



Session 1: The new OWASP Top 10 API Security 2023

James Lee | Security Solutions Architect | F5

API Security is in the Grey area...

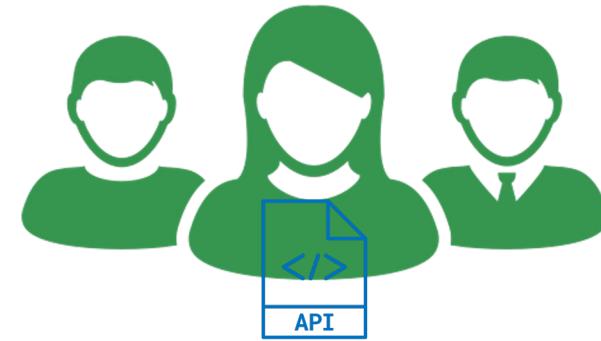
API Owners and SecOps

SecOps



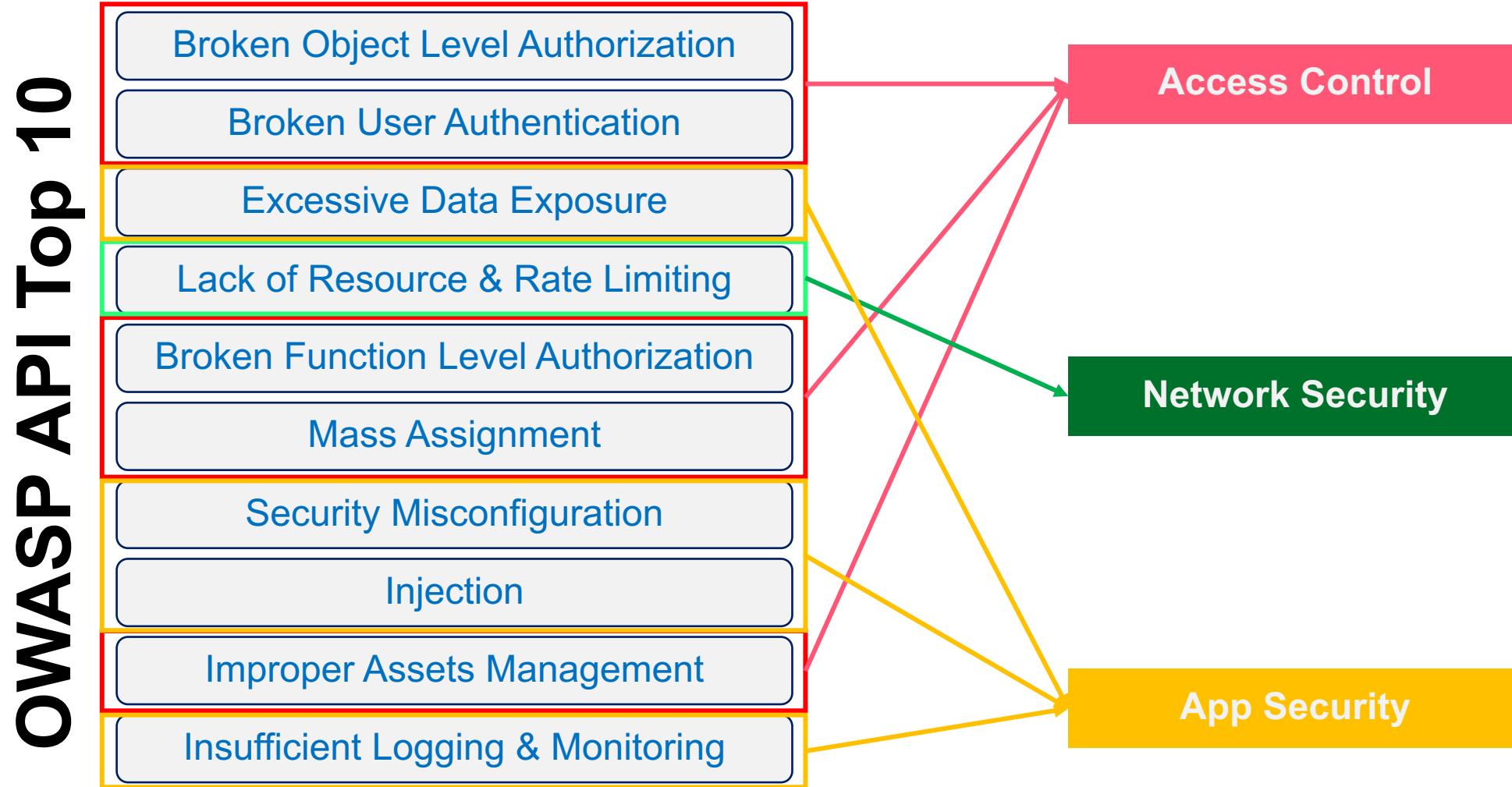
- Responsible for the organization's security policy and compliance
- Operating security policies in NG F/W, WAF, and Anti-Virus
- Managing all alerts, including FPs and FNs
- Normally, SecOps provides the guideline of API security to API owners but they don't have control and visibility of API G/W or API management.

API Owners (Dev Team)



- Responsible for designing the business logic with APIs
- Operating API G/W, API management and required IAM(Identity and Access Management) for APIs
- API Owners are not considered security experts.
- Normally, API Owners manage the API G/W policies and code-level security but do not manage the organization-wide security policy or compliance.

OWASP API Top 10 - 2019



OWASP API Top 10 - 2023

Broken Object Level Authorization

Broken User Authentication

Excessive Data Exposure

Lack of Resource & Rate Limiting

Broken Function Level Authorization

Mass Assignment

Security Misconfiguration

Injection

Improper Assets Management

Insufficient Logging & Monitoring

Broken Object Level Authorization

Broken Authentication

Broken Object Property Level Authorization

Unrestricted Resource Consumption

Broken Function Level Authorization

Unrestricted Access to Sensitive Business Flows

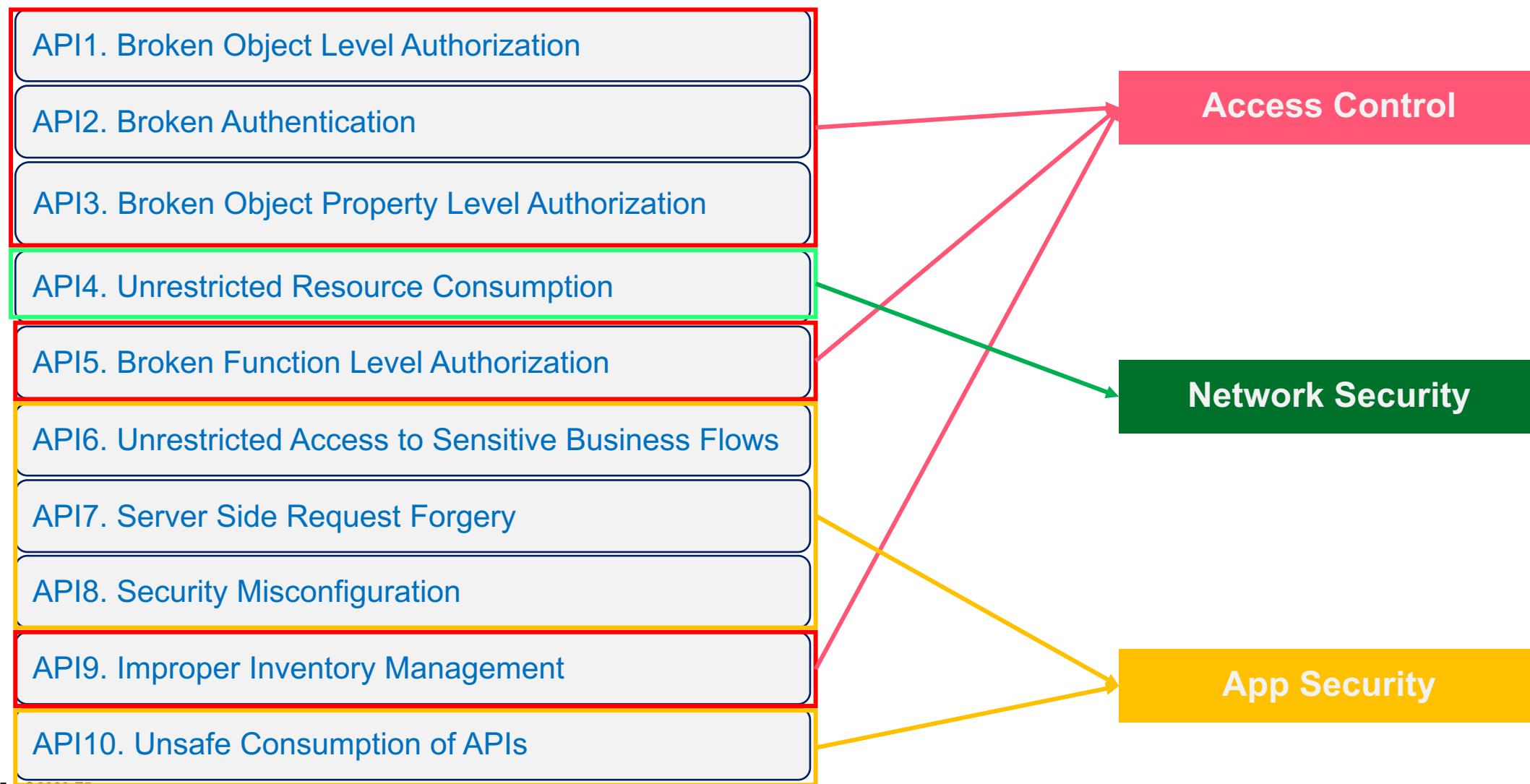
Server Side Request Forgery

Security Misconfiguration

Improper Inventory Management

Unsafe Consumption of APIs

OWASP API Top 10 - 2023

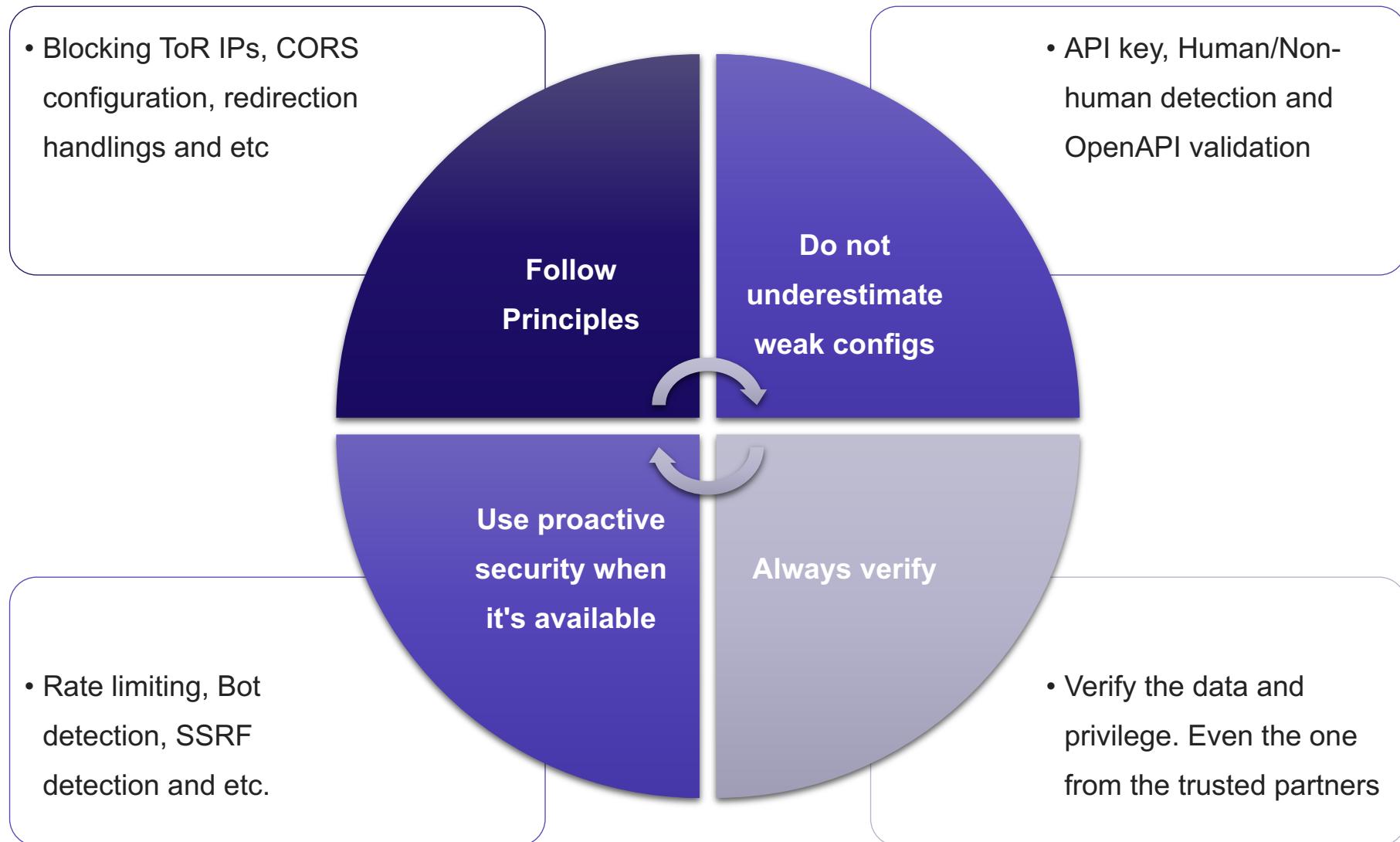


How Can You Comply with the new OWASP 2023?

The screenshot shows the OWASP API Top 10 - 2023 website. The top navigation bar includes a logo, the title "API8:2023 Security Misconfiguration", a search bar, and a navigation menu with items like "Home", "About", "Contact", "Logout". The main content area displays a list of bullet points under the heading "API8:2023 Security Misconfiguration".

- Ensure that all API communications from the client to the API server and any downstream/upstream components happen over an encrypted communication channel (TLS), regardless of whether it is an internal or public-facing API.
- Be specific about which HTTP verbs each API can be accessed by: all other HTTP verbs should be disabled (e.g. HEAD).
- APIs expecting to be accessed from browser-based clients (e.g., WebApp front-end) should, at least:
 - implement a proper Cross-Origin Resource Sharing (CORS) policy
 - include applicable Security Headers
 - Restrict incoming content types/data formats to those that meet the business/ functional requirements.
- Ensure all servers in the HTTP server chain (e.g. load balancers, reverse and forward proxies, and back-end servers) process incoming requests in a uniform manner to avoid desync issues.
- Where applicable, define and enforce all API response payload schemas, including error responses, to prevent exception traces and other valuable information from being sent back to attackers.

Guidelines of the new OWASP API Top 10 - 2023



API Security Maturity Model

API Security Maturity Model

Gartner.

Gartner's API Strategy Maturity Model



FOUNDATIONAL

Refreshed: 27 April 2021 | Published: 21 October 2019 ID: G00451168

Analyst(s): Saniye Alaybeyi, Mark O'Neill

IT organizations struggle to evolve their processes for developing, delivering and managing APIs for integration and digital business transformation. Application leaders must assess and improve their API strategy using five key dimensions explained in this research.



FOUNDATIONAL DOCUMENT

This research is reviewed periodically for accuracy. Last reviewed on 27 April 2021.

Key Findings

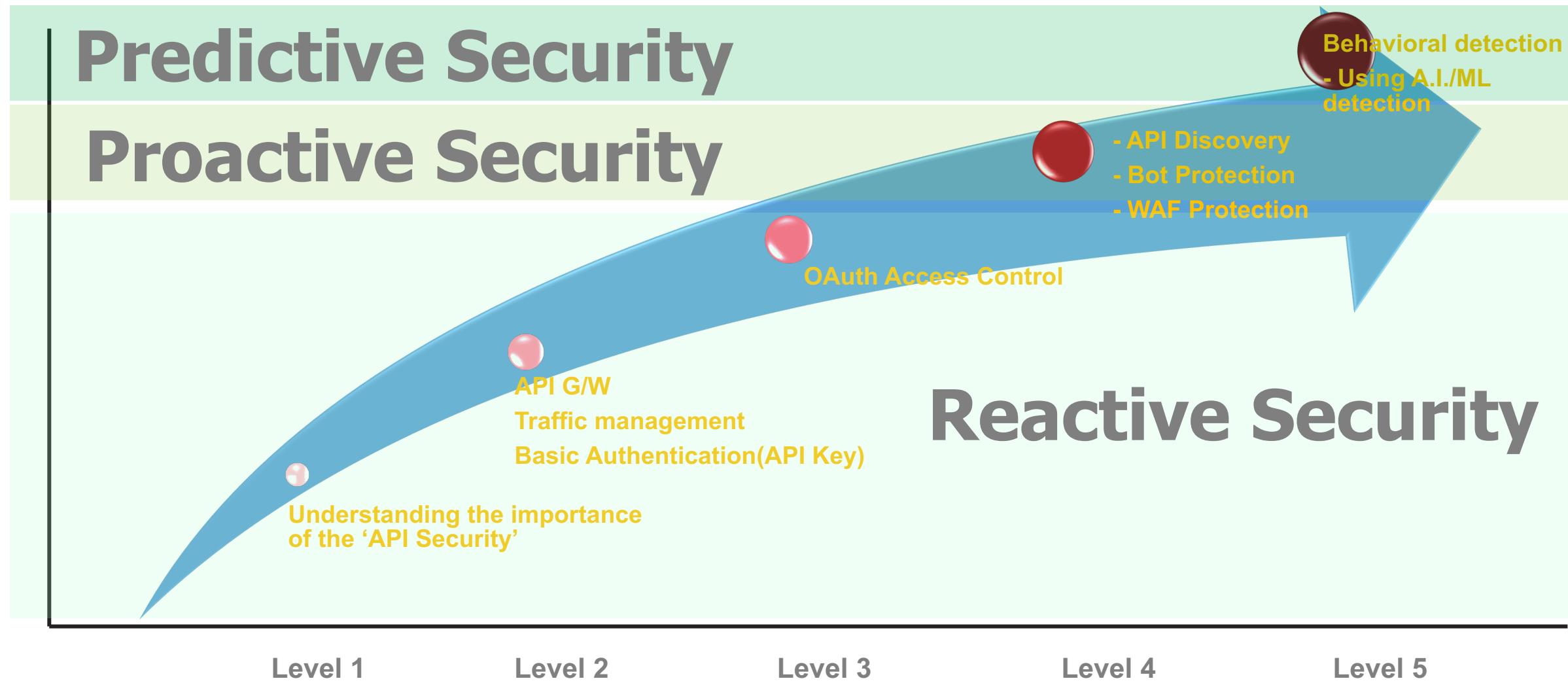
- Misalignment between API strategy and business goals results in failure to capture the benefits initially envisioned from APIs.
- Poor API design limits usage and results in deployment delays and cost overruns.
- Failing to plan for API life cycle management results in an inadequate feedback mechanism for API product managers and consumers.
- Without metrics for API usage and performance there is no visibility into how APIs are working and impacting customers and services, and inhibiting improvements.

API Security Maturity Model

Communications

- **Level 0:** There is no evidence or awareness for authentication, traffic management and privacy mechanisms or policies.
- **Level 1:** API teams understand the importance of an authentication, authorization, traffic management, quality of service (caching), interface protocols and security mechanisms and policies. Some of these basic mechanisms exist in a few silos within the organization and are implemented at various levels, but without API gateway in place.
- **Level 2:** Teams are implementing authentication, authorization and messaging mechanisms based on industry-accepted practices but in isolation of each other. Beyond individual team or solution implementation, standards and policies do not exist for security, interface, data privacy and traffic management (throttling). Instead, these are being implemented on a team-by-team or solution-by-solution basis, typically using API gateways.
- **Level 3:** Core principles, basic standards and policies are defined across the enterprise for API security, traffic management, quality of service, data privacy, service routing and orchestration. Standards are adopted at various levels across the organization, in many cases reactively.
- **Level 4:** API security, privacy, quality and communication standards are being proactively and consistently adopted across the organization. There is active monitoring and proactive intervention ensuring compliance with the defined standards. Developer consumption quotas and traffic prioritization mechanisms are in place. Robust caching, service orchestration and routing (load balancing) capabilities are in place. These may be provided by a full life cycle API management solution, which includes API gateways.
- **Level 5:** Organizations are transitioning their security and traffic management capabilities from proactive to predictive. Full life cycle API management and web application firewalls (WAFs) work together in unison to detect threats and anticipate traffic and required service levels. Network behavior analysis and content inspection mechanisms are in place to detect misuse and attacks. Visualization and advanced reporting monitoring mechanisms are in place.

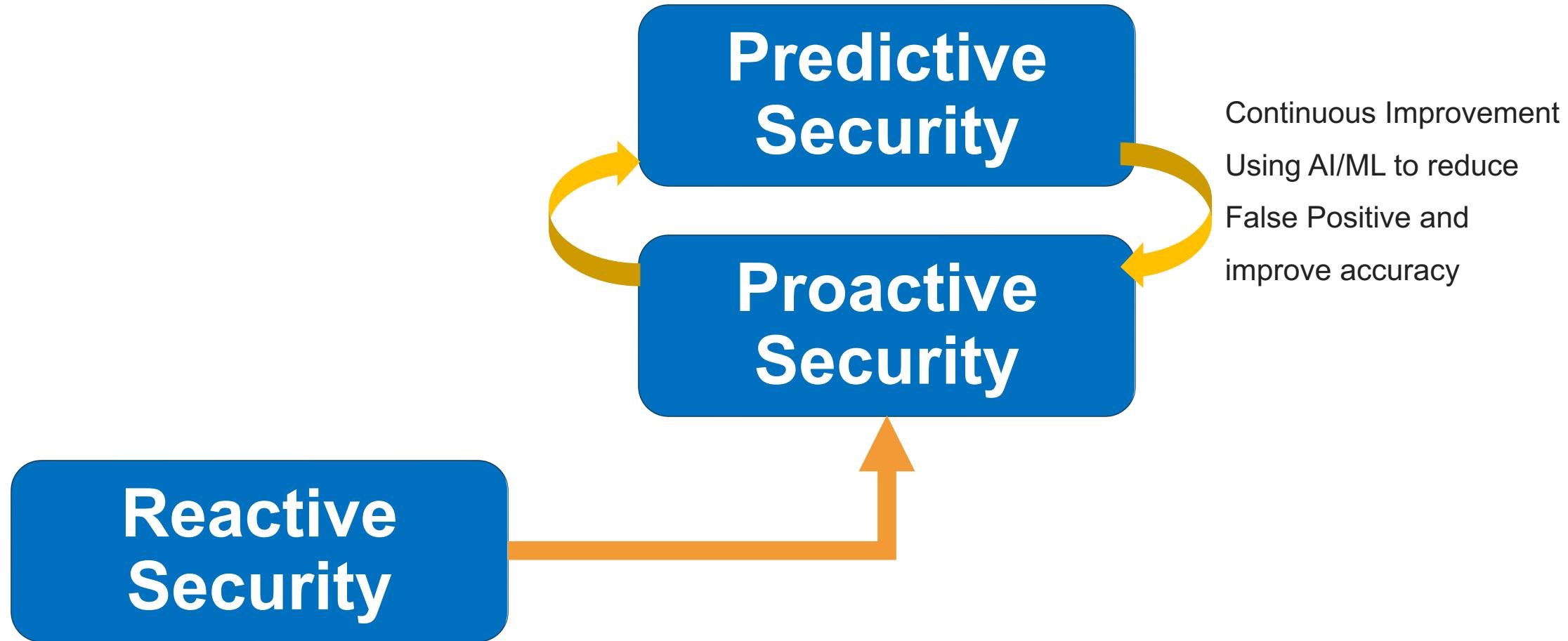
API Security Maturity Model



Reactive vs Proactive vs Predictive

	Reactive	Proactive	Predictive
Premise	<ul style="list-style-type: none"> Reactive cyber security introduces defending against attacks that have already happened. 	<ul style="list-style-type: none"> Proactive cyber security involves identifying and addressing security risks before an attack occurs. 	<ul style="list-style-type: none"> Using contextual analysis to identify threats before they become incidents
Examples	<ul style="list-style-type: none"> Adding IPs to Deny-list after 10 failed logon attempts 	<ul style="list-style-type: none"> Payload inspection by a WAF/WAAP 	<ul style="list-style-type: none"> Behavioral – Malicious or Suspicious user and API client detection
Benefits	<ul style="list-style-type: none"> Stop-gap when all controls fail 	<ul style="list-style-type: none"> Real runtime protection 	<ul style="list-style-type: none"> Advance warning and protection
Powered By	<ul style="list-style-type: none"> Logs and telemetry data indicating attacks 	<ul style="list-style-type: none"> Real time analysis of traffic 	<ul style="list-style-type: none"> Machine learning on telemetry data
Gartner Maturity Model Levels	<ul style="list-style-type: none"> Level 1 ~ Level 3 	<ul style="list-style-type: none"> Level 4 	<ul style="list-style-type: none"> Level 5

API Security Maturity Model at Runtime





Session 1: The new OWASP Top 10 API Security 2023

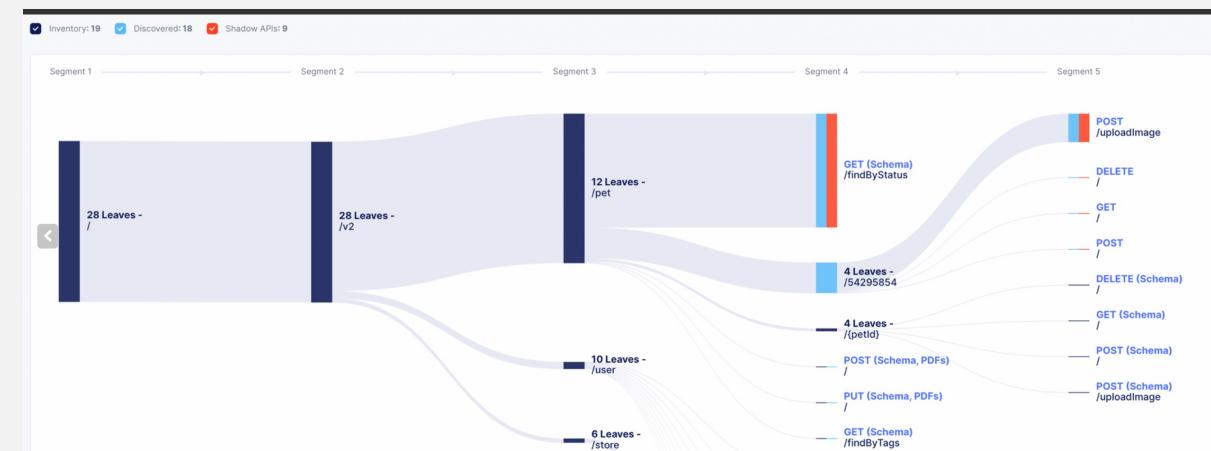
API8:2023 – Security Misconfiguration

API8: Security Misconfiguration

Security misconfiguration vulnerabilities occur when an API component is susceptible to attack due to a misconfiguration or nonsecure configuration

- API Inventory/ documentation incomplete or missing
- APIs don't conform to OpenAPI specifications
- Authentication token includes insecure configuration
- PII data exists in the JWT or API request body
- CORS misconfiguration

- An F&B outlet had an API not meant for external users exposed putting PII information for 100 Million users at risk
- Attacker used brute force to detect the vulnerable API endpoint



Source: samcurry.net

API8: Security Misconfiguration

Examples

Case#01.

- An attacker finds the .bash_history file under the root directory of the server, which contains commands used by the DevOps team to access the API.



Case#02.

- An attacker finds some API endpoints that are accessible from the internet but don't have the authentication mechanism.

Case#03.

- Some API endpoints include potentially critical data in their token or header or body, and an attacker can exploit those data potentially.

API8: Security Misconfiguration

How to Stop the Security Misconfiguration

API8:2023 Security Misconfiguration

X A Search

Contents

WASP

ard

ction

Notes

urity Risks

Top 10 API Security
2023

23 Broken Object Level
zation

23 Broken Authentication

23 Broken Object
y Level Authorization

23 Unrestricted Resource
noption

- Ensure that all API communications from the client to the API server and any downstream/upstream components happen over an encrypted communication channel (TLS), regardless of whether it is an internal or public-facing API.
- Be specific about which HTTP verbs each API can be accessed by: all other HTTP verbs should be disabled (e.g. HEAD).
- APIs expecting to be accessed from browser-based clients (e.g., WebApp front-end) should, at least:
- implement a proper Cross-Origin Resource Sharing (CORS) policy
- include applicable Security Headers
- Restrict incoming content types/data formats to those that meet the business/ functional requirements.
- Ensure all servers in the HTTP server chain (e.g. load balancers, reverse and forward proxies, and back-end servers) process incoming requests in a uniform manner to avoid desync issues.
- Where applicable, define and enforce all API response payload schemas, including error responses, to prevent exception traces and other valuable information from being sent back to attackers.

API8: Security Misconfiguration

How to Stop the Security Misconfiguration

OpenAPI Spec Validation

- It is always recommended to enable OpenAPI validation to perform a basic schema validation for your APIs.

Continuous Monitoring

- Continuous assessment of the security effectiveness of API endpoints.

TLS Policy Setting

- Need to have the capability to set up the different TLS policies based on the backend apps' requirements.

Strong AuthN/AuthZ

- Need to consider strong authN/authZ methods for your APIs with standard-based protocols such as OAuth/OIDC or SAML.

HTTP Methods Control

- Enable only relevant HTTP methods for the specific API endpoints.

CORS Policy

- Need to configure the correct CORS settings in your apps.

API8: Security Misconfiguration

Protection#01 – CORS Configuration

```
j.lee@C02FPFZPM6M ~ %  
j.lee@C02FPFZPM6M ~ % curl -I -H "Origin: https://www.evil.com"  https://james.apac-ent.f5demos.com  
HTTP/2 200  
access-control-allow-origin: *  
x-content-type-options: nosniff  
x-frame-options: SAMEORIGIN  
feature-policy: payment 'self'  
x-recruiting: /#/jobs  
accept-ranges: bytes  
cache-control: public, max-age=0  
last-modified: Thu, 20 Jul 2023 02:55:19 GMT  
etag: W/"7c3-18971392a77"  
content-type: text/html; charset=UTF-8  
content-length: 1987  
vary: Accept-Encoding  
date: Thu, 20 Jul 2023 12:24:33 GMT  
x-envoy-upstream-service-time: 7  
set-cookie: 8bd85=1689855873089-779805097; Path=/; Domain=f5demos.com; Expires=Fri, 20 Oct 2023 12:24:33 GMT  
set-cookie: 8bd803=HHC33JWEDc6kXUZPt2cFmIril9/iBbs+FTi7yazF16ENk9KHP+WVa6md5t+aoW2N1c9+HdZu0Ciauth989Gjxhn1F4wGZnfnxak12ZPLk0ivT0XPNDxbUgj5AgQ9QOHmluTsPTbNyFS8tau3dc6McRprJKtNS4wDTSzuLNXD9PDiu6B; path=/  
x-volterra-location: sg3-sin  
server: volt-adc
```

A red arrow points from the word "access-control-allow-origin" in the highlighted section to a callout box.

- The webserver has a vulnerable CORS setting with the wildcard(*).

API8: Security Misconfiguration

Protection#01 – CORS Configuration

CORS Policy Configuration

Order Allow Origin ⓘ Actions

There are no items added yet. Start by adding first item.

+ Add Item

Order	Allow Origin Regex ⓘ	Actions
1	https://([-a-z0-9]+)\.f5\.com	
2	https://([-a-z0-9]+)\.nginx\....	

+ Add Item

Allow Methods ⓘ

GET,POST

Allow Headers ⓘ

Expose Headers ⓘ

Maximum Age ⓘ

30

A red box highlights the table containing the two CORS policy entries. A red arrow points from the text "In XC WAAP, a user can configure the CORS policy. In this case, 'https://*.f5.com' and 'https://*.nginx.com' are only allowed." to the second row of the table.

- In XC WAAP, a user can configure the CORS policy. In this case, 'https://*.f5.com' and 'https://*.nginx.com' are only allowed.

API8: Security Misconfiguration

Protection#01 – CORS Configuration

Header Options

Add Request Headers ⓘ

There are no items added yet. Start by adding first item.

+ Add Item

Order Remove Request Headers ⓘ Actions

There are no items added yet. Start by adding first item.

+ Add Item

Add Response Headers ⓘ

There are no items added yet. Start by adding first item.

+ Add Item

Order Remove Response Headers ⓘ Actions

1 access-control-allow-origin ×

+ Add Item

- By default, XC WAAP does not replace the server's original header.
- A user needs to remove the server's original response using this menu. Then, XC will respond with its CORS configuration.

API8: Security Misconfiguration

Protection#01 – CORS Configuration

```
j.lee@C02FPFZPM6M ~ %  
j.lee@C02FPFZPM6M ~ % curl -I -H "Origin: https://www.f5.com" https://james.apac-ent.f5demos.com  
HTTP/2 200  
x-content-type-options: nosniff  
x-frame-options: SAMEORIGIN  
feature-policy: payment 'self'  
x-recruiting: /#/jobs  
accept-ranges: bytes  
cache-control: public, max-age=0  
last-modified: Thu, 20 Jul 2023 02:55:19 GMT  
etag: W/"7c3-18971392a77"  
content-type: text/html; charset=UTF-8  
content-length: 1987  
vary: Accept-Encoding  
date: Thu, 20 Jul 2023 12:17:25 GMT  
x-envoy-upstream-service-time: 7  
set-cookie: 8bd85=1689855445431-141961141; Path=/; Domain=f5demos.com; Expires=Fri, 20 Oct 2023 12:17:25 GMT  
set-cookie: 8bd803=j7PKaEUHKf1anRb6AsNw/pcd0JzZVzEniY2/ku3Q0o+05NnE3rqQ/fXC; path=/; expires=Fri, 20 Oct 2023 12:17:25 GMT  
access-control-allow-origin: https://www.f5.com  
x-volterra-location: sg3-sin  
server: volt-adc
```

• Since the request includes the correct origin server in the header, fXC responds accordingly.

API8: Security Misconfiguration

Protection#01 – CORS Configuration

```
j.lee@C02FPFZPM6M ~ %  
j.lee@C02FPFZPM6M ~ % curl -I -H "Origin: https://www.evil.com" https://james.apac-ent.f5demos.com  
HTTP/2 200  
x-content-type-options: nosniff  
x-frame-options: SAMEORIGIN  
feature-policy: payment 'self'  
x-recruiting: /#/jobs  
accept-ranges: bytes  
cache-control: public, max-age=0  
last-modified: Thu, 20 Jul 2023 02:55:19 GMT  
etag: W/"7c3-18971392a77"  
content-type: text/html; charset=UTF-8  
content-length: 1987  
vary: Accept-Encoding  
date: Thu, 20 Jul 2023 12:21:03 GMT  
x-envoy-upstream-service-time: 7  
set-cookie: 8bd85=1689855663446-855534115; Path=/; Domain=f5demos.com; Expires=Fri, 20 Oct 2023 12:21:03 GMT  
set-cookie: 8bd803=+IJPxIAYZF3NCq1hi7XlZTxbirrP63rHR02105yCXnyH9AbG5iuFM+9PKGWUc1K1/G1rft511Wb1L5w3kI1p5wpwlx/0VQ86qIzhtR+m/8LxwHayEYx6Dq5SzCspKaUbkx/mVdoXE3Sjj4QsnWzmTG+gqBy34XgbZ6YDUaI0GHFRL1; path=/  
x-volterra-location: sg3-sin  
server: volt-adc
```

- Since the request has the 'un-allowed' origin server, XC WAAP does not respond accordingly. Thus, the request would be rejected in the user's browser.

API8: Security Misconfiguration

Protection#02 – TLS Configuration

The screenshot shows the 'Domains and LB Type' configuration page. In the 'Domains' section, a domain 'jw-owasp.apac-ent.f5demos.com' is listed with an order of 1. Below this is an 'Add Item' button. In the 'Load Balancer Type' section, 'HTTPS with Automatic Certificate' is selected. Underneath, there are checkboxes for 'HTTP Redirect to HTTPS' and 'Add HSTS Header'. The 'HTTPS Listen Port Choice' section shows 'HTTPS Listen Port' set to '443'. A red box highlights the 'TLS Security Level' dropdown, which is set to 'Low'. A red arrow points from this highlighted field to a callout box containing the following text:

- Users can configure TLS security level in XC WAAP. In this example, we set it to 'Low' which means the TLS1.0 is allowed.

API8: Security Misconfiguration

Protection#02 – TLS Configuration

```
j.lee@C02FPFZPM6M ~ %  
j.lee@C02FPFZPM6M ~ % curl -vv --tlsv1.0 --tls-max 1.0 https://jw-owasp.apac-ent.f5demos.com/index1.php  
* Trying 72.19.3.189:443...  
* Connected to jw-owasp.apac-ent.f5demos.com (72.19.3.189) port 443 (#0)  
* ALPN: offers h2  
* ALPN: offers http/1.1  
* CAfile: /etc/ssl/cert.pem  
* CApth: none  
* [CONN-0-0][CF-SSL] TLSv1.0 (OUT), TLS handshake, Client hello (1):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS handshake, Server hello (2):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS handshake, Certificate (11):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS handshake, Server key exchange (12):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS handshake, Server finished (14):  
* [CONN-0-0][CF-SSL] TLSv1.0 (OUT), TLS handshake, Client key exchange (16):  
* [CONN-0-0][CF-SSL] TLSv1.0 (OUT), TLS change cipher, Change cipher spec (1):  
* [CONN-0-0][CF-SSL] TLSv1.0 (OUT), TLS handshake, Finished (20):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS change cipher, Change cipher spec (1):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS handshake, Finished (20):  
* SSL connection using TLSv1 / ECDHE-ECDSA-AES128-SHA  
* ALPN: server accepted h2  
* Server certificate:  
* subject: CN=jw-owasp.apac-ent.f5demos.com  
* start date: Jul 14 04:17:07 2023 GMT  
* expire date: Oct 12 04:17:06 2023 GMT  
* subjectAltName: host "jw-owasp.apac-ent.f5demos.com" matched cert's  
* issuer: C=US; O=Let's Encrypt; CN=R3  
* SSL certificate verify ok.
```

• TLS v1.0 works correctly.
"jw-owasp.apac-ent.f5demos.com"

API8: Security Misconfiguration

Protection#02 – TLS Configuration

The screenshot shows the 'Domains and LB Type' configuration page. A red box highlights the 'TLS Security Level' dropdown, which is set to 'High'. A red arrow points from this highlighted field to a callout box containing the following note:

- Setting the TLS level to 'High' means TLSv1.2+ and PFS will be only allowed.

The configuration includes:

- Domains and LB Type** section:
 - Order: 1
 - Domains: jw-owasp.apac-ent.f5demos.com
 - Add Item button
- * Load Balancer Type**: HTTPS with Automatic Certificate
 - HTTP Redirect to HTTPS (unchecked)
 - Add HSTS Header (unchecked)
- HTTPS Listen Port Choice**:
 - HTTPS Listen Port (dropdown menu)
 - HTTPS Listen Port (input field: 443)
- * TLS Security Level**: High (highlighted by a red box and arrow)

API8: Security Misconfiguration

Protection#02 – TLS Configuration

```
j.lee@C02FPFZPMD6M ~ %  
j.lee@C02FPFZPMD6M ~ % curl -vv --tlsv1.0 --tls-max 1.0 https://jw-owasp.apac-ent.f5demos.com/index1.php  
* Trying 72.19.3.189:443...  
* Connected to jw-owasp.apac-ent.f5demos.com (72.19.3.189) port 443 (#0)  
* ALPN: offers h2  
* ALPN: offers http/1.1  
* CAfile: /etc/ssl/cert.pem  
* CApth: none  
* [CONN-0-0][CF-SSL] TLSv1.0 (OUT), TLS handshake, Client hello (1):  
* [CONN-0-0][CF-SSL] TLSv1.0 (IN), TLS alert, protocol version (582):  
* LibreSSL/2.8.3: error:1400442E:SSL routines:CONNECT_CR_SRVR_HELLO:tlsv1 alert protocol version  
* Closing connection 0  
curl: (35) LibreSSL/2.8.3: error:1400442E:SSL routines:CONNECT_CR_SRVR_HELLO:tlsv1 alert protocol version  
j.lee@C02FPFZPMD6M ~ %  
j.lee@C02FPFZPMD6M ~ %
```

- *TLS v1.0 is NOT allowed in XC WAAP.*

API8: Security Misconfiguration

Protection#02 – TLS Configuration

The screenshot shows the configuration interface for a domain named "jw-owasp.apac-ent.f5demos.com". The "Load Balancer Type" is set to "HTTPS with Automatic Certificate". The "TLS Security Level" is set to "Custom" with "Minimum TLS version" as "TLSv1.0" and "Maximum TLS version" as "TLSv1.2". A note on the right states: "A user also can custom TLS settings in XC WAAP."

Domains and LB Type

Order *Domains Actions

1 jw-owasp.apac-ent.f5demos.com

Show Advanced Fields

*** Load Balancer Type**

HTTPS with Automatic Certificate

High
TLS v1.2+ with PFS ciphers and strong crypto algorithms.

Medium
TLS v1.0+ with PFS ciphers and medium strength crypto algorithms.

Low
TLS v1.0+ including non-PFS ciphers and weak crypto algorithms.

Custom
Custom selection of TLS versions and cipher suites

High

*** Load Balancer Type**

HTTPS with Automatic Certificate

HTTPS Listen Port Choice

HTTPS Listen Port

443

*** TLS Security Level**

Custom

Minimum TLS version

TLSv1.0

Maximum TLS version

TLSv1.2

Order *Cipher Suites

1 TLS_ECDHE_ECDSA_WITH_...

Add Item

A user also can custom TLS settings in XC WAAP.

API8: Security Misconfiguration

Protection#03 – Continuous Monitoring

Endpoint Details

- With 'Security Posture' feature, XC can discover the potential misconfiguration of API endpoints.
- XC WAAP can discover any sensitive data in the API request or response.

API Endpoint	/api/updatePaymentInfoByld...	API Category	Discovered, Shadow
Base Path	—	Authentication	Authenticated, JWT Header
Risk Score	70	Protection Rule	⚠️ Not Configured ⚙ Configure
Method	PUT	Rate Limit	⚠️ Not Configured ⚙ Configure

Overview **Discovered** **Inventory OpenAPI** **Security Posture**

Vulnerabilities

- Weak JWT: JWT without Expiration... ● High
Created: 11:10 AM, May 22
Last Observed: 7:40 AM, Jul 20
- Weak JWT: "aud" claim is missing ... ● Low
Created: 11:10 AM, May 22
Last Observed: 7:40 AM, Jul 20
- Sensitive Data Found in JWT Token... ● None
Created: 11:10 AM, May 22
Last Observed: 7:40 AM, Jul 20

State Open
Category Weak Authentication / Authorization
Description
This vulnerability is detected if the JWT does not have an expiration date. The access tokens should be short-lived and are not revoked. A malicious actor that has obtained an access token can use it for extent of its lifetime. Adjusting the lifetime of an access token is a trade-off between improving system performance and increasing the amount of time that the client retains access after the user's account is disabled. Expiration period may range between 15-20 minutes to 60-90 minutes.
Risk Score
Attack impact: 70 (High)
Evidence
[Review Evidence Detection](#)
Remediation
Set expiration period to between 15-20 minutes to 60-90 minutes.

Endpoint Details

API Endpoint: /api/getPaymentInfo/me
API Category: Authentication
Discovered, Shadow: Un-Authenticated
Risk Score: 0
Protection Rule: ⚠️ Not Configured | ⚙ Configure
Method: GET
Rate Limit: ⚠️ Not Configured | ⚙ Configure

Overview **Discovered** **Inventory OpenAPI** **Security Posture**

Sensitive Data

Type	Section	Field	Created By
Credentials	Response Body	username	Built-In

Request 2XX JSON Learnt schema

Response 2 "type": "object",
3 "required": [
4 "oauth_uid",
5 "user_phone",
6 "oauth_provider",
7 "email",
8 "id",
9 "last_login",
10 "active",
11 "first_name",
12 "last_name",
13 "photoUrl",
14 "username",
15 "created_on",
16 "photo"

Headers

Body

OpenAPI

Authentication

Details

Search

F5

API8: Security Misconfiguration

Protection#03 – Continuous Monitoring

API Endpoint ⓘ	Method	Sensitive Data ⓘ	Threat Level ⓘ	Authentication S...	Authentication T...	API Category	Risk Score
/api/auth/oidc	GET	—	● None	Authenticated	JWT:Header	Discovered Shadow	70
/api/getPaymentInfo/me	GET	Credentials	● None	Un-Authenticated	—	Discovered Shadow	0
/trading/login.php	GET	—	● High	Un-Authenticated	—	Discovered Shadow	0
/api/getStock	GET	—	● None	Un-Authenticated	—	Discovered Shadow	0
				Un-Authenticated	—	Discovered Shadow	0
				Un-Authenticated	—	Discovered Shadow	0

• XC WAAP can discover the Shadow APIs, API authentication types, and their authentication state.

• In this example, some APIs don't have a proper auth mechanism(Un-Authenticated) and they're Shadow APIs which means API is unknown to API owners.

• Those APIs can be categorized as high-risky APIs potentially, and likely it happened because of 'Security Misconfiguration'.

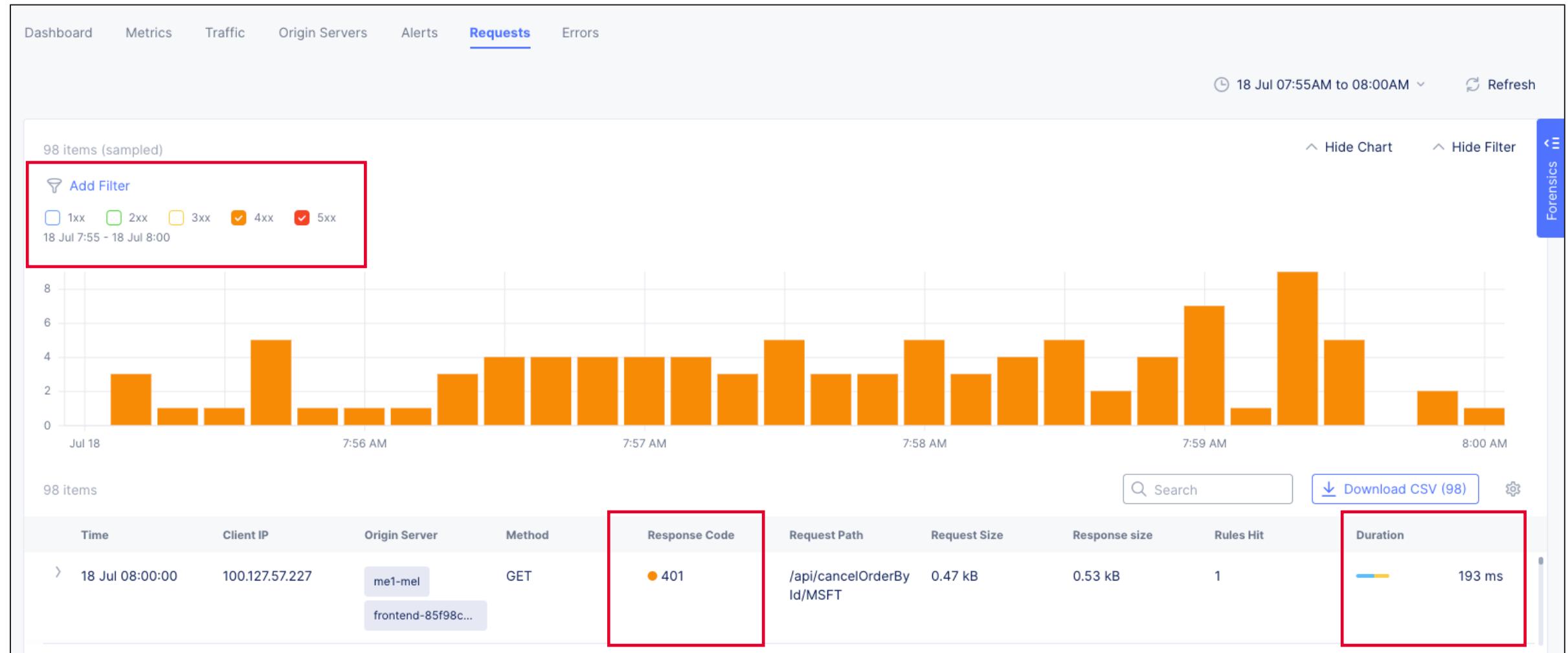
API8: Security Misconfiguration

Protection#04 – Monitoring API Metrics



API8: Security Misconfiguration

Protection#04 – Monitoring API Metrics



API8: Security Misconfiguration

Protection#04 – Monitoring API Metrics

Dashboard Metrics Traffic Origin Servers Alerts **Requests** Errors

⌚ 18 Jul 07:55AM to 08:00AM ⌚ Refresh

98 items (sampled)

98 items

Time	Client IP	Origin Server	Method	Response Code	Request Path	Request Size	Response size	Rules Hit
18 Jul 08:00:00	100.127.57.227	me1-mel frontend-85f98c...	GET	● 401	/api/cancelOrderBy Id/MSFT	0.47 kB	0.53 kB	1
18 Jul 07:59:55	100.127.57.227	me1-mel frontend-85f98c...	GET	● 401	/api/cancelOrderBy Id/AMZN	0.47 kB	0.53 kB	1
18 Jul 07:59:50	100.127.57.227	me1-mel frontend-85f98c...	GET	● 401	/api/cancelOrderBy Id/FFIV	0.47 kB	0.53 kB	1
18 Jul 07:59:39	192.42.116.186	me1-mel frontend-85f98c...	POST	● 404	/cart/checkout	0.69 kB	0.89 kB	1

Show Chart Show Filter

Search Download CSV (98)

Forensics Apply

Top country

- LU 4.1%
- NL 4.1%

Top asn

- 1337 Services GmbH(2105...) 4.1%
- PONYNET(53667) 4.1%

Top src_ip

- 192.42.116.186 67.3%
- 100.127.57.227 24.5%

Top tls_fingerprint

- d0ee3237a14bbd89ca4d... 67.3%
- 398430069e0a8ecfbc8d... 24.5%

10 50 100 items per page Page 1 of 10 < >



API8: Security Misconfiguration

Protection#05 – OpenAPI Validation

Time	Country,city	Src IP	Method	Rsp Code	Event Type	Mode	Authority	Request Path	Actions
14 Jul 15:05:33	SG,UNKNOWN	111.223.104.76	POST	302	● API	Report	james.apac-ent.f5demos.com	/profile/image/url	...
Information JSON									
Src					Request				
src_ip	111.223.104.76				req_id	6ea75fba-dc92-4c53-a384-e60930da1176			
city	UNKNOWN				authority	james.apac-ent.f5demos.com			
region	UNKNOWN				req_path	/profile/image/url			
country	SG				req_params	—			
asn	StarHub Ltd(4657)				method	POST			
browser_type	Chrom				req_size	/profile/image/url			
device_type	Other				rsp_size	2033			
user_agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/11...				rsn_code	598			
src_site	sa3-sin					302			

API8: Security Misconfiguration

Protection#06 – HTTP Verbs Control

The screenshot shows a configuration page for an API endpoint. Key fields include:

- Action**: Set to "Deny".
- Domain**: Set to "Any Domain".
- API Endpoint**: Set to "/anything".
- HTTP Methods**: Set to "POST", "PUT", and "DELETE".
- HTTP Query Parameters**: Placeholder text: "There are no items added yet. Start by adding first item." with an "Add Item" button.
- HTTP Headers**: Placeholder text: "There are no items added yet. Start by adding first item." with an "Add Item" button.

A red box highlights the "Action" field, and another red box highlights the "HTTP Methods" section. Red arrows point from these highlighted areas to the following descriptive text.

• A user can limit the HTTP methods, query and headers for specific API endpoints manually.

API8: Security Misconfiguration

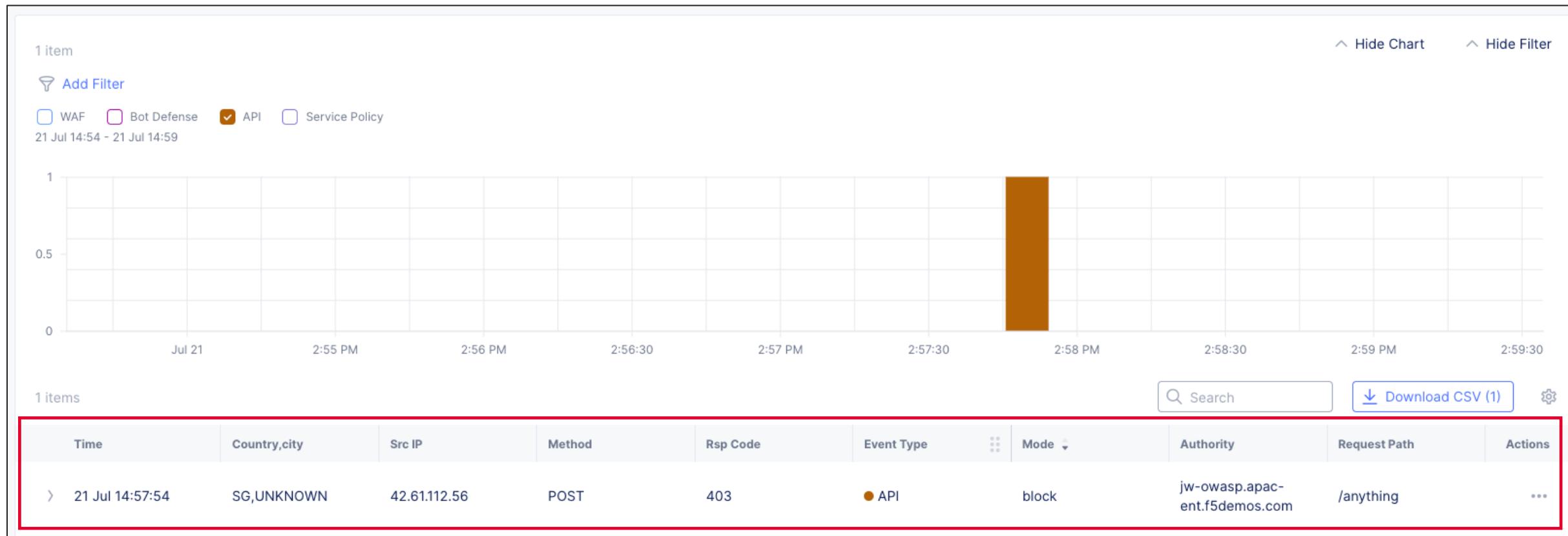
Protection#06 – HTTP Verbs Control

```
j.lee@C02FPFZPMD6M Desktop %
j.lee@C02FPFZPMD6M Desktop % curl -X POST https://jw-owasp.apac-ent.f5demos.com/anything
<html><head><title>Error Page</title></head>
<body>The requested URL was rejected. Please consult with your administrator.<br/><br/>
Your support ID is 2893bba9-7513-4d71-bf27-0febcd6ce174b<h2>Error 403 – Forbidden</h2>F5 site: sg3-sin<br/><br/><a href='javascript:history.back();'>[Go Back]</a></body></html>
j.lee@C02FPFZPMD6M Desktop %
j.lee@C02FPFZPMD6M Desktop %
j.lee@C02FPFZPMD6M Desktop % curl -X GET https://jw-owasp.apac-ent.f5demos.com/anything
{
  "args": {},
  "data": "",
  "files": {},
  "form": {},
  "headers": {
    "Accept": "*/*",
    "Host": "jw-owasp.apac-ent.f5demos.com",
    "User-Agent": "curl/7.87.0",
    "X-Envoy-External-Address": "42.61.112.56",
    "X-F5-Request-Id": "21e1e5d9-5381-44b5-90c3-ecac0c778c53"
  },
  "json": null,
  "method": "GET",
  "origin": "42.61.112.56",
  "url": "https://jw-owasp.apac-ent.f5demos.com/anything"
}
j.lee@C02FPFZPMD6M Desktop %
```

• *GET is allowed, but POST is blocked.*

API8: Security Misconfiguration

Protection#06 – HTTP Verbs Control





Session 1: The new OWASP Top 10 API Security 2023

API10:2023 – Unsafe Consumption of APIs

API10: Unsafe Consumption of APIs

Developers tend to trust data received from 3rd party APIs, especially for APIs from well-known companies. Thus, it sometimes has a lack of input validation and sanitization for those APIs.

- Does not properly validate and sanitize the data from 3rd party companies.
- Blinely follows redirections.
- No timeout mechanism with 3rd party APIs.
- No limit the number of resources.

Scenario #2

An API integrates with a third-party service provider to safely store sensitive user medical information. Data is sent over a secure connection using an HTTP request like the one below:

```
POST /user/store_phr_record
{
  "genome": "ACTAGTAG__TTGADDAAIICCTT..."
}
```

Bad actors found a way to compromise the third-party API and it starts responding with a 308 Permanent Redirect to requests like the previous one.

```
HTTP/1.1 308 Permanent Redirect
Location: https://attacker.com/
```

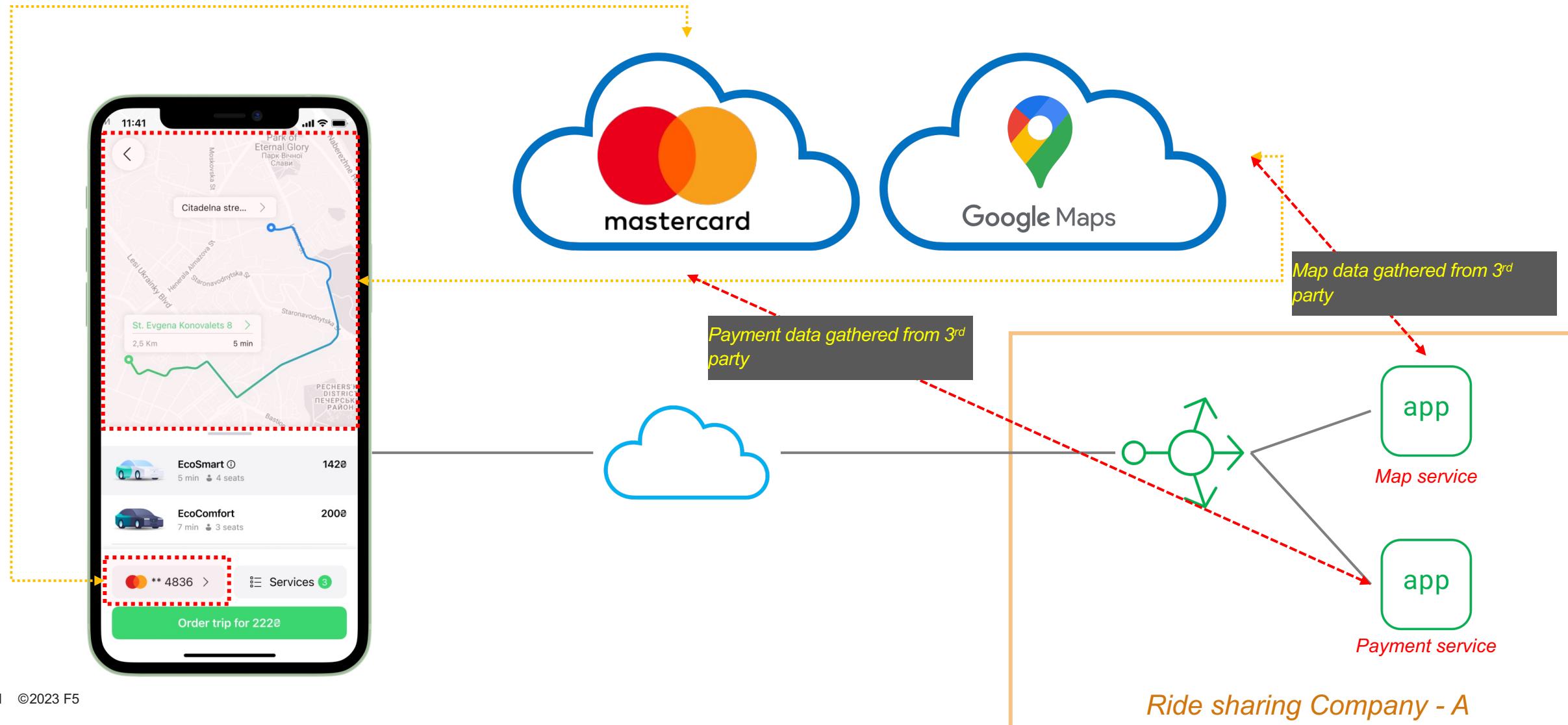
Since the API blindly follows the third-party redirects, it will repeat the exact same request including the user's sensitive data, but this time to the attacker's server.



**OWASP API
SECURITY TOP 10**

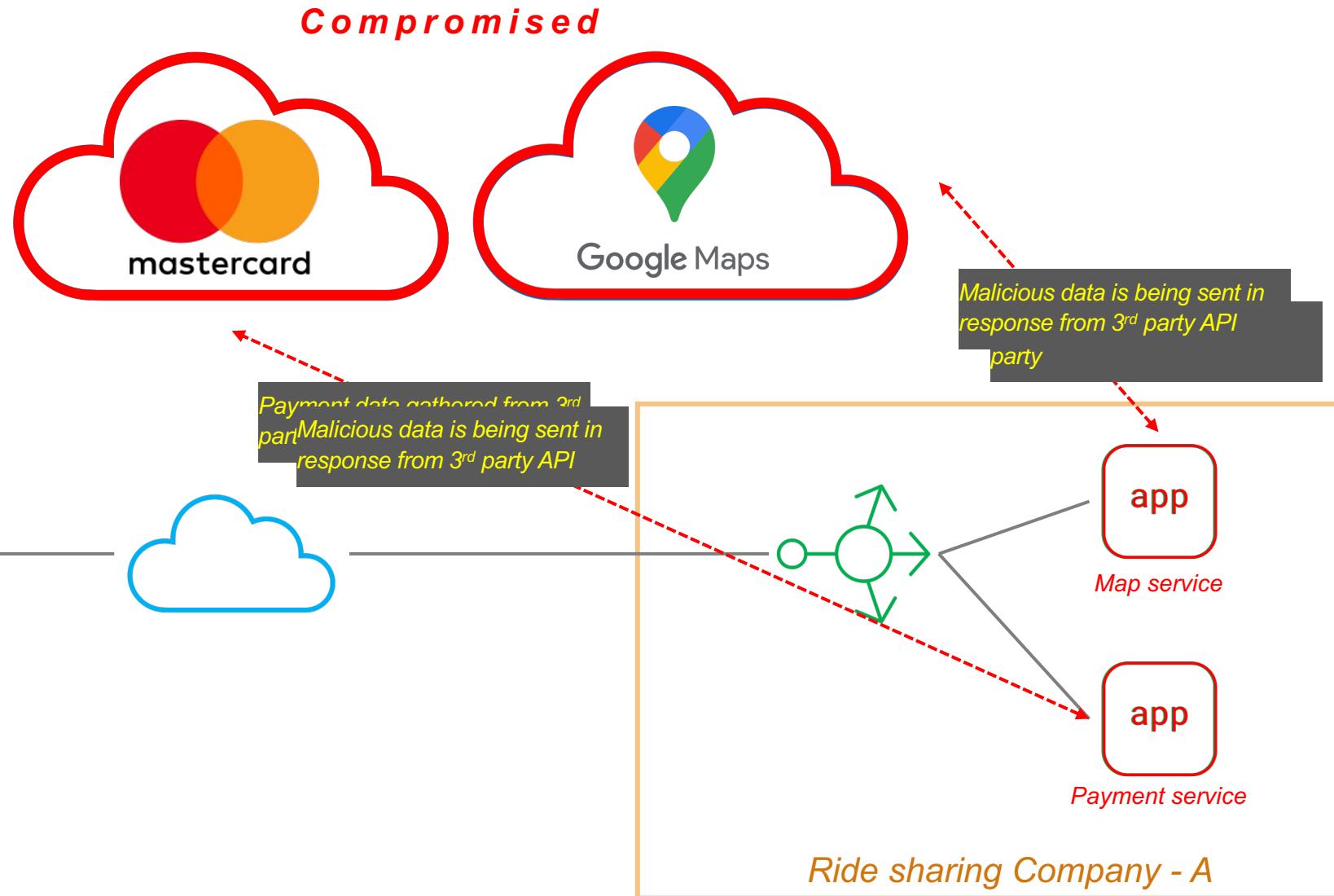
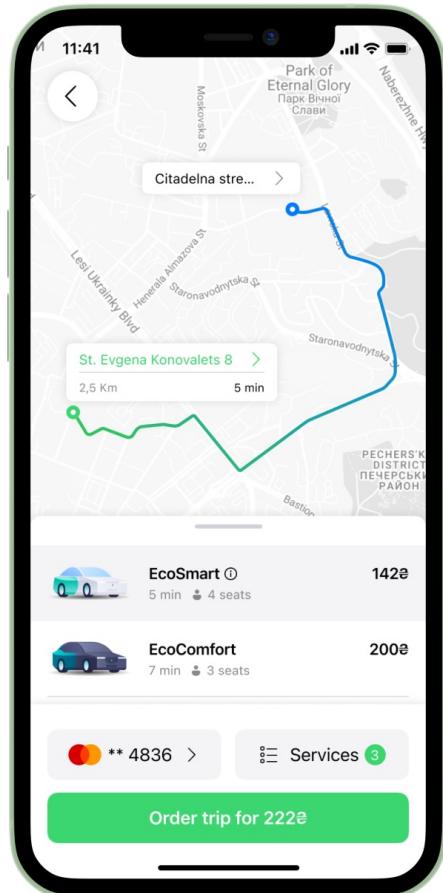
API10: Unsafe Consumption of APIs

How does it work? And why does it risky?



API10: Unsafe Consumption of APIs

How does it work? And why does it risky?



OWASP Recommendation



API10:2023 Unsafe Consumption of APIs



Search



OWASP/API-Security
☆ 1.8k ♫ 363

2023
Noti

Tabl

Abor

Fore

Intro

Rele

API :

OWA

Risk

API1

Example Attack Scenarios

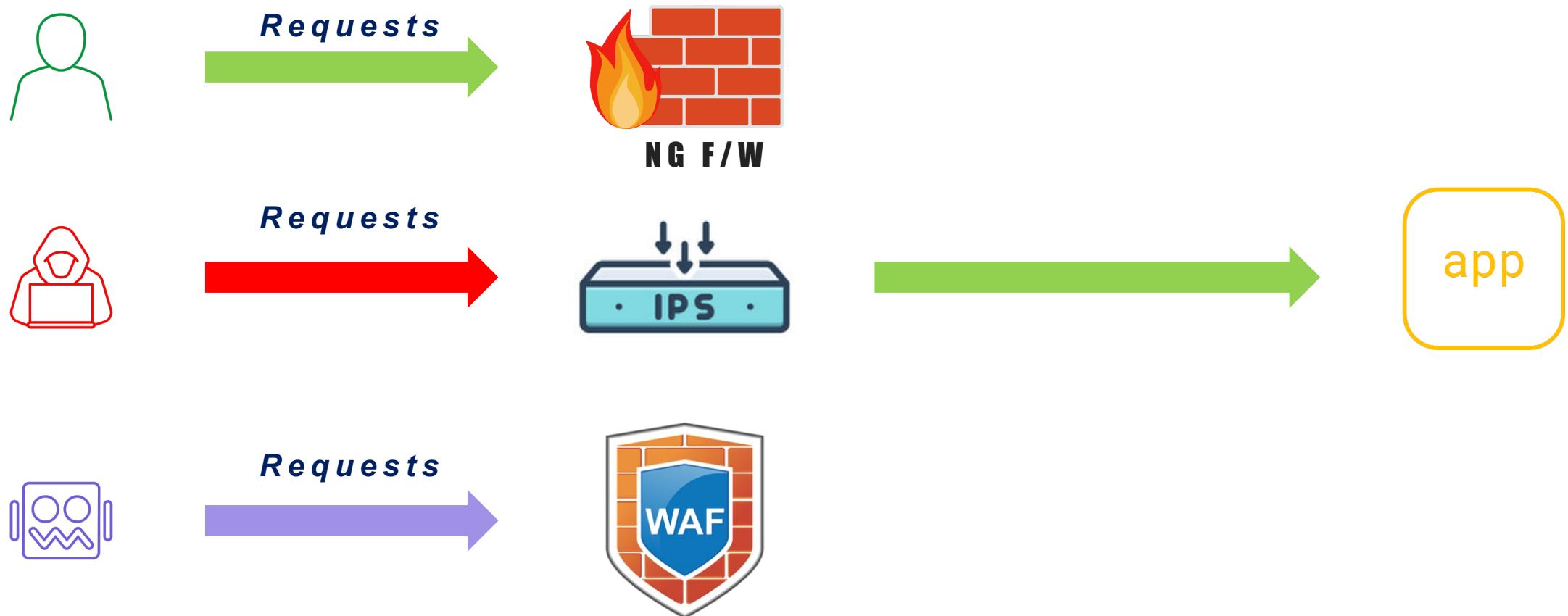
Scenario #1

An API relies on a third-party service to enrich user provided business addresses. When an address is supplied to the API by the end user, it is sent to the third-party service and the returned data is then stored on a local SQL-enabled database.

Bad actors use the third-party service to store an SQLi payload associated with a business created by them. Then they go after the vulnerable API providing specific input that makes it pull their "malicious business" from the third-party service. The SQLi payload ends up being executed by the database, exfiltrating data to an attacker's controlled server.

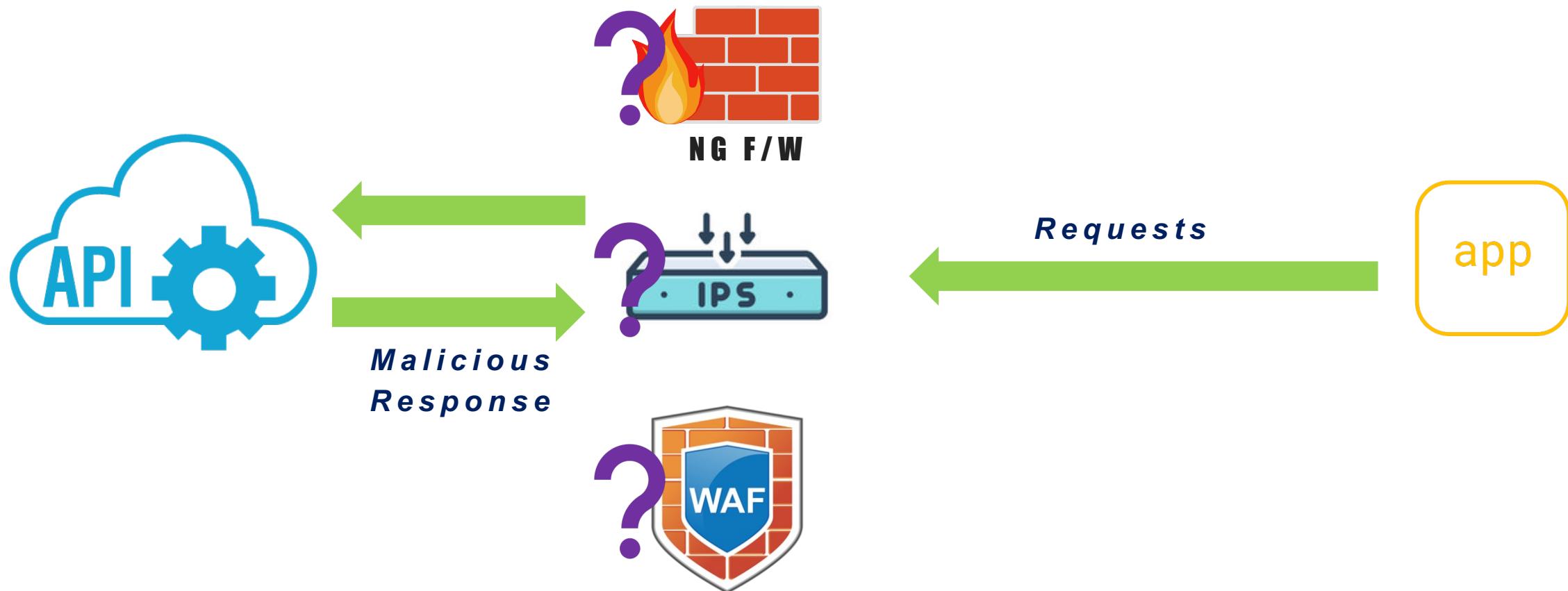
API10: Unsafe Consumption of APIs

How to Stop the ‘Unsafe Consumption of APIs’



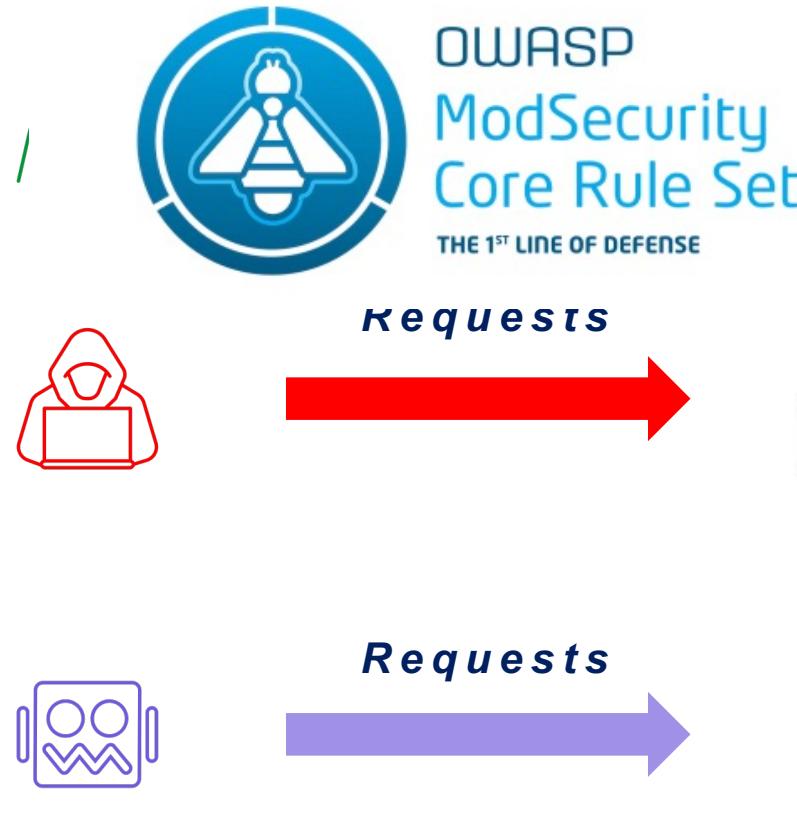
API10: Unsafe Consumption of APIs

How to Stop the ‘Unsafe Consumption of APIs’



API10: Unsafe Consumption of APIs

How to Stop the ‘Unsafe Consumption of APIs’



REQUEST-932-APPLICATION-ATTACK-RCE.conf REQUEST-933-APPLICATION-ATTACK-PHP.conf REQUEST-934-APPLICATION-ATTACK-GENERIC.conf

TODO

REQUEST-941-APPLICATION-ATTACK-XSS.conf TODO

| REQUEST-942-APPLICATION-ATTACK-SQLI | Configuration Path: [rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf](#) ||

Within this configuration file we provide rules that protect against SQL injection attacks. SQLi attackers occur when an attacker passes crafted control characters to parameters to an area of the application that is expecting only data. The application will then pass the control characters to the database. This will end up changing the meaning of the expected SQL query.

| REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION | Configuration Path:

[rules/REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf](#) || These rules focus around providing protection against Session Fixation attacks.



| RESPONSE-954-DATA-LEAKAGES-IIS | Configuration Path: [rules/RESPONSE-954-DATA-LEAKAGES-IIS.conf](#) || These rules provide protection against data leakages that may occur because of Microsoft IIS



| RESPONSE-952-DATA-LEAKAGES-JAVA | Configuration Path: [rules/RESPONSE-952-DATA-LEAKAGES-JAVA.conf](#) || These rules provide protection against data leakages that may occur because of Java

| RESPONSE-953-DATA-LEAKAGES-PHP | Configuration Path: [rules/RESPONSE-953-DATA-LEAKAGES-PHP.conf](#) || These rules provide protection against data leakages that may occur because of PHP

| RESPONSE-950-DATA-LEAKAGES | Configuration Path: [rules/RESPONSE-950-DATA-LEAKAGES.conf](#) || These rules provide protection against data leakages that may occur generically

API10: Unsafe Consumption of APIs

How to Stop the ‘Unsafe Consumption of APIs’

Secure Proxy for External API calls

- With proxy configuration with the custom script, reverse proxy can validate and sanitize the response data from external APIs.

OpenAPI Spec Validation

- With OpenAPI validation process, you can validate the response from 3rd party API vendors.

API10: Unsafe Consumption of APIs

Protection#01 – Secure Proxy for External API calls

```
ubuntu@ubuntu:~$ curl -k -H 'Content-Type: application/json' https://14c1fa7d-c460-4da3-ba49-a923dbb3a63d.access.udf.f5.com/normal.json
{
  "email": "james@test.com",
  "age": "32",
  "company": "F5"
}
ubuntu@ubuntu:~$
```

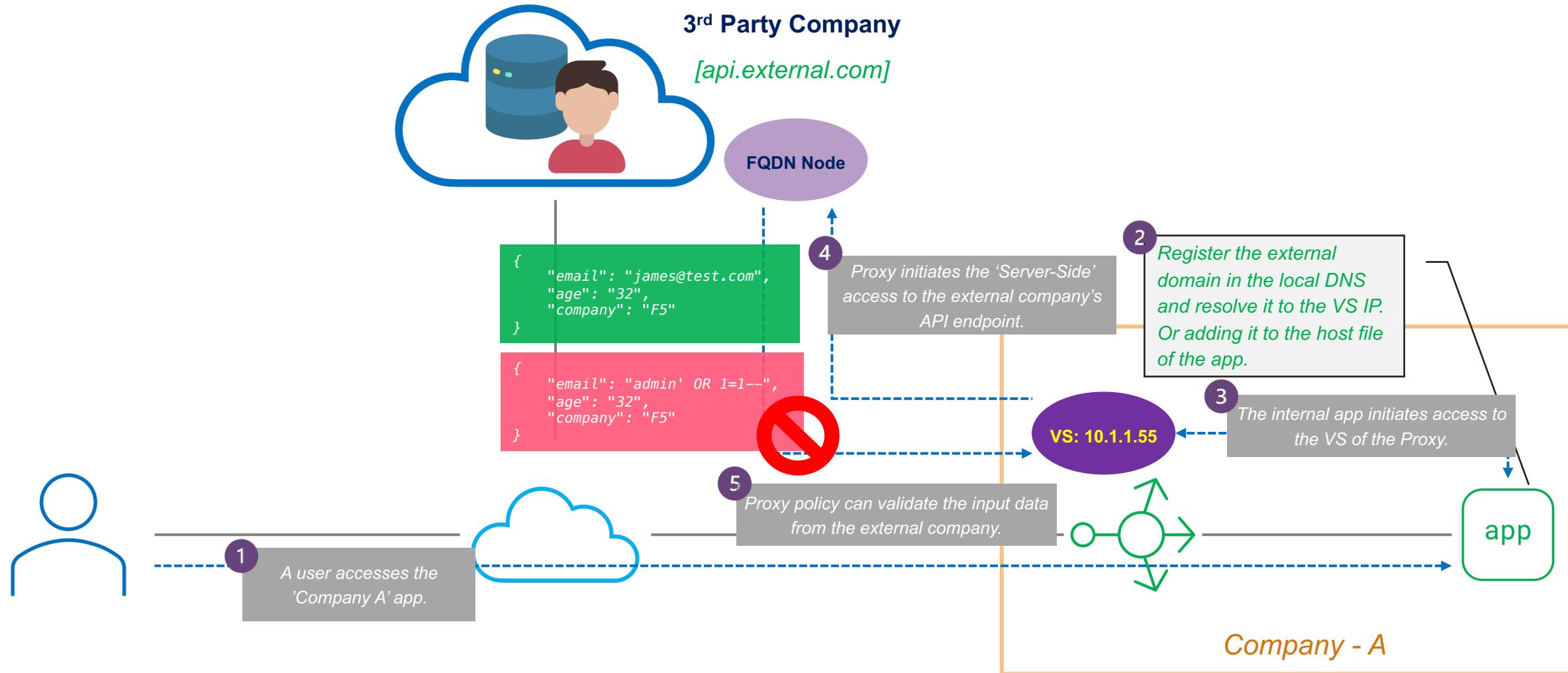
- *The internal app can fetch the user's data from the external company via API.*

```
ubuntu@ubuntu:~$ curl -k -H 'Content-Type: application/json' https://14c1fa7d-c460-4da3-ba49-a923dbb3a63d.access.udf.f5.com/malformed.json
{
  "email": "admin' OR 1=1--",
  "age": "32",
  "company": "F5"
}
ubuntu@ubuntu:~$
```

- *If the external company's app is compromised, it could send malicious content.*
- *In this example, the attacker put the SQLi payload in the 'email' field. If the internal app trusts the 3rd party company's data without additional validation, this could cause the SQLi attack.*
- *This example is the same example that OWASP described as 'Scenario #1'.*

API10: Unsafe Consumption of APIs

Protection#01 – Secure Proxy for External API calls



API10: Unsafe Consumption of APIs

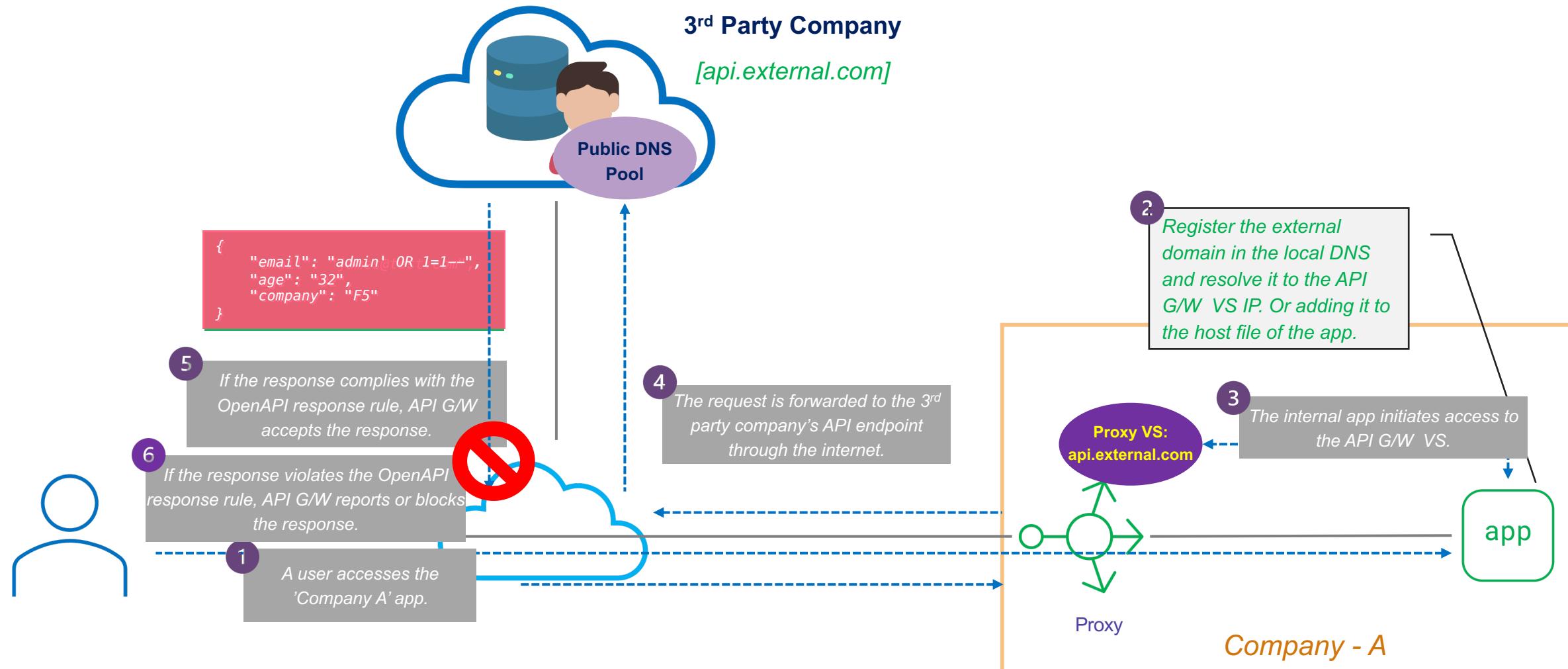
Protection#01 – Secure Proxy for External API calls

```
[ubuntu@ubuntu:~$  
[ubuntu@ubuntu:~$ curl -k -H 'Content-Type: application/json' https://14c1fa7d-c460-4da3-ba49-a923dbb3a63d.access.udf.f5.com/normal.json  
{  
    "email": "james@test.com",  
    "age": "32",  
    "company": "F5"  
}  
[ubuntu@ubuntu:~$  
[ubuntu@ubuntu:~$  
[ubuntu@ubuntu:~$ curl -k -H 'Content-Type: application/json' https://14c1fa7d-c460-4da3-ba49-a923dbb3a63d.access.udf.f5.com/malformed.json  
curl: (56) OpenSSL SSL_read: Connection reset by peer, errno 104  
[ubuntu@ubuntu:~$  
[ubuntu@ubuntu:~$
```

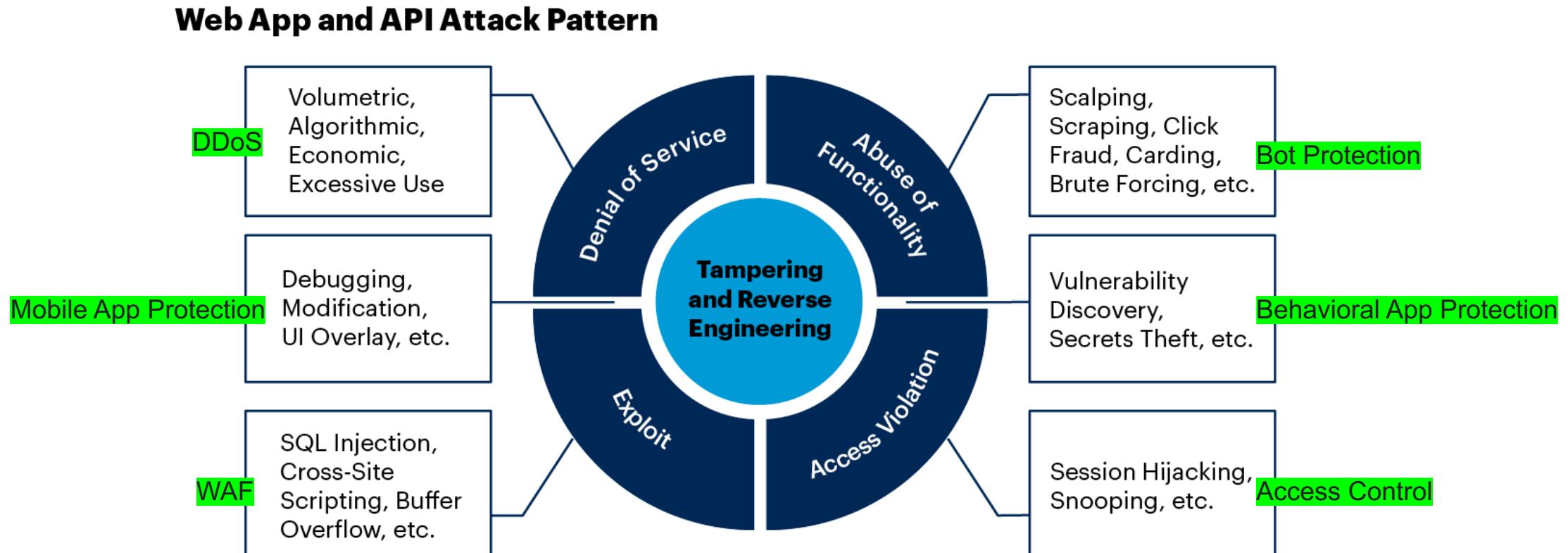
- If the received data is aligned with the defined format in the Proxy policy, the data is consumed as normal.
- However, if the data format violates the defined format in the Proxy policy, Proxy reset the session.

API10: Unsafe Consumption of APIs

Protection#02 – OpenAPI Validation



Web App and API Attack Pattern



Source: Gartner

747213_C

Gartner



