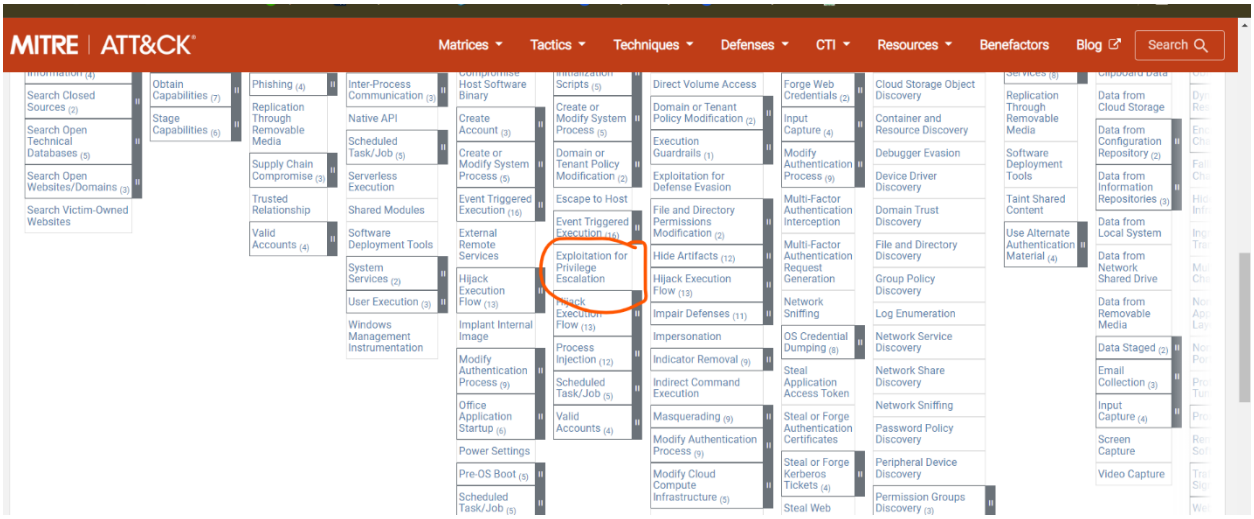
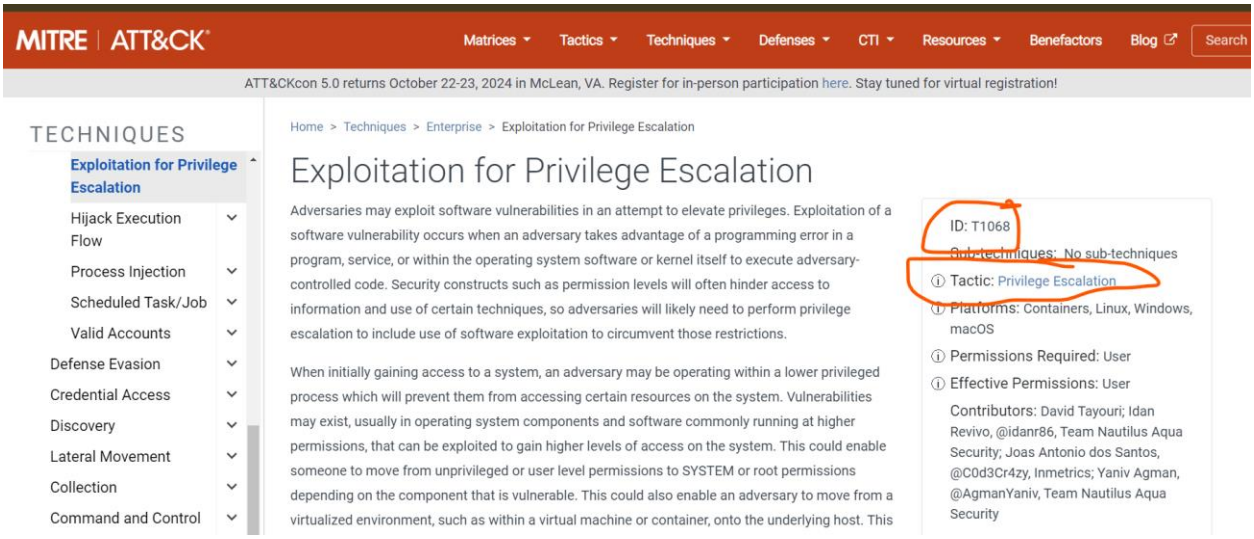
Part 1.

a. Exploitation for Privilege Escalation

   Technique ID: T1068

   Tactic: privilege escalations





Mitigation Strategies

1. Patching: To keep the system up to date and its software with latest new patches to fix known vulnerabilities that attacker can exploit.
2. User Account Control: Make efforts to restrict and control excessive privileges on assigned accounts to avert any likely sequences of elevation.

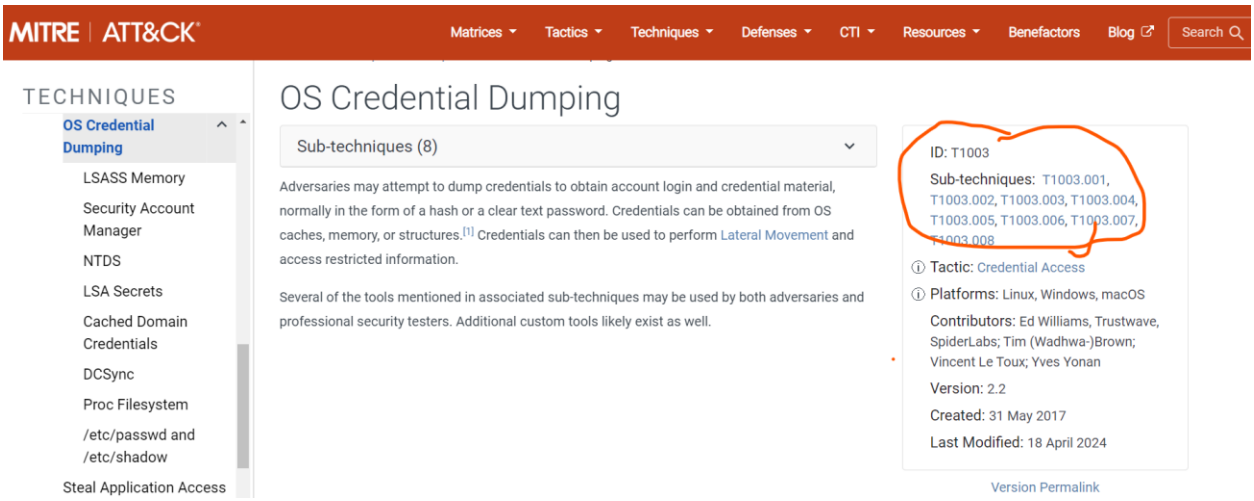
Detection Strategies

1. Analyze log Files: Examine and dissect some certain types of the logs of the system in search of some failed events regarding elevation of privileges. Like we mostly snort , Splunk tools for this. Or review the system logs to detect failed attempts at privilege escalations.
2. Monitor privilege changes: Safeguard oversight and management of any security monitoring processes and any ‘anomalies’ regarding singular or mass privilege elevation.

b. Credential Dumping

Technique ID: T1003

Tactic: Credential Access



Mitigation strategies:

1. Use Least Privilege Role: Least Privilege Principle (User account should have only the access that they need to perform their specific role, minimizing chance of credential exposure)
2. Use credential Guard: Turn on the Windows Credential Guard to stop credential from being stored in-memory.

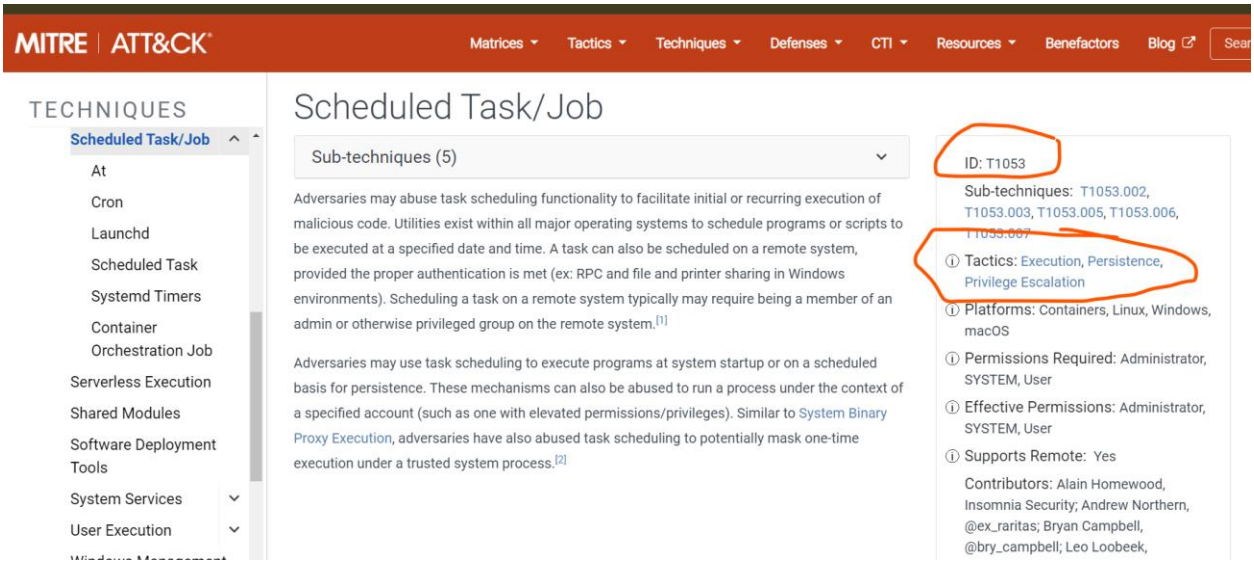
Detection strategies:

1. Regularly Audit access to the processes of LSASS: Examine the LSASS Process Access , Used frequently for credential dumping, look out for processes accessing the LSASS process without permission.
2. Monitoring Memory Access: Detecting processes reading memory from where you store your credentials using tools such as Sysmon.

c. Scheduled Task/Job

Technique ID: T1053

Tactic ID: Execution , Persistence , privilege Escalations



Mitigation Strategies:

1. Make clean, Audit scheduled tasks: Allows admins to review their scheduled tasks for unexpected changes or malicious additions.
2. Restrict task schedular usage: Allow only system administrators to create or change scheduled tasks.

Detections Strategies:

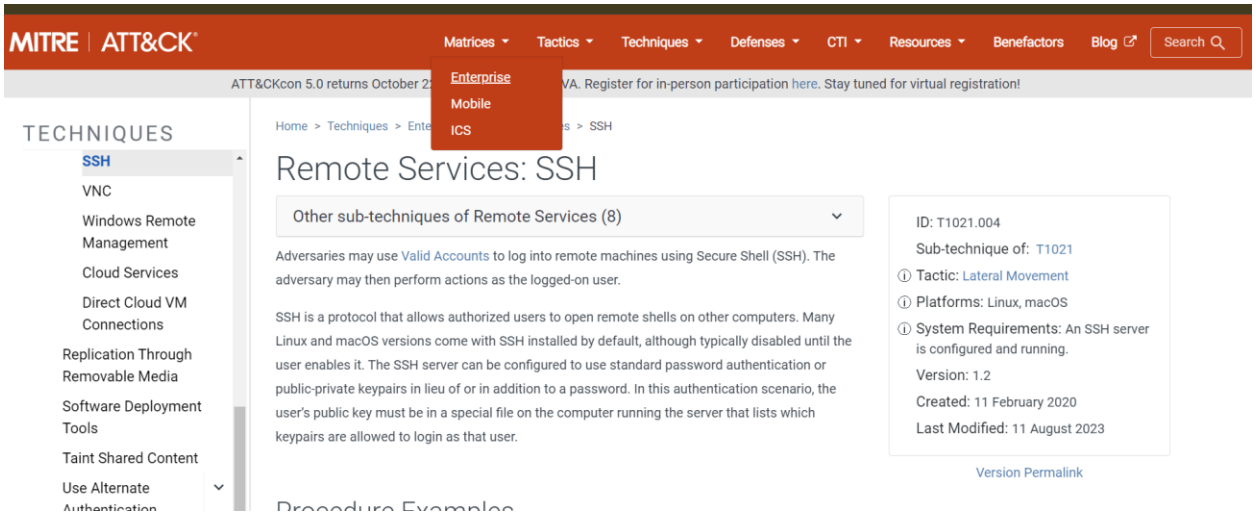
1. Analyze Logs and its Modifications: Pay attention to any changes in the scheduled tasks that may indicate a bad activity.

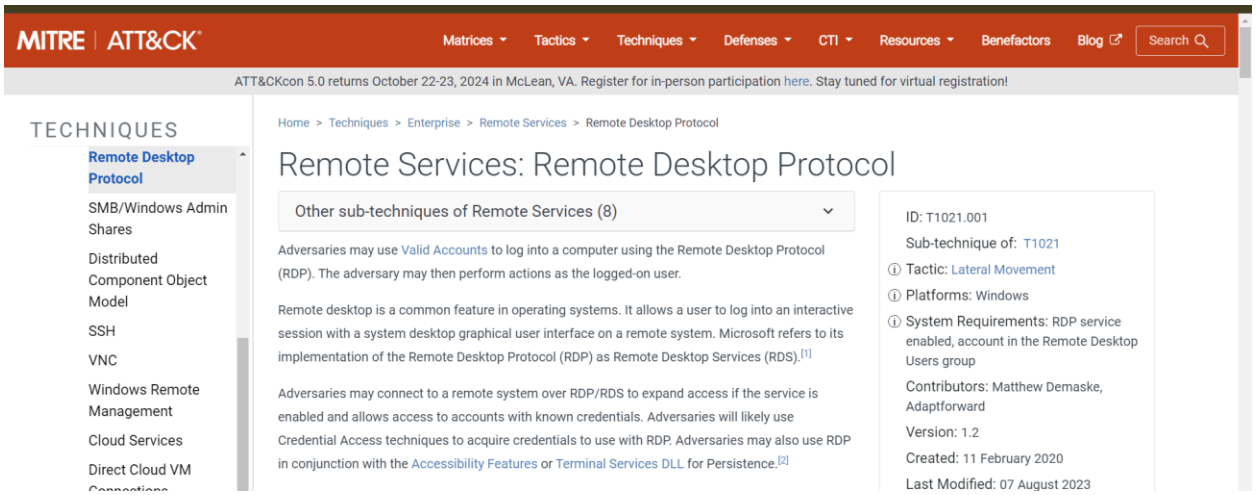
2. Monitoring task Creations: use the windows event logs for generating new scheduled tasks (Event ID 4698)

d. Remote Services (RDP, SSH,etc)

Technique ID: T1021

Tactic: Lateral Movement





Mitigation Strategies:

1. Disable unused remote services: Disable unnecessary service like RDP or SSH as it reduces potential attack surfaces.
2. Enable MFA: Add a second layer of security to your remote services; single sign on + another factor is less guessable than just the password.

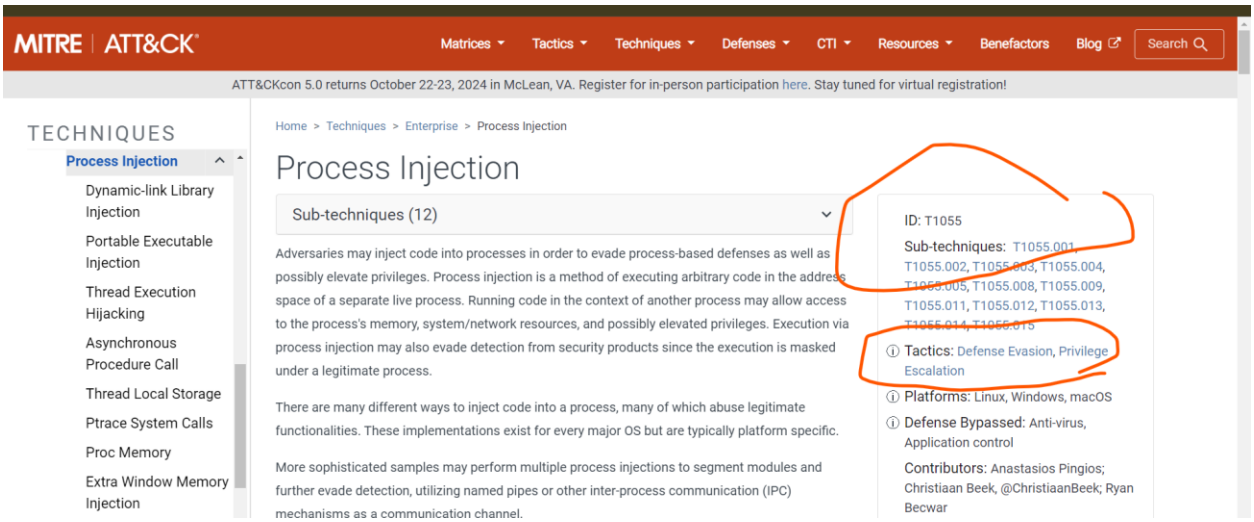
Detection Strategies:

1. Log you remote login attempts: Monitor unsuccessful and out-of-the-norm successful remote logins from unusual IPs.
2. Monitor network traffic: Network monitoring tools are used to discover rogue remote service connections.

E. Process Injections:

Technique ID: T1055

Tactic: Defense Evasion, Privilege Escalaltions



Mitigation Strategies

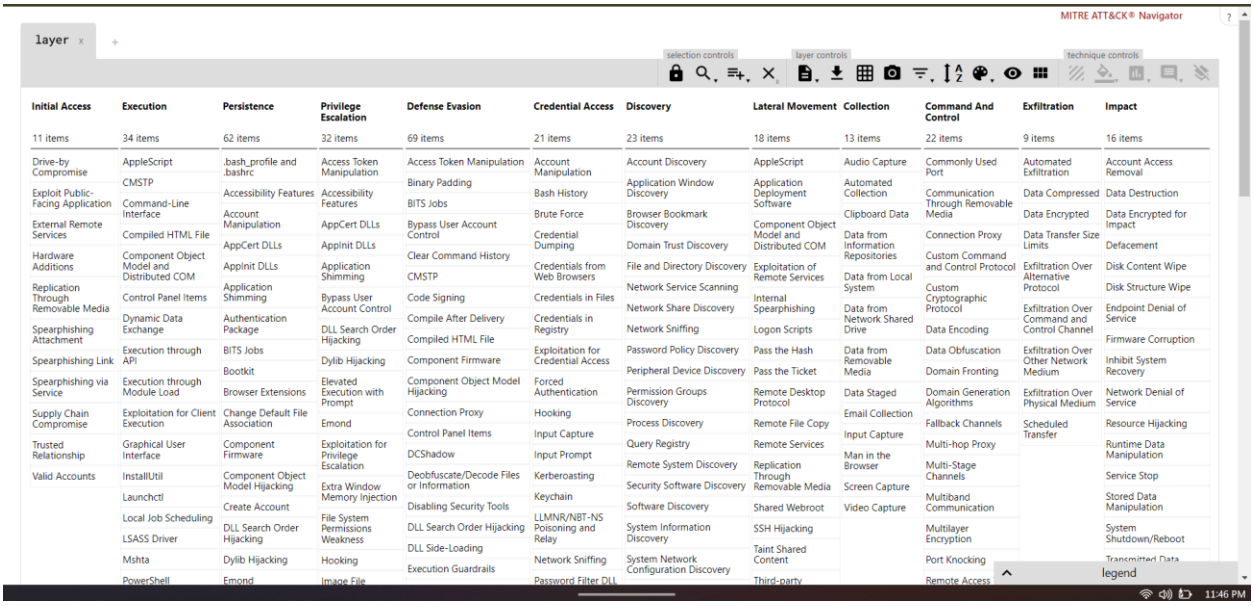
1. Use Data Execution Prevention (DEP): DEP is a security feature that prevents the execution of code from system memory locations.
2. Enforce code signing: Make sure only signed code is allowed to run in the memory, this way you lessen the threat of process injection.

Detection Strategies

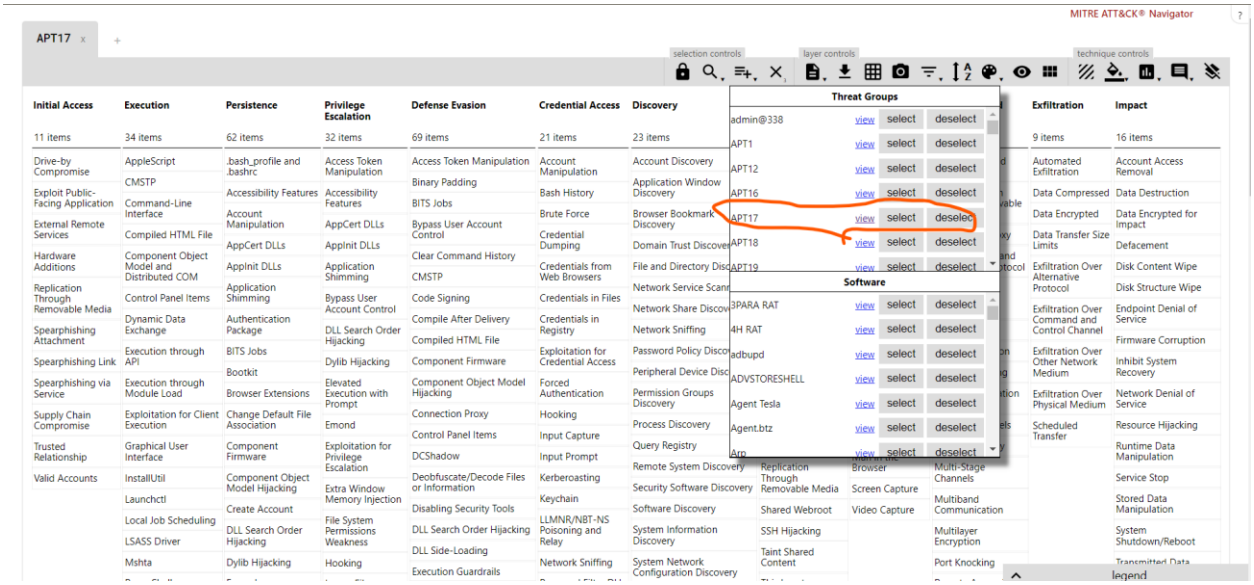
1. Monitor api calls : Look for suspicious API called such as Write Process Memory, Create Remote Thread that are used in process injection.
2. Tracking memory access patterns: to identify abnormal read/write operations on the processes that could have been related with process injection techniques.

Part 2:

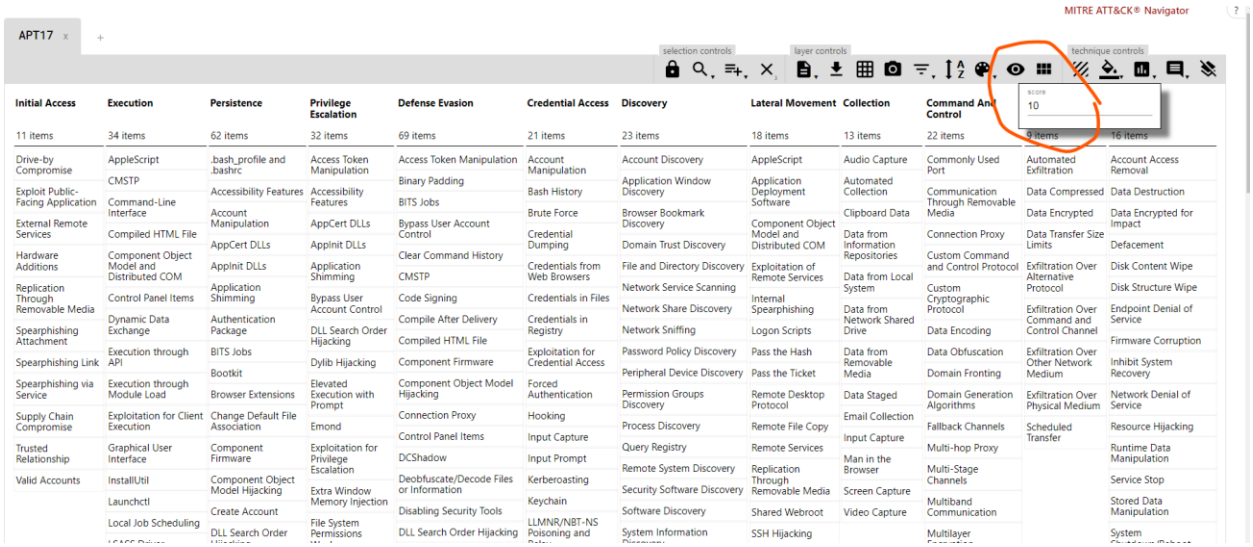
Navigator to https://mitre-attack.github.io/attack-navigator/v2/enterprise/



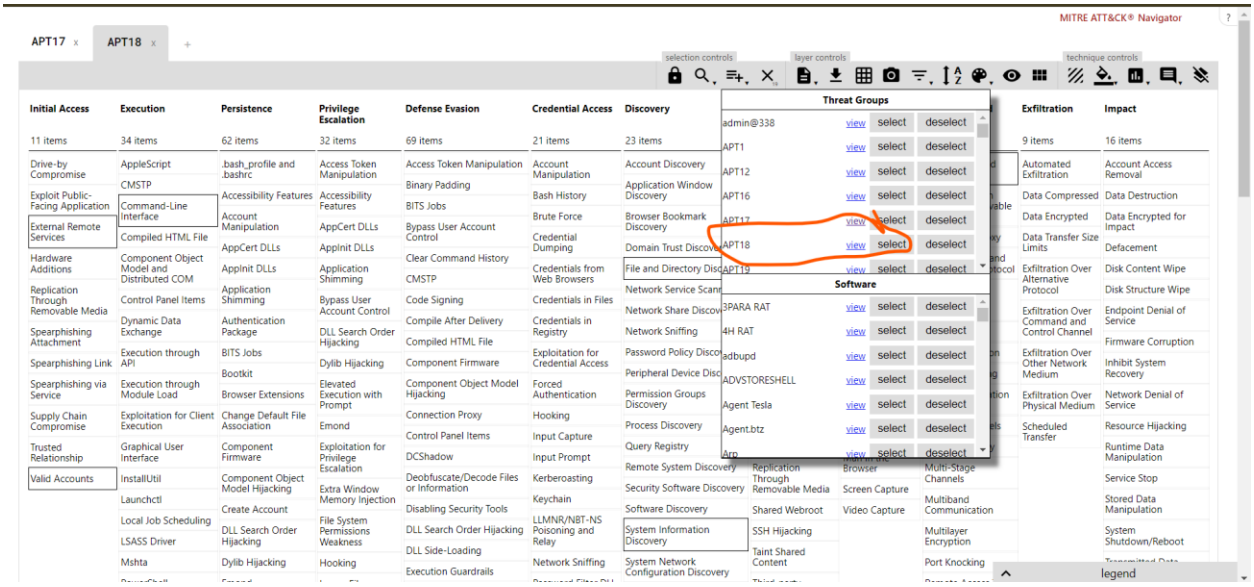
a) Create an APT17 layer



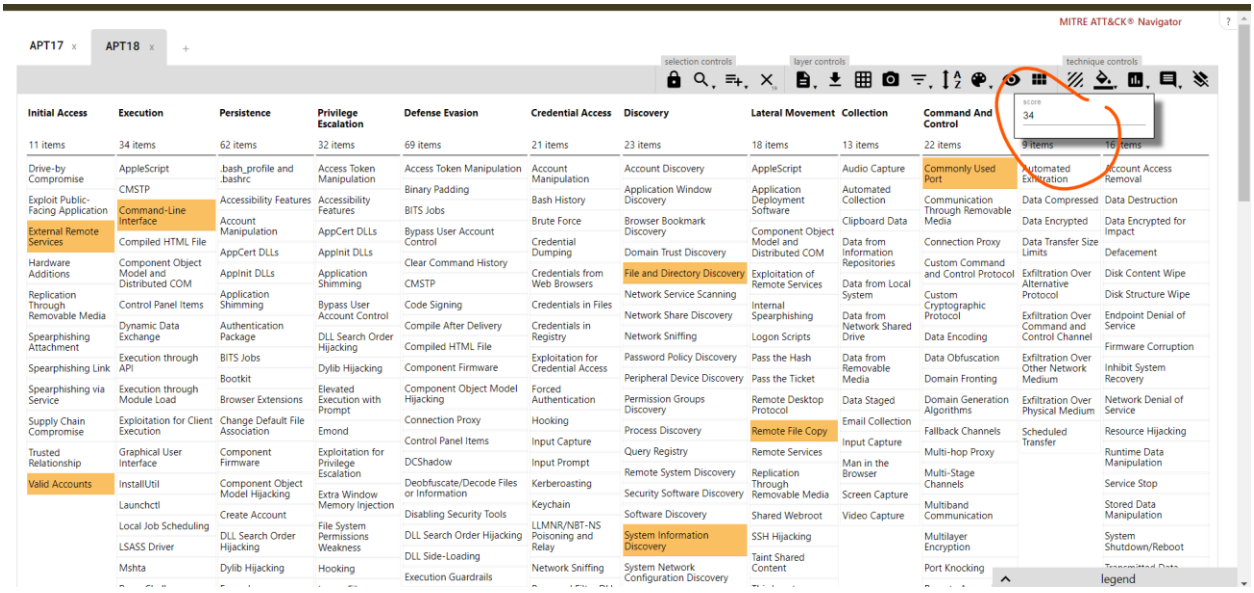
and assign the first two numbers of our student ID which is 10 as a score.



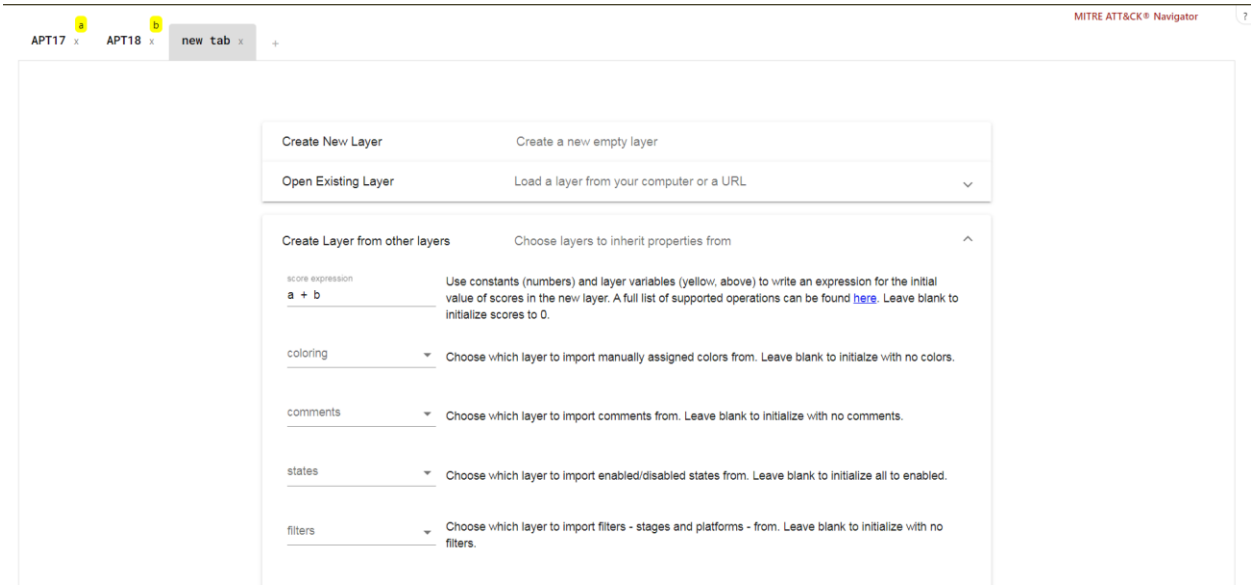
b. Create a second layer of APT18

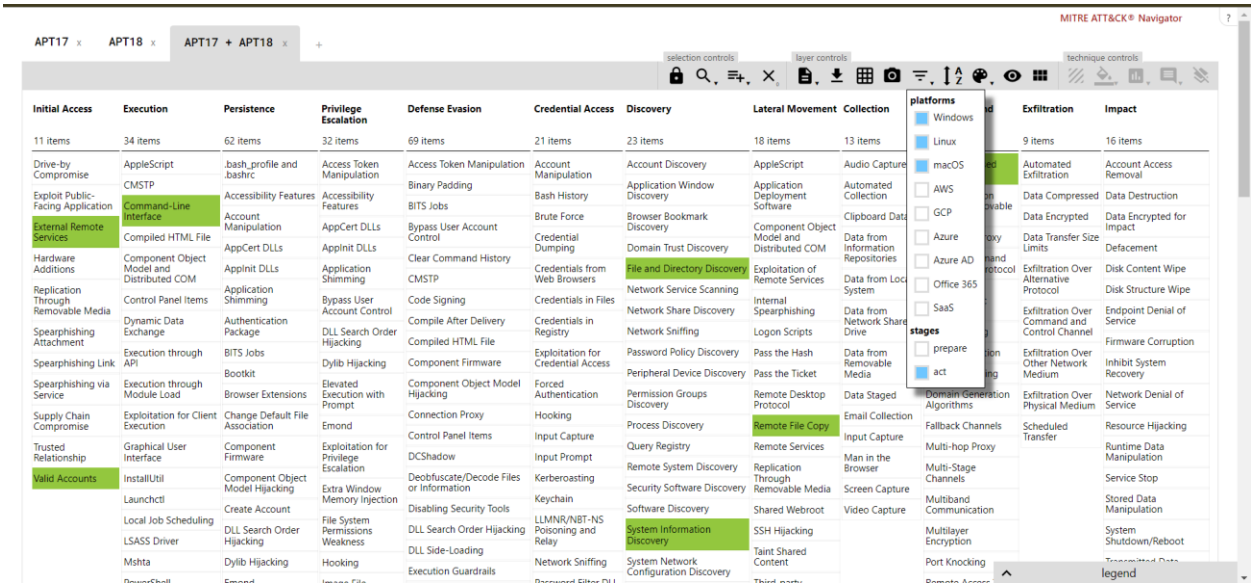


and assign the last two numbers of our student ID as a score which is 34.

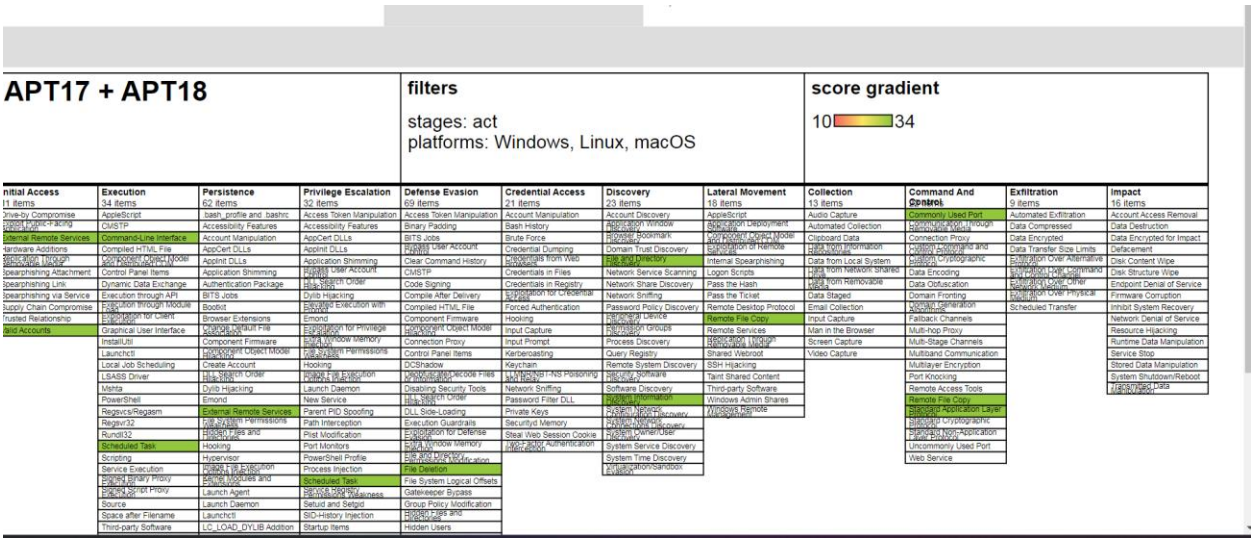


c. Combine the two using “Create Layer from other layers” using the expression “a + b”





d. Export the layer in the image format and add it to your report.



e. Provide the list of the techniques that overlap between the two groups

From the combined layer the technique with a score up to 10+34=44 show or represents the overlap

f. Explain how this information can be useful to the SOC team.

1. Threat Analysis:
   Understanding which tactics and techniques are shared between the diverse set of APT groups works to highlight those areas of interest most often targeted by threat actors, This will helps in striking the balance between which techniques to monitor and defend.

2. Improving Detection:
   When some of the techniques utilized by multiple groups, concentrating on them can enhance the overall detection capabilities in Security Operations Center (SOC). It makes possible to create more specify detection rules and alerts.

3. Enhancing Defense Strategies:
   The SOC team can then begin creating or updating defenses to prevent these techniques, as well as find other cases that overlap in usage and work together with Security Engineering to develop a specific coverage for the technique. Offensive: This can range from patching to bolstering configuration settings or increasing user education.

4. Resource Allocation:
   There are numerous techniques being used, and knowing what common practices will assist in spending resources properly. As an example, if a method is religiously practiced.

Part 3:

a. Navigate to the OWASP Top 10 (https://owasp.org/Top10).

b. At the time of this writing, the draft of latest list of Top 10 was published in 2021. Review the Top 10 categories. Pick 3 categories out of the Top 10. In the table below, list your chosen categories. Then briefly describe the category and some of the ways to prevent attacks in the category.

Top 10 Category	Description	Prevention
A02: Cryptographic Failure	Cryptographic shortcomings, formerly named "Sensitive Data Exposure," are works that involve abuse of, or erroneous implementation or processing of, any algorithm, protocol, or mechanism regarding the conversion of one form of information into a secure format in which the disclosure of information is not possible. Such a category incorporates weak encryption, poor key handling and other issues where data is not protected sufficiently.	Use Strong Encryption: Strong modern criteria and protocols should be used for the cryptographic treatment of sensitive data in active and passive states. Key Management: Best practices in key management should be employed including safe storage, key rotation, and key life cycle management Secure Protocols: Secure transport protocols such as TLS should be utilized and updated timely and correctly.
A05: Security Misconfigurations	Security Misconfiguration: Security misconfigurations is a risk that may arise as a result of improper structures within an application, server, or other system components. Misconfigurations means improper design or inappropriate security lets, excess access control, administration interfaces, and vulnerability in the security settings. This will in most cases result to loss of data, access to personal and sensitive information or other critical vulnerabilities.	Start with the most secure configuration possible: Address that default settings and credentials can give rise to certain vulnerabilities. Disable Unnecessary Features: Disable all of the features and services that cannot be used for the scope of the application or the environment to reduce the attack vectors. Apply security patches
A03: Injection	Injection flaws take place when a malicious user is allowed to inject ill-sourced data into a certain system which is then run by the application or server. This is common when user information is accepted without sufficient checking or cleansing. Types of injection attacks include SQL Injection, Command Injection and XML Injection, among others. Each of these can result in data leakage, unauthorized information or even total control of the system.	Prepared Statements: Use parameterized queries or prepared statements while dealing with database. This way it is ensured that input data is treated as a normal data rather than as an executable code. Sanitize Input: All data provided by the user should be normalized and cleansed appropriately or else none of these formats should be accepted.

Step 2: OWASP Community Pages

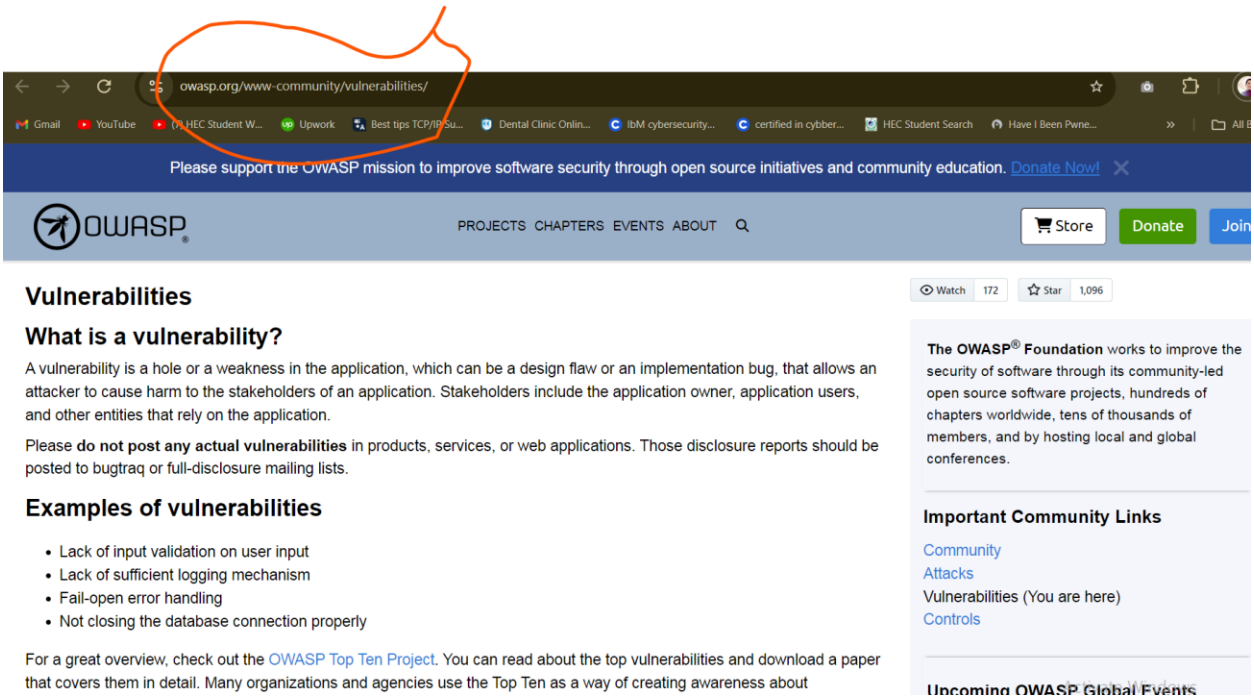
The OWASP Community Pages allows security-related contributions from the community. In this part, you will review the Vulnerabilities pages to investigate the attack techniques reported by the contributors.

a. Navigate to the OWASP Community Page for Vulnerabilities. (https://owasp.org/www-community/vulnerabilities/).

OWASP defines a vulnerability as flaw in the application that a threat actor can exploit.

Question:

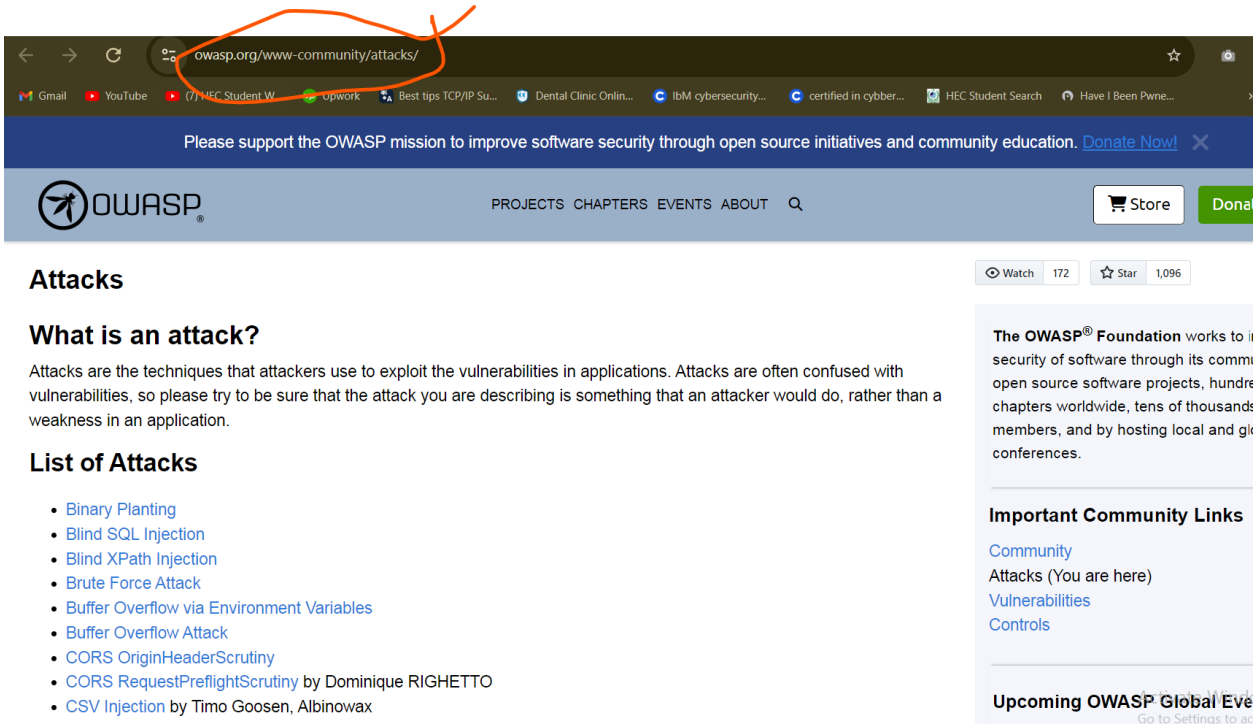
Review the List of Vulnerabilities and pick 3. In the table below, list your chosen vulnerabilities. Briefly describe the vulnerability and some of the ways to prevent exploitation.



Vulnerabilities	Descriptions	Preventions
Insecure Deserializations	When someone exploit flaws in the deserialization it will leading to the remote code executions or some other malicious attack	Avoiding from Untrusted data, use secure data serializations libraries that are safe and validate all input data.
Broken Authentications	Bugs or flaws in authentications mechanisms allow attackers to compromise user accounts or perform unwanted or unauthorized actions.	Implement to 2FA authentications and using secure passwords storage processes and limit login attempts.
Misconfigured Security Headers	Not correct security headers can expose applications to attacks such as xss , and clickjacking.	Implement security header proper loke CSP – content security policy, X-frame-options

b. Navigate to the OWASP Community Page for Attacks. (https://owasp.org/www-community/attacks).

According to OWASP, an attack is a technique used to exploit application vulnerabilities. Review the **List of Attacks** and pick 3. In the table below, list your chosen attacks. Then briefly describe the attack and some of the ways to prevent it.



Attacks	Descriptions	Preventions
CSRF-Cross-Site Request Forgery	The attackers tricks a user into performing actions on a web chrome applications where they are authenticated without the user or victims consent.	Using anti-token of the CSRF , validate referrer headers, and implement user expirations controls.
SQL Injections	In this the attacker Injects some malicious SQL payload into an application's query , it will leading to unauthorized access to the database.	Validate and sanitize user inputs, also use parametrize queries.
Cross site scripting	Some malicious scripts are injected into the input field in the trusted websites which will executed in a user' browsers.	Take output in encoding use input validations, employee content security policy and avoid includes not trusted data in HTML.