CEQUENCE SECURITY

Demystifying the OWASP API Security Top 10

INTERACTIVE GUIDE

About OWASP and the API Security Top 10

OWASP Foundation

The source for developers and technologists to secure the web

- Tools and Resources
- Community and Networking
- Education and Training

Other OWASP lists

- Top 10 Web App Security Risks
- Automated Threats to Web Applications

https://owasp.org/

API Security Top 10

Relatively new – published Dec 2019

Raises awareness of API security risks for both security and dev teams

Use as a framework for your API Security initiative

Many of the API Security Top 10 were exploited in 2020

https://owasp.org/www-project-api-security/

OWASP API Security Top 10

Organizations that are moving towards an API centric development methodology, making heavy use of containers and have seen their API usage explode should leverage the OWASP API Security Top 10 as an integral component for how to protect their APIs from automated attacks and vulnerability exploits. The table below lists the most common root cause of the respective OWASP API Security Top 10 threat. The remainder of the e-book describes in simple terms what the threat is, how threat actors might leverage it, then provides prevention tips and how Cequence Security can help.

OWASP API Top 10	Typical Root Cause
API 1: Broken Object Level Authorization	Weak Access Control
API 2: Broken User Authentication	Weak Access Control
API 3: Excessive Data Exposure	Business Logic Abuse
API 4: Lack of Resources & Rate Limiting	Insufficient Traffic Management
API 5: Broken Function Level Authorization	Weak Access Control
API 6: Mass Assignment	Business Logic Abuse
API 7: Security Misconfiguration	Business Logic Abuse
API 8: Injection	Application Vulnerability
API 9: Improper Assets Management	Lack of Holistic Visibility
API 10: Insufficient Logging & Monitoring	Lack of Operational Security Readiness

About Cequence Security

The Company

- Application security provider backed by Shasta, Dell Capital and T-Mobile
- Veteran leadership team from Palo Alto Networks, Symantec and Zscaler
- Founded in 2015 with offices in Sunnyvale, CA (HQ) and Cincinnati, OH

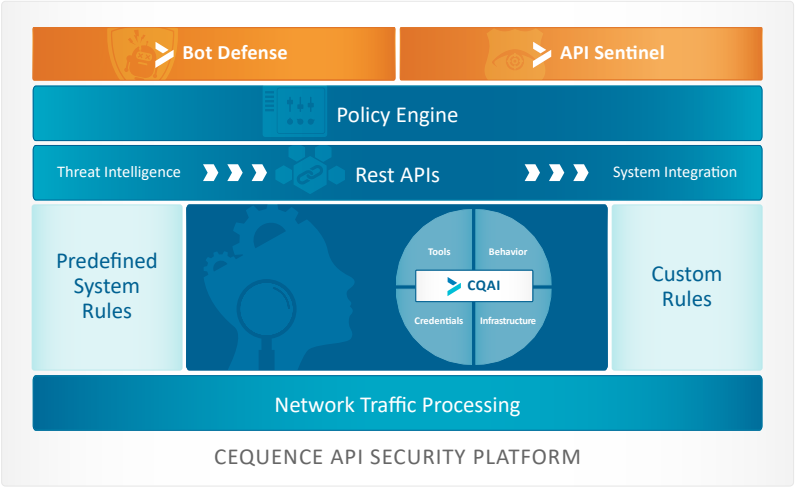
SHASTA

DELL Technologies CAPITAL

T-MOBILE VENTURES

Our Solution: Application Security Platform

- The only platform that unifies runtime API visibility, security risk monitoring, and behavioral fingerprinting
- Protects web apps and APIs from online fraud, business logic attacks, exploits and unintended data leakage
- Built for and deployed by large enterprises in financial services, retail, media and social media

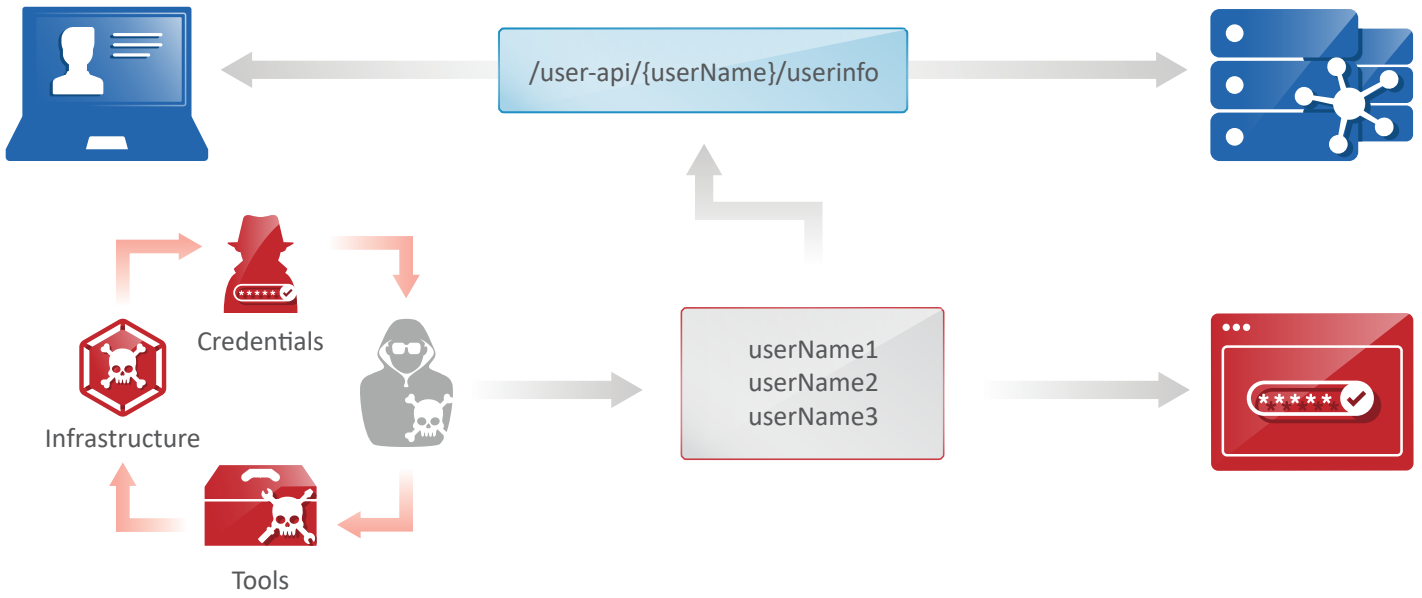


CEQUENCE API SECURITY PLATFORM

#1: Broken Object Level Authorization Flaw

Layperson's Description

Insufficient validation of an object access request allows a threat actor to perform an unauthorized action by reusing an access token.



How Threat Actors Leverage This Threat

Threat actors will use a variety of tools and techniques to discover your APIs, resource IDs or objects that do not have proper authorization and use them for malicious purposes. In some cases, the data used by the API has no user validation and is accessible to the public, in other cases error messages return too much information, telling threat actors how to abuse the API.

#1: Broken Object Level Authorization Flaw

Top 3 Prevention Tips

1. Application development should collaborate with security to implement strong authorization to ensure that the API validates user privileges for all functions.

2. API authorization requirements should be well defined in the API specification and include the use random/unpredictable IDs.

3. The API test plan should include function-level, security specific test cases for authorization related features.

How Cequence Can Help

API Sentinel

Detect API runtime usage patterns including IP addresses, organizations and countries

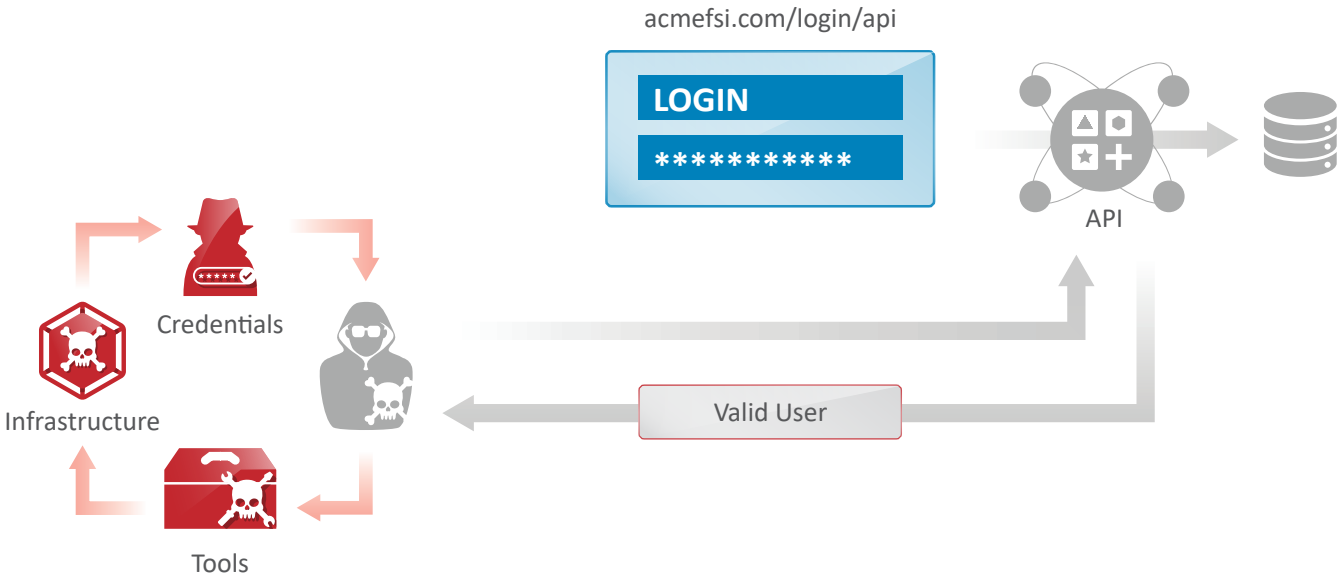
Bot Defense

Detect and block enumeration and token reuse/rotation attacks

#2: Broken User Authentication

Layperson's Description

Poorly implemented user authentication allows threat actors to impersonate legitimate users by exploiting implementation flaws in authentication mechanisms.



How Threat Actors Leverage This Threat

Threat actors will discover flawed, unprotected authentication APIs (e.g., login, registration, password reset) and target them with automated attacks or use them to gain system access and steal data. The types of API flaws that expose this threat include APIs that allow weak passwords, use error messages that return too much information, lack token validation or use weak or no encryption.

#2: Broken User Authentication

Top 3 Prevention Tips

1 Application development, security and business groups should understand, agree on and document authentication workflow and associated requirements.

2 API management requirements should ensure ALL authentication endpoints (e.g., login, reset, etc.) are identified and apply the same strong, standards-based, multi-factor authentication wherever possible.

3 Implement volumetric and account lockout protection mechanisms to prevent brute force attacks at both the global and individual user levels.

How Cequence Can Help

API Sentinel

Detect and flag unauthenticated endpoints as high-risk; work with development to update them

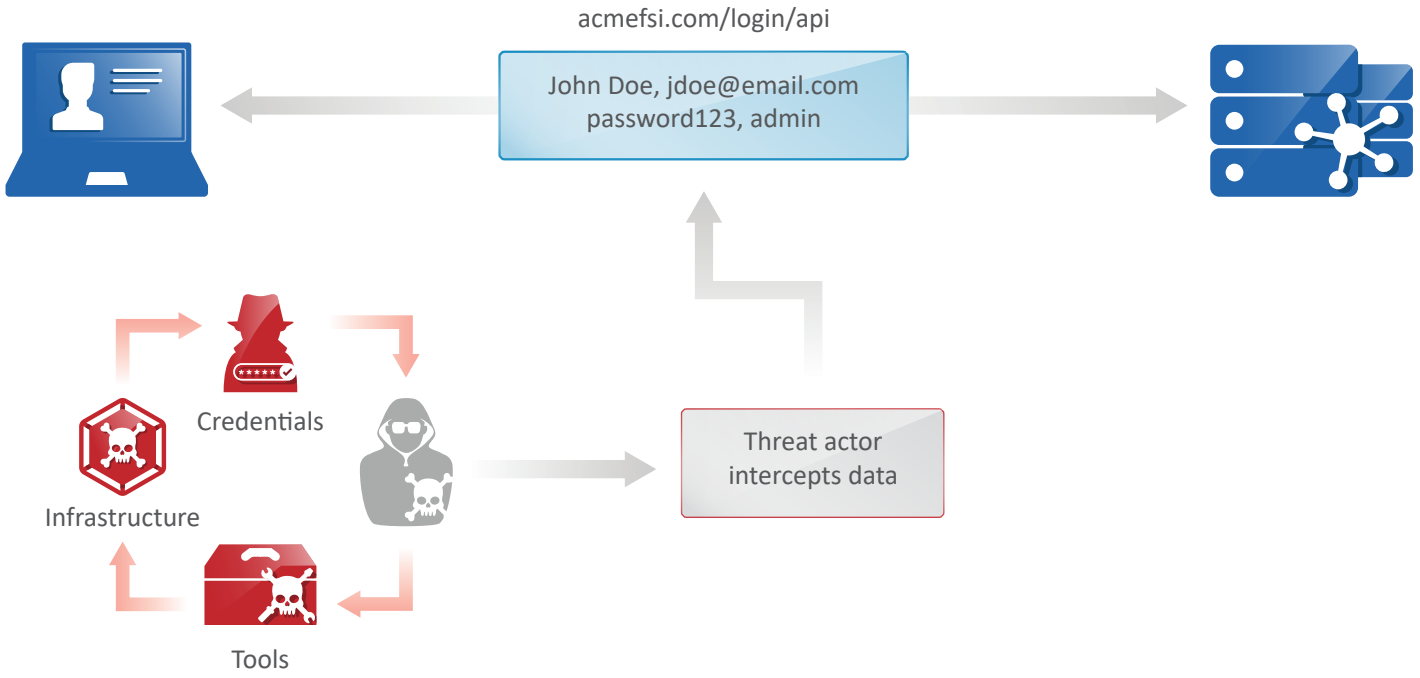
Bot Defense

Detect and block authentication requests that do not have valid tokens; prevent volumetric attacks

#3: Excessive Data Exposure

Layperson's Description

A published API might expose more data than necessary, relying on the client app to perform the necessary filtering.



How Threat Actors Leverage This Threat

This threat is founded on APIs that return too much information or rely on the client, not backend systems to filter the data. Threat actors use these errors to gain a greater understanding of bad good API inputs, allowing them to create an attack "cookbook" to steal data or use the information for larger attack.

#3: Excessive Data Exposure

Top 3 Prevention Tips

1 Design the API with a full understanding from business groups of who the consumers of the API and data are.

2 API design requirements should define the minimal data to be returned to the client and never rely on client-side filtering

3 Be very selective of the properties the API should return, avoid using generic methods and be sure to classify sensitive and PII data on the backend.

How Cequence Can Help

API Sentinel

Detect and flag endpoints leaking proprietary/sensitive data in cleartext, deviating from published schemas

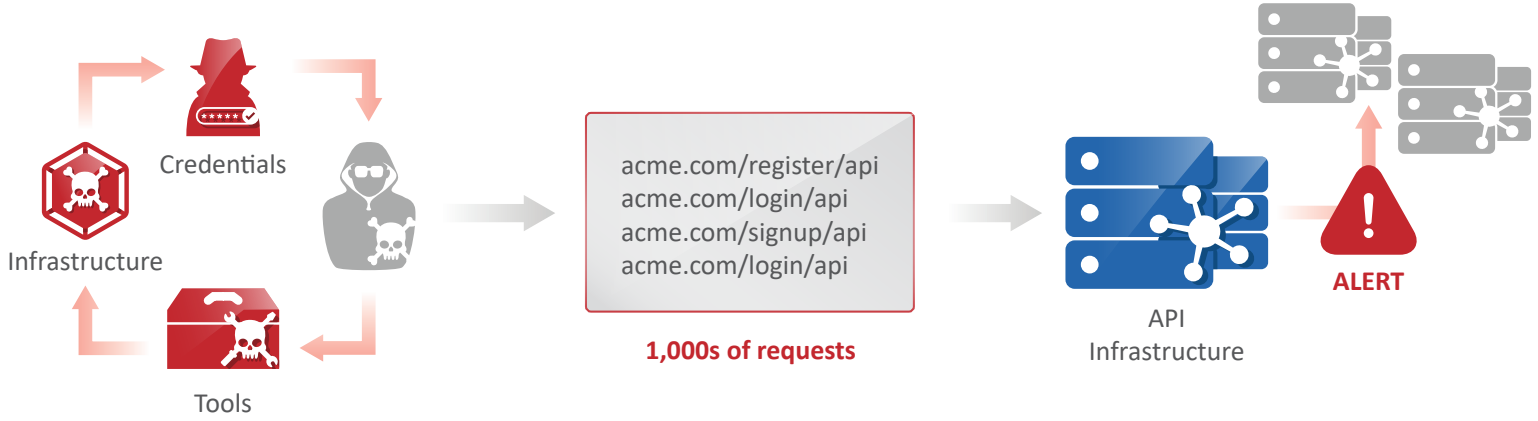
Bot Defense

Block requests containing headers or query parameters that are identified as data leakage

#4: Lack of Resources and Rate Limiting

Layperson's Description

By not implementing internal rate limiting policies, threat actors can overwhelm the backend with denial-of-service attacks.



How Threat Actors Leverage This Threat

Inadequate or no rate limiting (e.g., response timeouts, memory, payload size, number of processes, records, requests) allows threat actors to submit many API requests, rendering service unavailable (DoS, DDoS), crashing the application or purposely driving resource costs up.

#4: Lack of Resources and Rate Limiting

Top 3 Prevention Tips

1. Security, development and business groups collaborate to define and adhere to API resource consumption requirements and limits.

2. Implement limits on the number of API calls, client notifications (e.g., resets, lockout, etc.) and include server-side validation for response size (e.g., # of records, resource consumption, etc.).

3. Define and enforce maximum size of data on all incoming parameters and payloads (e.g., length for strings, number of array elements, etc.).

How Cequence Can Help

API Sentinel

Identify and alert on endpoints experiencing large usage spikes; determine traffic spike sources: IP, geography, organization

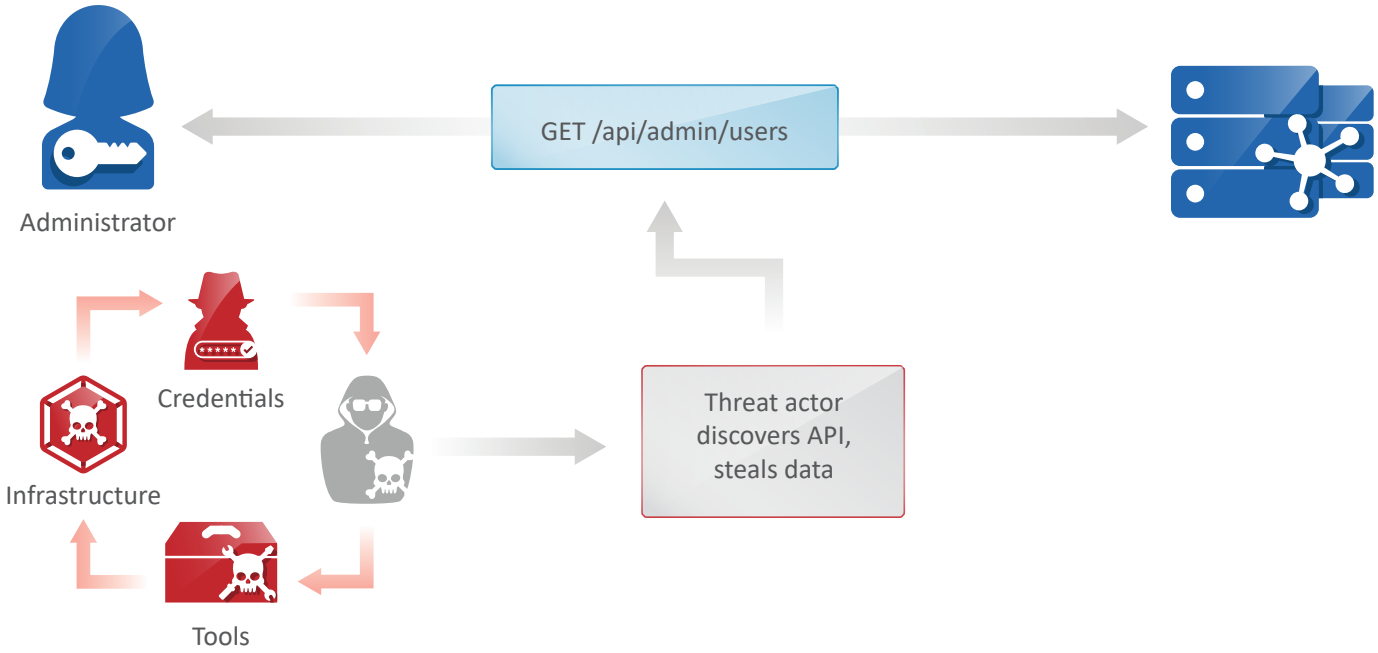
Bot Defense

Detect and block suspicious volumetric spikes based on custom/ pre-defined checks separating good vs bad requests

#5: Broken Function Level Authorization

Layperson's Description

This threat is a variation on API Threat #1 and is also an authorization vulnerability that allows a threat actor to execute actions by sending requests to functions they should be unauthorized to access.



How Threat Actors Leverage This Threat

User privileges are not adequately enforced or segregated (e.g., admin, superuser, helpdesk, etc.), allowing a threat actor to gain access to privileged commands or sensitive functions (e.g., PUT, DELETE, OPTIONS, etc.), enabling data theft to occur.

#5: Broken Function Level Authorization

Top 3 Prevention Tips

1. Application development, security and business groups should understand, agree on and document authentication workflow and associated requirements.

2. API enforcement design should default to positive security model – deny all, except those roles you want to allow.

3. Define and implement a strong and consistent access control/ authorization mechanism that flows from parent to child for policy consistency.

How Cequence Can Help

API Sentinel

Detect endpoint and methods usage at runtime, including sources such as IP address, org., and country

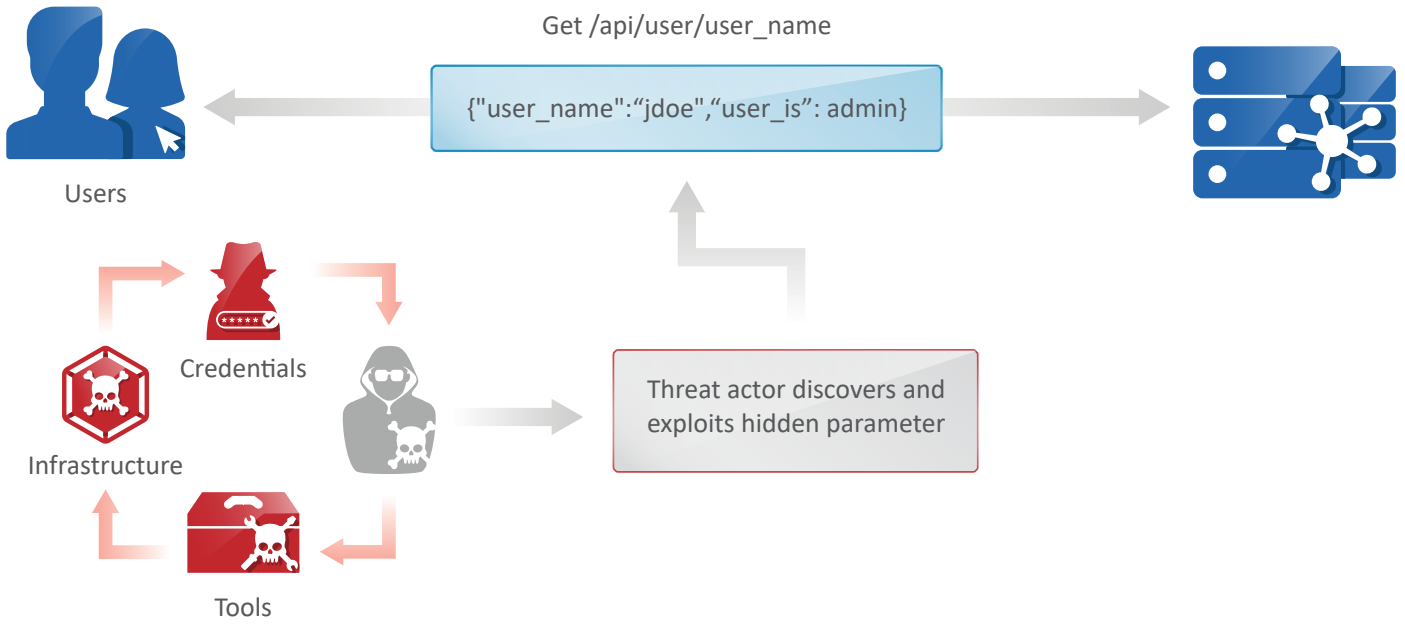
Bot Defense

Detect and block reuse and enumeration of session keys/tokens across functions

#6: Mass Assignment

Layperson's Description

Unfiltered data provided via APIs to client apps allows threat actors to guess object properties via requests.



How Threat Actors Leverage This Threat

Threat actors discover modifiable parameters and server-side variables, exploiting them by creating new users with elevated privileges, or modifying existing user profiles.

#6: Mass Assignment

Top 3 Prevention Tips

1. Build requirements into the design that limit or avoid the use of functions that bind inputs to objects or code variables.
2. Agree on and enforce adherence to published API schema that includes input data payloads.
3. Ensure client-updatable properties are whitelisted and those that should be restricted are blacklisted.

How Cequence Can Help

API Sentinel

Identify endpoints, headers, parameters deviating from schema definitions

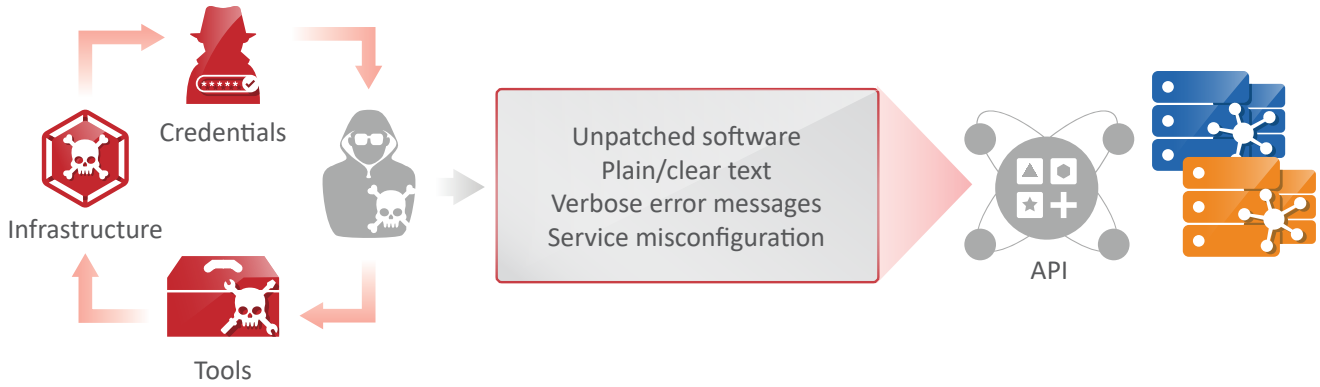
Bot Defense

Detect and block API requests or responses with specific non-conformant headers or query parameters

#7: Security Misconfiguration

Layperson's Description

Commonly a result of incomplete, ad-hoc or (insecure) default configurations, misconfigured HTTP headers, unnecessary HTTP methods, permissive Cross-Origin resource sharing (CORS), and verbose error messages containing sensitive information.



How Threat Actors Leverage This Threat

This threat is all too common and is oftentimes the result of human errors including: a lack of application hardening, poor patching practices, verbose messages, improper encryption, or is missing the Cross- Origin Resource Sharing (CORS) policy. Threat actors discover any one of these errors and leverage them to execute attacks that result in fraud or data loss.

#7: Security Misconfiguration

Top 3 Prevention Tips

1 The following elements should be part of an agreed upon API lifecycle: repeatable hardening process; a configuration review and update process encompassing orchestration files, API components, and cloud services; and a mechanism to continuously assess the configuration and settings effectiveness.

2 Define and enforce all API response payload schemas including error responses to prevent information from being sent back to threat actors.

3 Ensure Cross-Origin Resource Sharing (CORS) policies are in place for all browser-based use cases.

How Cequence Can Help

API Sentinel

Identify endpoints that use non-conformant headers, query params or verbose error messages leaking sensitive data

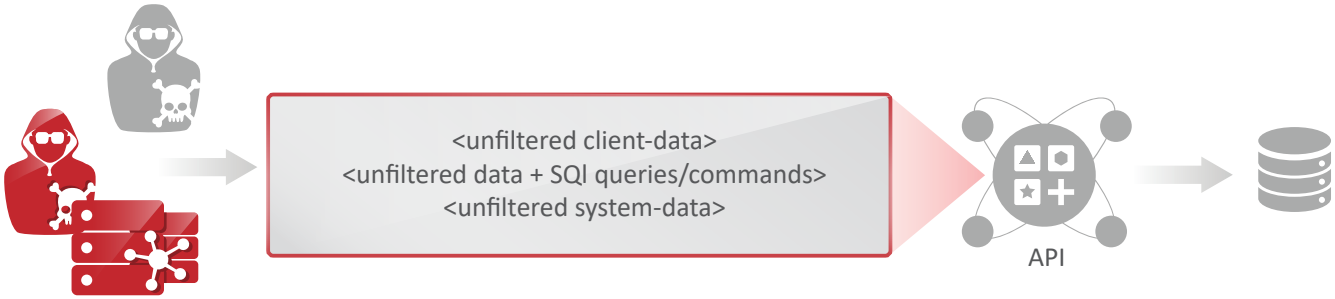
Bot Defense

Detect and block automated exploitation attempts of hidden methods or endpoints, automatically separating good vs. bad requests

#8: Injection

Layperson's Description

Untrusted injection of data, such as SQL, NoSQL, XML Parsers, ORM, LDAP, OS Commands, and JavaScript, into API requests can result in the execution of unintended commands or unauthorized data access.



How Threat Actors Leverage This Threat

This threat is a carryover from the OWASP Web Application top 10 and is leveraged by threat actors when the database or application lacks filtering, validation of client or machine data, allowing them to steal data, or inject malware by sending queries/commands directly to the database or application (e.g., SQL, NoSQL, LDAP queries, OS commands, XML parsers, etc.).

#8: Injection

Top 3 Prevention Tips

1. Security, development and business groups need to collaborate and agree on a design that maintains separation between data and commands, queries.
2. Define data types and parameter patterns; limit the number of records returned.
3. Validate, test and filter all data emanating from clients and external integrated systems.

How Cequence Can Help

API Sentinel

Roadmap: Flag requests containing anomalous values indicating potential Injection attacks

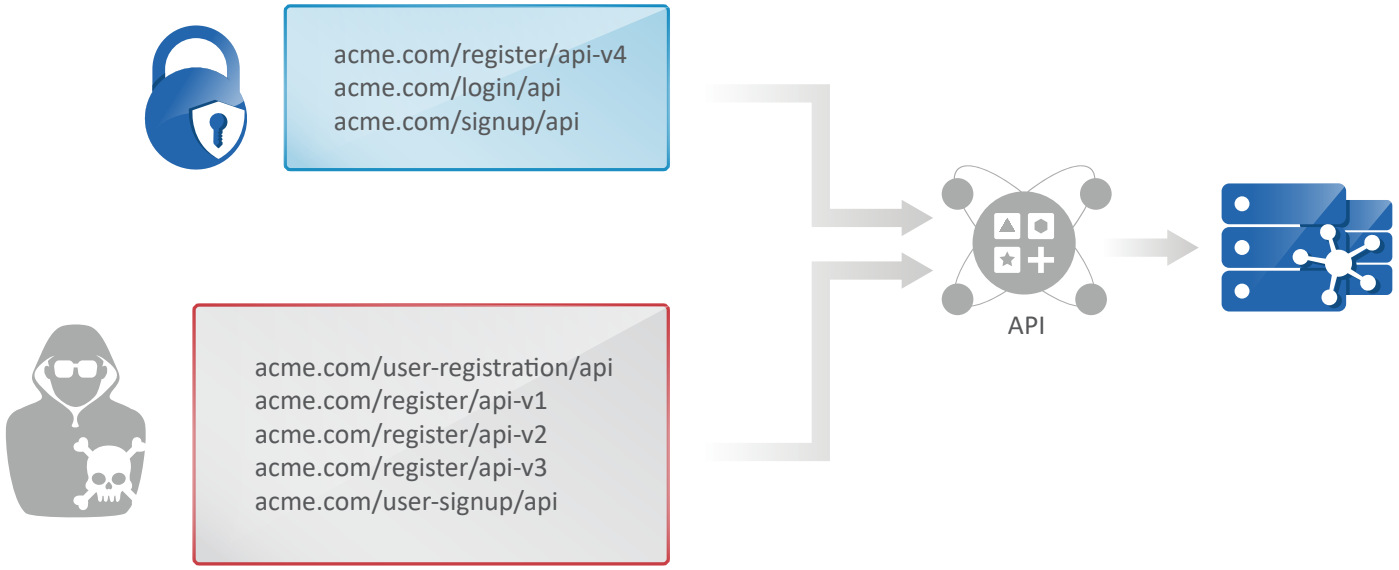
Bot Defense

Detect and block API requests containing potential injection attack patterns

#9: Improper Asset Management

Layperson's Description

Insufficient environment management and environment segregation allows threat actors to access under-secured API endpoints.



How Threat Actors Leverage This Threat

Threat actors discover shadow, deprecated, end-of-life APIs that have been deployed outside of, or are not in security view. Other threat vectors include pre-production APIs may have been inadvertently exposed to the public, or a lack of API documentation led to an exposed flaw (e.g., authentication, errors, redirects, rate limiting, Cross-Origin Resource Sharing (CORS) policy and endpoint details such as parameters, requests, and responses).

#9: Improper Asset Management

Top 3 Prevention Tips

1 Application development, security and business groups should agree upon, document and follow an API publication process that includes replace or update risk analysis as new APIs are released.

2 Implement continuous visibility and monitoring of the entire API environment (e.g., dev, test, stage, production, etc.) including services and data flow.

3 Use open standards to simplify API documentation that is then integrated into CI/CD pipeline.

How Cequence Can Help

API Sentinel

Detect shadow APIs and hidden endpoints with usage and API specification conformance analysis

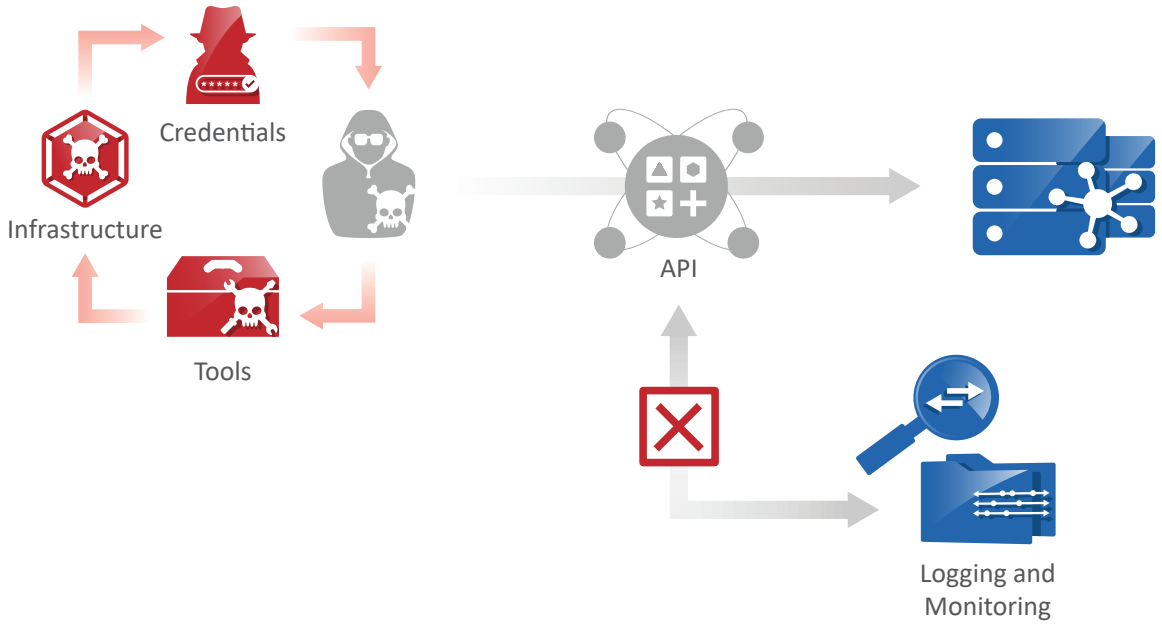
Bot Defense

Detect and block traffic to shadow APIs and hidden endpoints

#10: Insufficient Logging and Monitoring

Layperson's Description

Insufficient logging, monitoring, and alerts allows attacks in progress to go undetected.



How Threat Actors Leverage This Threat

Threat actors indirectly leverage this threat by successfully executing an attack without being detected during and after the event. Examples of insufficient logging and monitoring include misconfigured API logging levels; messages lack detail; log integrity is not guaranteed; APIs published outside of existing logging and monitoring infrastructure.

#10: Insufficient Logging and Monitoring

Top 3 Prevention Tips

1. Define and enforce logging/monitoring requirements for all APIs that includes sufficient detail to uncover malicious activity (e.g., failed authentication attempts, denied access, and input validation errors).

2. Treat logs as sensitive data at rest, in motion and in use.

3. Use a log format that is consumable by log management, SIEMs, analysis, and archival tools.

How Cequence Can Help

API Sentinel

Continuous API visibility and usage analysis, including hidden endpoints and those listed in OpenAPI schemas

Bot Defense

Integrate with SIEM and backend systems to facilitate detection and blocking of bad/attack traffic

Next Steps

Organizations are rapidly adopting an API-first development methodology because of APIs' power, flexibility, and efficiency. The shopping, finance, manufacturing, and marketing apps we use every day are all based on APIs, connecting back to compute resources located elsewhere — the cloud, the data center, or both. Unfortunately, threat actors love APIs for the same reasons that developers do. APIs are susceptible to a range of automated attacks and vulnerability exploits that can lead to data loss and system compromise. To protect existing and future APIs, organizations need to implement a forward looking API security solution that unifies runtime API visibility, security risk monitoring, and patented behavioral fingerprinting technology to consistently detect and protect against ever-evolving online attacks.

Protect Your APIs While Empowering Your Developers with Cequence Security

Schedule Your Cequence API Security Platform Demo at cequence.ai/demo

CEQUENCE® SECURITY

100 S. Murphy Avenue, Suite 300, Sunnyvale, CA 94086, 1-650-437-6338, info@cequence.ai, www.cequence.ai