# Information Security

Dr./ Ahmed Mohamed Rabie

# Chapter 4

# Threat Modeling

**Threat modeling** is the security process where potential threats are identified, categorized, and analyzed. Threat modeling can be performed as a proactive measure during design and development or as a reactive measure once a product has been deployed. In either case, the process identifies the potential harm, the probability of occurrence, the priority of concern, and the means to eradicate or reduce the threat.

Threat modeling isn't meant to be a single event. Instead it's common for an organization to begin threat modeling early in the design process of a system and continue throughout its life cycle. For example, Microsoft uses a Security Development Lifecycle (SDL) process to consider and implement security at each stage of a product's development. This supports the motto of "Secure by Design, Secure by Default, Secure in Deployment and Communication" (also known as SD3+C). It has two goals in mind with this process:

- To reduce the number of security-related design and coding defects .
- To reduce the severity of any remaining defects.

In other words, it attempts to reduce vulnerabilities and reduce the impact of any vulnerabilities that remain. The overall result is reduced risk. A **proactive approach** to threat modeling takes place during early stages of systems development, specifically during initial design and specifications establishment. This type of threat modeling is also known as a defensive approach. This method is based on predicting threats and designing in specific defenses during the coding and crafting process, rather than relying on post deployment updates and patches.

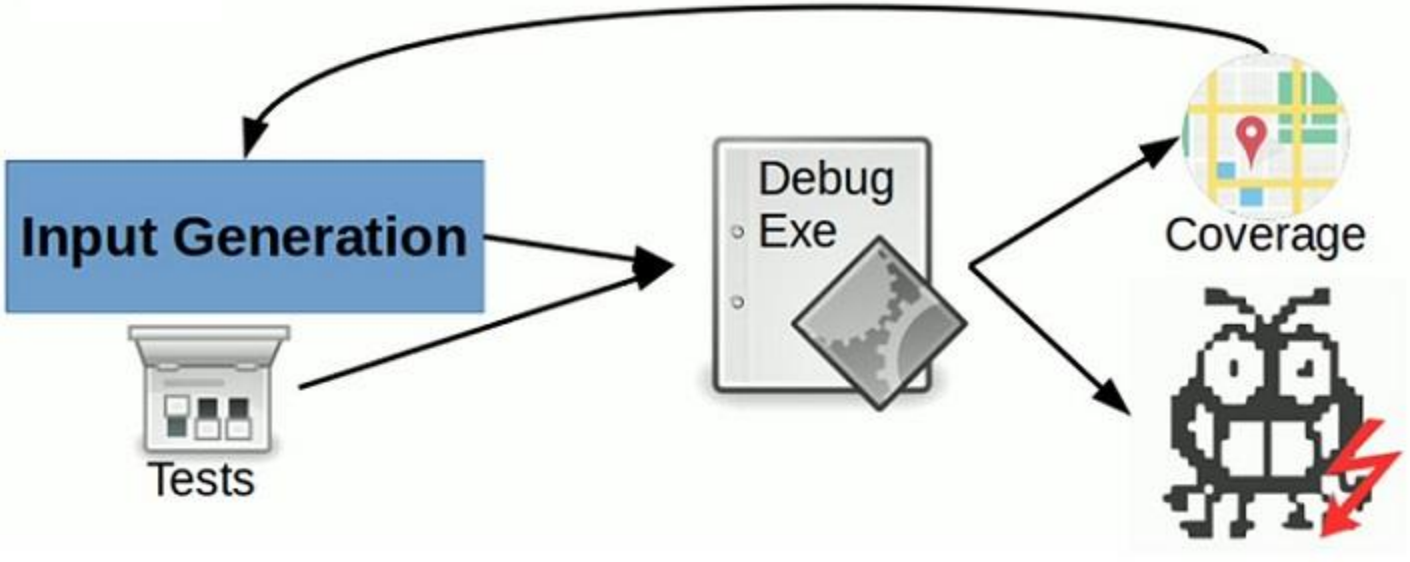
In most cases, integrated security solutions are more cost effective and more successful than those shoehorned in later. Unfortunately, not all threats can be predicted during the design phase, so reactive approach threat modeling is still needed to address unforeseen issues.

A **reactive approach** to threat modeling takes place after a product has been created and deployed. This deployment could be in a test or laboratory environment or to the general marketplace. This type of threat modeling is also known as the adversarial approach. This technique of threat modeling is the core concept behind ethical hacking, penetration testing, source code review, and fuzz testing. Although these processes are often useful in finding flaws and threats that need to be addressed, they unfortunately result in additional effort in coding to add in new countermeasures.

Returning back to the design phase might produce better products in the long run, but starting over from scratch is massively expensive and causes significant time delays to product release. Thus, the shortcut is to craft updates or patches to be added to the product after deployment. This results in less effective security improvements (over-proactive threat modeling) at the cost of potentially reducing functionality and user-friendliness.

**Fuzz testing** is a specialized dynamic testing technique that provides many different types of input to software to stress its limits and find previously undetected flaws. Fuzz testing software supplies invalid input to the software, either randomly generated or specially crafted to trigger known software vulnerabilities. The fuzz tester then monitors the performance of the application, watching for software crashes, buffer overflows, or other undesirable and/or unpredictable outcomes.

## FUZZ Testing



## Threat Modeling Methodologies

When performing threat modeling, there are **multiple methodologies you can use**. The right model for your needs depends on what types of threats you are trying to model and for what purpose. There's an almost infinite possibility of threats, so it's important to use a structured approach to accurately identify relevant threats.

For example, some organizations use one or more of the following three approaches:

**1- Assets Centric:** This method uses asset valuation results and attempts to identify threats to the valuable assets. For example, a specific asset can be evaluated to determine if it is susceptible to an attack. If the asset hosts data, access controls can be evaluated to identify threats that can bypass authentication or authorization mechanisms.

**2- Attackers Centric**: Some organizations are able to identify potential attackers and can identify the threats they represent based on the attacker’s goals. For example, a government is often able to identify potential attackers and recognize what the attackers want to achieve. They can then use this knowledge to identify and protect their relevant assets. A challenge with this approach is that new attackers can appear that weren’t previously considered a threat.

**3- Software Centric:** If an organization develops software, it can consider potential threats against the software. Although organizations didn't commonly develop their own software years ago, it's common to do so today. Specifically, most organizations have a web presence, and many create their own web pages. Fancy web pages drive more traffic, but they also require more sophisticated programming and present additional threats.

If the threat is identified as an attacker (as opposed to a natural threat), threat modeling attempts to identify what the attacker may be trying to accomplish. Some attackers may want to disable a system, whereas other attackers may want to steal data. Once such threats are identified, they are categorized based on their goals or motivations. Additionally, it's common to pair threats with vulnerabilities to identify threats that can exploit vulnerabilities and represent significant risks to the organization. An ultimate goal of threat modeling is to prioritize the potential threats against an organization's valuable assets.

## Threat Modeling Methods

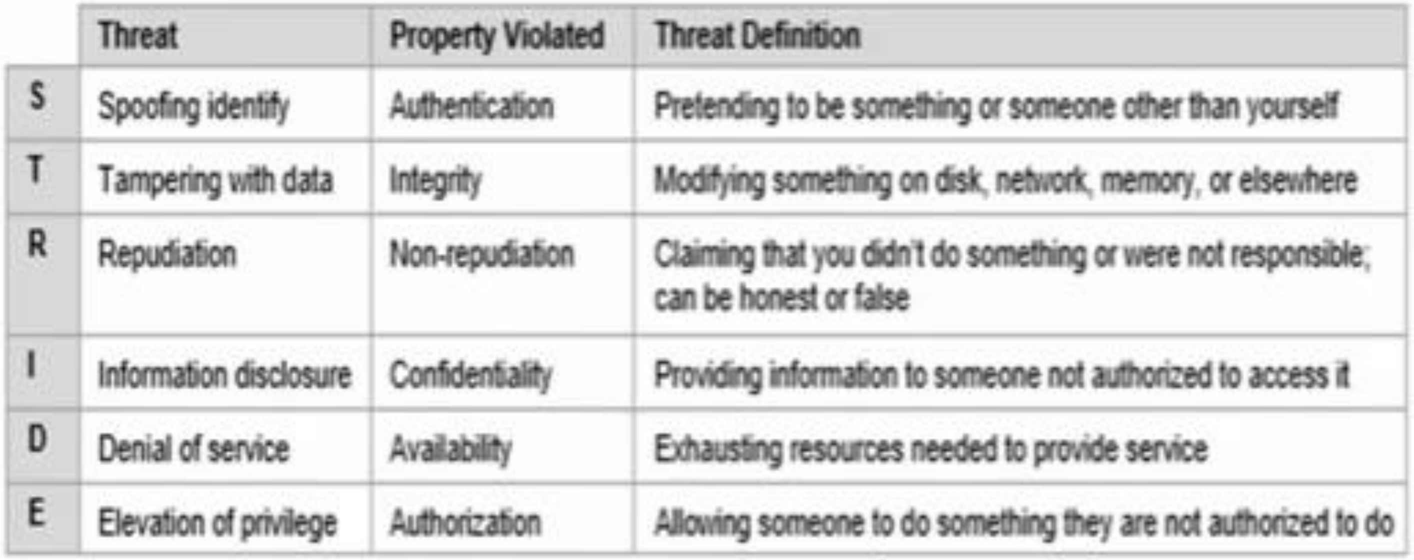
## 1- STRIDE

When attempting to inventory and categorize threats, it is often helpful to use a guide or reference. Microsoft developed a threat categorization scheme known as **STRIDE**. STRIDE is often used in relation to assessing threats against applications or operating systems. However, it can also be used in other contexts as well.

## STRIDE

Microsoft developed the STRIDE methodology in the late 1990s as a way to standardize the identification of threats across their product line.

**a classic software-centric threat model.**



| | Threat | Property Violated | Threat Definition |
|---|---|---|---|
| S | Spoofing identify | Authentication | Pretending to be something or someone other than yourself |
| T | Tampering with data | Integrity | Modifying something on disk, network, memory, or elsewhere |
| R | Repudiation | Non-repudiation | Claiming that you didn't do something or were not responsible; can be honest or false |
| I | Information disclosure | Confidentiality | Providing information to someone not authorized to access it |
| D | Denial of service | Availability | Exhausting resources needed to provide service |
| E | Elevation of privilege | Authorization | Allowing someone to do something they are not authorized to do |

STRIDE is an acronym standing for the following**:**

**1- Spoofing:** An attack with the goal of gaining access to a target system through the use of a falsified identity. Spoofing can be used against IP addresses, MAC address, usernames, system names, wireless network SSIDs, email addresses, and many other types of logical identification. When an attacker spoofs their identity as a valid or authorized entity, they are often able to bypass filters and blockades against unauthorized access. Once a spoofing attack has successfully granted an attacker access to a target system, subsequent attacks of abuse, data theft, or privilege escalation can be initiated.

**2-Tampering:** Any action resulting in the unauthorized changes or manipulation of data, whether in transit or in storage. Tampering is used to falsify communications or alter static information. Such attacks are a violation of integrity as well as availability.

**3- Repudiation**: The ability for a user or attacker to deny having performed an action or activity. Often attackers engage in repudiation attacks in order to maintain plausible deniability so as not to be held accountable for their actions. Repudiation attacks can also result in innocent third parties being blamed for security violations.

**4- Information disclosure**: The revelation or distribution of private, confidential, or controlled information to external or unauthorized entities. This could include customer identity information, financial information, or proprietary business operation details. Information disclosure can take advantage of system design and implementation mistakes, such as failing to remove debugging code, leaving sample applications and accounts, not sanitizing programming notes from client visible content (such as comments in HTML documents), using hidden form fields, or allowing overly detailed error messages to be shown to users.

**5- Denial of service (DoS):** An attack that attempts to prevent authorized use of a resource. This can be done through flaw exploitation, connection overloading, or traffic flooding. A DoS attack does not necessarily result in full interruption to a resource; it could instead reduce throughput or introduce latency in order to hamper productive use of a resource. Although most DoS attacks are temporary and last only as long as the attacker maintains the onslaught, there are some permanent DoS attacks.

A permanent DoS attack might involve the destruction of a dataset, the replacement of software with malicious alternatives, or forcing a firmware flash operation that could be interrupted or that installs faulty firmware. Any of these DoS attacks would render a permanently damaged system that is not able to be restored to normal operation with a simple reboot or by waiting out the attackers. A full system repair and backup restoration would be required to recover from a permanent DoS attack.

**6- Elevation of privilege:** An attack where a limited user account is transformed into an account with greater privileges, powers, and access. This might be accomplished through theft or exploitation of the credentials of a higher-level account, such as that of an administrator or root. It also might be accomplished through a system or application exploit that temporarily or permanently grants additional powers to an otherwise limited account.

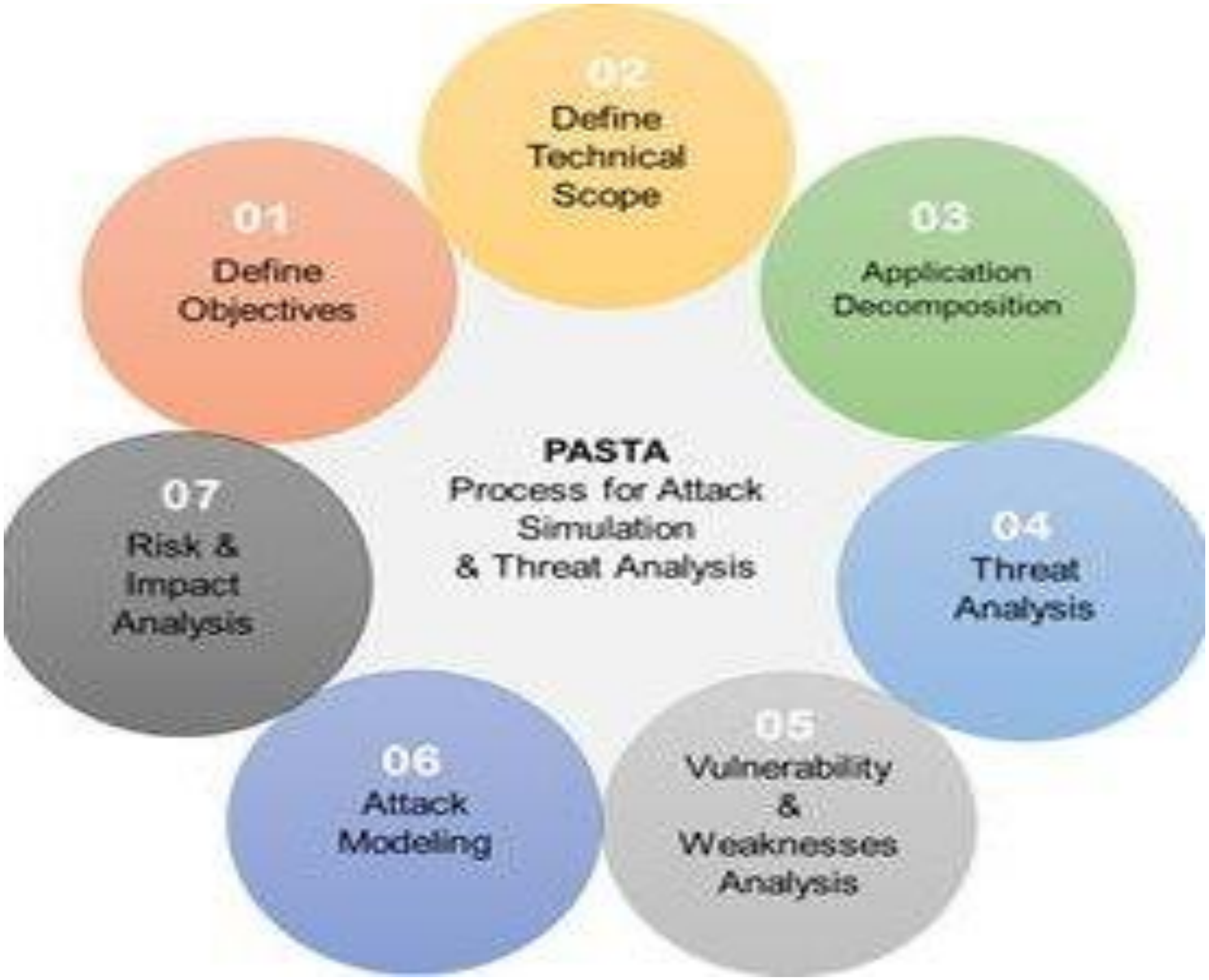
Although STRIDE is typically used to focus on application threats, it is applicable to other situations, such as network threats and host threats. Other attacks may be more specific to network and host concerns, such as sniffing and hijacking for networks and malware and arbitrary code execution for hosts, but the six threat concepts of STRIDE are fairly broadly applicable.

Generally, the purpose of STRIDE and other tools in threat modeling is to consider the range of compromise concerns and to focus on the goal or end results of an attack. Attempting to identity each and every specific attack method and technique is an impossible task—new attacks are being developed constantly. Although the goals or purposes of attacks can be loosely categorized and grouped, they remain relatively constant over time.

Potential threats to your business are broad and varied. A company faces threats from nature, technology, and people. Most businesses focus on natural disasters and IT attacks in preparing for threats, but it's also important to consider threat potential from individuals. Always consider the best and worst possible outcomes of your organization's activities, decisions, interactions. Identifying threats is the first step toward designing defenses to help reduce or eliminate downtime, compromise, and loss.

## 2- PASTA

**PASTA** is an attacker-centric methodology with seven steps. It is designed to correlate business objectives with technical requirements. PASTA's steps guide teams to dynamically identify, count, and prioritize threats. PASTA allows for collaboration between developer and business stakeholders to truly understand your application's inherent risk, its likelihood of attack, and the business impact if there was a compromise.


01
Define Objectives
02
Define Technical Scope
03
Application Decomposition
04
Threat Analysis
05
Vulnerability & Weaknesses Analysis
06
Attack Modeling
07
Risk & Impact Analysis
PASTA
Process for Attack Simulation & Threat Analysis

The steps of a PASTA threat model are:

1. Define business objectives
2. Define the technical scope of assets and components
3. Application decomposition and identify application controls
4. Threat analysis based on threat intelligence
5. Vulnerability detection
6. Attack enumeration and modeling
7. Risk analysis and development of countermeasures

| | |
|---|---|
| **1. Define Objectives** | • Identify Business Objectives<br>• Identify Security and Compliance Requirements<br>• Business Impact Analysis |
| **2. Define Technical Scope** | • Capture the Boundaries of the Technical Environment<br>• Capture Infrastructure \| Application \| Software Dependencies |
| **3. Application Decomposition** | • Identify Use Cases \| Define App. Entry Points & Trust Levels<br>• Identify Actors \| Assets \| Services \| Roles \| Data Sources<br>• Data Flow Diagramming (DFDs) \| Trust Boundaries |
| **4. Threat Analysis** | • Probabilistic Attack Scenarios Analysis<br>• Regression Analysis on Security Events<br>• Threat Intelligence Correlation and Analytics |
| **5. Vulnerability & Weaknesses Analysis** | • Queries of Existing Vulnerability Reports & Issues Tracking<br>• Threat to Existing Vulnerability Mapping Using Threat Trees<br>• Design Flaw Analysis Using Use and Abuse Cases<br>• Scorings (CVSS/CWSS) \| Enumerations (CWE/CVE) |
| **6. Attack Modeling** | • Attack Surface Analysis<br>• Attack Tree Development \| Attack Library Mgt.<br>• Attack to Vulnerability & Exploit Analysis Using Attack Trees |
| **7. Risk & Impact Analysis** | • Qualify & Quantify Business Impact<br>• Countermeasure Identification and Residual Risk Analysis<br>• ID Risk Mitigation Strategies |

**1- Define the Objectives:** The first step of the PASTA methodology is to define the objectives. These may be internally driven, externally driven, and/or driven by your user base. You should clearly understand the purpose of the application. How does it make your company money? Or maybe it does more back-end processing. What kind of regulations need to be incorporated? Unlike traditional, static threat modeling methods, stage one of PASTA allows you the opportunity to incorporate governance into these discussions and bake it in from the very beginning.

**2- Define the Technical Scope:** Stage two of PASTA is to understand your attack surface by defining your technical scope: know what you are protecting. A common theme for professionals in application security and product security is under-scoping because we are focused purely on the application domain. When you are defining an attack surface, you should understand what you are running with and what sort of dependencies you might have with third party services. This can include features created as a developer, systems maintained as an engineer, or components monitored in the infrastructure.

**3- Decompose the Application:** Stage three of PASTA is application decomposition. In stage two, we built context around what we are running. Stage three goes further by creating context around how everything communicates, how it all comes together. The key output of this stage is to understand if you have implicit trust models and where they are. It may be an IoT device talking to the cloud, or an embedded device talking to an automobile component. You may have an implicit trust model that could be a good conduit for exploitation.

In this stage, you should produce data flow diagrams. It is best to work with your architecture to understand the calls and integrations you discovered in stage two. Data flow diagrams alone are not threat modeling. A data flow diagram shows the flow of data between callers across trust boundaries, but it has no depiction of threats. It doesn't illustrate to a developer or to an engineer what they should be worried about, it provides a map for analysis.

**4- Analyze the Threats:** Stage four is analyzing the threats. The main output for stage four is to understand what the application does and what sort of threats are affecting your defined attack surface. The scope is based upon your technology selection defined in stage two. You also need to consider your data type, your data model, and your data consumption model. What sort of threats are more pervasive based upon how you're consuming data?

As a threat modeler and security champion, you must first understand what threats are relevant to you by analyzing threat intelligence that might provide an insight into attack behavior against your industry and your technology footprint. From there, you can start to build your own threat library.

**5- Vulnerability Analysis:** Stage five correlates the application's vulnerabilities to the application's assets. How are you going to sew together tools and best practices, in terms of volume management, volume assessment, static analysis, dynamic analysis, etc.? And in all the noise that you're seeing in the vulnerability analysis, what are the ones that are material to the threats in your threat library?

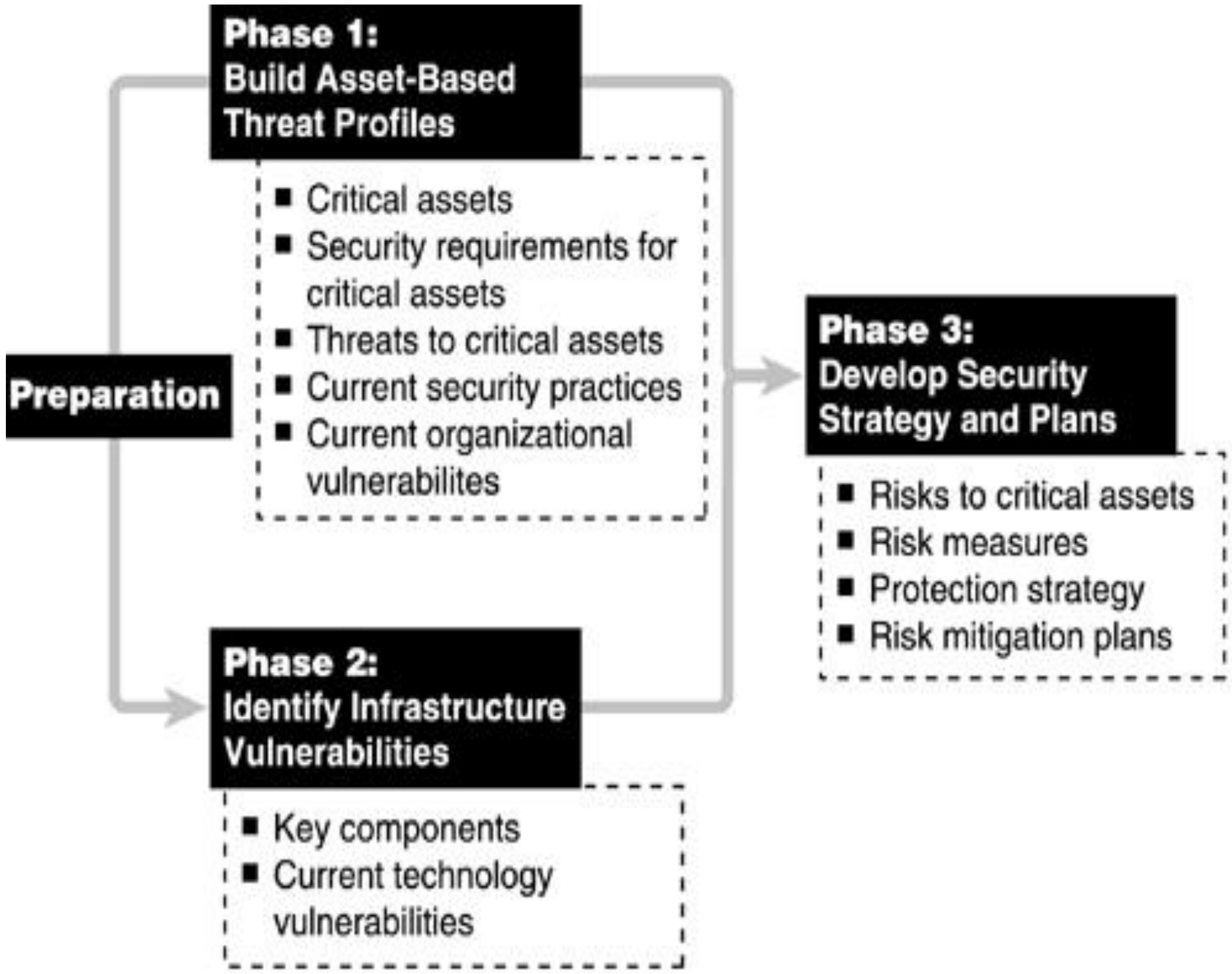
The key differentiator with PASTA is focusing on risks that will have the most impact to the business – all based upon stage one. In stage five, you identify what is wrong. What is wrong with the application in terms of not just vulnerabilities that might be in my code base through static analysis, but also what is wrong with my design? What's wrong with my trust model that I may have discovered in stage three? There are many factors to throw into the vulnerability bucket, including flaws or weaknesses that were identified during manual security testing, vulnerabilities or weaknesses in architecture stemming from your data flow diagram, and/or different types of vulnerability scanners to name a few.

**6- Attack Analysis:** The key objective for stage six of PASTA, is to prove that the things we found vulnerable in stage five, are actually viable. To blueprint a good model for attacks, you want to use attack trees. Using attack trees allows you to map known vulnerabilities to a node on the attack tree to determine it's likelihood.

**7- Risk and Impact Analysis:** At the end of the day, PASTA threat modeling is about reducing risks. The end goal for stage seven, is to build countermeasures that mitigate the threats that are important. To finalize the threat modeling exercise, we want to utilize and tie back in the information we found in stages one through six. By factoring all this information in, you will have access to the derived impact of threats through simulated attacks. By increasing your visibility into the impact of exploits and gaps in countermeasures, it becomes possible to make informed risk management decisions that save your organization time and money.

## 3- OCATVE

**The Operationally Critical Threat, Asset, and Vulnerability Evaluation** (OCTAVE) is a framework for identifying and managing information security risks. It defines a comprehensive evaluation method that allows an organization to identify the information assets that are important to the mission of the organization, the threats to those assets, and the vulnerabilities that may expose those assets to the threats. By putting together the information assets, threats, and vulnerabilities, the organization can begin to understand what information is at risk. With this understanding, the organization can design and implement a protection strategy to reduce the overall risk exposure of its information assets.


Phase 1:
Build Asset-Based
Threat Profiles
■ Critical assets
■ Security requirements for critical assets
■ Threats to critical assets
■ Current security practices
■ Current organizational vulnerabilites
Preparation
Phase 2:
Identify Infrastructure
Vulnerabilities
■ Key components
■ Current technology vulnerabilities
Phase 3:
Develop Security
Strategy and Plans
■ Risks to critical assets
■ Risk measures
■ Protection strategy
■ Risk mitigation plans

ΣV

**OCTAVE is organized around these three basic aspects** enabling organizational personnel to assemble a comprehensive picture of the organization's information security needs. The phases are:

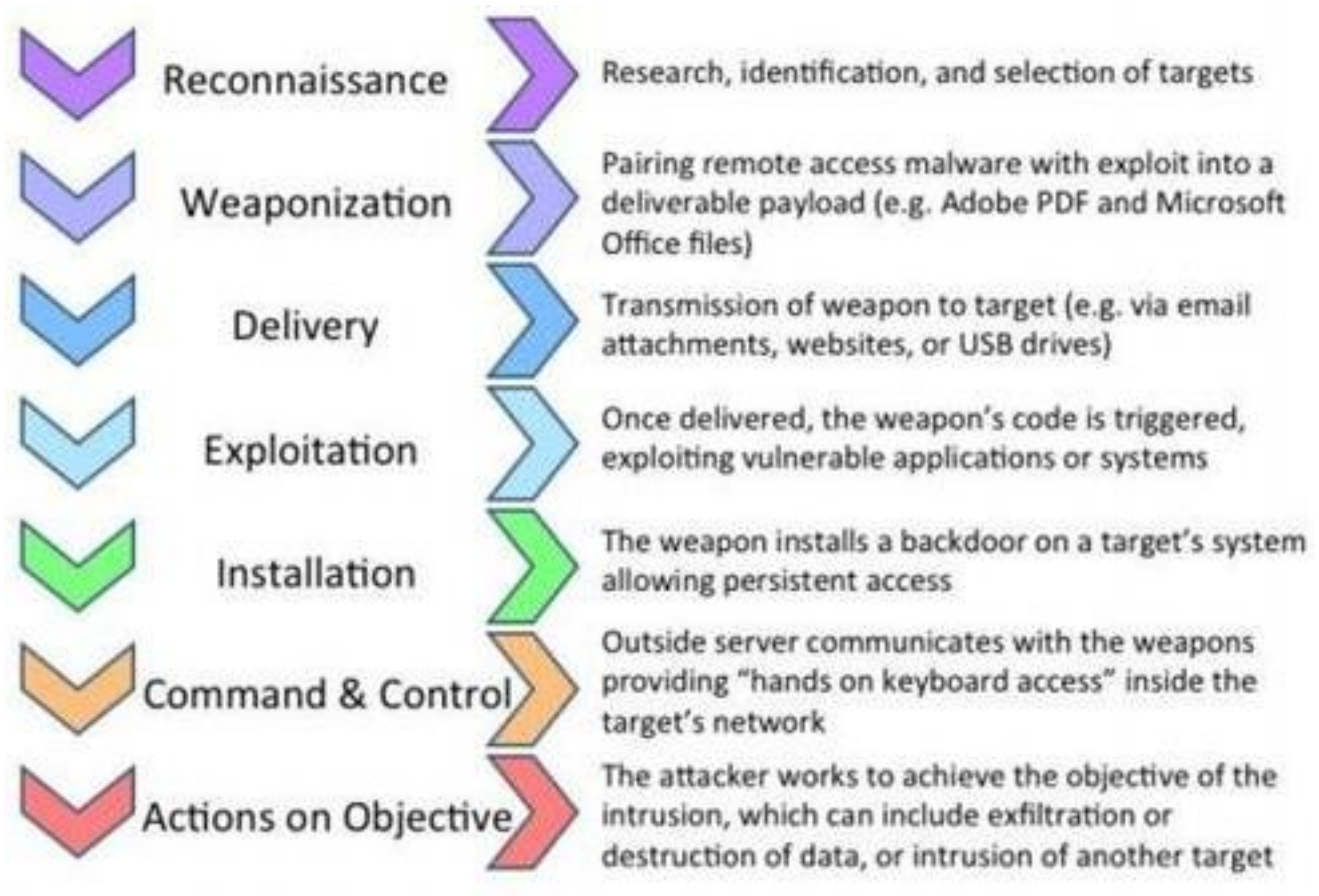
- Phase 1: Build Asset-Based Threat Profiles: This is an organizational evaluation. The analysis team determines what is important to the organization (information-related assets) and what is currently being done to protect those assets. The team then selects those assets that are most important to the organization (critical assets) and describes security requirements for each critical asset. Finally, it identifies threats to each critical asset, creating a threat profile for that asset.

- Phase 2: Identify Infrastructure Vulnerabilities: This is an evaluation of the information infrastructure. The analysis team examines network access paths, identifying classes of information technology components related to each critical asset. The team then determines the extent to which each class of component is resistant to network attacks.

- Phase 3: Develop Security Strategy and Plans: During this part of the evaluation, the analysis team identifies risks to the organization's critical assets and decides what to do about them. The team creates a protection strategy for the organization and mitigation plans to address the risks to the critical assets, based upon an analysis of the information gathered.

## Kill Chain Concept

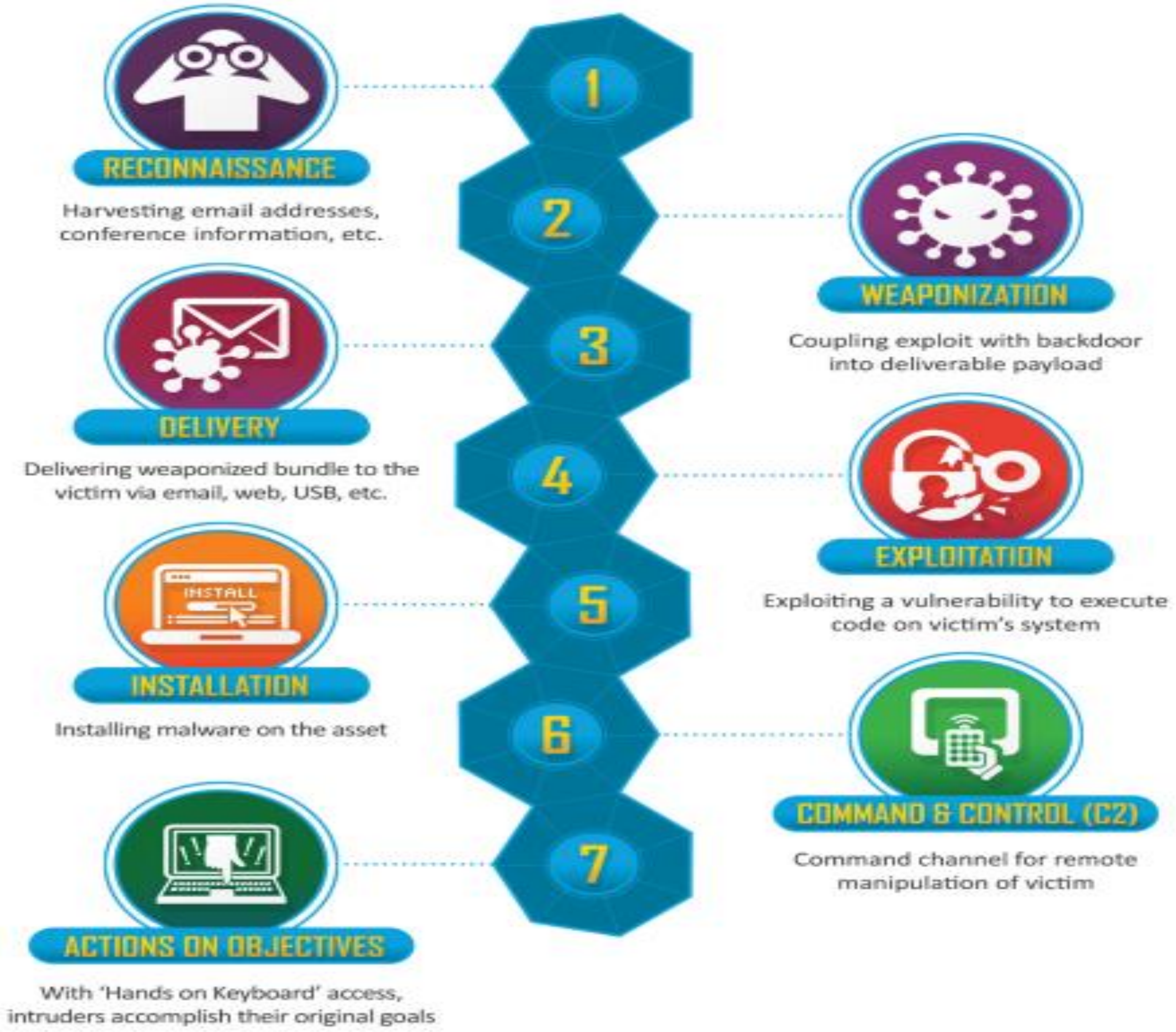
The term **kill chain** comes from a military concept that uses stages to outline the structure of an attack. “Breaking” the opponent’s kill chain refers to the ability to block an attack at any stage. The seven stages (phases) include: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control (C2), and Action on Objectives.



| | |
|---|---|
| Reconnaissance | Research, identification, and selection of targets |
| Weaponization | Pairing remote access malware with exploit into a deliverable payload (e.g. Adobe PDF and Microsoft Office files) |
| Delivery | Transmission of weapon to target (e.g. via email attachments, websites, or USB drives) |
| Exploitation | Once delivered, the weapon's code is triggered, exploiting vulnerable applications or systems |
| Installation | The weapon installs a backdoor on a target's system allowing persistent access |
| Command & Control | Outside server communicates with the weapons providing "hands on keyboard access" inside the target's network |
| Actions on Objective | The attacker works to achieve the objective of the intrusion, which can include exfiltration or destruction of data, or intrusion of another target |

**1. Reconnaissance:** This is the intelligence-gathering phase. The intruder selects a target, researches it, and looks for vulnerabilities within the target network. They review available information and resources about an organization and its public-facing assets.

**2- Weaponization:** The intruder analyzes the gathered information and plans the weapon to be used in the cyber-attack. For example, a deliverable payload, a type of weaponized file created to do something for the intruder, may be embedded into a PDF or Word document. In another example, a malicious URL that redirects users to a malware-laden website can be put inside an email. Individuals within an organization may be targeted through social-engineering attacks such as phishing.



RECONNAISSANCE

Harvesting email addresses, conference information, etc.

WEAPONIZATION

Coupling exploit with backdoor into deliverable payload

DELIVERY

Delivering weaponized bundle to the victim via email, web, USB, etc.

EXPLOITATION

Exploiting a vulnerability to execute code on victim's system

INSTALLATION

Installing malware on the asset

COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

ACTIONS ON OBJECTIVES

With 'Hands on Keyboard' access, intruders accomplish their original goals

**3. Delivery:** In this phase, the payload (weapon) is transmitted to the target via a communication vector.

**4. Exploitation:** Once inside the system, the intruder will attempt to move laterally to other systems/accounts on the network. The goal is to increase the permission level in order to reach more data.

**5- Installation:** The installation (or persistence) phase describes the actions taken by a threat actor to establish a backdoor into the targeted system. This gives the threat actor sustained and persistent access to the target – thus providing them a way to access the system whenever they desire.

**6. Command and Control (CnC):** Command-and-Control (CNC) occurs when an exploited host sends outbound beacons to an Internet-based controller in order to establish a communications channel. Once CNC is established with an exploited target, threat actors will have access to the target system and possibly the entire network itself. These channels allow the threat actor to issue commands to the malicious software that had been installed on the target (or targets).

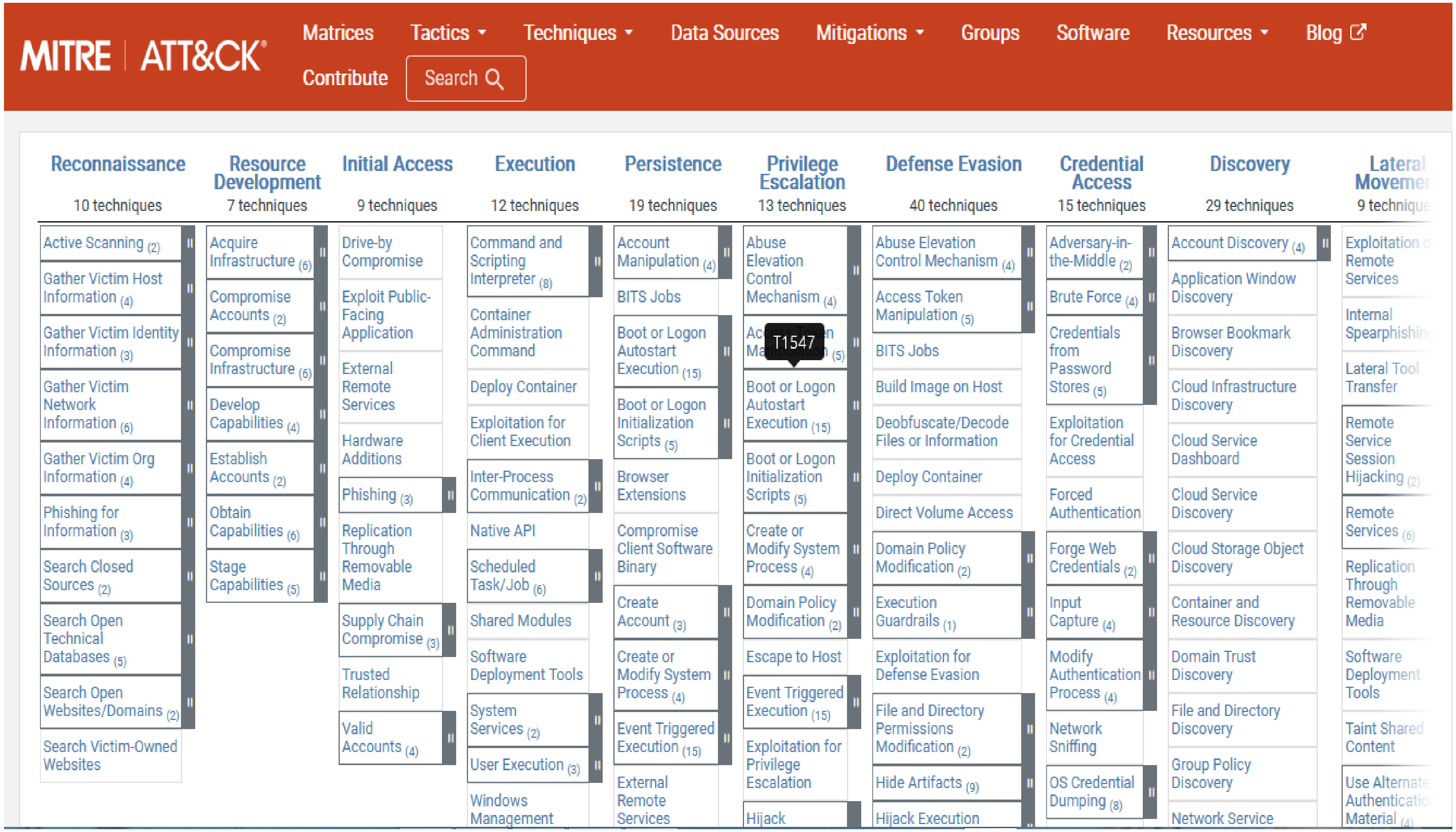
**7. Actions on Objectives:** Intruders take action to achieve their goals, such as:

- data exfiltration
- data destruction
- encryption for ransom

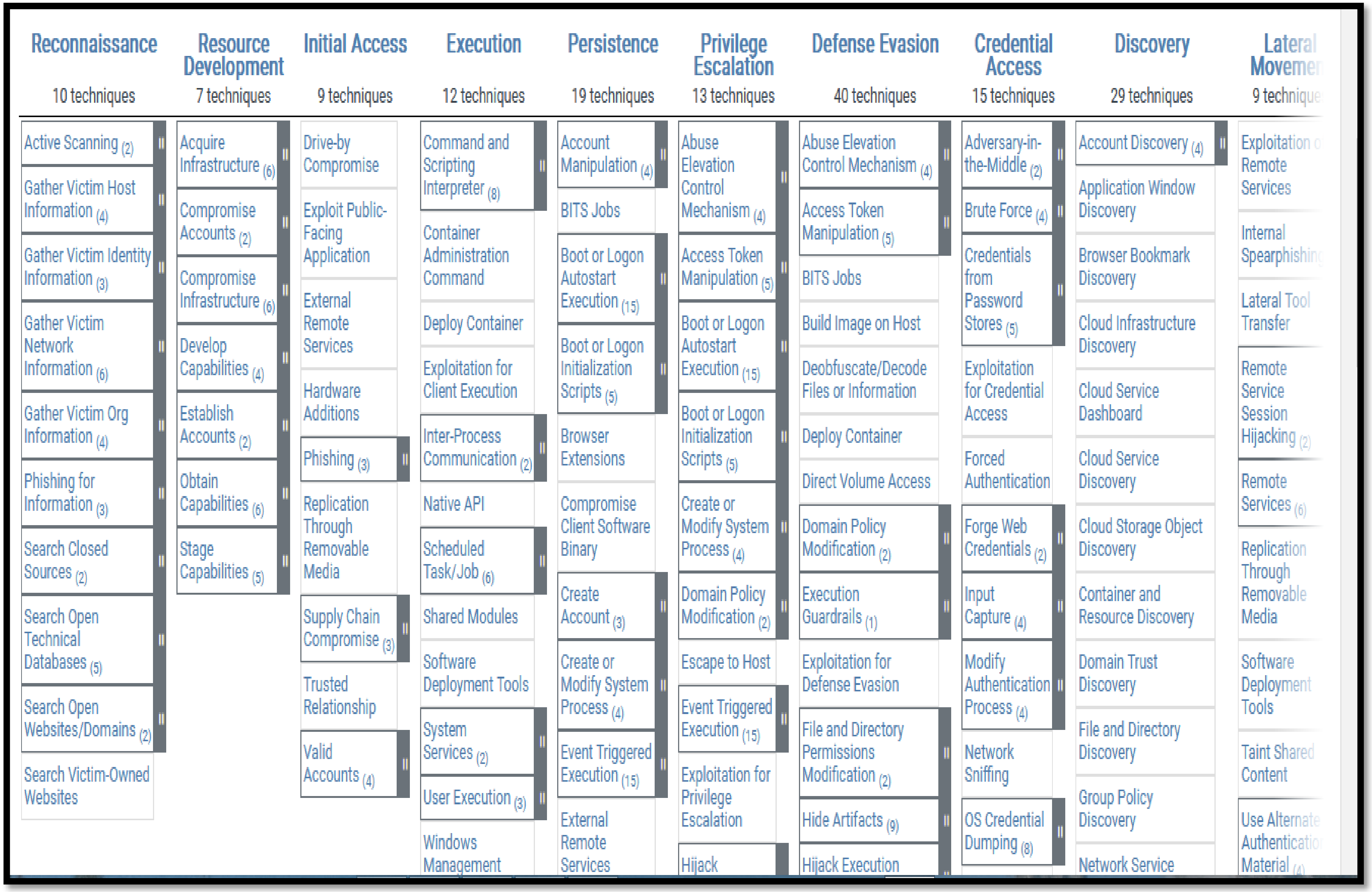
## MITRE ATT&CK MATRIX

**MITRE ATT&CK Framework** was created by to document attacker tactics and techniques based on real-world observations. This index continues to evolve with the threat landscape and has become a renowned knowledge base for the industry to understand attacker models, methodologies, and mitigation. Successful and comprehensive threat detection requires understanding common adversary techniques, which ones may especially pose a threat to your organization, and how to detect and mitigate these attacks

# https://attack.mitre.org/



| Reconnaissance<br>10 techniques | Resource Development<br>7 techniques | Initial Access<br>9 techniques | Execution<br>12 techniques | Persistence<br>19 techniques | Privilege Escalation<br>13 techniques | Defense Evasion<br>40 techniques | Credential Access<br>15 techniques | Discovery<br>29 techniques | Lateral Moveme<br>9 technique |
|---|---|---|---|---|---|---|---|---|---|
| Active Scanning (2) | Acquire Infrastructure (6) | Drive-by Compromise | Command and Scripting Interpreter (8) | Account Manipulation (4) | Abuse Elevation Control Mechanism (4) | Abuse Elevation Control Mechanism (4) | Adversary-in-the-Middle (2) | Account Discovery (4) | Exploitation o Remote Services |
| Gather Victim Host Information (4) | Compromise Accounts (2) | Exploit Public-Facing Application | Container Administration Command | BITS Jobs | Ac T1547 n Ma (5) | Access Token Manipulation (5) | Brute Force (4) | Application Window Discovery | Internal Spearphishin |
| Gather Victim Identity Information (3) | Compromise Infrastructure (6) | External Remote Services | Deploy Container | Boot or Logon Autostart Execution (15) | Boot or Logon Autostart Execution (15) | BITS Jobs | Credentials from Password Stores (5) | Browser Bookmark Discovery | Lateral Tool Transfer |
| Gather Victim Network Information (6) | Develop Capabilities (4) | Hardware Additions | Exploitation for Client Execution | Boot or Logon Initialization Scripts (5) | Boot or Logon Initialization Scripts (5) | Build Image on Host | Exploitation for Credential Access | Cloud Infrastructure Discovery | Remote Service Session Hijacking (2) |
| Gather Victim Org Information (4) | Establish Accounts (2) | Phishing (3) | Inter-Process Communication (2) | Browser Extensions | Create or Modify System Process (4) | Deobfuscate/Decode Files or Information | Forced Authentication | Cloud Service Dashboard | Remote Services (6) |
| Phishing for Information (3) | Obtain Capabilities (6) | Replication Through Removable Media | Native API | Compromise Client Software Binary | Domain Policy Modification (2) | Deploy Container | Forge Web Credentials (2) | Cloud Service Discovery | Replication Through Removable Media |
| Search Closed Sources (2) | Stage Capabilities (5) | Supply Chain Compromise (3) | Scheduled Task/Job (6) | Create Account (3) | Escape to Host | Direct Volume Access | Input Capture (4) | Cloud Storage Object Discovery | Software Deployment Tools |
| Search Open Technical Databases (5) | | Trusted Relationship | Shared Modules | Create or Modify System Process (4) | Event Triggered Execution (15) | Domain Policy Modification (2) | Modify Authentication Process (4) | Container and Resource Discovery | Taint Shared Content |
| Search Open Websites/Domains (2) | | Valid Accounts (4) | Software Deployment Tools | Event Triggered Execution (15) | Exploitation for Privilege Escalation | Execution Guardrails (1) | Network Sniffing | Domain Trust Discovery | Use Alternate Authenticatio Material (4) |
| Search Victim-Owned Websites | | | System Services (2) | External Remote Services | Hijack | Exploitation for Defense Evasion | OS Credential Dumping (8) | File and Directory Discovery | |
| | | | User Execution (3) | | | File and Directory Permissions Modification (2) | | Group Policy Discovery | |
| | | | Windows Management | | | Hide Artifacts (9) | | Network Service | |
| | | | | | | Hijack Execution | | | |



| Reconnaissance 10 techniques | Resource Development 7 techniques | Initial Access 9 techniques | Execution 12 techniques | Persistence 19 techniques | Privilege Escalation 13 techniques | Defense Evasion 40 techniques | Credential Access 15 techniques | Discovery 29 techniques | Lateral Movemen 9 technique |
|---|---|---|---|---|---|---|---|---|---|
| Active Scanning (2) | Acquire Infrastructure (6) | Drive-by Compromise | Command and Scripting Interpreter (8) | Account Manipulation (4) | Abuse Elevation Control Mechanism (4) | Abuse Elevation Control Mechanism (4) | Adversary-in-the-Middle (2) | Account Discovery (4) | Exploitation o Remote Services |
| Gather Victim Host Information (4) | Compromise Accounts (2) | Exploit Public-Facing Application | Container Administration Command | BITS Jobs | Access Token Manipulation (5) | Access Token Manipulation (5) | Brute Force (4) | Application Window Discovery | Internal Spearphishin |
| Gather Victim Identity Information (3) | Compromise Infrastructure (6) | External Remote Services | Deploy Container | Boot or Logon Autostart Execution (15) | Boot or Logon Autostart Execution (15) | BITS Jobs | Credentials from Password Stores (5) | Browser Bookmark Discovery | Lateral Tool Transfer |
| Gather Victim Network Information (6) | Develop Capabilities (4) | Hardware Additions | Exploitation for Client Execution | Boot or Logon Initialization Scripts (5) | Boot or Logon Initialization Scripts (5) | Build Image on Host | Exploitation for Credential Access | Cloud Infrastructure Discovery | Remote Service Session Hijacking (2) |
| Gather Victim Org Information (4) | Establish Accounts (2) | Phishing (3) | Inter-Process Communication (2) | Browser Extensions | Create or Modify System Process (4) | Deobfuscate/Decode Files or Information | Forced Authentication | Cloud Service Dashboard | Remote Services (6) |
| Phishing for Information (3) | Obtain Capabilities (6) | Replication Through Removable Media | Native API | Compromise Client Software Binary | Domain Policy Modification (2) | Deploy Container | Forge Web Credentials (2) | Cloud Service Discovery | Replication Through Removable Media |
| Search Closed Sources (2) | Stage Capabilities (5) | Supply Chain Compromise (3) | Scheduled Task/Job (6) | Create Account (3) | Escape to Host | Direct Volume Access | Input Capture (4) | Cloud Storage Object Discovery | Software Deployment Tools |
| Search Open Technical Databases (5) | | Trusted Relationship | Shared Modules | Create or Modify System Process (4) | Event Triggered Execution (15) | Domain Policy Modification (2) | Modify Authentication Process (4) | Container and Resource Discovery | Taint Shared Content |
| Search Open Websites/Domains (2) | | Valid Accounts (4) | Software Deployment Tools | Event Triggered Execution (15) | Exploitation for Privilege Escalation | Execution Guardrails (1) | Network Sniffing | Domain Trust Discovery | Use Alternate Authenticatio Material (4) |
| Search Victim-Owned Websites | | | System Services (2) | External Remote Services | Hijack | Exploitation for Defense Evasion | OS Credential Dumping (8) | File and Directory Discovery | |
| | | | User Execution (3) | | | File and Directory Permissions Modification (2) | | Group Policy Discovery | |
| | | | Windows Management | | | Hide Artifacts (9) | | Network Service | |
| | | | | | | Hijack Execution | | | |