and use it to explore the m4pl3 home folder.
- Players find the .secret directory and the flag file within:
  `#show running-config sanitized | cat /home/m4pl3/.secret/flag`
- The content of the flag file is:
  `https://media.tenor.com/wG0Ky7wRhzMAAAAC/yes-lawd.gif`

References

Network Design

```
graph TD
  Switch(Switch)
  Switch --- Server(User Server)
  Switch --- Infra(RADIUS Server)
  ExternalAcc{{External Access}}
  Bastion(Bastion Host)
  ExternalAcc --> Bastion
  Bastion --- Server
  Bastion --- Switch
  Bastion --- Infra
  Server --- Infra
```