ISAUnited
INSTITUTE OF SECURITY ARCHITECTURE UNITED

Technical Research Center

Cyber Threat Flow Model (CTFM)

Research Paper: ISAU-RP-913-2026-CTFM

Art Chavez
6-22-2026

An ISAUnited.org Published Whitepaper

Institute of Security Architecture United (ISAUnited.org)

Author or Task Group Number:

Art Chavez

Publishing Reviewer(s):

ISAUnited Master Fellow Committee

Affiliation: ISAUnited.org

Authored Date:

May 4, 2026

Published Date:

June 22, 2026

Document Registration Number:

ISAU-RP-913-2026-CTFM

Assigned by the Institute Document Management Register

Publication Notice and Disclosure

This publication is an independent work product of the ISAUnited Technical Research Center. It is offered in good faith to improve technical clarity and promote defensible practice, and it is not intended to disparage, diminish, or harm any organization, institution, standards body, vendor, or individual.

This publication is not affiliated with, sponsored by, endorsed by, or approved by any third party mentioned in the text unless explicitly stated. References to third-party standards, organizations, products, services, trademarks, trade names, and document identifiers are for identification and scholarly discussion only and do not imply endorsement, recommendation, certification, validation, or procurement guidance.

Any scoring, indices, comparative positioning, or analytical conclusions reflect ISAUnited-defined criteria and the authors' professional judgment, applied to the stated method and cited evidence. They are informational and do not constitute compliance attestations, certification decisions, legal determinations, or measures of institutional merit.

ISAUnited does not accept payment in exchange for publication decisions. Authors disclose funding and competing interests in the publication.

Abstract

Cybersecurity teams rely on many threat intelligence sources, including government advisories, vendor threat reports, actor catalogs, behavior frameworks, cloud security findings, vulnerability references, weakness libraries, commercial intelligence platforms, and internal incident data. The size and fragmentation of this source ecosystem can create confusion for Cybersecurity Architects and Engineers when threat information must be used during design-and-build activities. The core challenge is not solving the threat intelligence ecosystem itself; the challenge is applying available threat information in a functional order that supports architecture and engineering decisions.

The Cyber Threat Flow Model (CTFM) provides a six-layer structure for understanding how threats relate to architecture: Threat Actor / Actor Class, Threat Vector, Target Surface / Security Zone, Threat Path, TTPs, and Exposure Conditions: CVEs, CWEs, and Misconfigurations. By organizing threat information in this order, CTFM helps practitioners understand how threats may reach, move through, and act against an environment. This structured threat-flow view supports better architecture decisions, security zone planning, defensive control selection, detection planning, engineering implementation, and validation activities.

Keywords: Cyber Threat Flow Model, CTFM, threat vectors, threat paths, cybersecurity architecture, cybersecurity engineering, threat intelligence, TTPs, exposure conditions, CVE, CWE, misconfiguration, security zones.

1. Introduction

Cybersecurity teams have access to more threat intelligence than ever before. Government agencies, standards organizations, cloud providers, security vendors, research teams, intelligence communities, software platforms, and service providers publish information on threat actors, campaigns, tactics, techniques, procedures, indicators, vulnerabilities, weaknesses, and misconfigurations.

The size of this ecosystem is part of the problem, but it is not the problem this paper attempts to solve. Cybersecurity Architects and Engineers cannot eliminate the volume of threat intelligence sources used across the industry. However, they can use a structured model to organize available threat information into a functional order that supports better design-and-build decisions.

Cybersecurity Architects and Engineers are often expected to translate threat intelligence into architecture models, engineering build concepts, security zone planning, threat modeling, detection coverage, and determination of security controls. However, available sources do not always explain what to use first, what comes next, or how each threat concept relates to the architecture under design or review.

This creates a practical challenge. A team may have access to many advisories, vendor reports, actor names, cloud findings, CVEs, CWEs, and behavior frameworks, but still struggle to determine how to interpret those inputs together. Without a functional order, teams may move too quickly from threat information to defensive decisions without first understanding how the threat reaches, moves through, and acts against the environment.

The Cyber Threat Flow Model (CTFM) was created to help address this problem. It organizes available threat intelligence into a simple sequence that helps Cybersecurity Architects and Engineers use threat information more effectively during architecture, design, build, review, control selection, detection planning, and validation activities.

2. Problem Statement and Threat Intelligence Without Functional Order

The cybersecurity industry maintains many threat-related lists, catalogs, libraries, taxonomies, schemas, advisories, reports, findings, platforms, and services. Some describe threat actors. Some describe adversary behavior. Some describe campaigns. Some describe cloud findings. Some describe TTPs. Some describe technical exposure conditions, such as CVEs, CWEs, and misconfigurations.

This creates a problem with the threat intelligence source ecosystem. The industry has no shortage of threat-related information, but the available sources do not automatically organize themselves into an architecture-ready sequence. For Cybersecurity Architects and Engineers, the challenge is not only deciding which sources are credible. The challenge is determining how each source should be used, where it fits in the architecture discussion, and how it should influence defensive control selection, security zone planning, detection coverage, engineering implementation, and validation.

Each source may be useful, but they do not serve the same function.

A Cybersecurity Architect or Engineer may ask:

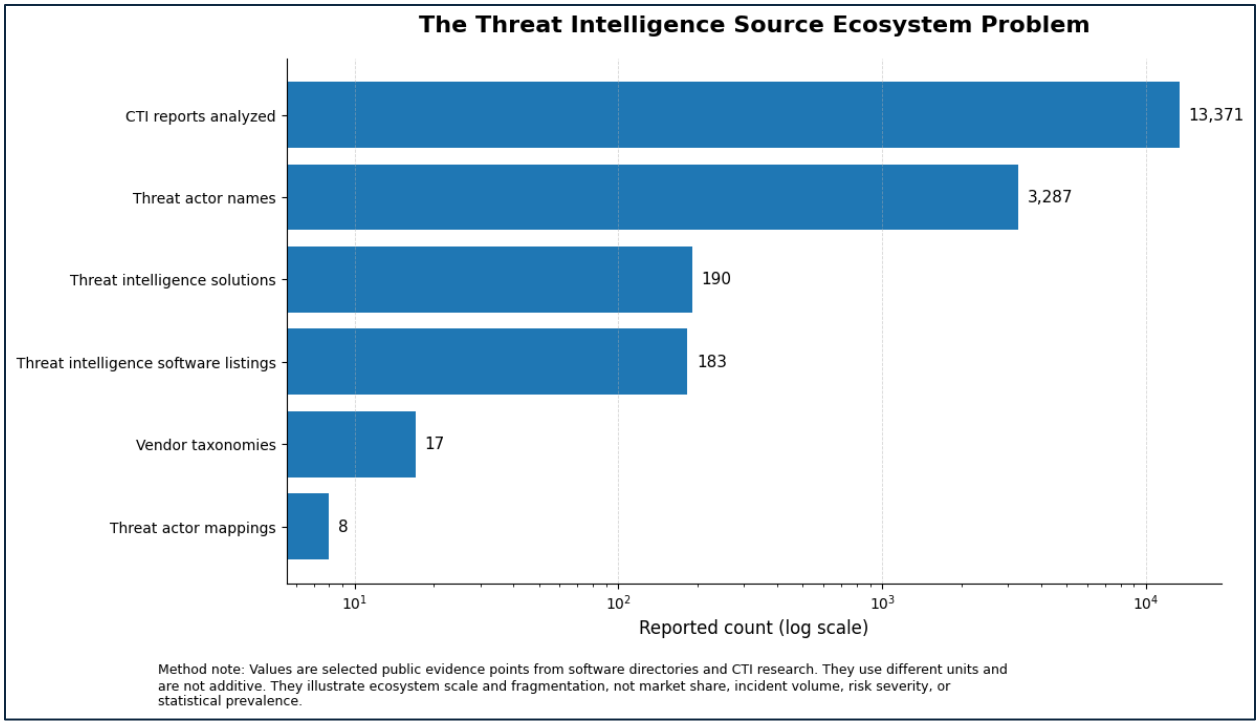
- Do we start with the threat actor or actor class?
- Do we start with the threat vector?
- Do we start with the affected surface or security zone?
- Do we start with TTPs?
- Do we start with a CVE, CWE, or misconfiguration?
- How do we know which source belongs to which part of the threat discussion?

Without a functional order, teams may rush from intelligence to defensive decisions. They may know the actor's name but miss the actual vector. They may know a TTP but fail to identify where it applies within the architecture. They may know a CVE but misunderstand the broader threat path. They may discuss controls before understanding how the threat reaches, moves through, or impacts the environment.

CTFM addresses this gap by providing CAEs with a structured approach to organizing threat information before making design-and-build decisions.

Figure 1. Threat Intelligence Source Ecosystem Context:

Figure 1 presents selected public evidence points that illustrate the scale and fragmentation of the threat intelligence source ecosystem. These values are not additive and do not represent a total market count. They are included to show why Cybersecurity Architects and Engineers need a systematic way to organize available threat information during design-and-build activities.



Note: Values are selected public evidence points from software directories and CTI research. They use different units, including product listings, solution results, CTI reports, threat actor names, vendor taxonomies, and mappings. They illustrate ecosystem context and source fragmentation, not incident volume, threat frequency, vendor market share, risk severity, or statistical prevalence.

3. Research Basis and Threat Sources Serve Different Functions

The Cyber Threat Flow Model is based on a practical research observation: widely used threat sources do not all address the same aspect of threat understanding. Each source may be useful, but each does not serve the same function for Cybersecurity Architects and Engineers.

MITRE ATT&CK is widely used to organize adversary tactics and techniques based on observed adversary behavior. CISA cybersecurity advisories provide information on cyber threats, including threat actor activity, tactics, techniques, and procedures, as well as indicators of compromise. NIST SP 800-150 describes cyber threat information sharing and identifies multiple types of cyber threat information, including indicators of compromise, TTPs used by threat actors, suggested actions, and incident analysis findings.

Other sources serve different functions. Microsoft, CrowdStrike, MISP, Google/Mandiant, and other threat intelligence providers maintain actor naming, attribution, intelligence, or visibility models. Cloud provider findings may identify suspicious or malicious activity within a provider-specific environment. CVE and CWE sources describe vulnerabilities and weakness types that may create exposure conditions, but they do not, on their own, describe the full threat flow.

This creates a source-function problem for CAEs. A naming source may help identify who is conducting the activity. A behavior framework may help explain what an adversary may do. A cloud finding may show where suspicious activity was observed. A CVE or CWE may explain a technical condition that enables exploitation. However, none of these sources alone provides the complete functional flow from actor to vector, target surface, path, behavior, and exposure condition.

CTFM addresses this issue by assigning threat information to the layer where it is most useful. This helps CAEs avoid treating all threat sources as a single input type.

Table 1. Source Types Mapped to CTFM Layers:

Source Type	Primary CTFM Layer	Secondary CTFM Layer
Threat actor reports	Threat Actor / Actor Class	TTPs
Government advisories	Threat Actor / Actor Class	Threat Vector, TTPs
Adversary behavior frameworks	TTPs	Threat Path
Cloud security findings	Threat Vector	Target Surface / Security Zone
Incident reports	Threat Path	TTPs, Exposure Conditions
CVE references	Exposure Conditions	Target Surface
CWE references	Exposure Conditions	Engineering build concept
Internal architecture reviews	Target Surface / Security Zone	Threat Path

The source-to-layer mapping explains why a single threat list is difficult to use across every architecture and engineering need. CTFM does not require every source to be the central authority. Instead, it provides a central organizing model that allows the strongest source type to best support the layer it describes.

4. Why Cybersecurity Architects and Engineers Need CTFM

Cybersecurity Architects and Engineers (CAEs) are responsible for translating security concerns into architectural decisions, design patterns, build concepts, and the implementation of technical standards and security controls. To do this well, they need more than threat intelligence. They need a practical way to understand how threat information applies to the systems, zones, identities, applications, data flows, and trust boundaries they are designing or reviewing.

CTFM helps CAEs answer six practical questions:

1. Who may conduct the activity?
2. How could the threat reach the environment?
3. Where does the threat touch the architecture?
4. Where could the threat move next?
5. What adversary behavior may occur along the path?
6. What conditions allow the threat path to work?

This creates a bridge between threat intelligence and architectural thinking. By clarifying the threat actor or actor class, threat vector, target surface, threat path, TTPs, and exposure conditions, CTFM helps CAEs determine which defensive security controls, design patterns, and engineering mechanisms are appropriate for the threat being addressed.

CTFM gives CAEs a simple way to organize the threat side of the problem, enabling better decisions during defensive design, control selection, engineering implementation, detection planning, and validation.

5. The Cyber Threat Flow Model

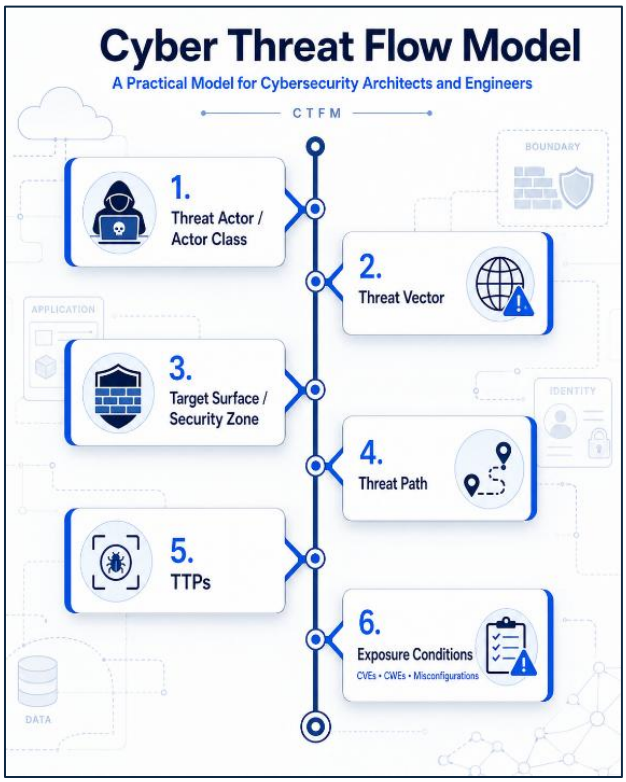
The Cyber Threat Flow Model organizes threat understanding into six layers. The model is intended to be used in order, so Cybersecurity Architects and Engineers can move from general threat intelligence to architecture-relevant threat understanding.

The CTFM sequence is:

Each layer answers a different question. Together, the layers help practitioners understand who may conduct the activity, how the threat may reach the environment, where it may affect the architecture, what path it may follow, what behaviors may occur, and what exposure conditions may enable the path to succeed.

CTFM stops at structured threat understanding. Defensive control selection, detection engineering, validation, and response planning may follow, but they fall outside the model's scope.

Figure 2. Cyber Threat Flow Model Six-Layer Structure:
This figure presents the six-layer Cyber Threat Flow Model. The model organizes threat understanding into a functional sequence, enabling Cybersecurity Architects and Engineers to interpret threat information before selecting defensive controls or engineering mechanisms.



6. Layer Summary

CTFM uses six layers to organize threat understanding. Each layer has a specific purpose and should be interpreted before moving to the next layer.

6.1 Threat Actor / Actor Class

This layer identifies the bad actor or actor type that may perform the activity. A named threat group may be useful, but an actor class is often more practical for architecture and engineering work.

Examples include ransomware affiliates, initial access brokers, nation-state actors, insider-enabled actors, social engineering crews, supply chain actors, and cloud-focused intrusion actors.

6.2 Threat Vector

This layer describes how the actor may reach, pressure, or attack the environment.

Examples include phishing and credential theft, remote access abuse, third-party access abuse, internet-exposed application attacks, identity federation abuse, cloud control plane abuse, API abuse, supply chain compromise, and SaaS administration abuse.

6.3 Target Surface / Security Zone

This layer identifies where the threat intersects the architecture. It links threat intelligence to systems, zones, identities, data flows, and trust boundaries.

Examples include the external edge, remote access zone, identity plane, API gateway, application zone, data zone, backup environment, OT zone, and the development pipeline.

6.4 Threat Path

This layer describes the possible route the threat may take through the environment.

A threat path may span users, identities, applications, cloud services, partner connections, internal systems, data stores, or security zones.

Example:

6.5 TTPs

This layer outlines the tactics, techniques, and procedures the adversary uses along the threat path. TTPs explain what the adversary may do, not just where the adversary may go.

Examples include credential harvesting, valid account abuse, discovery, privilege escalation, lateral movement, defense evasion, data staging, exfiltration, and impact.

6.6 Exposure Conditions: CVEs, CWEs, and Misconfigurations

This layer identifies the technical, architectural, or operational conditions that enable the threat path to function.

Examples include known CVEs and CWEs, misconfigured services, weak MFA, exposed management interfaces, excessive permissions, flat network design, weak API authorization, poor segmentation, insecure remote access, and weak logging at trust boundaries.

CVEs, CWEs, and misconfigurations are not the full story of the threat. They are exposure conditions that may enable the threat actor to use a vector, follow a path, and execute TTPs.

7. How to Use CTFM

CTFM should be used in sequence. Its purpose is to help Cybersecurity Architects and Engineers understand the threat before moving into defensive design, control selection, detection planning, or validation.

The sequence is:

This order helps prevent common mistakes. Teams should not start with a CVE and assume they fully understand the threat. They should not start with a threat actor name and immediately select controls. They should not use TTPs without first understanding where those behaviors apply within the architecture.

Table 2. CTFM Layer Questions for Cybersecurity Architects and Engineers:

CTFM Layer	CAE Question
Threat Actor / Actor Class	Who may conduct the activity?
Threat Vector	How could the threat reach the environment?
Target Surface / Security Zone	Where does the threat touch the architecture?
Threat Path	Where could the threat move next?
TTPs	What behavior could occur along the path?
Exposure Conditions	What allows the threat path to work?

The model can be used during architecture reviews, cloud and application design, threat modeling, security zone planning, and design-and-build discussions. It gives teams a common way to discuss the threat side of the problem before deciding how to defend against it.

8. Applying CTFM to Architecture Before and After View

Cybersecurity Architects and Engineers often start with a technical architecture diagram that shows systems, components, zones, and data flow. This view is necessary for design-and-build work, but it may not explain how a threat reaches the environment, where it affects the architecture, what path it may follow, or what exposure conditions may enable it to succeed.

CTFM adds that missing threat-flow view.

The baseline architecture shows the standard outside-in application and data flow from the client side into the environment. It includes common components such as the client app, internet, and DNS path, edge gateway, load balancer, API gateway, identity provider, application service, message queue, database, and object storage or backup layer.

Figure 3. Baseline Cloud / Data Center Architecture and Data Flow.
This figure shows the standard outside-in application and data flow across a vendor-neutral cloud or data center environment. The diagram provides the technical foundation for understanding how users, network entry points, identity services, application services, and data components interact.

The CTFM after-view applies the threat-flow model to the same architecture. The normal data flow remains visible, but the model now helps identify the threat actor or actor class, the likely threat vector, the affected target surface or security zone, the possible threat path, the TTPs used along that path, and the exposure conditions that may enable the threat.

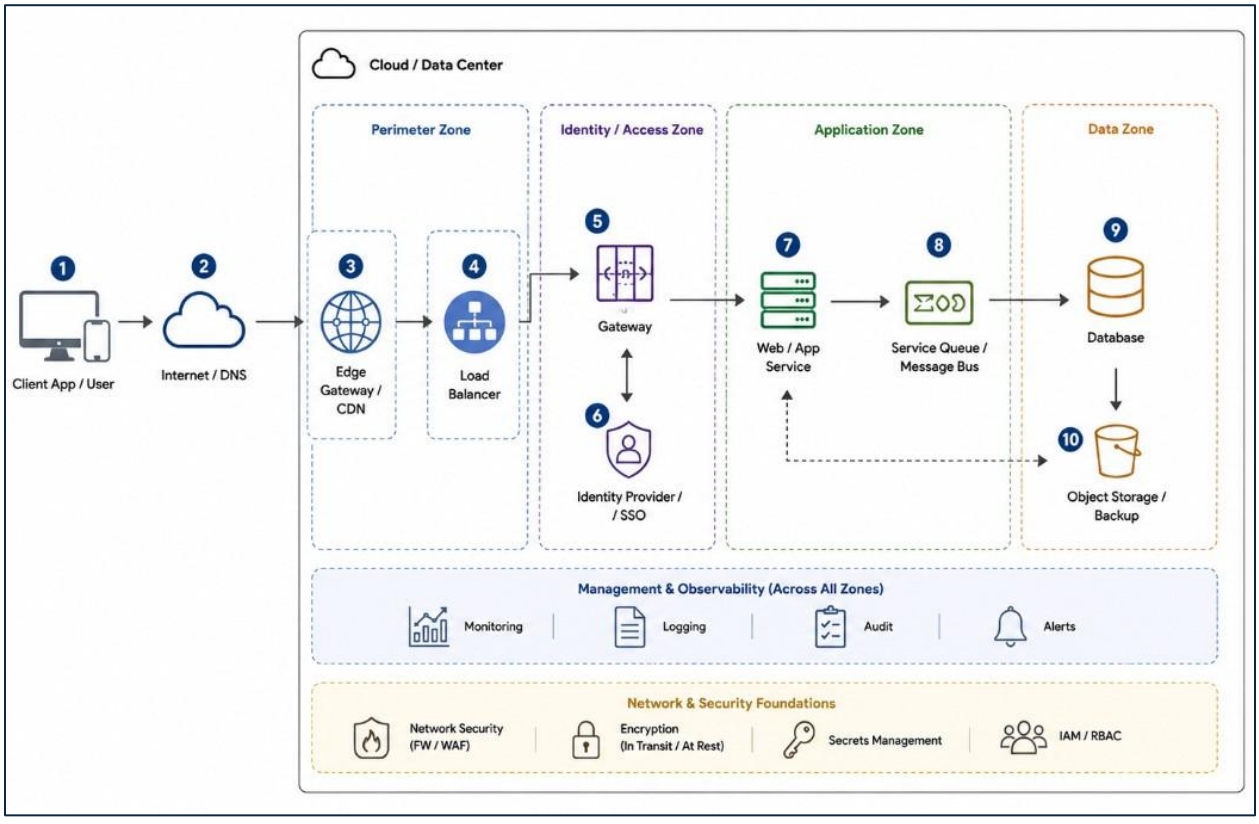
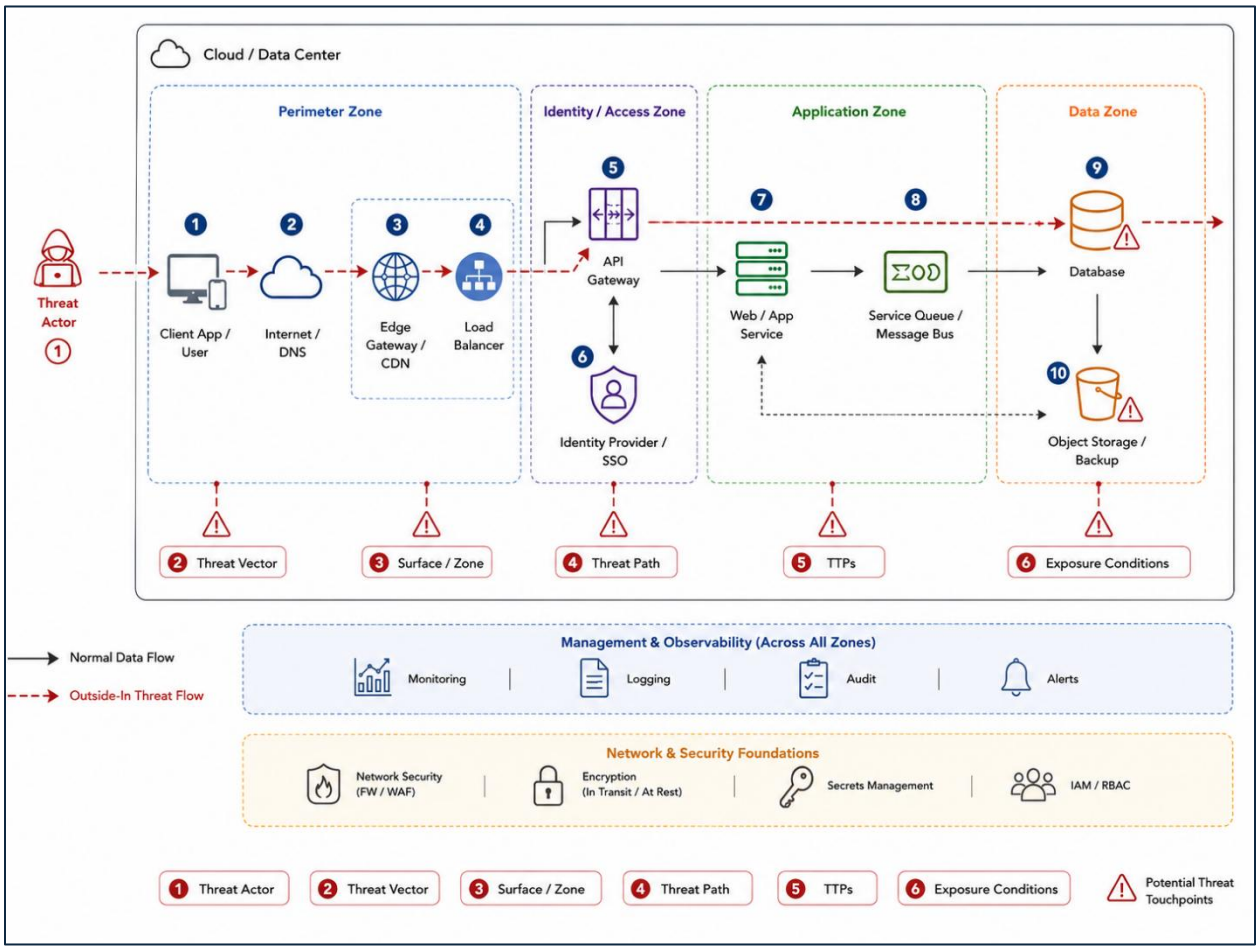


Figure 4. CTFM Outside-In Threat Flow Applied to Baseline Architecture.
This figure applies the Cyber Threat Flow Model to the baseline architecture. The outside-in threat flow highlights potential threat touchpoints, affected zones, likely movement paths, adversary behaviors, and exposure conditions that may influence subsequent design-and-build decisions.

Together, the before-and-after views illustrate how CTFM helps CAEs shift from a component-based architecture view to a threat-informed architecture view. This threat-informed view helps teams identify where defensive architecture, security zone configuration, control selection, detection coverage, engineering mechanisms, and validation activities should be considered based on how the threat may reach, move through, and act against the environment.



9. Outside-In and Boundary-Aware Design Thinking

CTFM supports outside-in and boundary-aware design thinking. This means the architect or engineer considers how a threat may approach the environment, which surface or zone it may first touch, and how it may move across systems, identities, applications, data flows, and trust boundaries.

Outside-in does not only refer to internet-based attacks. The “outside” may include a vendor connection, a SaaS administration console, an identity federation trust, a remote access path, a cloud control plane, a partner integration, an exposed API, or a compromised user account.

Using CTFM, CAEs can ask:

- What is the first surface or zone the threat may touch?
- What trust boundary could be crossed?
- What identity or access path could be abused?
- Where could the threat move next?
- What TTPs may appear along the path?
- What CVEs, CWEs, or misconfigurations may allow the path to work?

Once the threat flow is understood, the team can move into defensive architecture, security zone configuration, control selection, detection planning, or validation through separate defensive and engineering processes.

10. Relationship to Existing Threat Sources

CTFM does not replace existing threat sources. It organizes them.

Threat intelligence sources, advisories, actor catalogs, behavior frameworks, cloud findings, incident reports, CVE references, and CWE references each provide useful information. However, they do not all address the same part of the threat flow. CTFM helps Cybersecurity Architects and Engineers place each source in the layer where it provides the most value.

Table 3. Existing Threat Sources Mapped to CTFM Layers:

Existing Source Type	Where It Fits in CTFM
Threat actor reports	Threat Actor / Actor Class
Vendor adversary catalogs	Threat Actor / Actor Class
Government advisories	Threat Actor, Threat Vector, TTPs
MITRE ATT&CK	TTPs
Cloud threat findings	Threat Vector, Target Surface / Security Zone, TTPs
Incident reports	Threat Path, TTPs, Exposure Conditions
CVE references	Exposure Conditions
CWE references	Exposure Conditions
Internal architecture reviews	Target Surface / Security Zone, Threat Path

This relationship is important because it prevents teams from treating all threat sources as the same type of input. A threat actor report may help identify who may be conducting the activity. A behavior framework may help explain what the adversary may do. A cloud finding may show suspicious activity in a specific environment. A CVE or CWE may describe a condition that enables the threat path.

CTFM gives CAEs a practical way to use these sources together without forcing a single source to answer every question. The model provides the functional order, while the sources provide supporting intelligence for each layer.

11. Conclusion

Cybersecurity teams have access to many threat catalogs, advisories, libraries, schemas, vendor reports, cloud findings, vulnerability references, and internal reviews. These sources are valuable, but they do not always explain how threat information should be used in a functional order by Cybersecurity Architects and Engineers.

The Cyber Threat Flow Model addresses this gap by organizing understanding of threats into a simple sequence:

This sequence helps practitioners understand how a threat may reach, move through, and act against an environment. It also clarifies where CVEs, CWEs, and misconfigurations belong, ensuring those details support the threat model rather than dominate it.

CTFM assigns a functional order to threat intelligence sources. By placing actor intelligence, threat vectors, target surfaces, threat paths, adversary behaviors, and exposure conditions in the appropriate layers, CAEs can develop a clearer, threat-informed view of the architecture. This view helps teams determine appropriate defensive security controls, design patterns, security zone decisions, detection coverage, engineering mechanisms, and validation activities.

CTFM's value lies in its practical application during design-and-build work. It helps teams move from scattered threat lists to structured threat understanding, and from that understanding to better security decisions. Organizations should adopt CTFM as a lightweight model for architecture reviews, threat-modeling discussions, control selection, and engineering design activities that require threat-informed decisions.

References

1. Roa, G., Suarez-Roman, M., & Tapiador, J. (2025). Hesperus is Phosphorus: Mapping threat actor naming taxonomies at scale. arXiv
2. Amazon Web Services. (n.d.). *GuardDuty finding types*. AWS Documentation. Retrieved June 22, 2026.
3. Cybersecurity and Infrastructure Security Agency. (n.d.). *Cybersecurity alerts & advisories*. U.S. Department of Homeland Security. Retrieved June 22, 2026.
4. CVE Program. (n.d.). *CVE: Common vulnerabilities and exposures*. Retrieved June 22, 2026.
5. CrowdStrike. (2025, June 2). *CrowdStrike and Microsoft unite to deconflict cyber threat attribution*.
6. Google Cloud. (n.d.). *Google Threat Intelligence: Know who's targeting you*. Retrieved June 22, 2026.
7. Microsoft. (n.d.). *How Microsoft names threat actors*. Microsoft Learn. Retrieved June 22, 2026.
8. Microsoft Security. (2025, June 2). *Announcing a new strategic collaboration to bring clarity to threat actor naming*. Microsoft Security Blog.
9. MISP Project. (2025, January 4). *Introducing the MISP Threat Actor Naming Standard*. MISP Standards.
10. MITRE. (n.d.). *MITRE ATT&CK*. Retrieved June 22, 2026.
11. MITRE. (n.d.). *Common Weakness Enumeration*. Retrieved June 22, 2026.
12. National Institute of Standards and Technology. (2016). *Guide to cyber threat information sharing* (NIST Special Publication 800-150). U.S. Department of Commerce.
13. National Institute of Standards and Technology. (2024). *The NIST Cybersecurity Framework (CSF) 2.0*. U.S. Department of Commerce.
14. The MISP Project. (n.d.). *Threat actor naming standard*. GitHub. Retrieved June 22, 2026.

End of Document
IO.