**Tactics to Technical Impacts Mapping Rationale**

In order to connect mitre’s 11 attack tactics to Common Weakness Enumerations (CWEs), it is necessary to find a common link between the consequences of CWEs and what the tactics are attempting to exploit in a software system. This mapping explores which attack tactics can be used to exploit the 8 technical impacts caused by CWEs detectable using static analysis, which includes 18 different CWEs. Traversing between attack tactics and technical impacts provides a way to connect the tactics to the CWEs.

This mapping was created by analyzing mitre’s definition of each attack tactic and which techniques fall under each tactic (https://attack.mitre.org/wiki/Main\_Page) against examples of CWEs that result in each of the 8 technical impacts (<https://cwe.mitre.org/community/swa/detection_methods.html)>. CWEs list a number of code examples that result in the weakness arising as well as their scope, such as Access Control, Authentication, or other aspects of Security. Using this evidence along with intuition enabled an initial mapping to be drawn between the tactics and technical impacts.

Future work could investigate how these mappings may change or develop over time as more examples of attack tactics are witnessed in conjunction with exploited CWEs. For instance, it may be found that the tactic Credential Access is often used to exploit CWEs causing the technical impact Bypass Protection Mechanism, and in this case, an edge should be added in the mapping to link the two vertices Credential Access and Bypass Protection Mechanism.

Below is the list of technical impacts, broken down by which attack tactics they are mapped to in the Tactics to Technical Impacts association graph. For each attack tactic, there is a brief explanation describing the connection between the two vertices. Note that Discovery is not listed under any technical impact because it has no *direct* means of exploiting common weaknesses in software; rather, many tactics depend on Discovery, and therefore its place is in the dependency graph.

1. **Execute unauthorized code or commands**
   1. *Execution* – By mitre’s definition, “Execution” includes techniques that result in execution of adversary-controlled code, which directly corresponds to this technical impact.
2. **DoS: unreliable execution**
   1. *Execution* – In order to execute their adversary-controlled code, attackers could use any number of “Execution” techniques, including Command-Line Interface and PowerShell, and these techniques could be used to exploit buffer overflows or other weaknesses in order to cause Denial of Service.
3. **DoS: resource consumption**
   1. *Execution* – Many of the “Execution” techniques could be used with the intention of resource consumption. For instance, the ‘Execution through API’ technique can “start processes with proper path and argument parameters”, and these processes may consume enough of the system’s resources to result in Denial of Service.
   2. *Lateral Movement* – By mitre’s definition, “Lateral Movement” can be used for remote execution of tools across a network, and this tactic could be employed to crash an entire system. Moving across a network often involves the exploitation of sockets programming (see CWE-400), which can cause Denial of Service from resource consumption.
4. **Modify data**
   1. *Exfiltration* – By mitre’s definition, “Exfiltration” allows an attacker to remove files and information from a target network. This removal of information is one way an adversary can modify data through a software vulnerability.
5. **Read data**
   1. *Collection* – Techniques categorized under “Collection” are used to identify and gather information from a target network. The direct goal of this tactic is to read sensitive data, which can be achieved by exploiting CWEs such as CWE-134: Use of Externally Controlled Format String to view the contents on the stack.
6. **Bypass protection mechanism**
   1. *Initial Access* – This attack tactic is necessary for gaining an initial foothold into the target system as a prerequisite to “Execution”. Several of its techniques involve Spearphishing, which can be used to bypass protection mechanisms through URL redirection (see CWE-601).
   2. *Persistence* – “Persistence” allows an adversary to maintain their presence on a system even in the face of system restarts or loss of credentials, according to mitre; therefore, this tactic is used to bypass protection mechanisms that would normally kick an adversary out of the system.
   3. *Defense Evasion* – By mitre’s definition, “Defense Evasion” encompasses any actions used to subvert a particular defense or mitigation, ie. Bypass User Account Control. The direct goal of this tactic is to bypass protection mechanisms, and thus is an important aspect of attacks like XSS and XSRF.
7. **Gain privileges/assume identity**
   1. *Privilege Escalation* – This attack tactic is composed of techniques that allow the adversary to obtain a higher level of permissions than they would otherwise be given. The direct goal of “Privilege Escalation” is to gain privileges in order to exploit CWEs within the scope of authorization.
   2. *Credential Access* – By mitre’s definition, “Credential Access” allows the adversary to assume the identity of a user or an admin with their legitimate credentials. Taking on this identity gives the attacker the permissions to exploit CWEs relating to authorization.
8. **Hide activities**
   1. *Defense Evasion* – Many of the techniques associated with “Defense Evasion” serve the direct purpose of hiding activities that the adversary wishes to keep concealed; these techniques include Hidden Files, Hidden Users, Hidden Windows, and more in order to evade detection.
   2. *Command and Control* – By mitre’s definition, “Command and Control” consists of techniques making it difficult to distinguish between an attack and benign traffic. Some such techniques are Data Encoding, Commonly Unused Port, and Custom Cryptographic Protocol, which all serve to hide the attacker’s activity as they transfer information.

**Tactic Dependencies Mapping Rationale**

Due to the nature of mitre’s attack tactics, a simple bipartite graph relating tactics to technical impacts does not show the entire picture. By examining the relationships between tactics, a clearer understanding of why and how tactics are used to exploit CWEs can be drawn. Many of the tactics depend on other tactics before they can be employed by an attacker; for instance, Execution depends on Initial Access so that the attacker can gain an initial foothold into the target network before executing their adversary-controlled code or commands.

Below is a breakdown of tactics that are dependent on other tactics before they can be successfully employed by an attacker. Some dependencies are purely contextual, as in the case of Privilege Escalation’s dependency on Lateral Movement – if the attacker can gain privileges in the initial system, then there is no need to move across the network in order to employ this tactic. Note that Defense Evasion is not dependent on any other tactics nor do other tactics depend on it per se; however, this tactic is often used *in parallel* withother tactics. The Tactic Dependencies graph does not illustrate parallel usage of tactics. Future work could investigate which tactics are most often used in conjunction with one another outside of the below dependencies.

1. **Persistence**
   1. *Credential Access* – The “Persistence” tactic is useful for attackers wishing to maintain their presence in the target network even in the face of system restarts or loss of credentials; in order to latch into the system, the attacker first needs to employ “Credential Access”. For example, the Browser Extensions technique first steals credentials, which then allow the attacker to implement an installer that gives him persistence.
2. **Execution**
   1. *Initial Access* – According to mitre, “Initial Access” is often used as the attack tactic preceding “Execution” so that the attacker can gain an initial foothold into the target system before employing their adversary-controlled code and commands.
   2. *Lateral Movement* – In cases where the attacker cannot successfully compromise the system from their initial gateway into the target network, the adversary will move across the network by using the “Lateral Movement” tactic prior to executing their code and commands.
3. **Privilege Escalation**
   1. *Lateral Movement* – There are cases in which an attacker cannot gain privileges within the entry point of the target system, and will therefore need to employ “Lateral Movement” before “Privilege Escalation”. The Exploitation of Remote Services technique is one such example, as well as SSH Hijacking.
4. **Exfiltration**
   1. *Collection* – In order to implement “Exfiltration”, an attacker will often need to be able to first gather the sensitive data in the system through “Collection” before it can be removed from the system.
5. **Collection**
   1. *Discovery* – The use of the “Discovery” tactic precedes “Collection” because an adversary must gain an understanding of the system in order to figure out where he should gather and collect the sensitive data.
6. **Command and Control**
   1. *Discovery* – The “Command and Control” tactic serves to hide communication of information from the target network back to the attacker. For a successful exploit, the adversary should first employ the “Discovery” tactic so that he understands the system well enough to avoid detection. For instance, the Port Knocking technique relies on the attacker knowing which ports are currently open.

For more information on Attack Tactics and Techniques, see <https://attack.mitre.org/wiki/Main_Page>.