
THREAT MODELING
CONNECT
THREAT MODCON 2023
THREAT MODELING IS FOR EVERYONE

THREAT MODCON 2023

EVERYONE IS A THREAT MODELER: AN AI-ENABLED JOURNEY FOR BEGINNERS

Wael Ghandour

THREAT MODCON 2023

Wael Ghandour

Security Engineer

Curious where this AI thing is going


wael@bitsavant.com


https://linkedin.com/in/waelsv

HELLO OLD FRIEND...

THREAT MODCON 2023

THE OSI MODEL

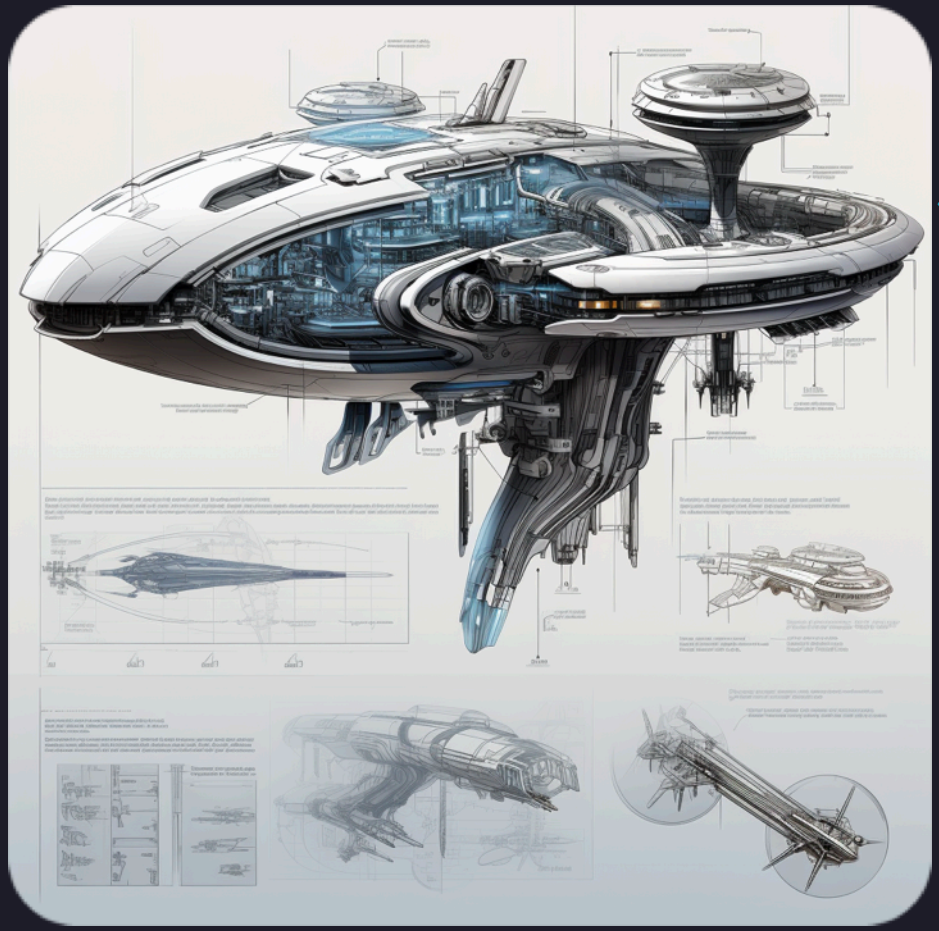
Works well for networking but was looking for a general-purpose equivalent for systems…

…namely to break things down before trying to secure them.

THREAT MODCON 2023

BREAKING THINGS DOWN

INTRODUCING CCCS



Components of useful systems are organized in a fairly repetitive pattern, one that holds at various scales.

THREAT MODCON 2023

Control

Compute

Communication

Storage

THREAT MODCON 2023

Control

Responsible for managing and orchestrating the overall system's resources and configurations.

MANAGEMENT CONSOLES, APIS, CONTROL PLANE ELEMENTS

POLICY ENFORCEMENT, TASK COORDINATION, PERFORMANCE MONITORING

THREAT MODCON 2023

Compute 

Encompasses the processing power and execution environment.

PHYSICAL SERVERS, VIRTUAL MACHINES, CONTAINERS, SERVERLESS

RUN APPLICATIONS AND WORKLOADS

THREAT MODCON 2023

Communication

Handles the exchange of data and information between the different components and systems.

NETWORKING PROTOCOLS, IPC MECHANISMS, LOAD BALANCERS

ENABLE DATA TRANSMISSION AND FACILITATE COMMUNICATION

THREAT MODCON 2023

Storage 

Responsible for managing data storage and retrieval.

FILE SYSTEMS, BLOCK STORAGE, OBJECT STORAGE, DATABASES

PERSIST, ORGANIZE, AND ACCESS DATA

THREAT MODCON 2023

Let’s see how it works in practice.

THREAT MODCON 2023

CCCS BREAKDOWN

MICROPROCESSOR

DECOMPOSITION USING CCCS

THREAT MODCON 2023

Control

The Control Unit (CU) manages instruction execution, while the Clock synchronizes operations.

Compute

The Arithmetic Logic Unit (ALU) performs arithmetic and logical operations.

Communication

Buses connect internal components, and I/O interfaces handle external communication.

Storage

Registers store data for computations, and the Cache holds frequently accessed data and instructions.

THREAT MODCON 2023

CCCS BREAKDOWN

OPERATING SYSTEM

DECOMPOSITION USING CCCS

THREAT MODCON 23

Control

The kernel and system-level services manage resources, coordinate tasks, and enforce policies (e.g., scheduling, access control, and security).

Compute

The operating system provides an execution environment for applications and services, allocating CPU time and managing the execution of processes.

Communication

Inter-process communication (IPC) mechanisms (e.g., pipes, message queues, and shared memory) enable data exchange between processes, while the networking stack manages data transmission between the system and other networked devices.

Storage

The OS handles file systems and storage devices, providing a hierarchical structure for organizing data and managing storage resources (e.g., hard drives, SSDs).

THREAT MODCON 2023

CCCS BREAKDOWN

MODERN CLOUD SERVICE

DECOMPOSITION USING CCCS

THREAT MODCON 23

Control

Cloud service management platforms and tools (e.g., AWS Management Console, Azure Portal) provide control over resource provisioning, configuration, monitoring, and security.

Compute

Cloud services offer various compute options, from virtual machines (e.g., EC2 and Azure VMs) and containers to server-less functions (e.g., AWS Lambda and Azure Functions).

Communication

Networking services (e.g., VPCs, virtual networks, and load balancers) enable communication between cloud resources and external systems.

Storage

Cloud storage options include object storage (e.g., Amazon S3 and Azure Blob Storage), block storage (e.g., Amazon EBS and Azure Disk Storage), file storage (e.g., Amazon EFS and Azure Files), and databases (e.g., Amazon RDS and Azure SQL Database).

Great! We now have a framework that provides a convenient starting point for analyzing a system we may not be familiar with.

THREAT MODCON 2023

It is also:

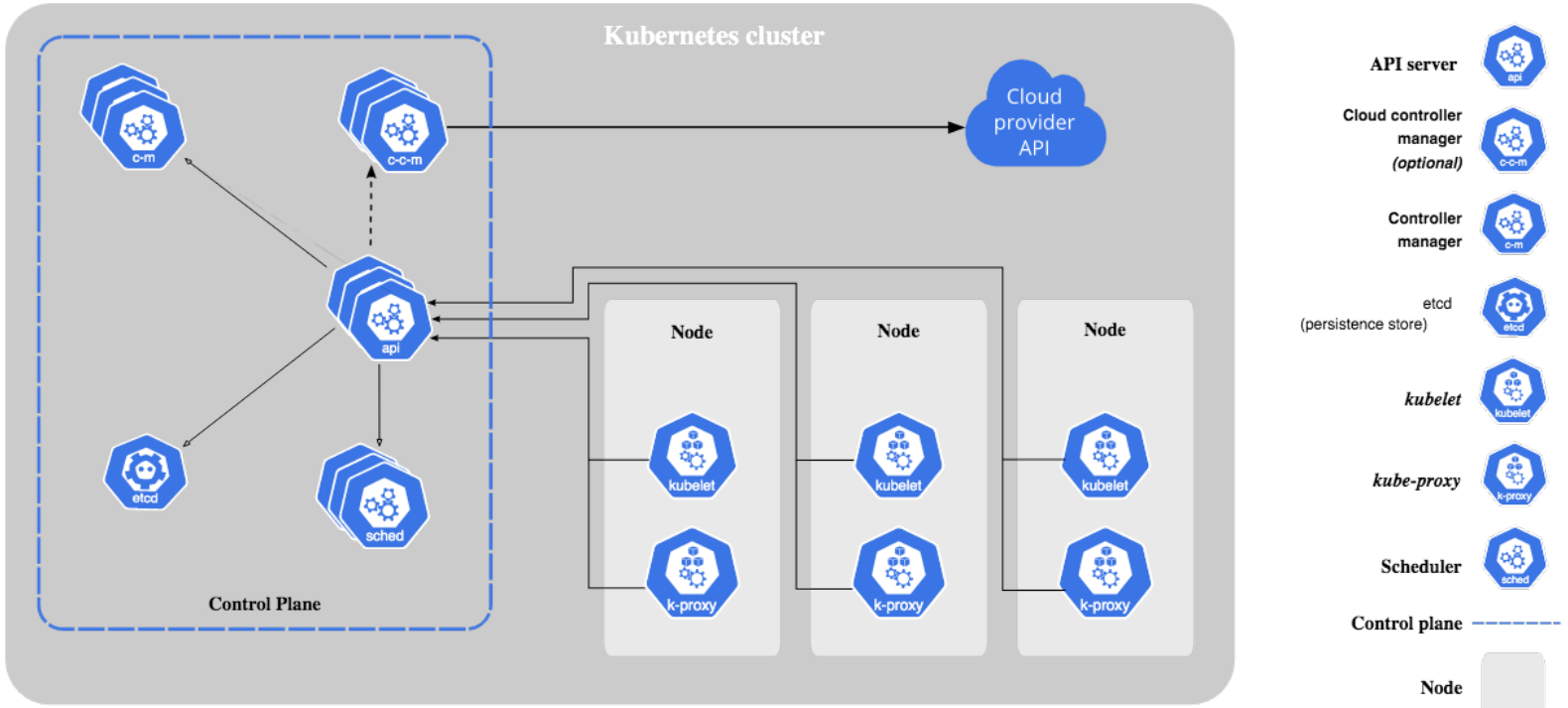
Structured	Predictable analysis
Modular	Allows for isolation and specialization
Scaleable	Provides flexibility

Great as a learning tool, but tedious to operationalize.
Time to leverage AI for that.

THREAT MODCON 2023

CCCS APPLIED

BREAKING DOWN KUBERNETES



SOURCE: HTTPS://KUBERNETES.IO/DOCS/CONCEPTS/OVERVIEW/COMPONENTS/

CONTROL

This layer deals with the management and orchestration of the system. In Kubernetes, the control layer is primarily managed by the control plane components.

THREAT MODCON 23

API Server

Serves as the entry point for commands and queries. It's the main interaction point for administrators and users with the cluster.

etcd

A consistent and highly-available key-value store used as Kubernetes' backing store for all cluster data.

Kube-controller-manager

Manages different controllers that regulate the state of the system, ensuring the current state matches the desired state.

Kube-scheduler

Assigns work, in the form of pods, to worker nodes based on multiple factors such as resource availability and user-defined constraints.

ACTUAL GPT4 OUTPUT

COMPUTE

This layer is responsible for executing and running the workloads. In Kubernetes, the compute layer is primarily represented by the nodes and the pods running on them.

THREAT MODCON 23

Nodes

These are the worker machines, VMs, or physical computers that run the workloads. Each node is managed by the control plane.

Kubelet

An agent that runs on each node in the cluster. It ensures that containers are running in a Pod.

Pods

The smallest deployable units in Kubernetes that can be created and managed. They encapsulate one or more containers.

COMMUNICATION

This layer ensures seamless communication within the components and also with external entities. Kubernetes has a robust networking model.

THREAT MODCON 23

Service

An abstract way to expose an application running on a set of Pods as a network service.

Ingress

Manages external access to the services in a cluster, typically HTTP.

Network Policies

Define how pods communicate with each other and with other network endpoints.

Container Network Interface (CNI)

Plugins that are responsible for connecting containers to the host network.

STORAGE

This layer deals with data storage and management. Kubernetes provides persistent storage solutions.

THREAT MODCON 23

Persistent Volumes (PVs)

Provides an API for users and administrators that abstracts details of how storage is provided and consumed.

Persistent Volume Claims (PVCs)

A user's request for storage resources.

Storage Classes

Allow administrators to describe the "classes" of storage they offer.

StatefulSets

Workloads that need a persistent identity or persistent storage.

THREAT MODCON 2023

CCCS FOUNDATION

BUILDING THE THREAT-MODEL

With the decomposition in hand, we can now analyze the components of each layer for threats.

Based on the decomposition results, we asked GPT4 to conduct a threat-model.

Here's what we got.

CONTROL

THREAT MODCON 2023

API Server

1. Unauthorized API Access

- **Description:** Attackers gaining unauthorized access to the Kubernetes API.
- **Evidence:** The API Server is the main entry point for commands and queries in a Kubernetes cluster.
- **Threat Likelihood:** Medium, depending on the configuration and access controls.
- **STRIDE Category:** Tampering, Information Disclosure
- **ASF Category:** Spoofing, Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Unauthorized access can lead to cluster-wide disruptions and data breaches.
- **Countermeasures:** Implement RBAC, use strong authentication methods, and regularly audit access logs.
- **Practical Attack Example:** An attacker exploiting weak credentials to access the API and retrieve secrets.

2. Denial of Service (DoS)

- **Description:** Attackers overwhelming the API server with requests, causing it to become unresponsive.
- **Evidence:** The API Server handles all requests to the cluster, making it a prime target for DoS.
- **Threat Likelihood:** Medium, depending on the rate limiting and security configurations.
- **STRIDE Category:** Denial of Service
- **ASF Category:** Denial of Service
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** A DoS attack can disrupt cluster operations but may not lead to data breaches.
- **Countermeasures:** Implement rate limiting, use a WAF, and monitor traffic patterns.
- **Practical Attack Example:** An attacker flooding the API server with requests using a botnet.

CONTROL

THREAT MODCON 2023

etcd

1. Data Tampering

- **Description:** Unauthorized modifications to the data stored in etcd.
- **Evidence:** etcd is the primary key-value store for Kubernetes, storing all cluster data.
- **Threat Likelihood:** Medium, if etcd is not properly secured.
- **STRIDE Category:** Tampering
- **ASF Category:** Tampering
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Tampering with etcd data can disrupt the entire cluster.
- **Countermeasures:** Secure etcd access, enable encryption, and regularly backup data.
- **Practical Attack Example:** An attacker modifying service configurations in etcd to redirect traffic.

2. Data Exfiltration

- **Description:** Unauthorized extraction of data from etcd.
- **Evidence:** etcd contains sensitive cluster data, including secrets and configurations.
- **Threat Likelihood:** Medium, depending on access controls and encryption.
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Exfiltration of data can lead to data breaches and further attacks.
- **Countermeasures:** Encrypt data at rest and in transit, monitor access patterns, and implement strict access controls.
- **Practical Attack Example:** An attacker accessing etcd to retrieve stored secrets.

CONTROL

THREAT MODCON 23

Controller Manager

1. Malicious Controller Actions

- **Description:** If an attacker gains control over the Controller Manager, they can initiate malicious actions.
- **Evidence:** The Controller Manager automates and manages various cluster operations.
- **Threat Likelihood:** Low to Medium, depending on the security posture.
- **STRIDE Category:** Tampering
- **ASF Category:** Tampering, Elevation of Privilege
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** While gaining control is challenging, successful attacks can disrupt cluster operations.
- **Countermeasures:** Monitor the Controller Manager's actions, secure its access, and regularly update Kubernetes.
- **Practical Attack Example:** An attacker exploiting a vulnerability in the Controller Manager to delete pods.

2. Misconfiguration Exploits

- **Description:** Exploiting misconfigurations in the Controller Manager to disrupt cluster operations.
- **Evidence:** Misconfigurations can expose vulnerabilities or unintended behaviors.
- **Threat Likelihood:** Medium, depending on the configuration and monitoring.
- **STRIDE Category:** Tampering
- **ASF Category:** Tampering
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Misconfigurations can lead to disruptions but may not always result in data breaches.
- **Countermeasures:** Regularly audit configurations, apply best practices, and monitor for anomalies.
- **Practical Attack Example:** An attacker exploiting a misconfiguration to gain elevated privileges.

CONTROL

THREAT MODCON 23

Scheduler

1. Malicious Pod Placement

- **Description:** Manipulating the scheduler to place pods on specific nodes for malicious intent.
- **Evidence:** The scheduler determines where to run pods based on resources and constraints.
- **Threat Likelihood:** Medium, if the scheduler is compromised or misconfigured.
- **STRIDE Category:** Tampering
- **ASF Category:** Tampering
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Manipulating pod placement can lead to targeted attacks on specific nodes.
- **Countermeasures:** Monitor scheduler decisions, secure its access, and ensure it's updated regularly.
- **Practical Attack Example:** An attacker influencing the scheduler to place a malicious pod on a node with sensitive data.

2. Denial of Service (DoS) through Scheduler Exhaustion

- **Description:** Overloading the scheduler with numerous pod placement requests, causing it to become unresponsive or make inefficient decisions.
- **Evidence:** The scheduler is responsible for determining the placement of pods, and overwhelming it can disrupt its operations.
- **Threat Likelihood:** Medium, depending on rate limiting and monitoring mechanisms.
- **STRIDE Category:** Denial of Service
- **ASF Category:** Denial of Service
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** A DoS attack can disrupt cluster operations but may not lead to data breaches.
- **Countermeasures:** Implement rate limiting, monitor scheduler activity, and set up alerts for unusual patterns.
- **Practical Attack Example:** An attacker flooding the scheduler with pod placement requests to exhaust its resources.

COMPUTE

THREAT MODCON 2023

Nodes

1. Unauthorized Node Registration

- **Description:** Malicious nodes registering with the cluster to gain access to workloads and data.
- **Evidence:** Nodes are the worker machines in Kubernetes where containerized applications run.
- **Threat Likelihood:** Medium, if proper authentication mechanisms aren't in place.
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Unauthorized nodes can access sensitive data and workloads.
- **Countermeasures:** Use mutual TLS for node authentication and regularly audit registered nodes.
- **Practical Attack Example:** An attacker registering a compromised node to intercept network traffic.

2. Node Misconfiguration

- **Description:** Nodes that are improperly configured can expose vulnerabilities.
- **Evidence:** Nodes require proper configuration to ensure security.
- **Threat Likelihood:** High, especially in large clusters with multiple nodes.
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Misconfigured nodes can lead to data breaches.
- **Countermeasures:** Regularly audit node configurations and apply security best practices.
- **Practical Attack Example:** An attacker exploiting a misconfigured node to access sensitive data.

COMPUTE

THREAT MODCON 2023

Kubelet

1. Kubelet API Exploitation

- **Description:** Unauthorized access to the Kubelet API, leading to control over node operations.
- **Evidence:** Kubelet has an API that can be accessed if not properly secured.
- **Threat Likelihood:** Medium, if the API is exposed and not secured.
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Gaining control over Kubelet can lead to control over node operations.
- **Countermeasures:** Secure the Kubelet API with authentication and limit its exposure.
- **Practical Attack Example:** An attacker using the Kubelet API to start or stop pods maliciously.

2. Kubelet Credentials Compromise

- **Description:** Theft of Kubelet credentials, allowing unauthorized access.
- **Evidence:** Kubelet requires credentials to communicate with the API server.
- **Threat Likelihood:** Medium, if credentials are stored insecurely.
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Stolen credentials can lead to unauthorized control over Kubelet.
- **Countermeasures:** Store credentials securely and rotate them regularly.
- **Practical Attack Example:** An attacker using stolen credentials to impersonate Kubelet.

COMPUTE

THREAT MODCON 23

Containers

1. Container Breakout

- **Description:** Escaping the container to gain access to the host or other containers.
- **Evidence:** Containers are isolated environments, but vulnerabilities can lead to breakout.
- **Threat Likelihood:** Medium, depending on the container runtime and its configuration.
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** A successful breakout can compromise the entire node.
- **Countermeasures:** Use secure container runtimes, regularly update them, and apply security best practices.
- **Practical Attack Example:** An attacker exploiting a vulnerability in the container runtime to access the host system.

2. Insecure Container Images

- **Description:** Using outdated or vulnerable container images.
- **Evidence:** Containers are created from images, which can have vulnerabilities.
- **Threat Likelihood:** High, especially if images aren't regularly updated.
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Vulnerable images can lead to breaches within the container.
- **Countermeasures:** Regularly update images and scan them for vulnerabilities.
- **Practical Attack Example:** An attacker exploiting a known vulnerability in an outdated container image.

COMPUTE

THREAT MODCON 2023

Pods

1. Inter-Pod Traffic Snooping

- **Description:** Intercepting traffic between pods to gain unauthorized information.
- **Evidence:** Pods can communicate with each other, and this traffic can be intercepted if not encrypted.
- **Threat Likelihood:** Medium, if network policies and encryption aren't applied.
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Intercepted traffic can lead to data breaches.
- **Countermeasures:** Use network policies and encrypt inter-pod traffic.
- **Practical Attack Example:** An attacker in a compromised pod snooping on traffic to another pod.

2. Pod Resource Exhaustion

- **Description:** Consuming all resources allocated to a pod, causing it to crash or become unresponsive.
- **Evidence:** Pods have resource limits, and exceeding them can lead to issues.
- **Threat Likelihood:** Medium, especially if resource limits aren't set or monitored.
- **STRIDE Category:** Denial of Service
- **ASF Category:** Denial of Service
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Resource exhaustion can disrupt services but may not lead to data breaches.
- **Countermeasures:** Set and monitor pod resource limits and use horizontal pod autoscaling.
- **Practical Attack Example:** An attacker running processes in a pod to consume all its allocated memory.

COMMUNICATION

THREAT MODCON 2023

Service

1. Service Exposure

- **Description:** Exposing services unintentionally to the public.
- **Evidence:** Services can be exposed using NodePort or LoadBalancer types.
- **Threat Likelihood:** High, especially if not properly configured.
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Unintended exposure can lead to unauthorized access.
- **Countermeasures:** Ensure services are properly configured and use internal types when necessary.
- **Practical Attack Example:** An attacker accessing a database service that was unintentionally exposed.

2. Service Spoofing

- **Description:** Malicious services impersonating legitimate ones.
- **Evidence:** Services can be created by any user with the right permissions.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Spoofing
- **ASF Category:** Spoofing
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Impersonation can lead to data interception.
- **Countermeasures:** Implement RBAC and monitor service creation activities.
- **Practical Attack Example:** An attacker creating a service that mimics a legitimate one to intercept data.

COMMUNICATION

THREAT MODCON 2023

Ingress

1. Insecure Ingress Configuration

- **Description:** Exposing sensitive applications due to misconfigured ingress rules.
- **Evidence:** Ingress controllers and resources define how traffic should be routed.
- **Threat Likelihood:** High, if not properly secured.
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Misconfigurations can lead to unauthorized access.
- **Countermeasures:** Regularly audit ingress configurations and apply security best practices.
- **Practical Attack Example:** An attacker accessing sensitive applications due to lax ingress rules.

2. SSL/TLS Misconfiguration

- **Description:** Weak or misconfigured SSL/TLS settings in ingress.
- **Evidence:** Ingress often handles SSL/TLS termination for applications.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Weak encryption can lead to data interception.
- **Countermeasures:** Ensure strong ciphers are used and certificates are valid.
- **Practical Attack Example:** An attacker exploiting weak SSL/TLS configurations to intercept data.

COMMUNICATION

THREAT MODCON 2023

Network Policies

1. Lack of Network Policies

- **Description:** Not defining network policies, leading to open communication between pods.
- **Evidence:** Network policies define how pods communicate with each other.
- **Threat Likelihood:** High, if not defined.
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Open communication can lead to lateral movement in attacks.
- **Countermeasures:** Define granular network policies to restrict unnecessary communication.
- **Practical Attack Example:** An attacker moving laterally between pods due to lack of network segmentation.

2. Misconfigured Network Policies

- **Description:** Incorrectly defined network policies that don't restrict communication as intended.
- **Evidence:** Even with network policies in place, misconfigurations can occur.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Misconfigurations can lead to unintended communication paths.
- **Countermeasures:** Regularly audit and test network policies.
- **Practical Attack Example:** An attacker exploiting misconfigured network policies to communicate with sensitive pods.

COMMUNICATION

THREAT MODCON 23

CNI

1. Misconfigured CNI Plugins

- **Description:** Using misconfigured or outdated CNI plugins can lead to network vulnerabilities.
- **Evidence:** CNIs are responsible for the networking of containers, and misconfigurations can expose the network.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Network vulnerabilities can lead to unauthorized access or data leakage.
- **Countermeasures:** Regularly update and audit CNI plugins, ensuring they are configured according to best practices.
- **Practical Attack Example:** An attacker exploiting a vulnerability in an outdated CNI plugin to gain unauthorized network access.

2. Lack of Network Isolation

- **Description:** Not implementing network isolation can allow pods to communicate without restrictions.
- **Evidence:** CNIs provide the capability to isolate network traffic between pods, but if not set up correctly, pods might have more access than necessary.
- **Threat Likelihood:** High
- **STRIDE Category:** Elevation of Privilege
- **ASF Category:** Elevation of Privilege
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Lack of network isolation can lead to lateral movement in attacks.
- **Countermeasures:** Use CNI features to implement network segmentation and isolation between pods.
- **Practical Attack Example:** An attacker moving laterally between pods due to lack of network isolation.

STORAGE

THREAT MODCON 2023

Persistent Volumes (PVs)

1. Unauthorized Access to Persistent Volumes

- **Description:** If not properly configured, unauthorized users or pods might access sensitive data stored in PVs.
- **Evidence:** PVs are designed to provide long-term storage, often containing sensitive data.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Unauthorized access can lead to data breaches.
- **Countermeasures:** Implement RBAC for PV access and regularly audit access logs.
- **Practical Attack Example:** An attacker exploiting misconfigured permissions to read data from a PV.

2. Data Corruption in Persistent Volumes

- **Description:** Data stored in PVs can be corrupted due to various reasons, including hardware failures or malicious attacks.
- **Evidence:** PVs are susceptible to the same risks as any storage mechanism.
- **Threat Likelihood:** Low
- **STRIDE Category:** Tampering
- **ASF Category:** Data Alteration
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Data corruption can lead to loss of critical data.
- **Countermeasures:** Regular backups, monitoring, and using reliable storage backends.
- **Practical Attack Example:** An attacker injecting malicious data to corrupt the PV content.

STORAGE

THREAT MODCON 23

Persistent Volume Claims (PVCs)

1. Unauthorized PVC Expansion

- **Description:** Unauthorized users might request more storage than required, leading to a Denial of Service by consuming all available storage.
- **Evidence:** PVCs allow users to request storage from the defined PVs.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Denial of Service
- **ASF Category:** Resource Exhaustion
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Can lead to resource unavailability for other legitimate services.
- **Countermeasures:** Implement quotas and monitor PVC requests.
- **Practical Attack Example:** A malicious user continually expanding PVCs to exhaust available storage.

2. Misconfigured PVC Access Modes

- **Description:** Incorrectly setting access modes can allow unauthorized pods to write to a PVC.
- **Evidence:** PVCs have access modes like ReadWriteOnce, ReadOnlyMany, etc.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Tampering
- **ASF Category:** Data Alteration
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Unauthorized writes can lead to data corruption or data breaches.
- **Countermeasures:** Regularly audit and correctly configure PVC access modes.
- **Practical Attack Example:** A pod writing malicious data to a PVC due to misconfigured access modes.

STORAGE

THREAT MODCON 2023

Storage Classes

1. Misconfigured Storage Class Provisioners

- **Description:** Using misconfigured or outdated storage provisioners can lead to vulnerabilities.
- **Evidence:** Storage classes define how PVs are provisioned.
- **Threat Likelihood:** Low
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Information Disclosure
- **Suggested Priority:** Medium
- **Justification for Suggested Priority:** Can lead to unauthorized access or data leakage.
- **Countermeasures:** Regularly update and audit storage class provisioners.
- **Practical Attack Example:** An attacker exploiting a vulnerability in an outdated storage provisioner.

2. Over-Provisioning with Storage Classes

- **Description:** Over-provisioning storage resources can lead to resource wastage and increased costs.
- **Evidence:** Storage classes can be set to dynamically provision storage.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Denial of Service
- **ASF Category:** Resource Exhaustion
- **Suggested Priority:** Low
- **Justification for Suggested Priority:** Mainly leads to increased costs rather than a direct security threat.
- **Countermeasures:** Monitor storage usage and set appropriate limits.
- **Practical Attack Example:** A user creating numerous unnecessary PVs due to misconfigured storage classes.

STORAGE

THREAT MODCON 23

StatefulSets

1. Inconsistent State due to Pod Rescheduling

- **Description:** When a node fails and a pod managed by a StatefulSet is rescheduled, there's a potential for the application to start in an inconsistent state if the storage backend isn't immediately consistent or if the application doesn't handle such scenarios gracefully.
- **Evidence:** StatefulSets ensure that each pod gets a stable hostname based on its ordinal number. If a pod is rescheduled, it might access data that hasn't been fully synchronized or committed.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Tampering
- **ASF Category:** Data Alteration
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Inconsistent data can lead to application errors, data corruption, or system crashes.
- **Countermeasures:** Use storage solutions that guarantee data consistency. Ensure applications can handle node failures gracefully.
- **Practical Attack Example:** A database in a multi-node setup where one node fails and, upon recovery, starts with stale or unsynchronized data.

2. Data Leakage through Persistent Volume Snapshots

- **Description:** If snapshots of persistent volumes used by StatefulSets are taken for backup or cloning purposes, and these snapshots are not properly secured, it could lead to data exposure.
- **Evidence:** Kubernetes allows for snapshots of persistent volumes. If these snapshots are misconfigured or stored insecurely, they can be accessed by unauthorized entities.
- **Threat Likelihood:** Medium
- **STRIDE Category:** Information Disclosure
- **ASF Category:** Data Exposure
- **Suggested Priority:** High
- **Justification for Suggested Priority:** Exposure of sensitive data can lead to significant breaches, especially if the StatefulSet manages sensitive applications like databases.
- **Countermeasures:** Secure snapshots with appropriate access controls, use encryption at rest, and regularly audit snapshot configurations and storage locations.
- **Practical Attack Example:** An attacker accessing a snapshot of a database's persistent volume and extracting sensitive information.

These threats are more closely aligned with the unique characteristics and potential vulnerabilities of StatefulSets in Kubernetes.

Quick Recap

THREAT MODCON 2023

Define CCCS to GPT4 so it can be used for system decomposition.

Instruct GPT4 to decompose Kubernetes.

Instruct GPT4 to threat-model the decomposed system.

Closing Thoughts

On the positive side…

THREAT MODCON 2023

- Relatively easy to get started with AI-based threat-modeling.
- Output can be tailored to different audiences.
- Output can be customized based on the desired methodology.
- Quickly get to a usable template to base the rest of the analysis on.
- Updating the threat-model due to new changes is straightforward.
- Great tool for those just starting out.

On the downside…

THREAT MODCON 2023

- Need to verify output; LLMs get creative and not always in a good way. Don't just trust the output!
- Can take several iterations before you get to what you need.
- Sometimes it just won't consistently follow the output format you asked for.
- Inconsistent quality of threat analysis at times.
- Getting it to where you want it to be in terms of providing the exact output, e.t.c. requires some effort. I'm working on something to help with that!

THREAT MODCON 2023

Read more on CCCS here:
https://github.com/bitsavant/opensdl

Contact me if you're interested in exploring AI-guided threat-modeling.


wael@bitsavant.com


https://linkedin.com/in/waelsv

Thank You!