Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Advanced Security (Information Security Management) - Threat Modelling

Prof. Dr.-Ing. Sebastian Schlesinger

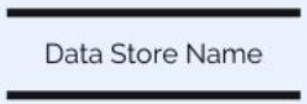
Professor of Business Computer Science (Security and Embedded Systems Engineering)

General Idea

Data Flow Diagram (DFD)

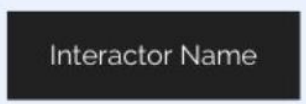
Symbols

Data Flow: Depicts data flowing from the source to the destination component.



Data Store: Storage location of information used by the system.

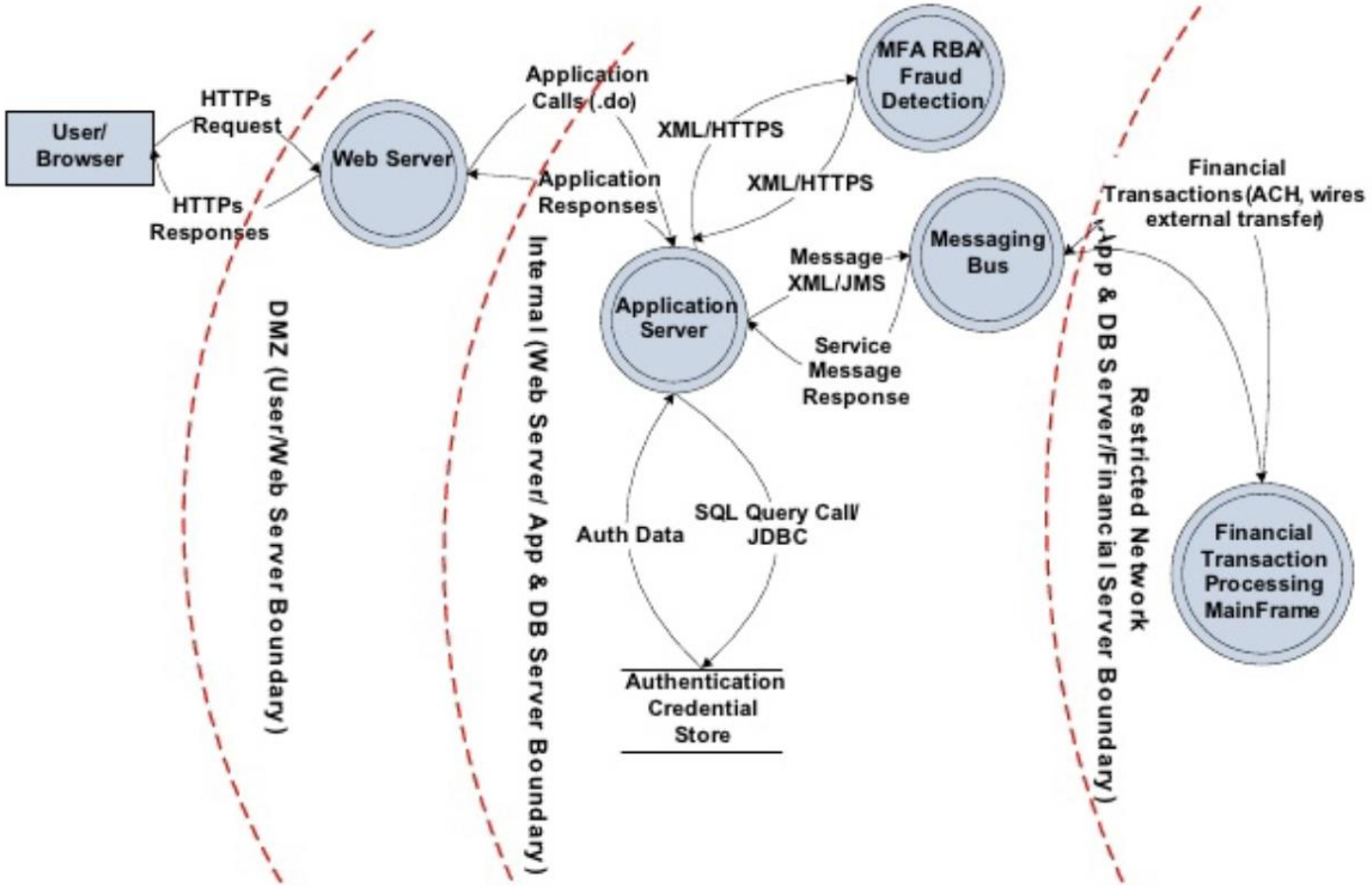
Process: Applications or processes that apply logic and/or change data.



Interactor: Endpoint (person or system) that interacts or uses the process.

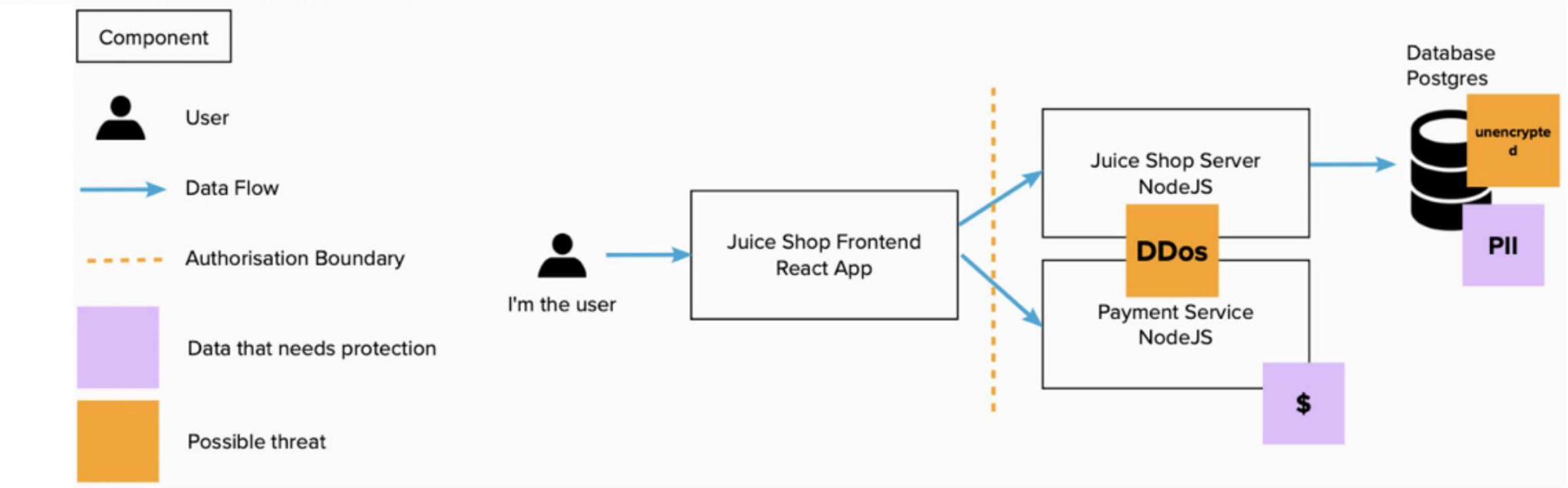
Trust Boundary: Boundary between trust zones (i.e., the boundary from trusted to non-trusted zone).

An example DFD



Trust boundaries show where a level of trust changes to either elevated or lowered levels of trust. Identifying your trust boundaries helps you clarify which components likely have a similar attack vectors, similar threat models.

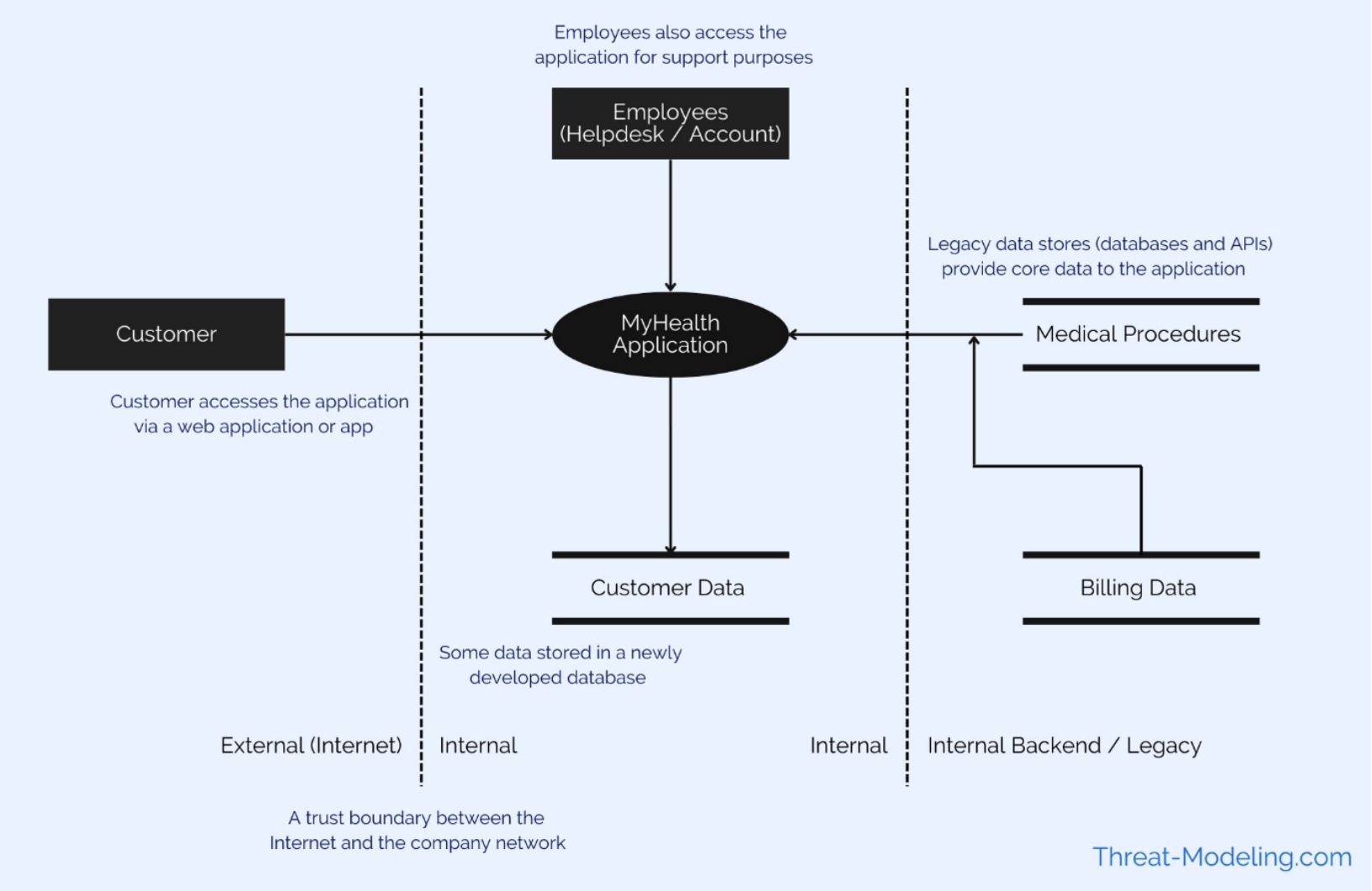
Another simple DFD



How to obtain a data flow diagram?

1. Understand Processes (and / or application components), Data Flows (Communications), and Actors (e.g. different user groups with different privileges)
2. Draw the main processes / applications etc.
3. Draw the Interactors, e.g. customer, external service, external backup service, internal employee
4. Draw the Data Stores
5. Draw the Trust Boundaries
6. Draw the Connectors

Another DFD

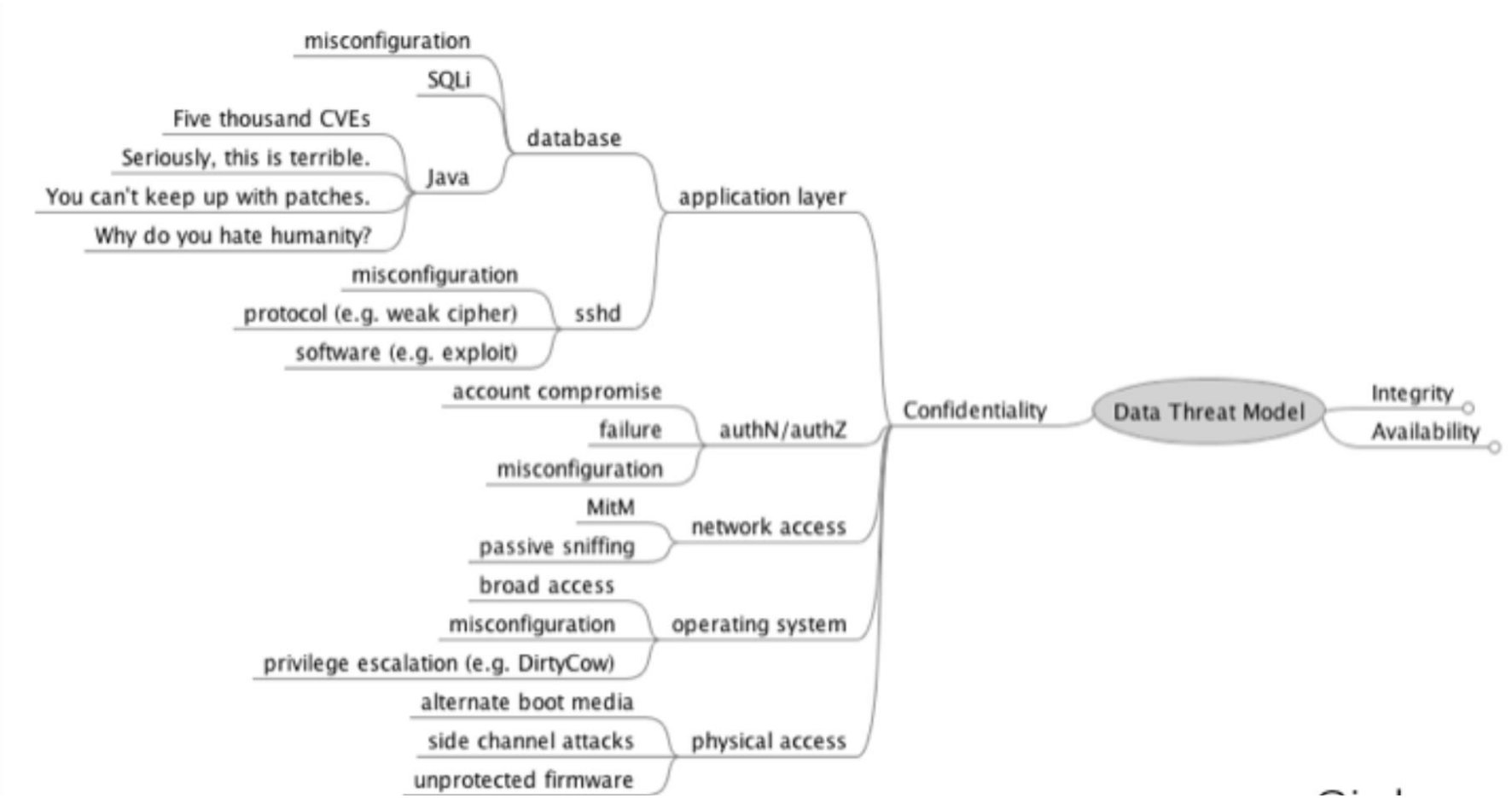

Employees also access the application for support purposes
Employees (Helpdesk / Account)
Legacy data stores (databases and APIs) provide core data to the application
Customer
MyHealth Application
Medical Procedures
Customer accesses the application via a web application or app
Customer Data
Billing Data
Some data stored in a newly developed database
External (Internet)
Internal
Internal
Internal Backend / Legacy
A trust boundary between the Internet and the company network
Threat-Modeling.com

STRIDE – One Approach to systematically contemplate about threats

STRIDE

Threat	*Property*
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-Repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Decomposing attacks to model the attack vectors

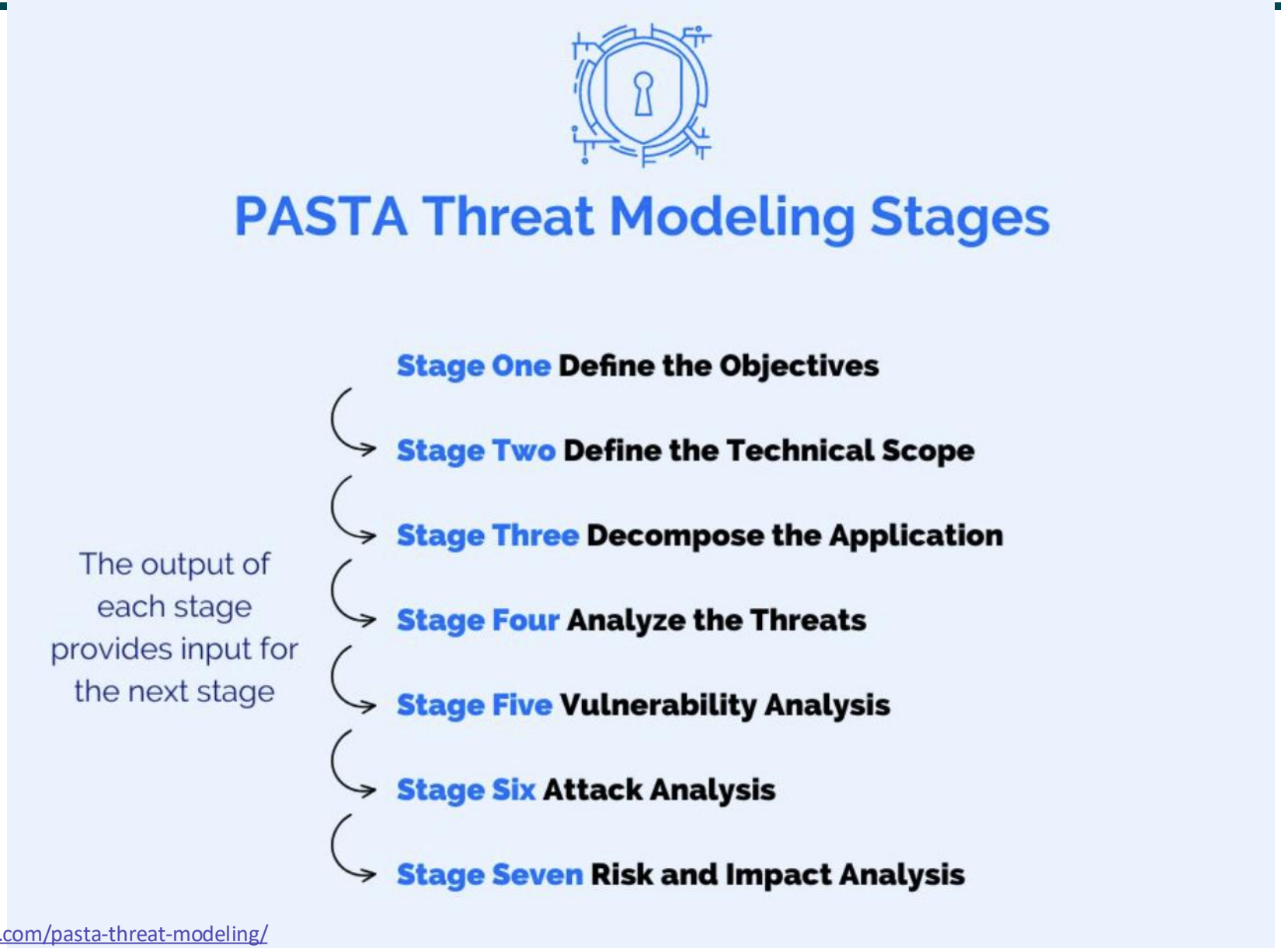


Assessing the risks

DREAD+D

Damage	How bad would the attack be?
Reproducability	How easy to recreate the attack?
Exploitability	How easy to launch the attack?
Affected Users	How many are impacted?
Discoverability	How easy to discover for attacker?
Detection	How hard to detect for defender?

Another approach

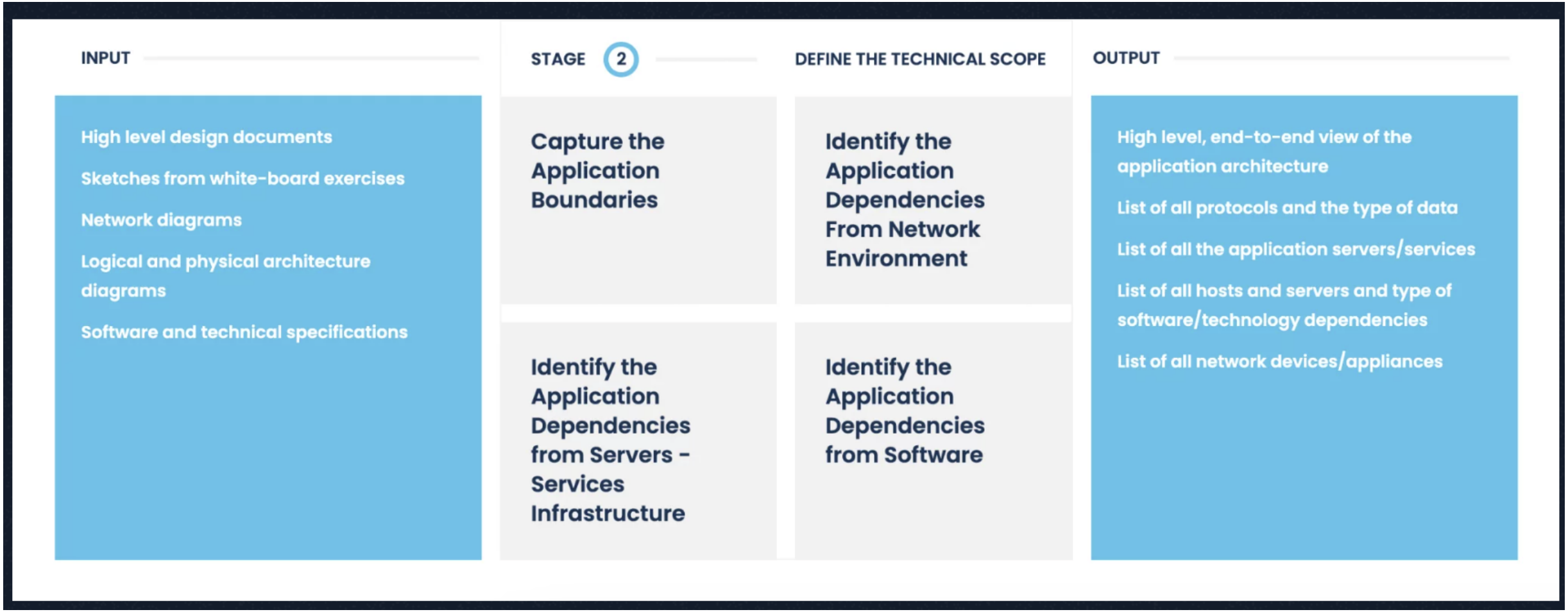


Source: https://threat-modeling.com/pasta-threat-modeling/

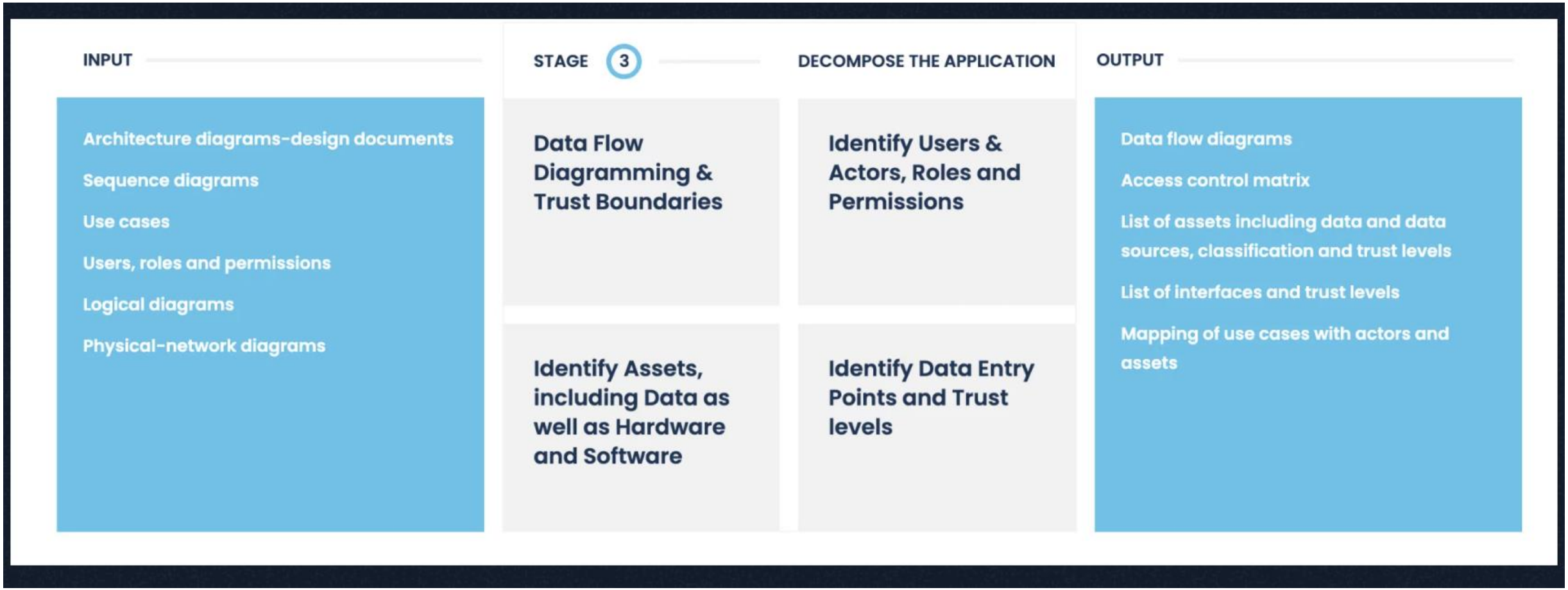
PASTA – Stage One (Define the Objectives)

Source: https://versprite.com/blog/what-is-pasta-threat-modeling/

PASTA – Stage Two (Define the Technical Scope)

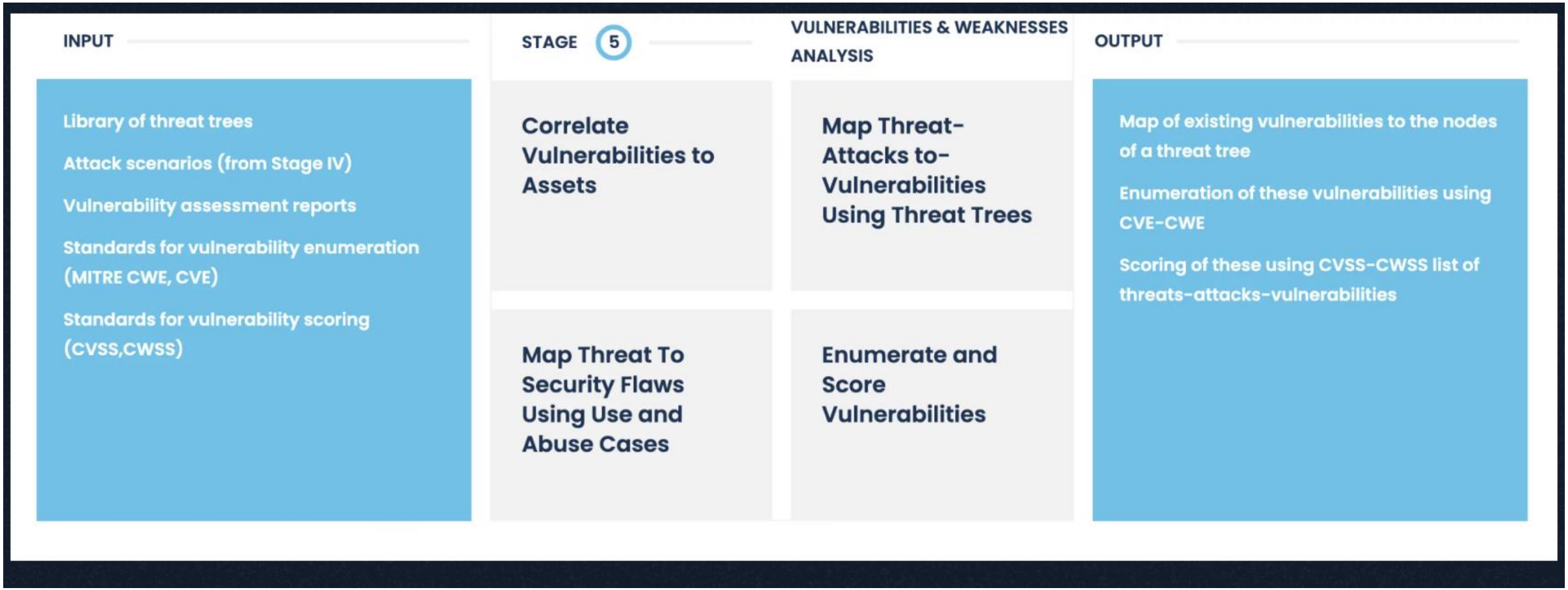


PASTA – Stage Three (Decompose the Application)

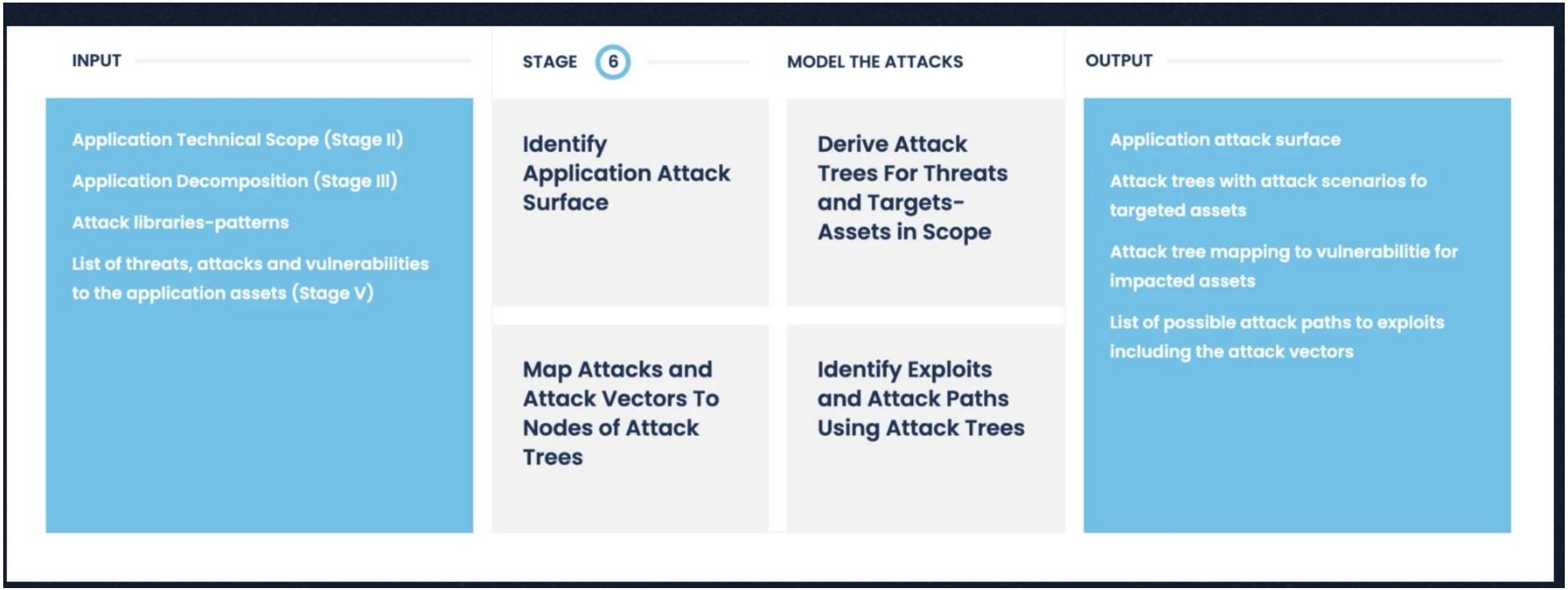


PASTA – Stage Four (Analyze the Threats)

PASTA – Stage Five (Vulnerability Analysis)

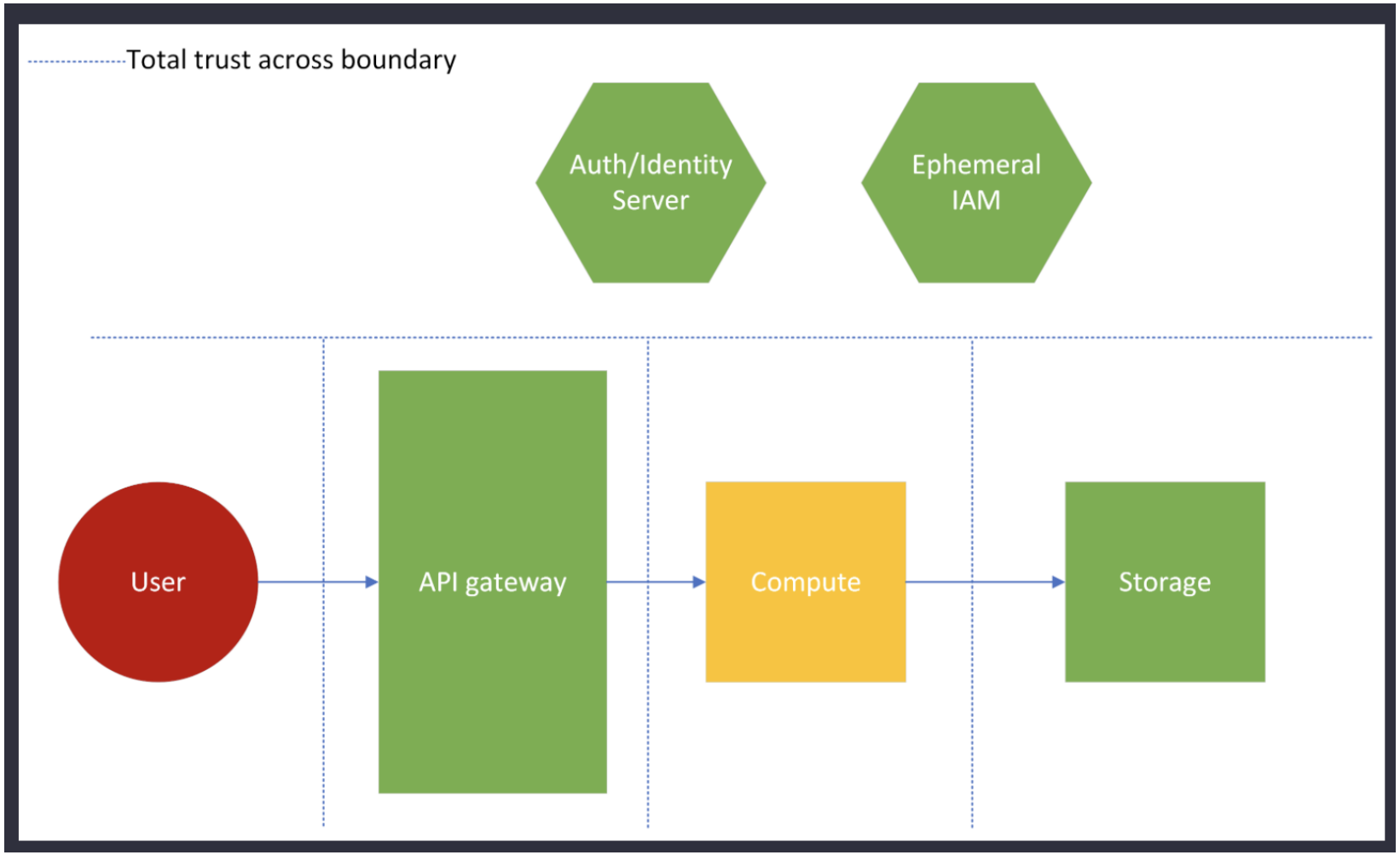


PASTA – Stage Six (Attack Analysis)



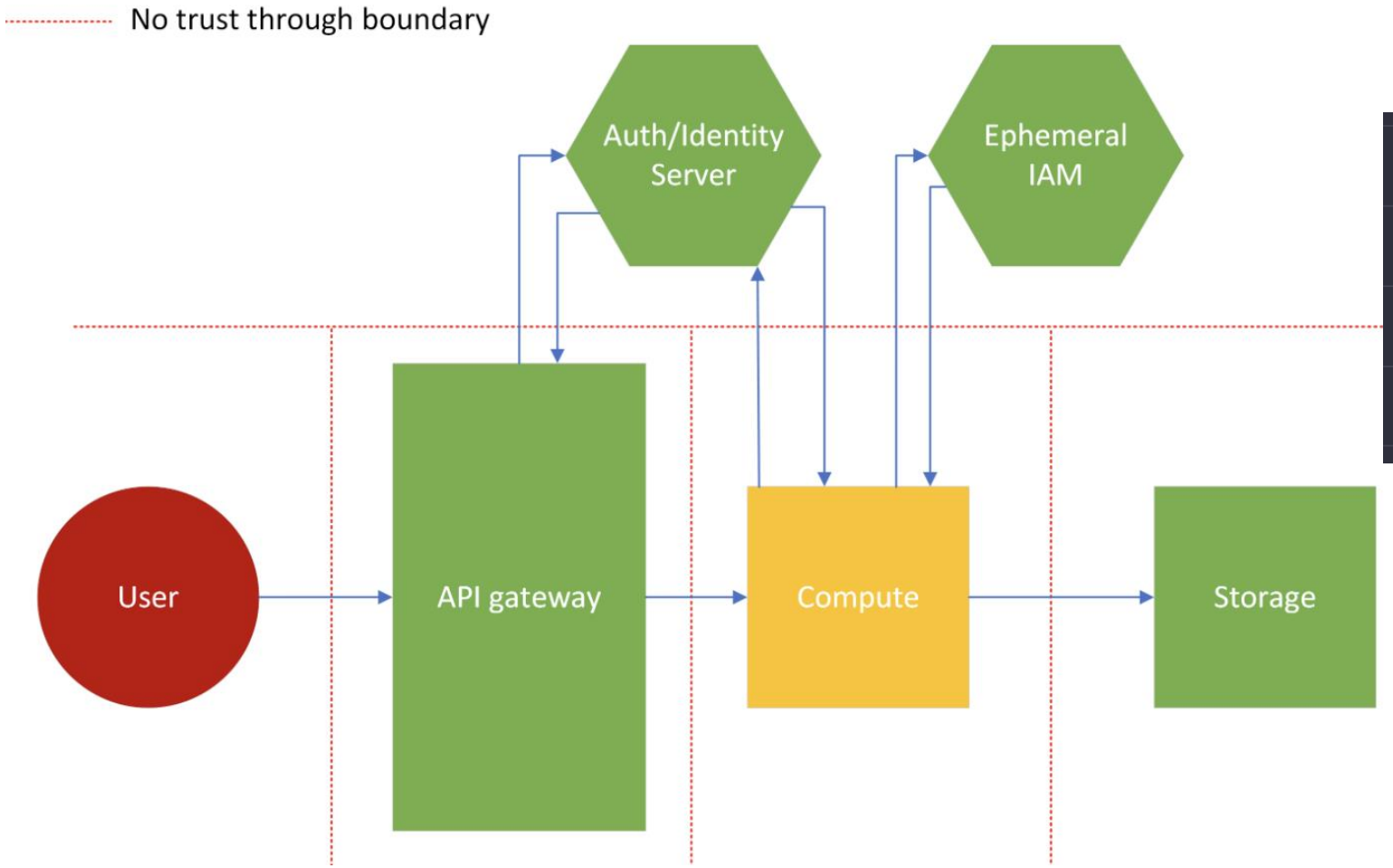
PASTA – Stage Seven (Risk and Impact Analysis)

Various levels of trusts – High Trust (low security)



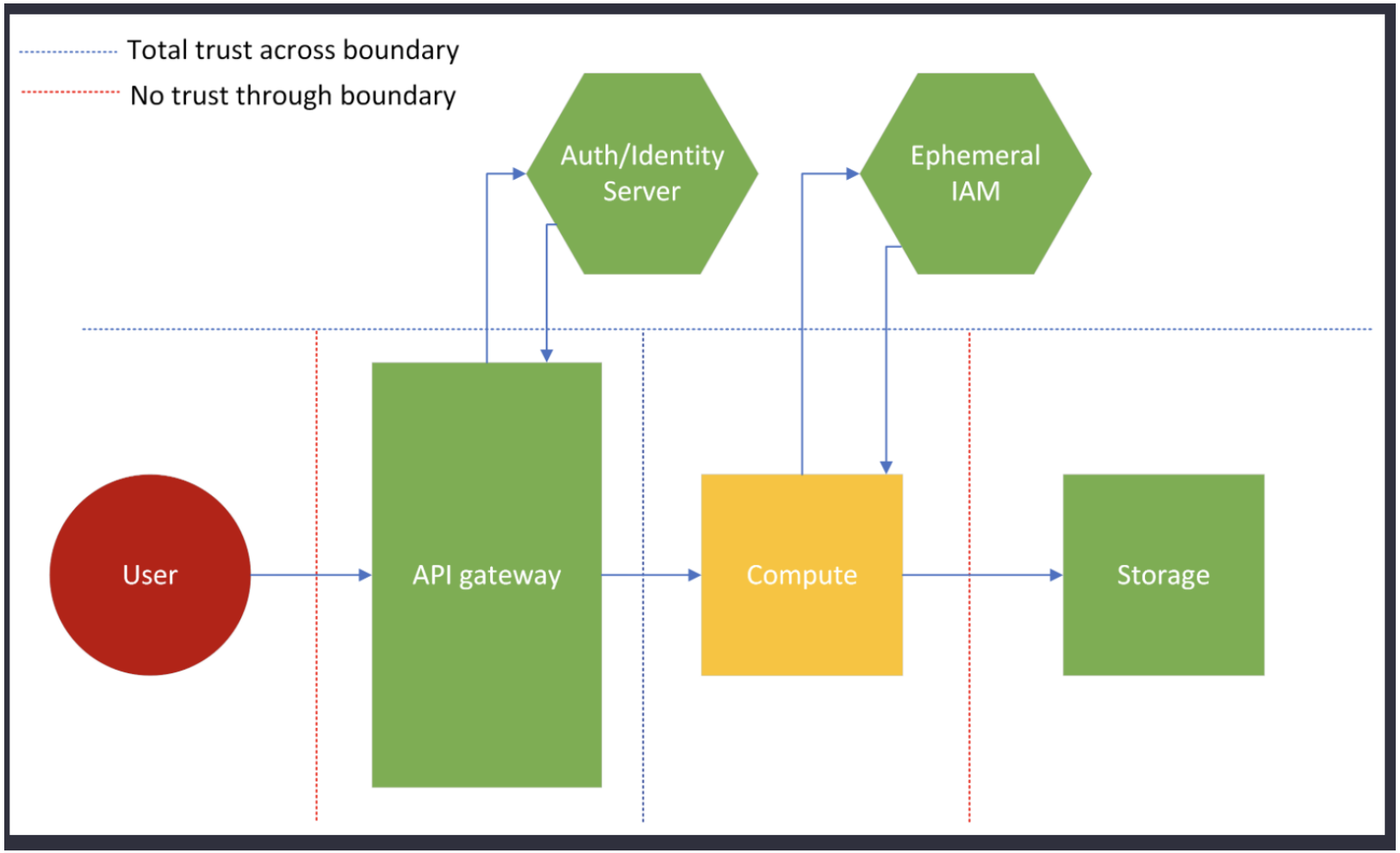
Pros	Cons
Efficient	Insecure
Simple	Potentially Wasteful
	High risk of compromise

Various levels of trusts – Zero Trust (highest security)



Pros	Cons
High assurance of data integrity	Slow and inefficient
Defense in depth	Complicated
	Likely more expensive

Various levels of trusts – Medium Trust



Pros	Cons
Secured based on risk	Known gaps in security
Cost/Efficiency derived from criticality	

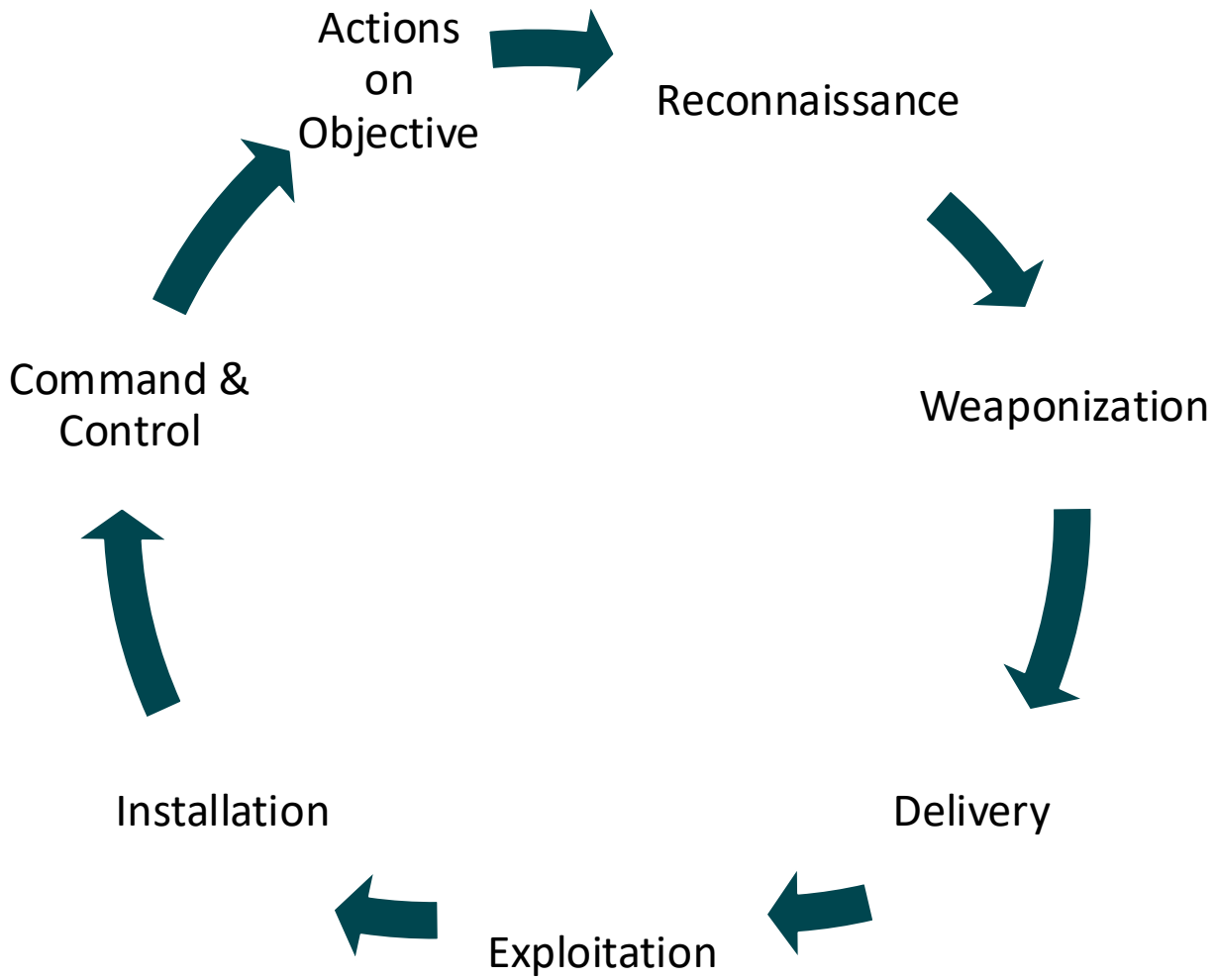
SYSTEM HACKING

Hacking a system is done in three steps. Attackers first collect enough information which can be used to gain access to the system. Once they have the right privileges, their goal is to maintain the access for as long as possible during which time they execute malicious programs, steal information, or simply tamper with the system. After they are done with their attack, attackers hide their tracks by modifying the system logs.

The objectives of system hacking are to:

- Gain access to the target system
- Escalate privileges
- Execute applications
- Hide files
- Cover tracks

Cyber Kill Chain

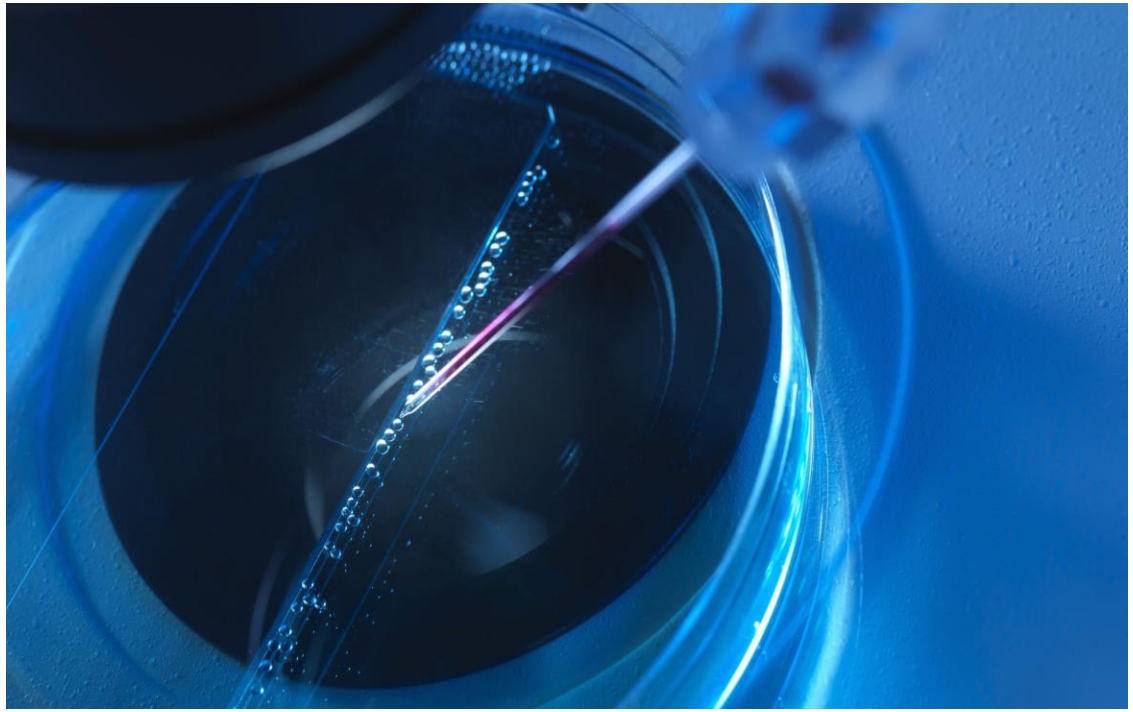


MITRE Attack Framework

Reconnaissance

- Using search engines to gather information
- Using social networking platforms
- Perform Google hacking
- Perform DNS interrogation
- Using social engineering

Scanning and Enumeration

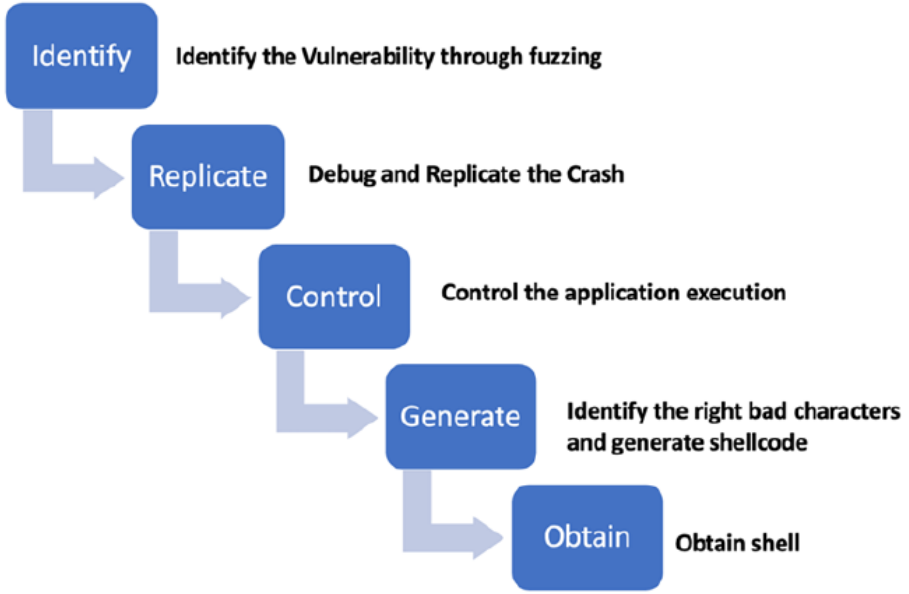


- Checking for any live systems
- Checking for firewalls and their rules
- Checking for open network ports
- Checking for running services
- Checking for security vulnerabilities
- Creating a network topology of the target network

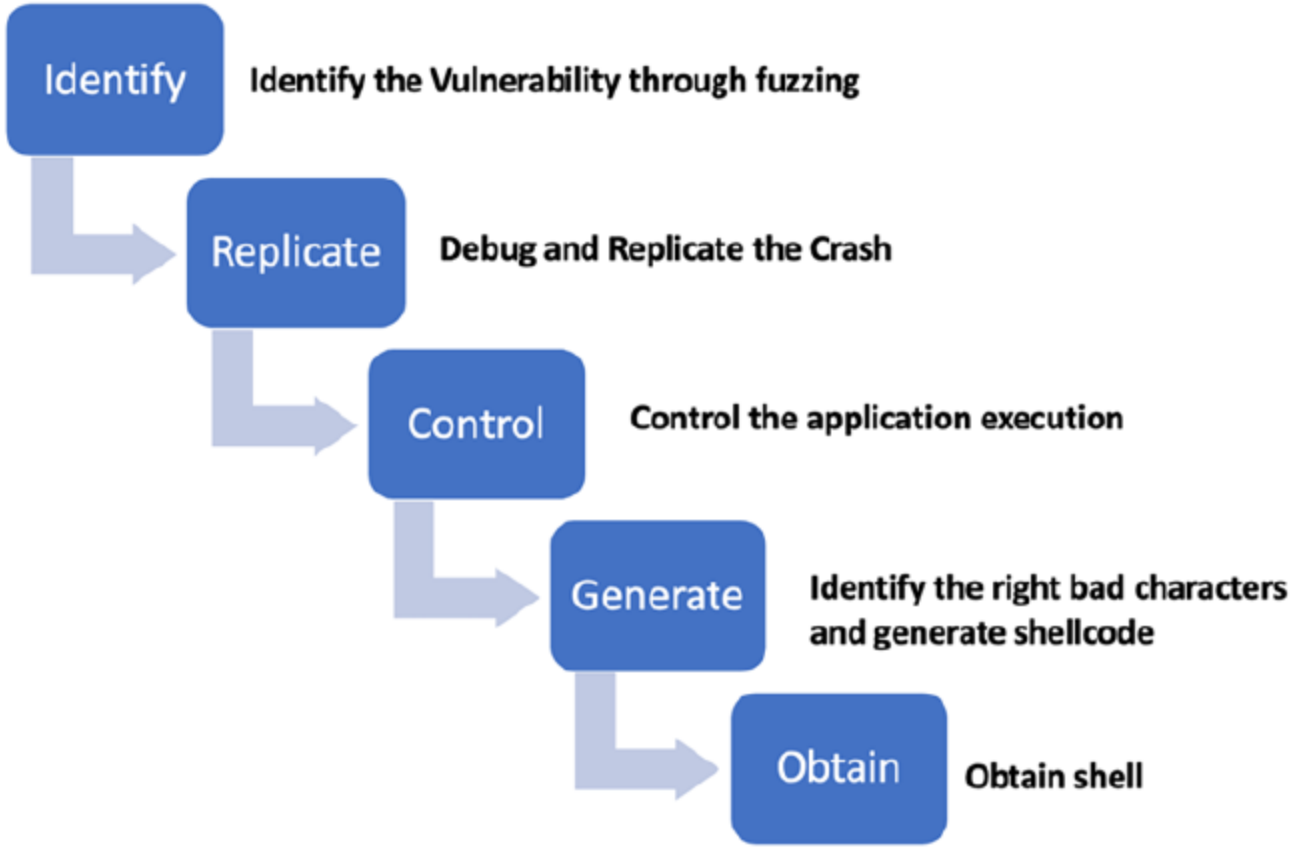
DNS Reconnaissance → Trace / Route Mapping → Enumerating the Edge → Scanning beyond Firewalls → Life Host Discovery and Port Scanning

Exploit the Vulnerability, Gaining Access

- Password Cracking
- Exploiting vulnerabilities
- Escalating privileges
- Hiding files



Custom Exploit Development



CRACKING PASSWORDS

Gaining Access

Cracking passwords refers to recovering passwords from the transmitted or stored data on computer systems

Non-electronic Attacks	Active Online Attacks	Passive Online Attacks	Offline Attacks
Dumpster Diving	Dictionary Attack	Wire Sniffing	Rainbow Table Attack
Shoulder Surfing	Brute Force Attack	MITM Attack	Distributed Network Attack
Social Engineering	Rule-based Attack	Replay Attack	
	Password Guessing		
	Trojan/Spyware/Keylogger		
	Hash Injection		
	LLMNR/NBT-NS Poisoning		

NON-ELECTRONIC ATTACKS

Gaining Access

Dumpster Diving	Dumpster diving is a technique which requires going through the target's trash bins, printer trash bins, and work desks and looking for notes or anything that can help in cracking the password.
Shoulder Surfing	Shoulder surfing refers to observing the target while they type in their passwords, that is, looking at their keyboard or screen.
Social Engineering	Social engineering requires the attacker to interact with the target and trick them into revealing their passwords.

ACTIVE ONLINE ATTACKS

Gaining Access

Dictionary Attack	Dictionary attack loads a dictionary file into a password cracking program which then checks the passwords against user accounts
Brute Force Attack	Brute force attack requires the attacker to run every combination of characters until the password is cracked
Rule-based Attack	Rule-based attack is used when the attacker has some information about the password, such as the length, if there are any digits, and similar
Password Guessing	In this attack, the attacker uses all information they have gathered about the target to create a list of possible passwords and then tries each password on the target's machine
Trojan/Spyware/Keylogger	Attackers install trojans, spyware, and keyloggers so that they could get the target's passwords and usernames
Hash Injection	Hash injection attack is an attack on systems that use hash functions for the user authentication
LLMNR/NBT-NS Poisoning	In this attack, attackers exploit the vulnerability when DNS fails to resolve name queries.

PASSIVE ONLINE ATTACKS

Gaining Access

Wire Sniffing	Wire sniffing is an attack in which attackers sniff credentials by capturing packets that are being transmitted. During the packet transmission, attackers are able to capture packets and extract sensitive information such as passwords and emails and thus gain access to the target system.
MITM Attack	Man-in-the-middle attack is an attack in which the attacker gains access to the communication channel between the target and server. Then, the attacker is able to extract information and data they need to gain unauthorized access.
Replay Attack	Replay attack involves using a sniffer to capture packets and authentication tokens. Once the relevant data is extracted, the tokens are placed back on the network in an attempt to gain unauthorized access.

OFFLINE ATTACKS

Gaining Access

Rainbow Table Attack	Rainbow table refers to a table of word and brute force lists and their hashes. The attack requires the attacker to create a rainbow table prior to the attack, and then use the information from the table to crack the password.
Distributed Network Attack	Distributed network attack utilizes the processing power of machines that are on the network in order to decrypt the password. This attack is used for recovering passwords from hashes. It works by installing a DNA manager in a central location from which it is possible to coordinate the attack by allocating portions of the key search to machines which are on the network.

ROOTKITS

Gaining Access

Rootkit is a program designed to help the attacker gain access to a system without being detected. It is designed to create a backdoor to the system and thus enable the attacker to access the system and perform malicious activities.

The objectives of a rootkit include:

- Gaining remote backdoor access
- Hiding traces of the attack
- Collect confidential data
- Install other malicious programs on the machine

ROOTKIT TYPES

Gaining Access

HYPERVISOR LEVEL ROOTKIT	Hypervisor level rootkit is designed to act as a hypervisor and load the target OS as a virtual machine
HARDWARE/FIRMWARE ROOTKIT	Hardware/Firmware rootkit is designed to conceal itself in hardware devices that are not inspected
KERNEL LEVEL ROOTKIT	Kernel level rootkit is designed to add malicious code or replace portions of the core operating system with some modified code
BOOTLOADER LEVEL ROOTKIT	Boot loader level rootkit is designed to replace the original bootloader with a malicious one
APPLICATION LEVEL ROOTKIT	Application level rootkit is designed to change the behavior of the target application
LIBRARY LEVEL ROOTKIT	Library level rootkit is designed to replace the original system calls in order to hide the attacker's activities

MALWARE

Gaining Access

Malware refers to a malicious program designed to cause damage to systems. Attackers use malware to gain access to target systems.

TROJAN	Trojan is a program which contains malicious code and has the ability to cause damage to the target system. They are contained inside seemingly harmless programs and activated when such programs are executed
CRYPTER	Crypter is a program which hides malware from antiviruses by encrypting the program's original binary code.
VIRUS	Virus is a program designed to replicate itself to other programs and documents on the infected machine. Viruses spread to other computers with the transfer of the infected files or programs. They are transmitted through file transfers, infected flash drives, and email attachments.
WORM	Worm is a program which replicates itself across network connections. Worms are designed to exploit vulnerabilities on the victim machines and then spread to other computers as the infected files are transferred.
RANSOMWARE	Ransomware is a type of malware in which hackers restrict access to files and folders on the target system until a payment is made. Victims are usually required to pay a certain sum of money in order to be able to access their files.

MALWARE ANALYSIS

Gaining Access

Malware analysis refers to a process of reverse engineering of a malware program. The purpose of the analysis is to determine how the malware works and assess the potential damage it could cause.

STATIC MALWARE ANALYSIS

Static analysis refers to analyzing the malware without running or installing it. The malware's binary code is examined to determine if there are any data structures or function calls that have malicious behavior.

DYNAMIC MALWARE ANALYSIS

Dynamic analysis requires the malware program to be running in a monitored environment, such as a sandbox or virtual machine. This type of analysis helps in understanding how the malware works by monitoring its activities on the system.

ANTI-MALWARE SOFTWARE

Gaining Access

Anti-malware software helps detect, prevent, and remove malware on the system. It also helps repair any damage that the malware may have caused.

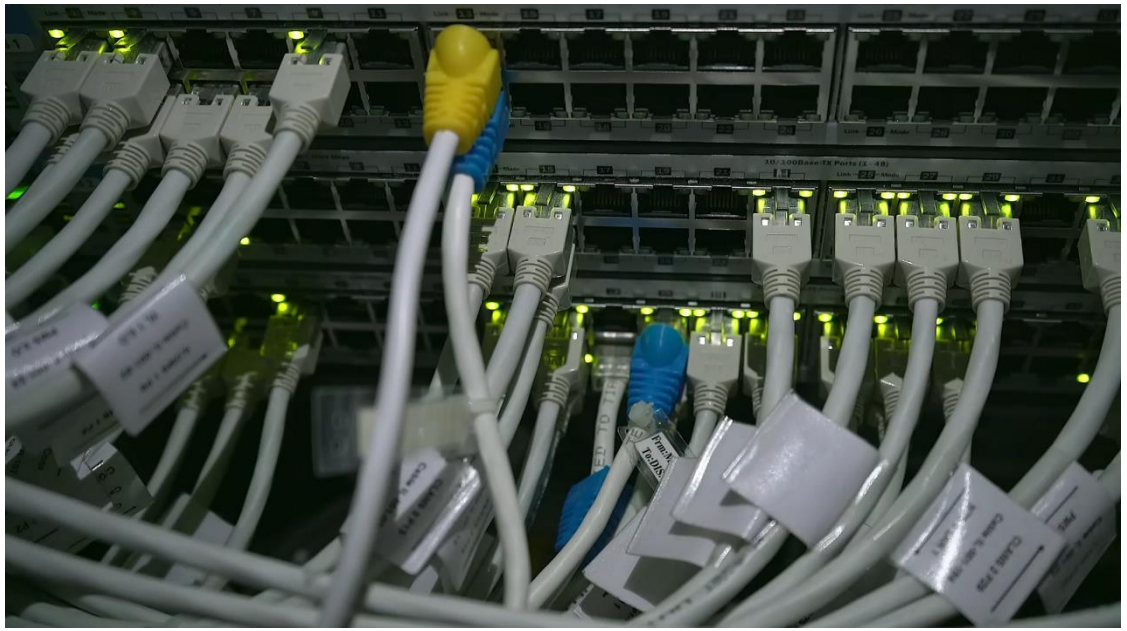
Signature-based Detection	Signature-based detection refers to comparing the hash of a suspicious code against a database of already known malware.
Behavior-based Detection	Behavior-based detection refers to detecting the malware based on its behavior and characteristics.
Sandboxing	Sandboxing refers to running unknown applications or possible threats in an isolated environment and monitoring the behavior. If the application appears to be malicious, then it gets terminated.

ANTIVIRUS vs ANTI-MALWARE

Gaining Access

ANTIVIRUS	ANTI-MALWARE
Antivirus is designed to detect and remove viruses from the system. It focuses more on already known and older viruses, trojans, and worms.	Anti-malware is designed to detect and protect the system from all malware, including viruses, worms, trojans, rootkits, and so on. The focus is shifted to the latest threats that could be even more dangerous than the ones that already exist.

Maintaining Access



- Lateral Movement
- Exfiltration of Data
- Creating backdoor and persistent connections

PRIVILEGE ESCALATION

Maintaining Access

Privilege escalation refers to taking advantage of the operating system and software vulnerabilities which enable the attacker to gain admin privileges. Becoming an admin on the target system allows the attacker to perform all sorts of malicious activities.

Horizontal Privilege Escalation	Vertical Privilege Escalation
Horizontal privilege escalation refers to acquiring the privileges of the same level.	Vertical privilege escalation refers to acquiring higher privileges.

REMOTE ACCESS

Maintaining Access

Once the attacker has gained access to the system and elevated privileges, they proceed to the next step in which they remotely execute malicious programs.

Programs that attackers install include:

- Backdoors to collect information and gain unauthorized access to the system
- Crackers to crack passwords
- Keyloggers to record keystrokes
- Spyware to capture screenshots and send them to the attacker

KEYLOGGERS

Maintaining Access

Keylogger is a program or hardware device designed to record every keystroke on the target's keyboard, logs them into a file, and sends them to a remote location.

Hardware Keylogger

Hardware keyloggers are devices that look like USB drives and are designed to record keystrokes, which are stored on the device. They are placed between a keyboard plug and USB socket and cannot be detected by antispyware or antivirus programs.

Software Keylogger

Software keyloggers are programs installed on the target's machine. Recorded keystrokes are logged into a log file on the target's machine which is then sent to the attacker using email protocols.

HARDWARE KEYLOGGERS

Maintaining Access

PC/BIOS EMBEDDED	PC/BIOS Embedded keylogger refers to modifying the BIOS level firmware to capture the keystrokes
KEYLOGGER KEYBOARD	Keylogger keyboard refers to attaching the hardware circuit with the keyboard cable connector
EXTERNAL KEYLOGGER	External keylogger refers to attaching the keylogger between a keyboard and computer

SOFTWARE KEYLOGGERS

Maintaining Access

APPLICATION KEYLOGGER	Application keylogger is designed to observe the target's activity whenever type something
KERNEL, ROOTKIT, DEVICE DRIVER KEYLOGGERS	Kernel keylogger is designed to exist on a kernel level and act as a keyboard device driver, which allows it to record everything that is typed on the keyboard. Rootkit keylogger refers to a forged Windows device driver which records keystrokes. Device driver keylogger is designed to replace the driver that has the keylogging functionality, logs the keystrokes, and send the file to a remote location
HYPERVISOR BASED KEYLOGGER	Hypervisor-based keylogger is designed to work within a malware hypervisor that is operating on the OS
FORM GRABBING BASED KEYLOGGER	Form grabbing based keylogger is designed to record web browsing when the Submit event is triggered

SPYWARE

Maintaining Access

Spyware is a stealthy program designed to record the target's interaction with the computer and Internet and then send the recorded data to the attacker. This program is also able to take and send screenshots. The program is hidden when installed

COVERING TRACKS

Covering Tracks

Covering tracks is a phase in which the attacker attempts to hide their presence on the system. To avoid detection, the attacker needs to modify the system logs and delete their activity during the attack, and also ensure that future activities are not logged too. It is very important that the system appears uncompromised.

REVERSE HTTP SHELLS	Reverse HTTP shells are designed to ask the master system for commands which, when received, are executed on the target's machine
REVERSE ICMP TUNNELS	Reverse ICMP tunnel is a technique in which the attacker accesses the system by using ICMP echo and reply packets as carriers of TCP payload
DNS TUNNELING	DNS tunneling refers to adding data payload to the target's DNS server in order to create a back channel through which it is possible to steal information from the server
TCP PAREMETERS	TCP parameters refers to using TCP parameters for payload distribution. Fields in which data can be hidden are IP identification field, TCP acknowledgement number, and TCP initial sequence number

SNIFFING

Maintaining Access

Packet sniffing refers to the process of capturing data packets on a network using a program or a device.

Packet sniffing programs are called sniffers and they are designed to capture packets that contain information such as passwords, router configuration, and traffic.

Passive Sniffing

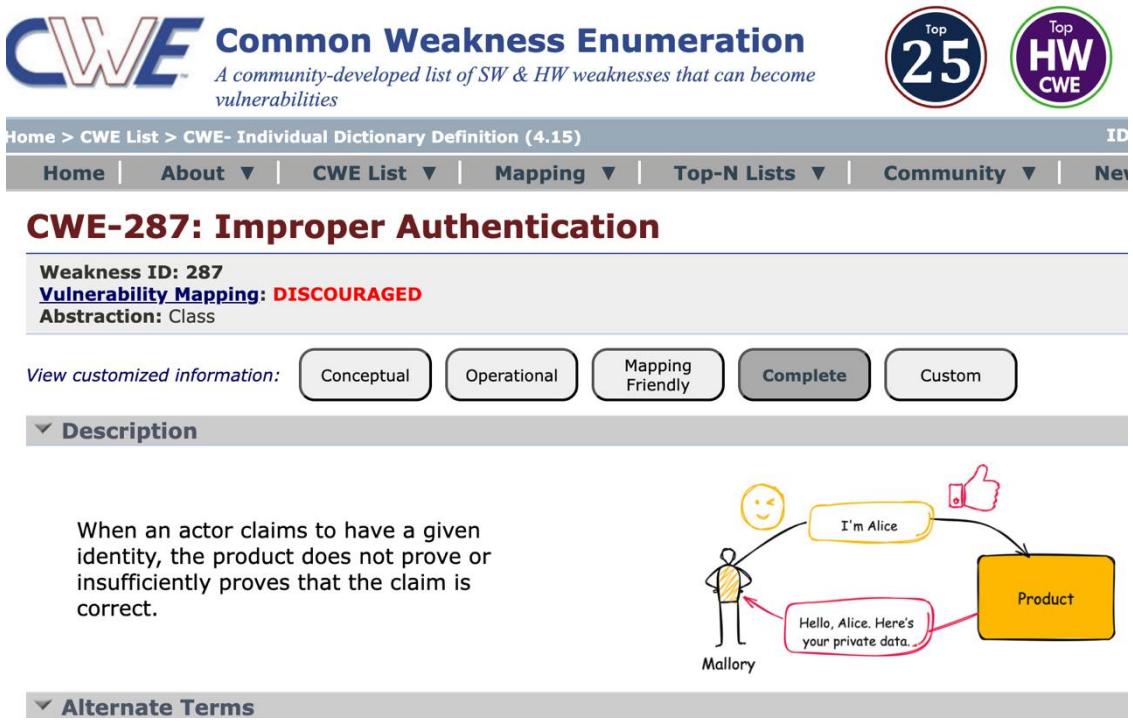
Passive sniffing is used in networks which use hubs to connect systems. Such networks allow their hosts to see all the traffic passing through the network, which makes it easy for attackers to capture that traffic. Passive sniffing does not require any packets to be sent. Instead, the packets coming into the network are monitored and captured.

Active Sniffing

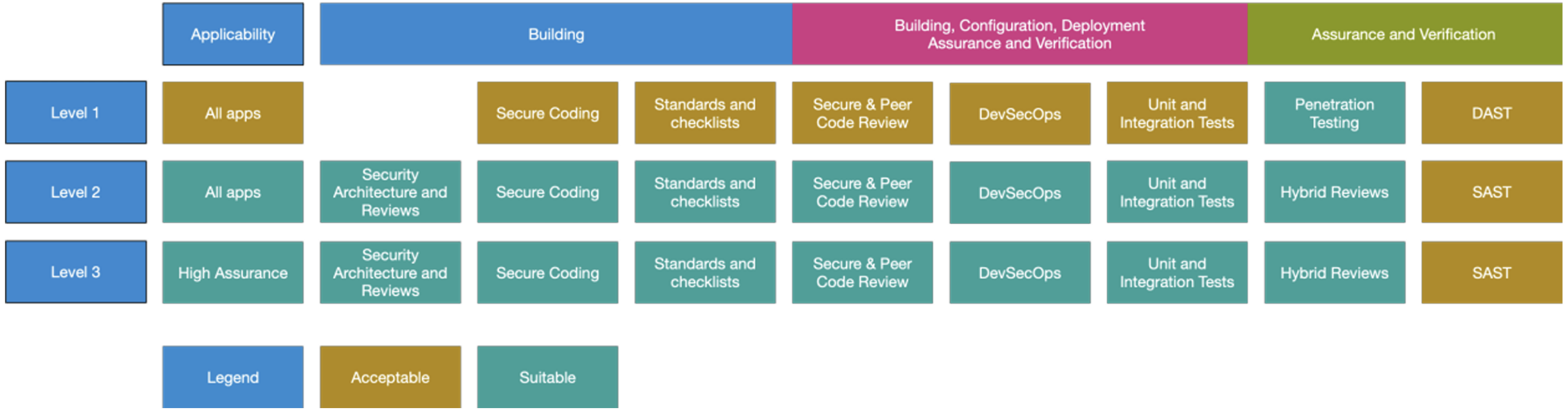
Active sniffing is used in networks which use switches. Switches require a packet to have a source and destination addresses in order to be sent to its destination, which makes it more difficult for attackers to sniff. However, it is possible to sniff a switch by flooding its Content Addressable Memory (CAM) table, which contains the information of which port belongs to which host. Flooding the CAM table is done by actively injecting ARP traffic into a LAN and then capturing the traffic.

Some useful sources

- https://owasp.org/www-community/Threat_Modeling_Process
- OWASP ASVS
- STRIDE GPT
- MITRE Common Weakness Enumeration https://cwe.mitre.org/data/index.html to determine vulnerabilities
- OWASP Top 10



Another guidance for threat modelling and secure software development – OWASP ASVS



V1.2 Authentication Architecture

When designing authentication, it doesn't matter if you have strong hardware enabled multi-factor authentication if an attacker can reset an account by calling a call center and answering commonly known questions. When proving identity, all authentication pathways must have the same strength.

#	Description	L1	L2	L3	CWE
1.2.1	Verify the use of unique or special low-privilege operating system accounts for all application components, services, and servers. (C3)		✓	✓	250
1.2.2	Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed. (C3)		✓	✓	306

Check it out yourself…
https://github.com/OWASP/ASVS/raw/v4.0.3/4.0/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.pdf

Other sources

- Specifically for Cloud, AWS:
- https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html?did=wp_card&trk=wp_card
- AWS Well-Architected Framework, Security Pillar
- Check it out…

Now it’s practice time…

- Start designing a web application or decompose an existing application; try following a more or less PASTA-oriented approach
  - Gather use cases, requirements (functional and non-functional)
  - Set up an architecture diagram, make design choices and argue why you took the decision
  - Determine the boundaries and external connections
  - Determine the data flows and trust boundaries
  - Determine the assets, assess their value, protection level
  - Model the vulnerabilities
  - Propose an architecture; cloud-native (or hybrid / on prem); provider of your choice
  - Example for AWS architecture: https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/high-performance-computing-simulation-workload-for-renewable-energy-data-ra.pdf?did=wp_card&trk=wp_card
  - Make specific suggestions to address the vulns, harden your architecture

Now it’s practice time…

- To get ideas for hardening your architecture, use all your background knowledge we gathered so far and use frameworks such as OWASP ASVS as orientation, e.g. topics like
  - Authentication
  - Authorization / Access Control
  - API Protection (Sanitization for instance) and Session Management
  - Data at rest and data in transit protection and Key Management
  - Supply Chain Security and malicious code protection
  - Logging and Monitoring
  - Backup & Recovery

Now it’s practice time…

- Use sources like OWASP Threat Modelling https://owasp.org/www-community/Threat_Modeling,
- Also, sources like https://cwe.mitre.org/data/index.html may help you in the process to gather vulnerabilities
- Maybe STRIDE GPT https://stridegpt.streamlit.app/ can help you getting into some dialog-oriented exploration mode
- Any online available source that may help you

Now it’s practice time…

- Make reasonable assumptions for your use case
- Perform a scientific presentation; 20 mins max presentation, 10 mins questions
  - Intro
  - Background if there are any specific settings / different cloud provider etc.
  - Overview of the process
  - Details of the process phases
  - Make a deep dive in one of the sections (e.g. a specific module with details of vulns and control mechanisms)
  - Conclusion