CYB631 Automating Information Security with Python and Shell Scripting

Lab 8: Advanced Persistent Threat

Your Name: **Vaibhav Mayekar**

Exercises:

[Exercise I: Explore MITRE Att&ck Framework]

1. MITRE ATT&CK (https://attack.mitre.org/) is a knowledge base of adversary tactics and techniques based on real-world observations. Explore the Enterprise Matrix for Windows. (https://attack.mitre.org/matrices/enterprise/windows/).
2. There are 8 stages in this matrix. For each stage, there are about 10-20 techniques.
3. Please review at least 4 stages out of the 8 stages. For each stage reviewed, please choose one technique (or sub-technique if available) to explore. For example, under "Initial Access" stage, you will find "Drive-by Compromise" under "Initial Access" stage. Click on the link for the technique and read thoroughly about the techniques including the adversary groups who have exploited the technique. Please use the following format to report your results.

Pick one technique in **Initial Access stage** to review.

Name of the technique: **Exploit Public-Facing Application**, **ID: T1190**

a. Brief description of the technique:

Public-facing applications: what are they?

Any application that can be accessible online is considered public-facing. Websites, web servers, databases, network devices, and other software that is accessible to the general public over the internet can all be considered applications.

Why do attackers target applications with a public facing profile?

Attackers target public-facing applications because they are frequently weak points in their defenses. This is due to the fact that these apps are frequently created and released fast, and their degree of security might not be as high as that of apps that are not accessible over the internet. Furthermore, sensitive data is frequently processed and stored in public-facing applications, making them a desirable target for attackers.

How are public-facing applications exploited by attackers?

Attackers use software flaws to their advantage when targeting programs with a public interface. Numerous things, including bad development methods, software defects, and misconfigurations, can lead to these vulnerabilities. Once a vulnerability has been identified, an attacker can utilize it to either obtain unauthorized access to the underlying system or network or inject malicious code into the application.

What happens if you abuse an application that is intended for the public?
Exploiting an application with a public face can have serious repercussions. With the help of these exploits, an attacker may be able to take over a machine or network, disrupt operations, steal confidential data, and install malware.

How do businesses defend themselves against exploits for applications that are visible to the public?

- The following actions can be taken by organizations to safeguard themselves against exploits for public-facing applications:
- Establish a vulnerability management program. Part of this program should be finding and fixing holes in applications that are used by the public.
- Employ a web application firewall (WAF) to prevent assaults on apps that are visible to the public.
- Employee education: Workers need to be taught how to spot suspicious activity and report it.
- Update your software often to help patch any vulnerabilities that hackers might find.
- Put security measures in place: To identify and stop threats, this involves utilizing firewalls, intrusion detection systems, and other security tools.

    b. Name of an adversary group who has used the technique: **BlackTech** ID: **G0098**
    c. Name of one mitigation to this technique: **Application Isolation and Sandboxing**, ID: **M1048**
    d. Name of one way to detect it: **Application Log**, ID: **DS0015**

4. Pick one technique in **Execution stage** to review.
    a. Name of the technique: **System Services: Launchctl** , ID: **T1569.001**
    b. Brief description of the technique:
    Launchctl is a potent tool for managing macOS services. Regrettably, enemies may misuse this authority to install harmful software on a system. Attackers can install malware, run code continuously, and even conceal their actions from system administrators by taking advantage of launchctl flaws.

    Loading malicious Launch Agents or Launch Daemons is one frequent way adversaries abuse launchctl. These particular plist file formats instruct launchctl, the macOS service management framework, what applications to execute and when. Attackers can command launchd to run their own code and take control of the system by creating malicious Launch Agents or Launch Daemons.

    Launchctl can also be abused by opponents who utilize it to carry out commands directly. The launchctl command can be used to accomplish this by passing it the -w flag, which instructs launchctl to write the given command to

a plist file and then load it. Launchd will carry out the designated command after the plist file has loaded.

Launchctl can also be abused by adversaries to conceal their actions. The StandardOutPath and StandardErrorPath properties of the launchd job can be changed to /dev/null by attackers to stop their malicious code from logging anything to the console. System administrators will find it considerably more difficult to identify and look into attacks as a result.

Employers can safeguard themselves against launchctl misuse by implementing a number of safety measures. They should first keep an eye out for any illegal modifications to Launch Daemons and Launch Agents. You can accomplish this by creating a custom script or by using a system monitoring tool. Secondly, in order to stop unauthorized users from executing launchctl commands, they need limit access to launchctl by utilizing access control lists (ACLs). Ultimately, they ought to employ a security solution capable of identifying and stopping malevolent Launch Agents and Launch Daemons.

c. Name of an adversary group who has used the technique: **Calisto**, ID: **S0274**
d. Name of one mitigation to this technique: **User Account Management** , ID: **M1018.**
e. Name of one way to detect it: **File,** ID: **DS0022**

5. Pick one technique in **Privilege Escalation stage** to review.
   a. Name of the technique: **Escape to Host** , ID: **T1611**
   b. Brief description of the technique:
      Attackers can breach a container and access the host system by using the escape to host approach. Because it enables attackers to go over the security isolation offered by containers and access other containers as well as the host system, this poses a severe security risk. Attackers can get away to the host in a few different methods. Making a container set up to mount the host's filesystem is one popular technique. This makes it possible for the attacker to install and run malicious files on the host machine. Using a privileged container to execute commands or load a malicious kernel module on the host system is another popular technique. Additionally, attackers may misuse system calls like unshare.

      Attackers can breach a container and access the host system by using the escape to host approach. Because it enables attackers to go over the security isolation offered by containers and access other containers as well as the host system, this poses a severe security risk.

      Attackers can get away to the host in a few different methods. Making a container set up to mount the host's filesystem is one popular technique. This

makes it possible for the attacker to install and run malicious files on the host machine. Using a privileged container to execute commands or load a malicious kernel module on the host system is another popular technique. Additionally, attackers may misuse system calls like unshare.

Use a container runtime that supports resource isolation: You can restrict the resources, including CPU, RAM, and disk space, that a container can access by using resource isolation. This can lessen the likelihood that attackers will use excessive resources to get to the host.

Employ a container runtime that is capable of picture scanning:
Vulnerabilities in container images can be found by using image scanning. This can lessen the likelihood that attackers will use weak pictures to get to the host.
Inform users about the dangers associated with Escape to Host: It is important for users to understand the dangers of Escape to Host and how to avoid it. They ought to know how to spot the telltale symptoms of a compromised container.

c. Name of an adversary group who has used the technique: **Hildegard** , ID: **S0601**
d. Name of one mitigation to this technique: **Execution Prevention** , ID: **M1038**
e. Name of one way to detect it: **Kernel**, ID: **DS0008**

6. Pick one technique in **Discovery stage** to review.
   a. Name of the technique: **Account Discovery: Local Account**, ID: **T1087.001**
   b. Brief description of the technique:
   Account discovery is the process of identifying the user accounts that exist on a system. This information can be used by adversaries to plan follow-on attacks, such as password cracking or privilege escalation.

   There are a number of ways to discover local accounts, including:
   Using system utilities: Most operating systems provide utilities that can be used to list local accounts. For example, the net user command can be used to list local accounts on Windows systems.

   Inspecting system files: On Linux systems, local accounts are stored in the /etc/passwd file. macOS systems use the Directory Service to store local account information.

   Using network tools: You can use some network tools, like Nmap, to search for open ports and services on a system. Then, with this data, possible flaws that might be leveraged to find local accounts could be found.
   The data acquired through account discovery might be utilized by adversaries to strategize subsequent attacks. They might utilize the data, for instance, to:

Target particular accounts for password cracking: An adversary can launch password cracking operations against individual accounts provided they are aware of the names of the local accounts.

Identify high-privilege accounts: Adversaries may attempt to increase their level of access by focusing on high-privilege accounts, like administrative accounts.

Use access control lists (ACLs) or disable the system utilities completely to limit access to those that can be used to list local accounts.
Establish robust password guidelines: Robust passwords are harder to figure out, which makes it harder for adversaries to access local accounts without authorization.

c. Name of an adversary group who has used the technique: **HyperStack**, ID: **S0537**
d. Name of one mitigation to this technique**: Operating System Configuration**, ID: **M1028**
e. Name of one way to detect it: **OS API Execution**, ID: **DS0009**

[Exercise II: Splunk with MITRE Att&ck]

7. Review the document "Top Cybersecurity Threat Detections With Splunk and MITRE ATT&CK," included in the lab folder.
8. The document is from Splunk. Please go to splunk.com and login using your splunk account.
9. The document describes different threat tactics mapped to MITRE Att&ck framework.
10. Pick one threat tactic, for example, Reconnaissance (TA0043) on page 6.
11. Briefly describe what this threat tactic is (please use your own words, not copy and paste from the document)

Credential Access (TA0006)

What is Credentials Access ?
Cybercriminals utilize a basic technique called credential access, or Initial Access Broker (IAB), to obtain account credentials, which include usernames and passwords. This strategy is essential to the majority of cyberattacks since it allows for the illegal access to networks, systems, and private information. Once credentials are obtained, attackers can use malware, exploit vulnerabilities, and deploy malware.

Why is Credential Access a Risk?
The threat posed by access is significant, for both organizations and individuals due to key factors;

1. Usage; In todays landscape credentials are used everywhere from email and bank accounts to enterprise systems.
2. Exploitation; Attackers find credential theft to be an effective method of unauthorized access since it often relies on human errors or weaknesses in security protocols.
3. Stealthy Nature; Credential access attacks can be challenging to detect as they involve exploiting user accounts and mimicking behavior.
4. Impactful Consequences; Once attackers gain access to credentials they can cause harm, including data breaches, financial losses and damage to reputation.

Common Techniques Used for Credential Access

Attackers employ techniques to steal credentials. These include;
1. Phishing; Attackers send emails or texts that appear legitimate tricking users into revealing their credentials or clicking on links.
2. Malware; Malicious software like keyloggers and trojans can be installed on victims devices capturing keystrokes (including passwords) or stealing files containing credentials.
3. Credential Stuffing; This technique involves using stolen lists to attempt logins, across different websites or services.
4. Brute Force Attacks;
Brute force attacks consist of attempting combinations of usernames and passwords in order to gain unauthorized access.
5. Social Engineering; Techniques, like pretexting and impersonation can be employed to manipulate users into revealing their login credentials.

Detecting and preventing credential access attacks can be achieved through measures;

1. Multi Factor Authentication (MFA); Implementing MFA requires users to provide verification factors beyond a username and password. This added layer of security makes it more challenging for attackers to gain access.
2. Password Management; Strengthening password policies by enforcing the use of passwords and regular password changes can significantly increase the difficulty for attackers attempting to guess or crack passwords.
3. User. Awareness; Educating users about phishing attempts, social engineering tactics and the importance of maintaining passwords can empower them to recognize and avoid falling victim to these types of attacks.
4. Network Monitoring and Intrusion Detection Systems (IDS); Keeping an eye on network traffic for any activities, such as abnormal login attempts or data exfiltration helps in early identification of potential credential access attacks.
5. Security. Event Management (SIEM); Utilizing SIEM solutions allows for data collection, analysis and correlation from sources. This enables organizations to gain insights into security threats including attempts at accessing credentials.

In conclusion preventing access to credentials is crucial in mitigating cybersecurity risks as it forms a step, in many cyberattacks. Organizations can greatly decrease their

vulnerability to access attacks by implementing security measures educating users and consistently monitoring for any suspicious activity.

12. Briefly describe an example of this threat tactic. (In the grey box on the lower right "How to detect and respond with Splunk", the Splunk Analytics Story is an example of this tactic.)

    Example:
    A company's employees receive a phishing email from an attacker pretending to be the company's IT department. The email claims that their passwords have expired and provides a link for them to click to reset them. However, this link leads them to a website designed to look like the company's password reset page. Unaware of the deception the employees enter their login credentials, on this website unknowingly handing them over to the attacker. With these stolen credentials the attacker can illicitly access the employees' accounts. Enter the company's network.

13. From this review, identify one advantage that MITRE Att&ck brings for SIEM or SAOR tools like Splunk.

    To improve security planning and response it is crucial for organizations to have an understanding of the tactics, techniques and procedures (TTPs) employed by adversaries. This knowledge enables organizations to enhance their security defenses and effectively respond to any incidents that may occur.

[Optional: Exercise III: MITRE Caldera]

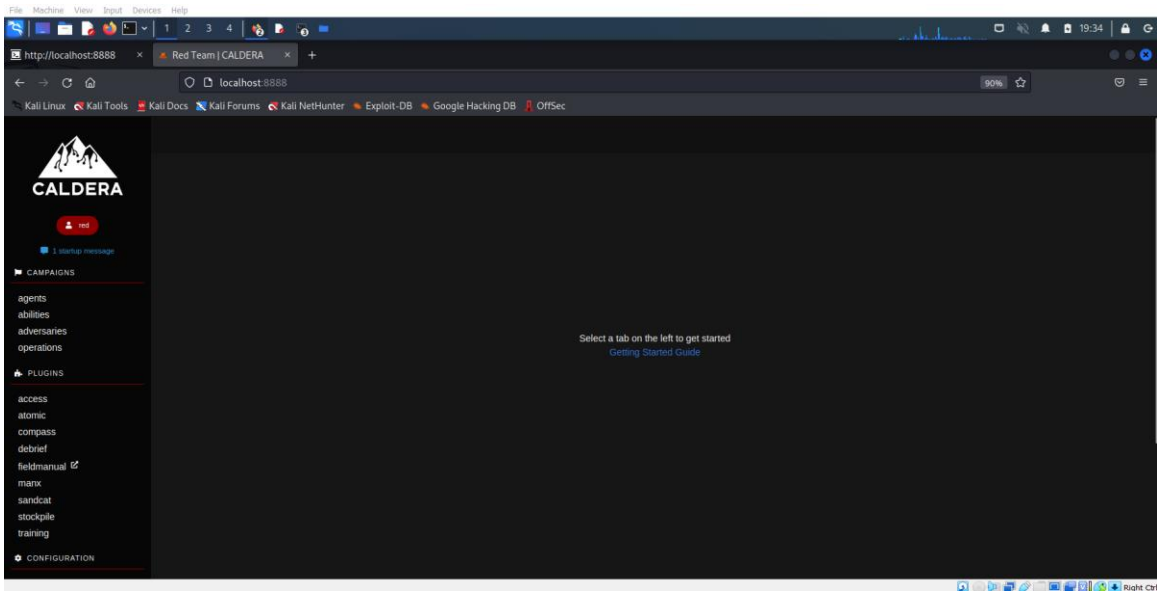
1. Caldera is an open-source attack emulation tool. It is an automation tool written in Python. It runs on either Linux or Mac.
2. The installation and documentation of the tool is available on https://caldera.readthedocs.io/en/latest/.
3. To install it, you will have to clone the github repository, install all dependency requirements and then run the python server program.

   git clone https://github.com/mitre/caldera.git --recursive --branch 4.0.0

   cd caldera

   sudo pip3 install -r requirements.txt

   python3 server.py –insecure
4. The GUI runs on http://localhost:8888. The username and password can be found in the conf/local.yml file (this file is generated on server start).
5. Paste a screenshot of the GUI after logging in.



6. Run one of the example scenarios from https://caldera.readthedocs.io/en/latest/Getting-started.html.

7. Show evidence that your scenario experiment works.


```
{
  "name": "vaibhavlinux",
  "host_group": [
    {
      "created": "2023-11-24T00:39:50Z",
      "server": "http://0.0.0.0:8888",
      "available_contacts": [
        "HTTP"
      ],
      "paw": "rcxyjj",
      "pid": 8969,
      "watchdog": 0,
      "proxy_receivers": {},
      "exe_name": "splunkd",
      "last_seen": "2023-11-24T00:52:31Z",
      "username": "kali",
      "upstream_dest": "http://0.0.0.0:8888",
      "location": "/home/kali/caldera/splunkd",
      "platform": "linux",
      "executors": [
        "proc",
        "sh"
      ],
      "privilege": "User",
      "host": "kali",
      "sleep_min": 30,
      "proxy_chain": [],
      "architecture": "amd64",
      "trusted": true,
      "origin_link_id": "",
      "group": "red",
      "deadman_enabled": true,
      "pending_contact": "HTTP",
      "sleep_max": 60,
      "contact": "HTTP",
      "links": [
        {
```

```
{
  "command": "PiAkSE9NRS8uYmFzaF9oaXN0b3J5ICYmIHVuc2V0IEhJU1RGSUxF",
  "relationships": [],
  "paw": "rcxyjj",
  "status": 0,
  "id": "e8bccfd3-361c-4002-a53a-6960c0132e39",
  "pid": "8977",
  "agent_reported_time": "2023-11-24T00:39:50Z",
  "ability": {
    "requirements": [],
    "tactic": "defense-evasion",
    "additional_info": {},
    "plugin": "stockpile",
    "description": "Stop terminal from logging history",
    "delete_payload": true,
    "access": {},
    "ability_id": "43b3754c-def4-4699-a673-1d85648fda6a",
    "singleton": false,
    "buckets": [
      "defense-evasion"
    ],
    "executors": [
      {
        "code": null,
        "payloads": [],
        "command": "> $HOME/.bash_history && unset HISTFILE",
        "language": null,
        "build_target": null,
        "uploads": [],
        "parsers": [],
        "cleanup": [],
        "variations": [],
        "timeout": 60,
        "platform": "darwin",
        "name": "sh",
```