



Secure Apps and APIs Everywhere

Andre Iswanto

Director, Solution Engineering

August 2023

The Importance of API Security

Landscape of API Traffic

Over 90% of developers use APIs

91% of Organizations Had An API Security Incident in 2020

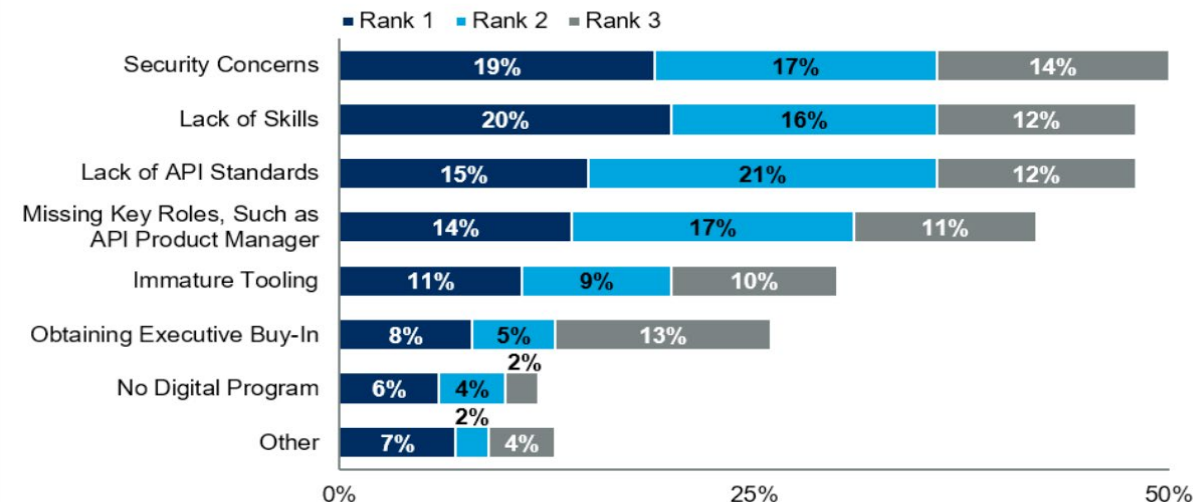
83% of All Internet Traffic Belongs to API-Based Services

93.4% of API Developers Are Still Using REST

GraphQL Is Used By 22.5% of API Developers

GraphQL is still sometimes susceptible to broken object-level authorization, which is the number one API vulnerability

Top Three Challenges of an Organization's API Strategy
Percentage of Respondents



n = 127

Source: Gartner (March 2018)

Base: Gartner Research Circle Members; excludes "Don't know"

Q: What are the top three challenges your organization needs to address regarding its API strategy?

ID: 404900



257%

increase in web applications and API attacks



81%

increase in bot activity



Insurance Company

- Data breach happened in Jan 2022
- 1.8 million accounts exposed
- BFLA(Broken Function Level Authorization) exploit

Digital Scheduling Platform

- Data breach happened in Jan 2022
- 3.7 million accounts exposed
- BOLA(Broken Object Level Authorization) exploit

Social Media Platform

- Data breach happened in July 2022
- 5.4 million accounts exposed
- BOLA(Broken Object Level Authorization) exploit

Online Marketing Platform

- Data breach happened in Feb 2022
- 7 million accounts exposed

Telco Company

- Data breach happened in Sep 2022
- 10 million accounts exposed
- BFLA(Broken Function Level Authorization) exploit

Notable API breaches

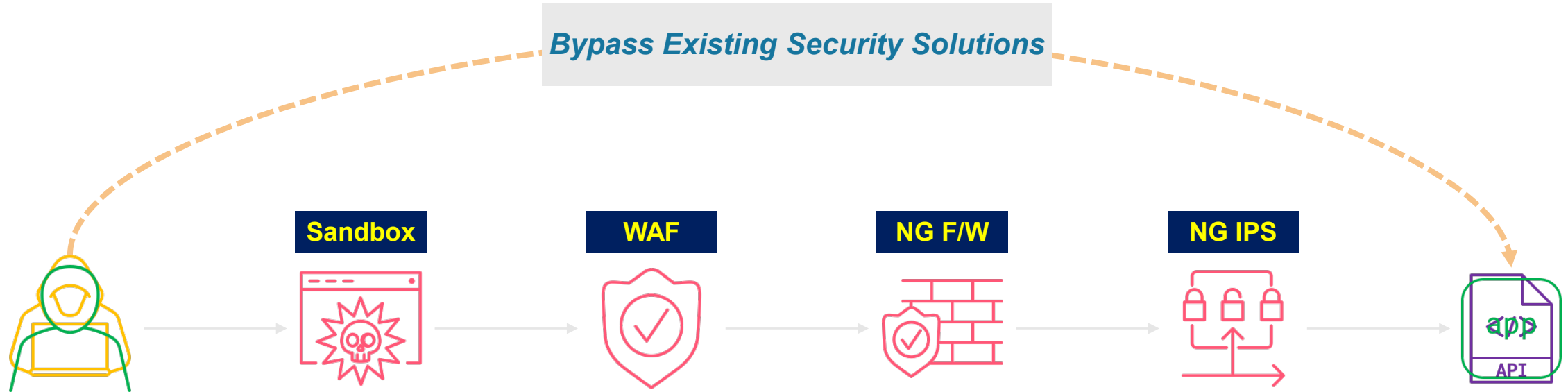
Vulnerabilities

Lack of proper access controls

Misconfigurations

Inadequate controls for API specific security

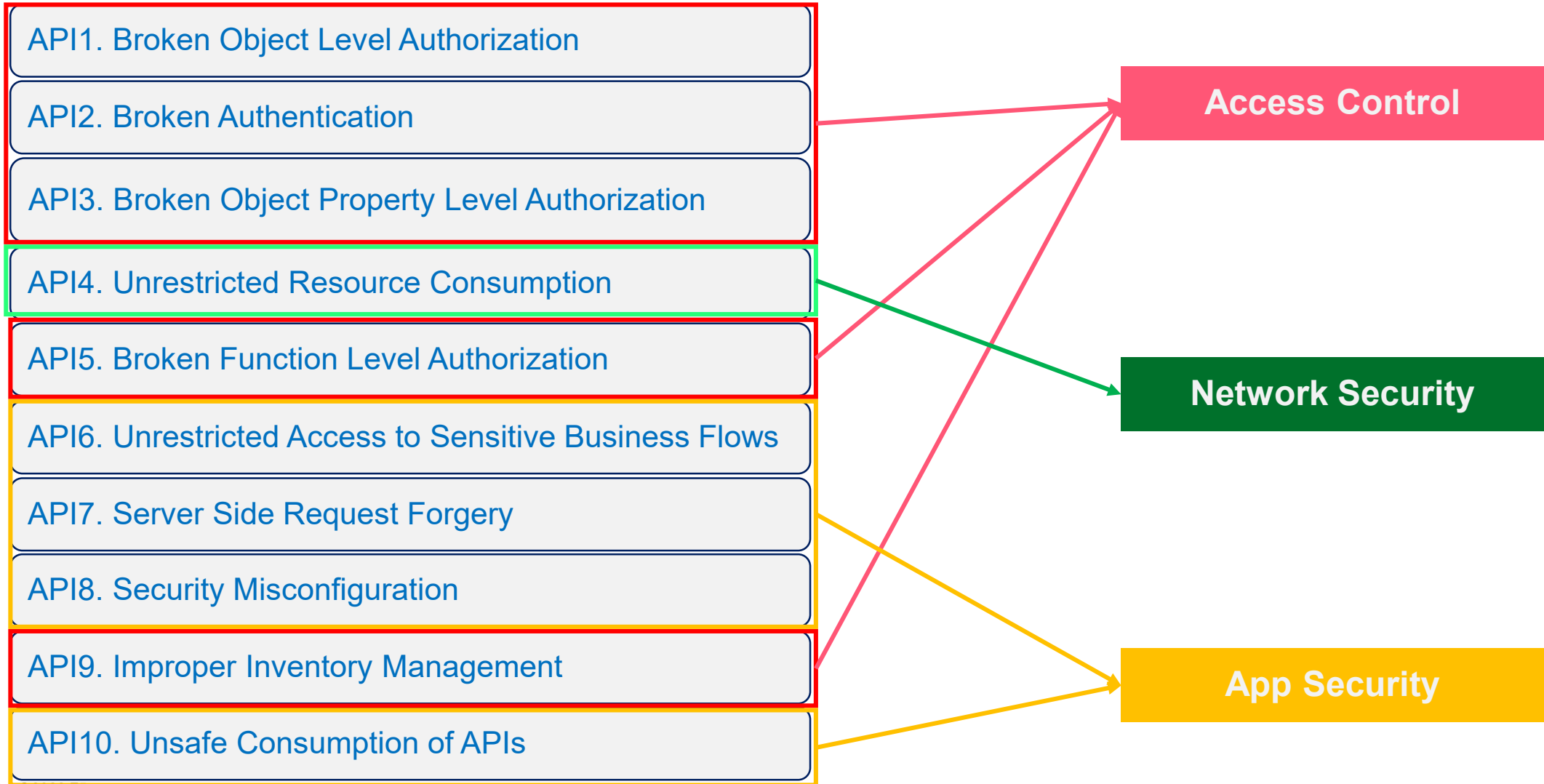
Why Are APIs Primary Target for Attackers?



Non malicious (or suspicious) bots

Using Bot to perform API Abuse (Sneaker Bots, ATO, 3rd Party Payment APIs, Content Scaping)

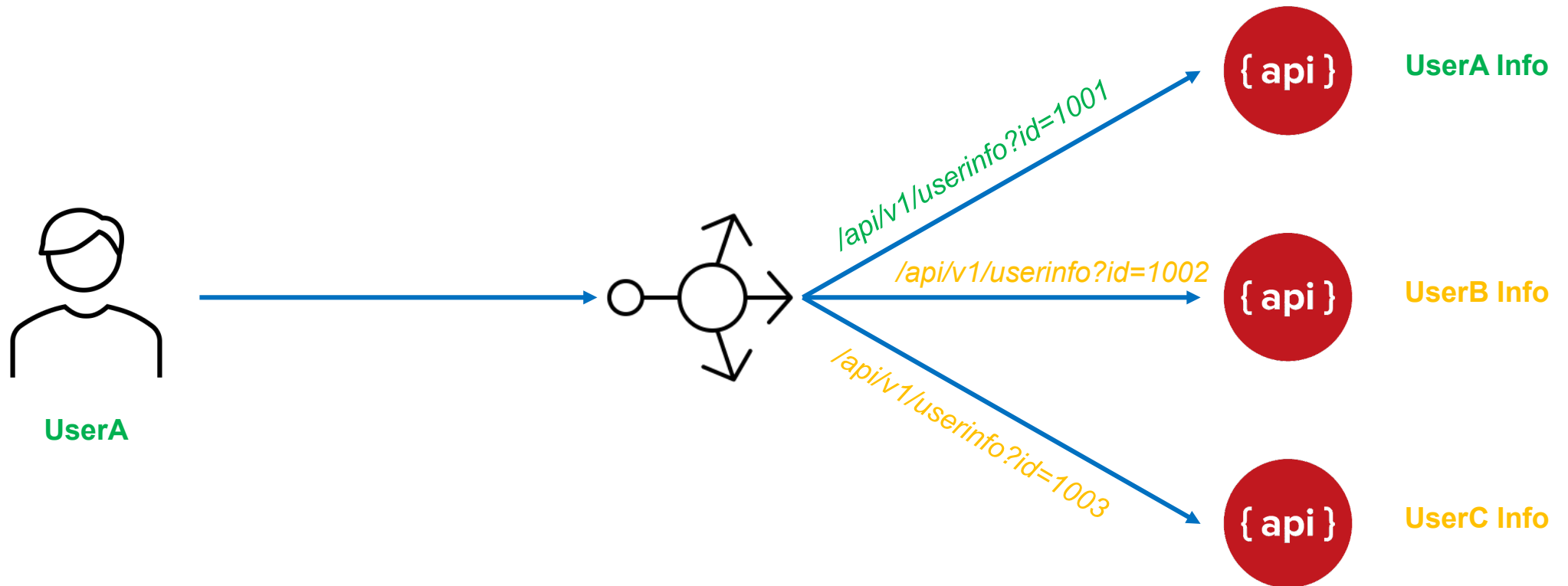
OWASP API Top 10 - 2023



What is BOLA? 1st Vulnerabilities of OWASP API

How BOLA Exploit Works?

BOLA(Broken Object Level Authorization) is a security vulnerability that is targeting API endpoints. The exploit works when the application does not correctly enforce the access control for the requests of specific objects from the users.



Object ID

What is the Object ID?

- The 'Object ID' is the key value of the object used as a unique identification in the database.

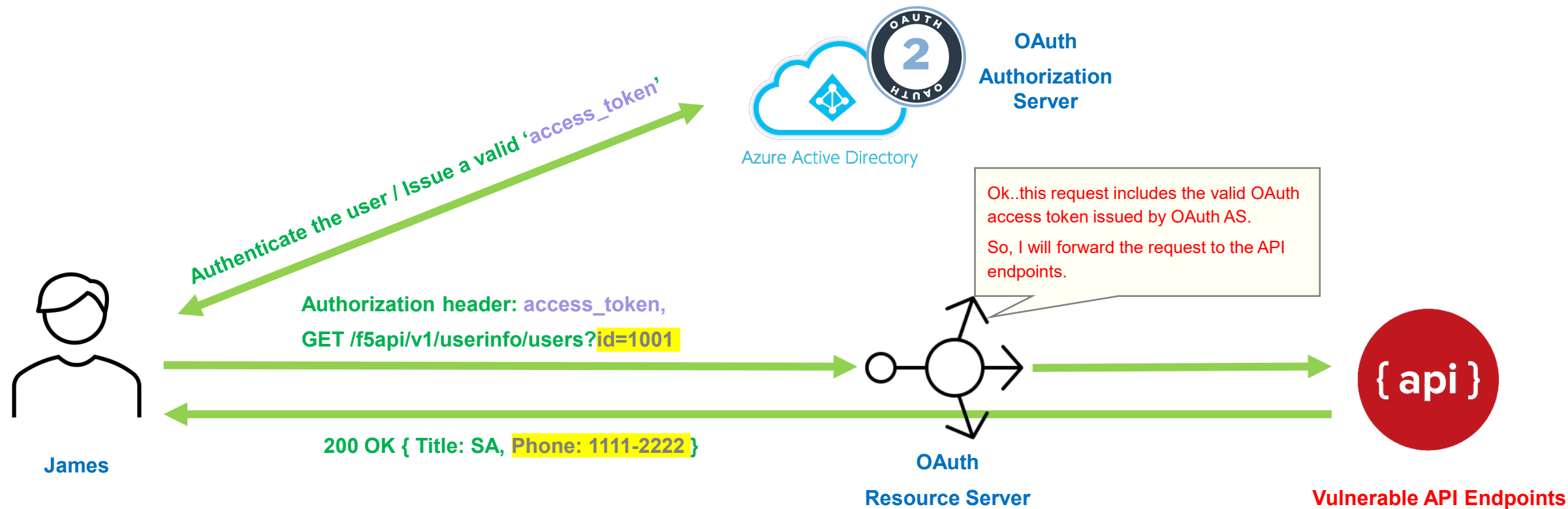
Object ID	User Name	Title	Location	Phone Number
1001	James	SA	Singapore	1111-2222
1002	Andre	SED	Singapore	2222-3333
1003	Michael	SA	Singapore	3333-4444
1004	Darren	SE	Singapore	4444-5555
1005	Shahn	SA	Singapore	5555-6666

- Normally, the 'Object ID(=key)' is used to identify the specific user or the object in the API request.

What is BOLA?

How BOLA Exploit Works?

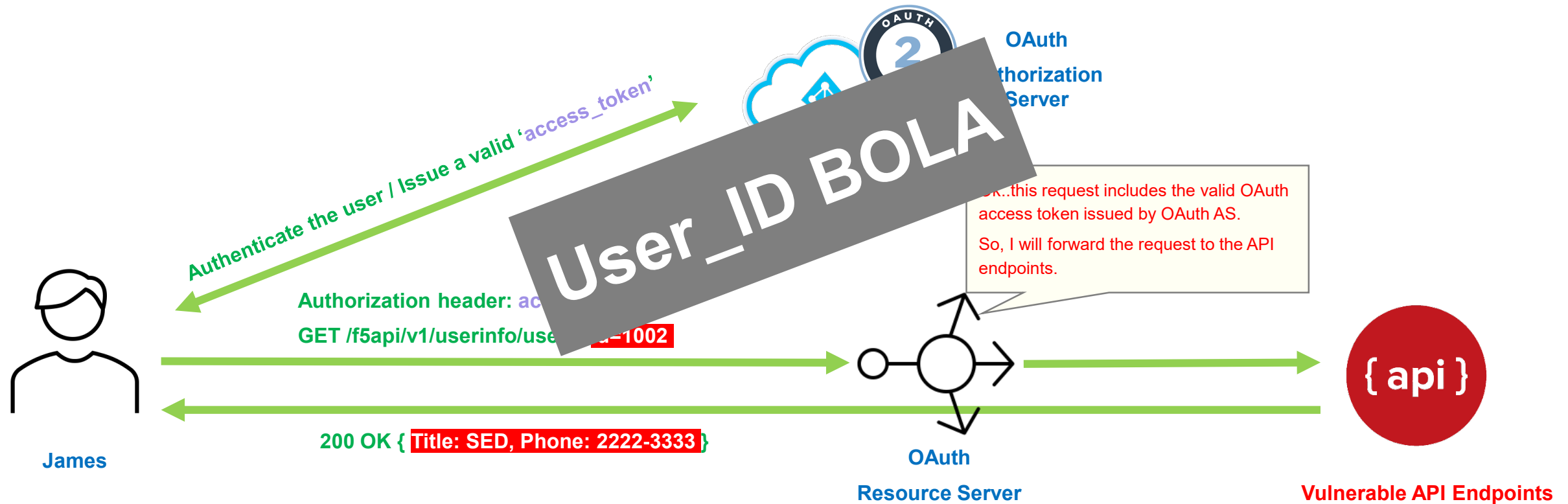
Object ID	User Name	Title	Location	Phone Number
1001	James	SA	Singapore	1111-2222
1002	Andre	SED	Singapore	2222-3333



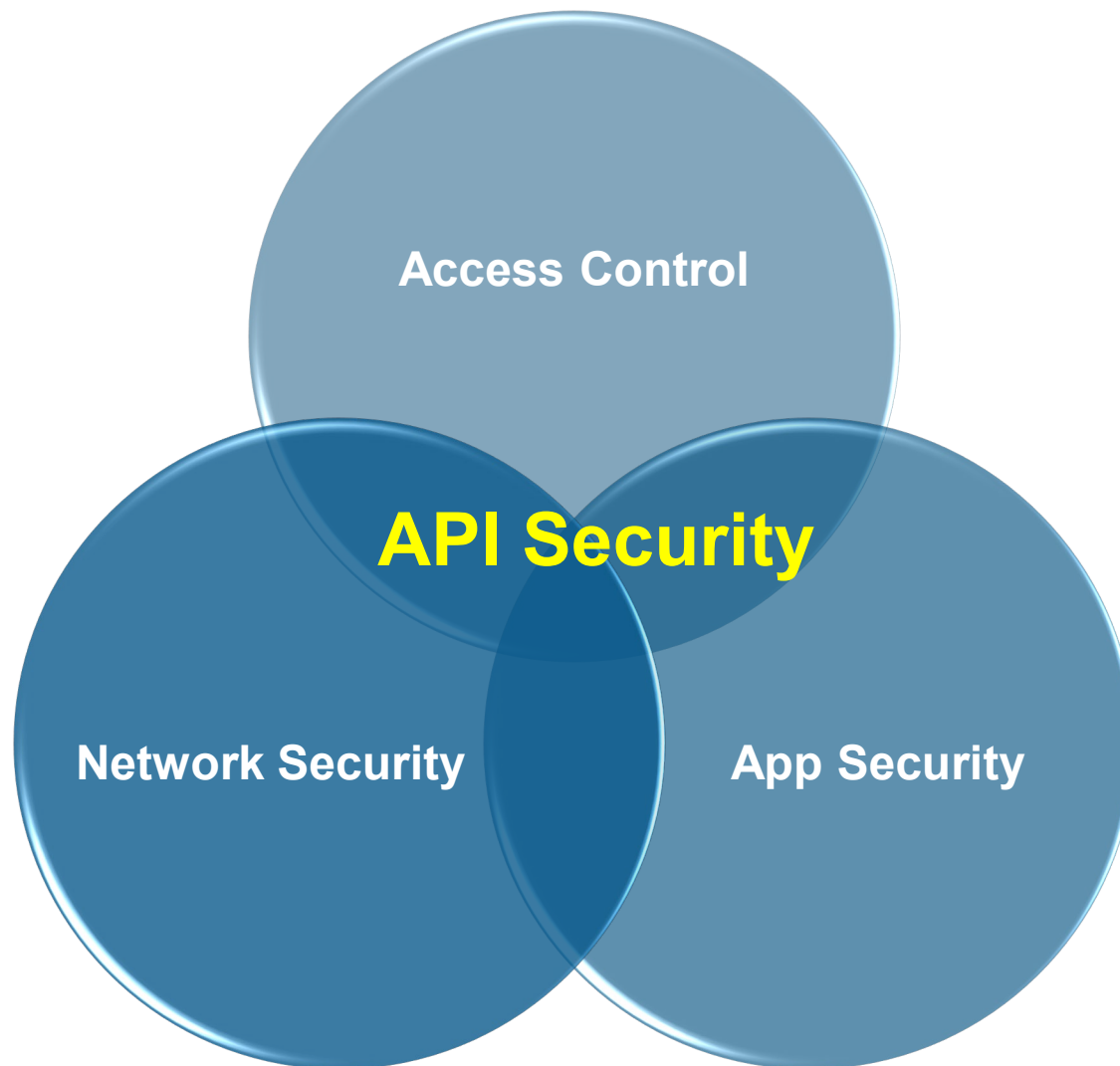
What is BOLA?

How BOLA Exploit Works?

Object ID	User Name	Title	Location	Phone Number
1001	James	SA	Singapore	1111-2222
1002	Andre	SED	Singapore	2222-3333



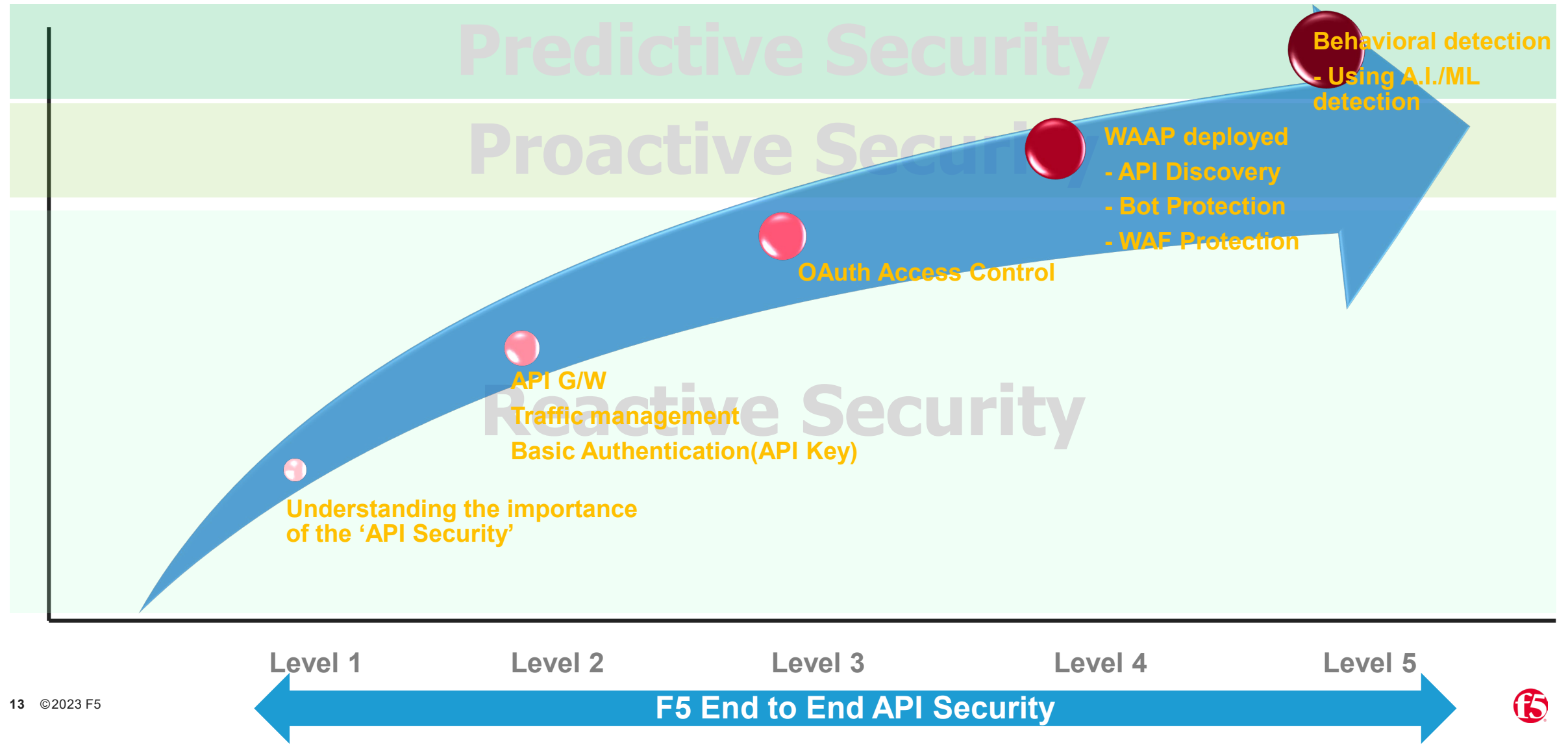
End-to-End API Security



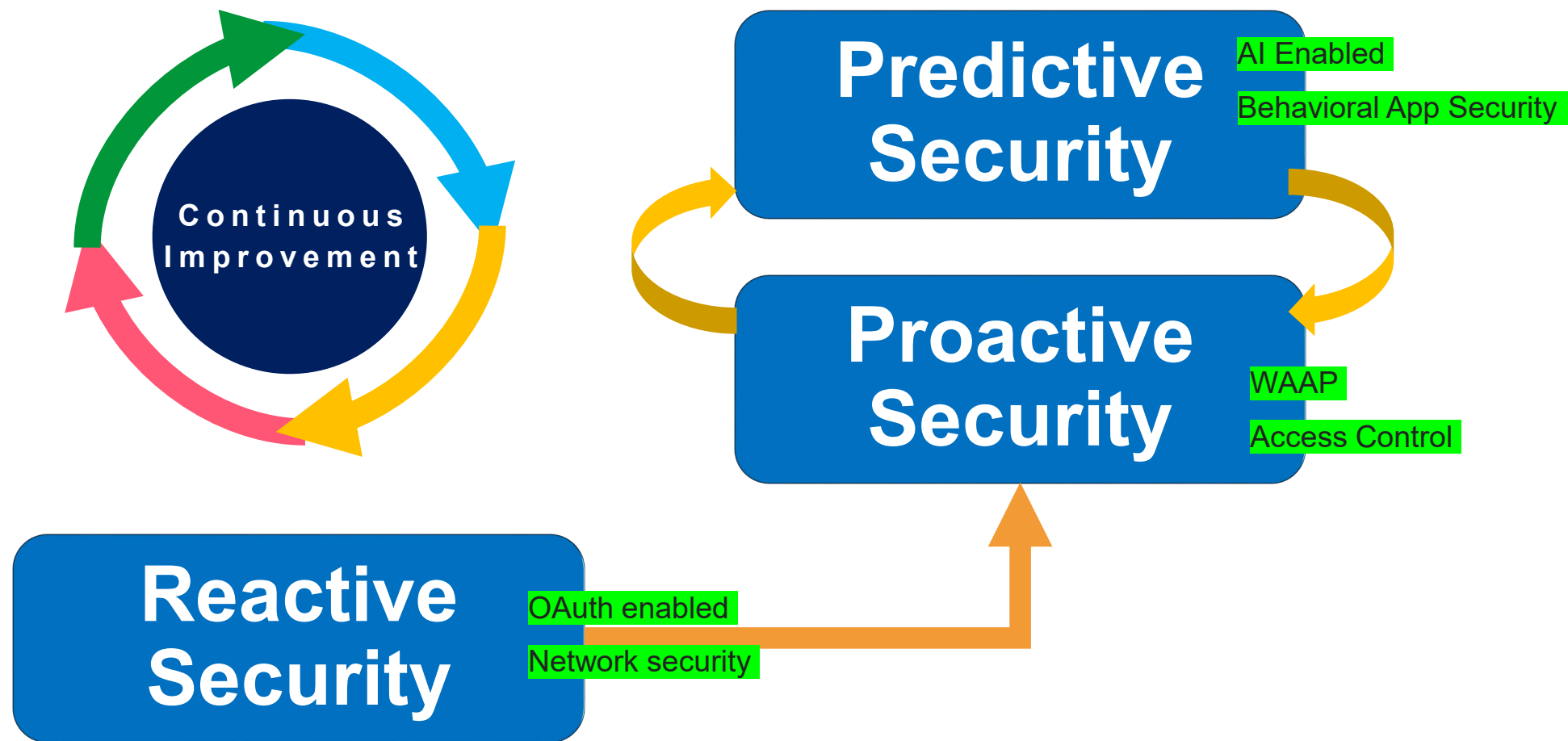
API Security Maturity Model



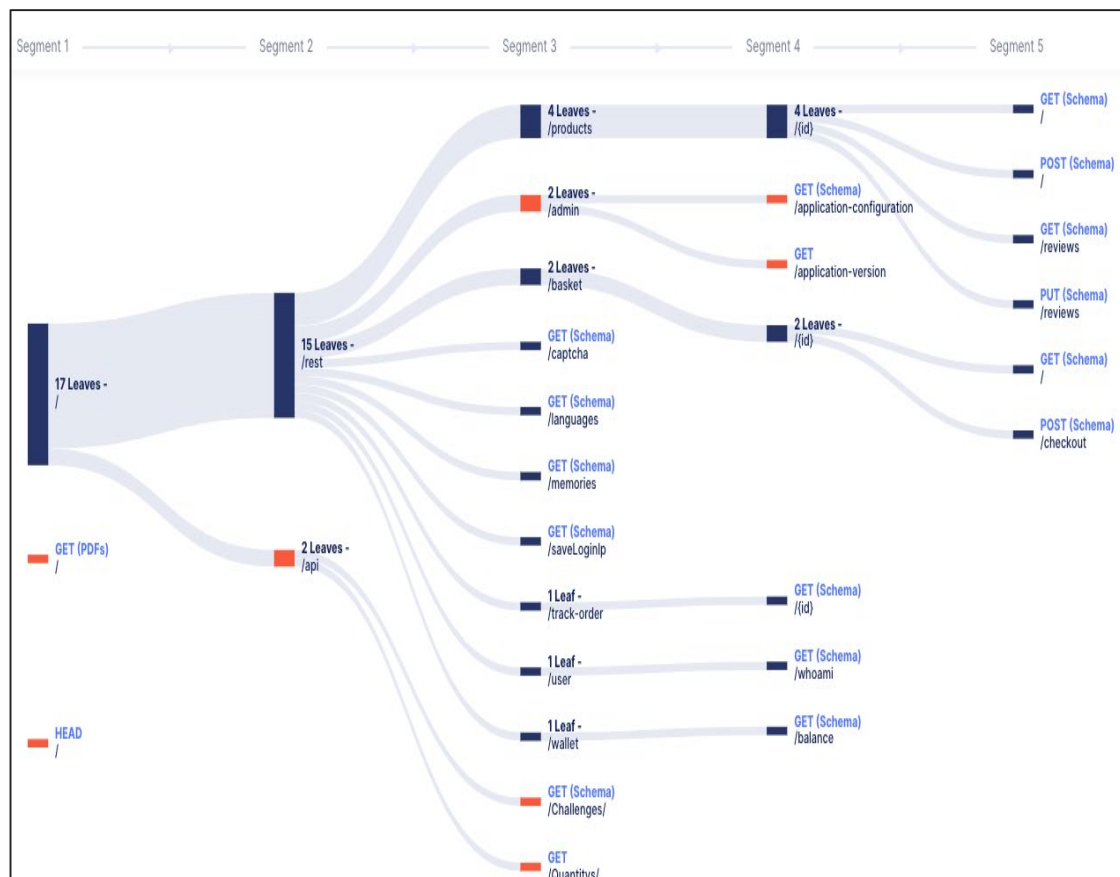
API Security Maturity Model



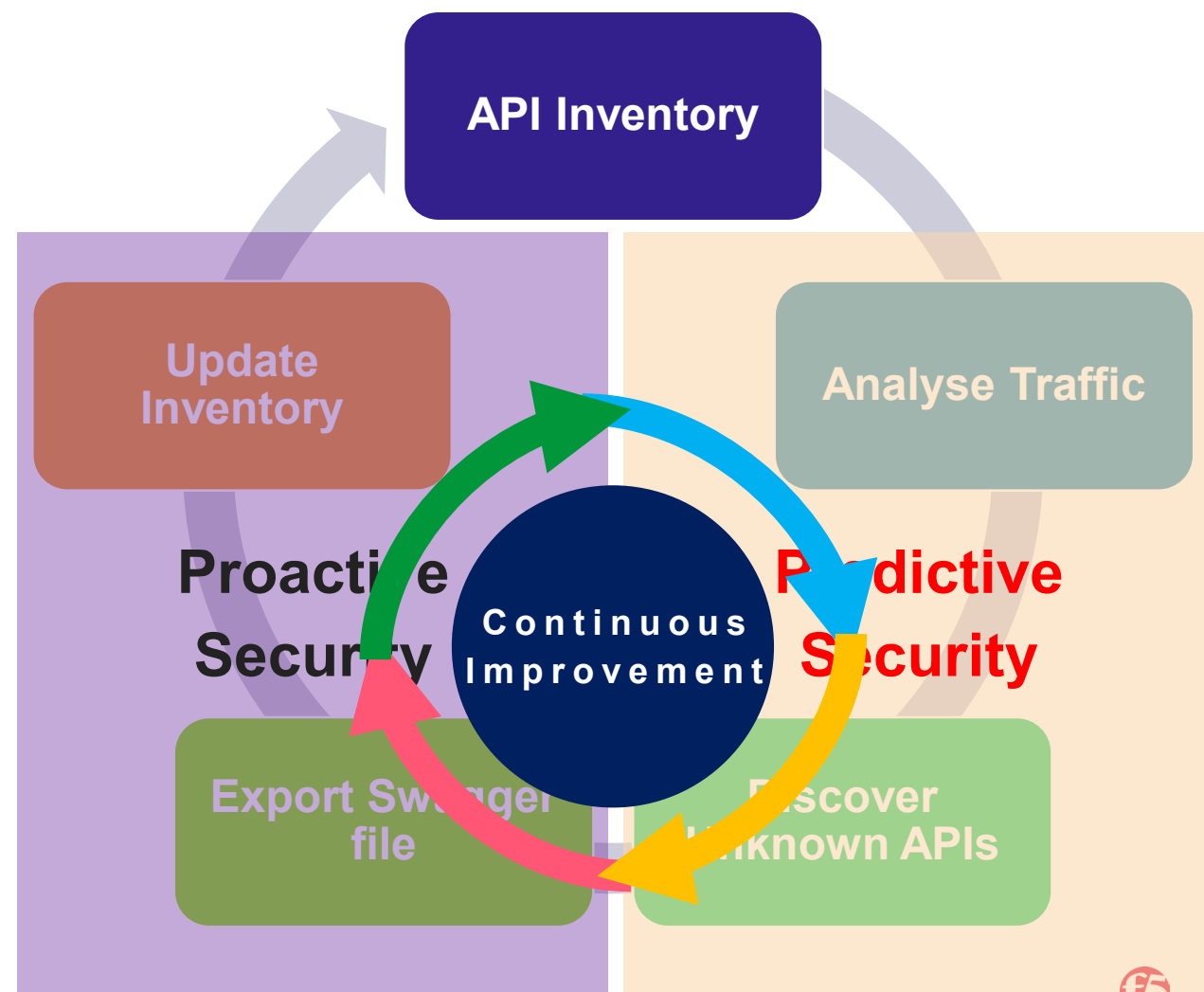
API Security Maturity Model at Runtime



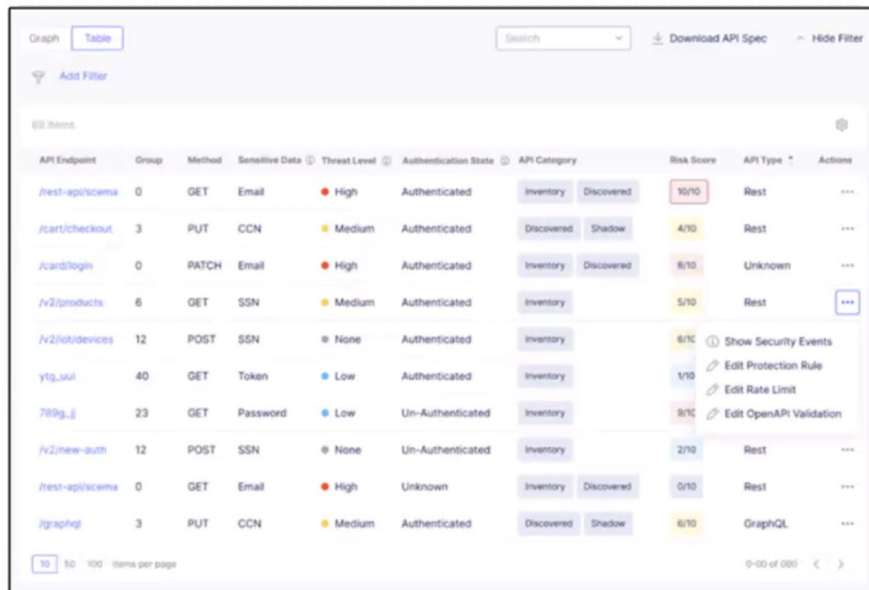
Proactive & Predictive Security: API Discovery



F5 XC WAAP Shadow API Discovery Example



API Discovery



API Endpoint	Group	Method	Sensitive Data	Threat Level	Authentication State	API Category	Risk Score	API Type	Actions
/rest-api/schema	0	GET	Email	High	Authenticated	Inventory Discovered	10/10	Rest	...
/cart/checkout	3	PUT	CCN	Medium	Authenticated	Discovered Shadow	4/10	Rest	...
/card/login	0	PATCH	Email	High	Authenticated	Inventory Discovered	8/10	Unknown	...
/v2/products	6	GET	SSN	Medium	Authenticated	Inventory	5/10	Rest	...
/v2/sid/devices	12	POST	SSN	None	Authenticated	Inventory	6/10		Show Security Events
ytg_uul	40	GET	Token	Low	Authenticated	Inventory	1/10		Edit Protection Rule
780g_j	23	GET	Password	Low	Un-Authenticated	Inventory	9/10		Edit Rate Limit
/v2/new-auth	12	POST	SSN	None	Un-Authenticated	Inventory	2/10	Rest	Edit OpenAPI Validation
/rest-api/schema	0	GET	Email	High	Unknown	Inventory Discovered	0/10	Rest	...
/graphql	3	PUT	CCN	Medium	Authenticated	Discovered Shadow	6/10	GraphQL	...

Un-Authenticated API and Sensitive Data Detection

Vulnerabilities

Weak JWT: Expired Tokens are Ac... ● Medium

Created: 7:16 PM, May 15
Last Observed: 3:14 AM, May 16

Weak JWT: Inadequate JWT Expir... ● Medium

Created: 7:16 PM, May 15
Last Observed: 3:14 AM, May 16

Weak JWT: "aud" claim is missing ... ● Low

Created: 7:16 PM, May 15
Last Observed: 3:14 AM, May 16

Weak JWT: "sub" claim is missing ... ● Low

Created: 7:16 PM, May 15
Last Observed: 3:14 AM, May 16

Weak JWT: "iss" claim is missing (... ● Medium

Created: 7:16 PM, May 15
Last Observed: 3:14 AM, May 16

State
Open

Category
Weak Authentication / Authorization

Description
This vulnerability is reported if API server does not check JWT expiration. Expired JWTs are accepted as valid. A malicious actor that has obtained an expired access token can use it to bypass authorization. This weakness does not expose API to an immediate attack as attackers need to obtain the token first. Therefore, the impact is Medium.

Risk Score
Attack impact: 40 (Medium)

Evidence
[Review Evidence Detection](#)

Remediation
API server should implement JWT expiration check.

BOT Defence

Threat Landscape, Bad Actor Evolution & Shape Rol

KEY DIFFERENTIATORS

Scaled Manual Attacks

- Human "Click Farms"
- Suspicious human behaviors
- VMs /

Emulated and Rooted Mobile Devices

Professional Automation

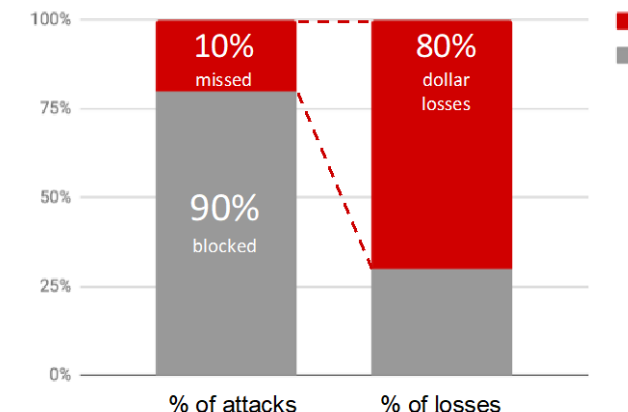
- Highly motivated and skilled
- Persistent
- Customized bot tools

- Selenium Web Driver / Puppeteer /
Browser Automation Studio

Amateur Automation

- Less motivated and skilled
- Off-the-shelf tools

- Sentry MBA, cURL / Python / Powershell



The "last 10%" includes the hardest attacks to DETECT and PREVENT, and contains the majority of dollar losses.

1 2 3 Shape

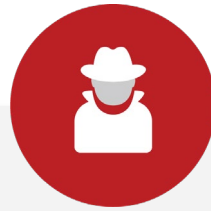
1 Other Solutions

Instead of *rigid rules*, we need *durable descriptions* of how fraud flows across the systems

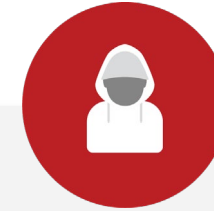
Across network security, application security, fraud and identity, B2C enterprises today needs to answer 3 fundamental questions:



**Are you
human?**



**Are you
good or
bad?**



**Are you
who you
say you
are?**

Shape Performs Transactional Analysis & Deterministic Detection

Risk scoring provides room for uncertainty

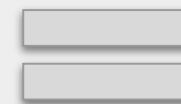
Browser
Signals



Behavior
Signals

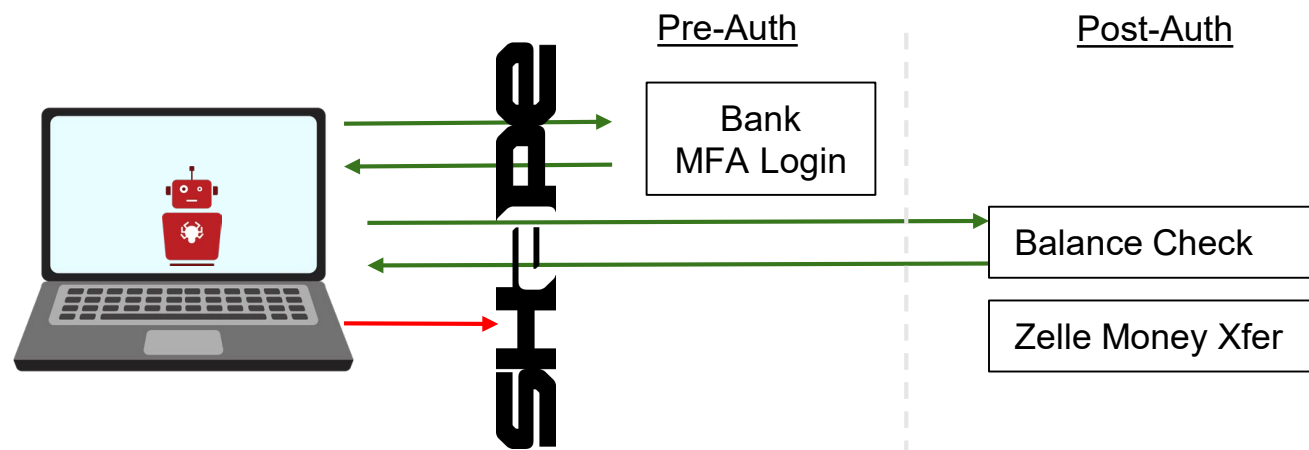
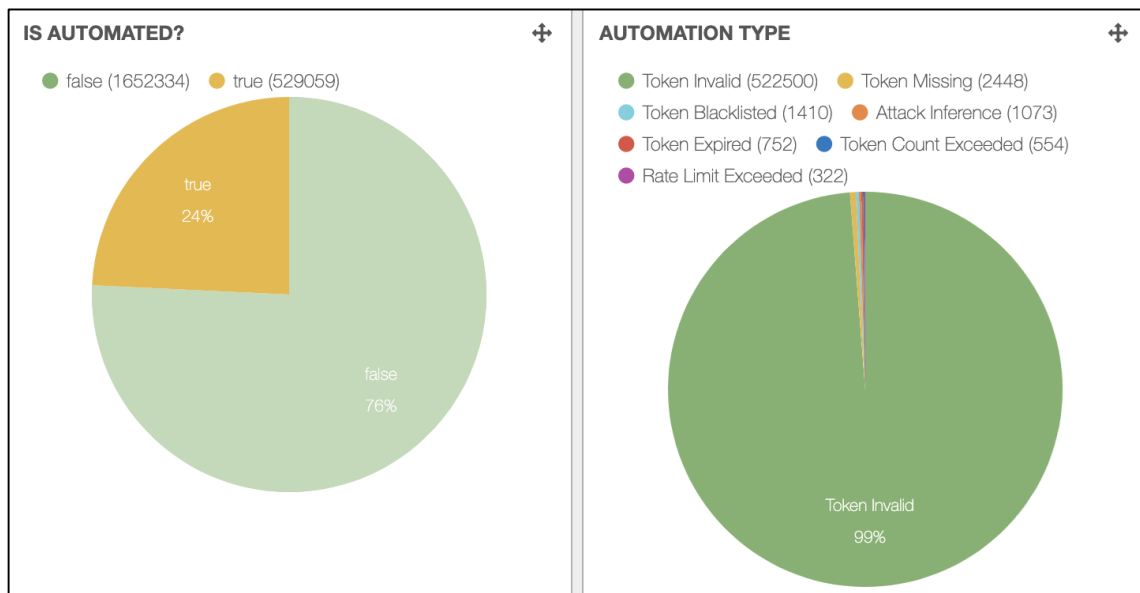


Network
Signals



Automated
True / False

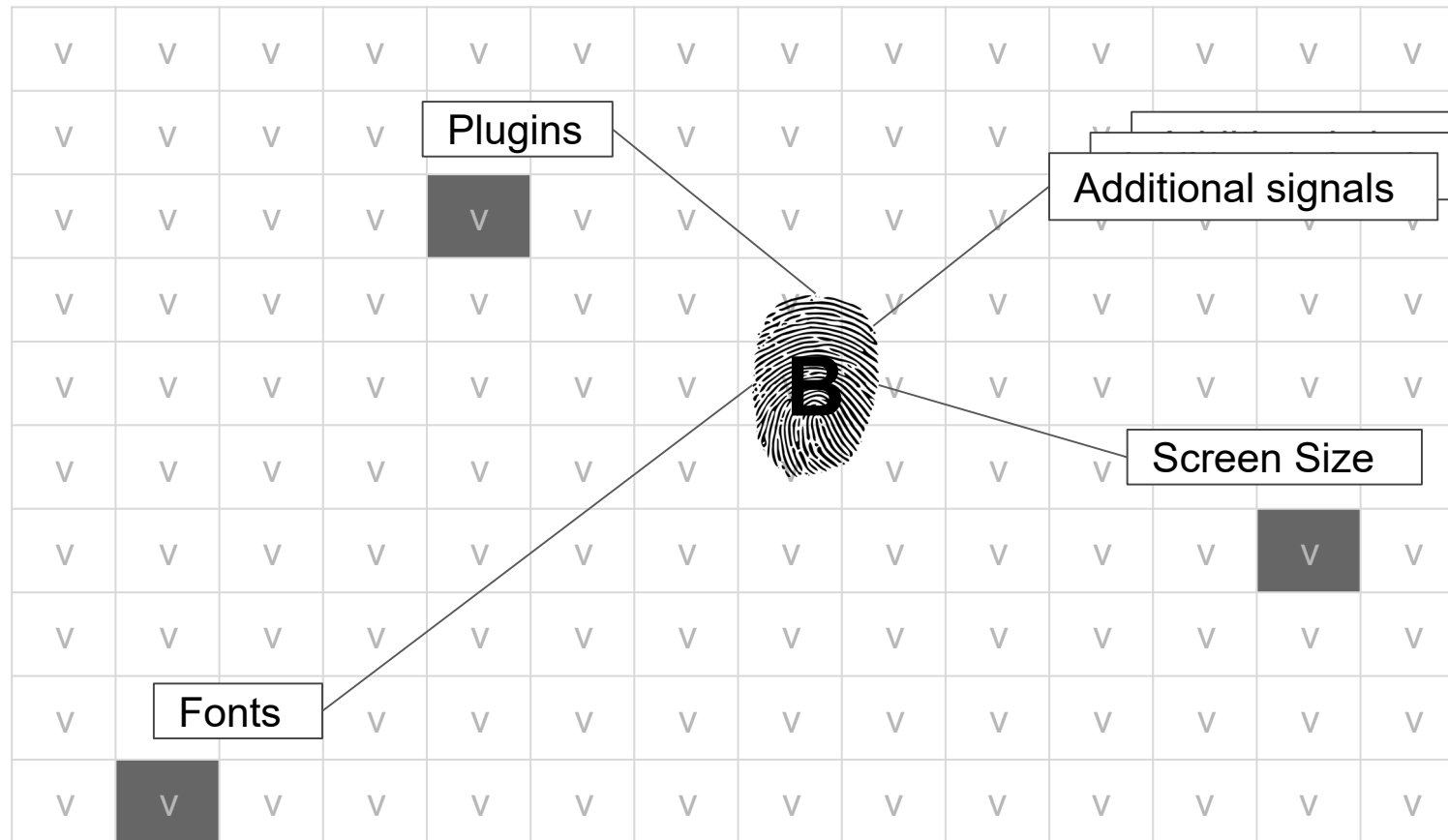
Shape evaluates each transaction independently,
regardless of what came before.



Ex) Malware infected client

- Good human accessing their account => Allowed!
- Malware performs automated session hijack and attempts to transfer money out => Prevented by Shape!

Browser Environment Signals (Web & Mobile Web)



Bot or Not?



Unique or repeat device?

Does the reported browser type/version & OS match the collected signals?

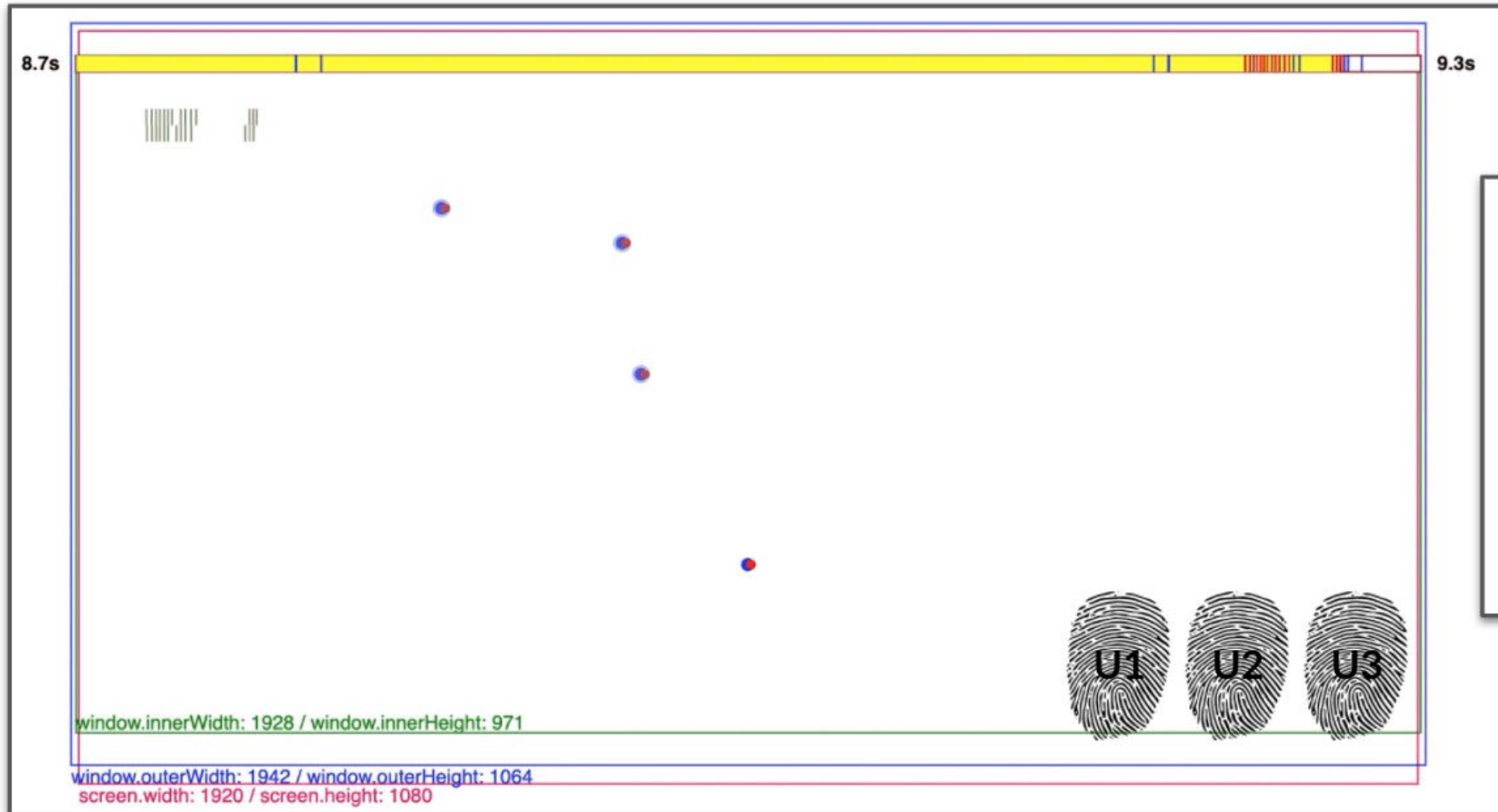
Browser Environment Signal: Emojis render differently on different platforms/apps

;-)



Visualization of automated interaction

Consistent & predictable delays consistent with amateur programmer



Sign in

Email

Password

[Show](#)

[Forgot password?](#)

By signing in to your account, you agree to our [Privacy Policy](#) and [Terms & Conditions](#).

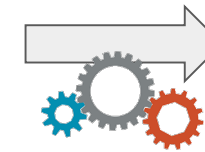
Sign in

Network Signal: Even the HTTP header contains useful signals

Fraudsters struggle to spoof everything correctly



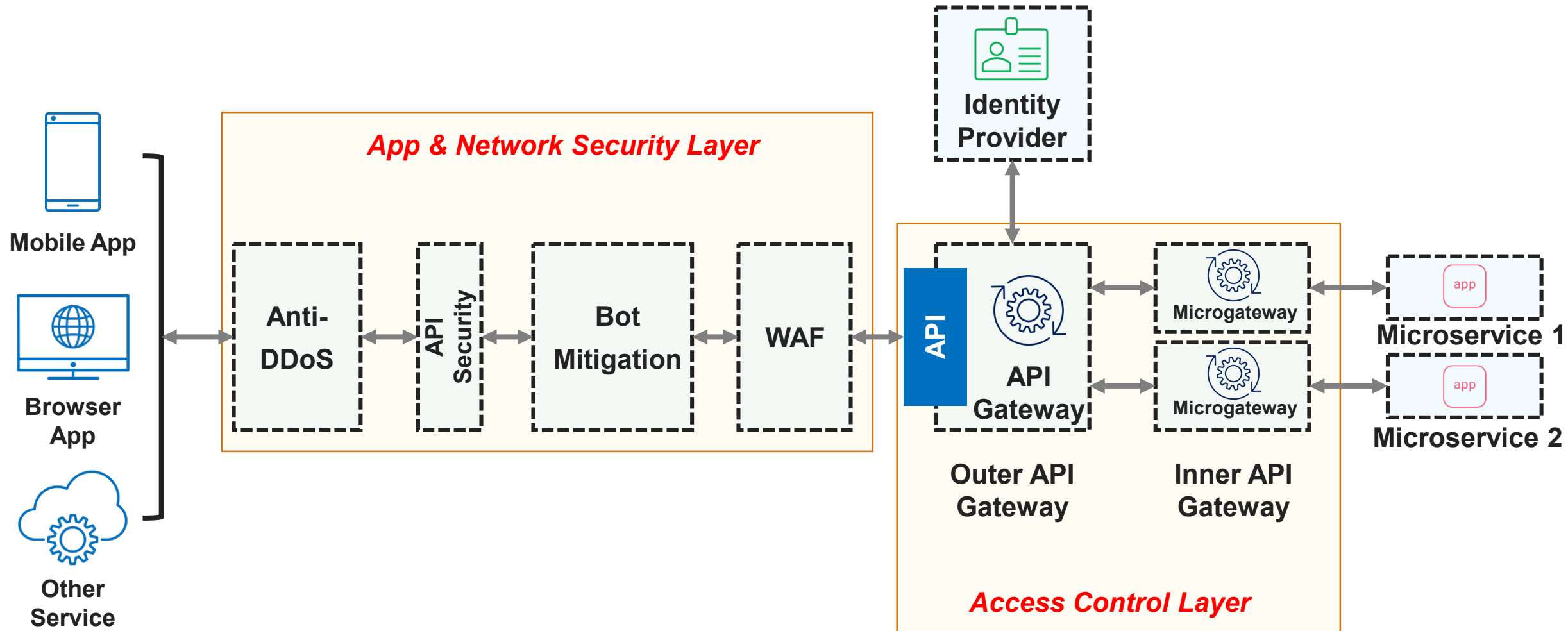
Header Field	Description	Example
Accept	Content-Types that are acceptable for the response. See Content negotiation .	Accept: text/plain
Accept-Charset	Character sets that are acceptable.	Accept-Charset: utf-8
Accept-Encoding	List of acceptable encodings. See HTTP compression .	Accept-Encoding: gzip, deflate
Accept-Language	List of acceptable human languages for response. See Content negotiation .	Accept-Language: en-US
Accept-Datetime	Acceptable version in time.	Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT
Authorization	Authentication credentials for HTTP authentication .	Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Cache-Control	Used to specify directives that <i>must</i> be obeyed by all caching mechanisms along the request-response chain.	Cache-Control: no-cache
Connection	Control options for the current connection and list of hop-by-hop request fields. ^[7] Must not be used with HTTP/2. ^[8]	Connection: keep-alive Connection: Upgrade



LIE
?

API Security Reference Architecture

Distributed Enforcement Model for Optimal Security





Gartner API Security Deployment Recommendation

API SECURITY: WHAT YOU NEED TO DO TO PROTECT YOUR TRADITIONAL AND MODERN APIS

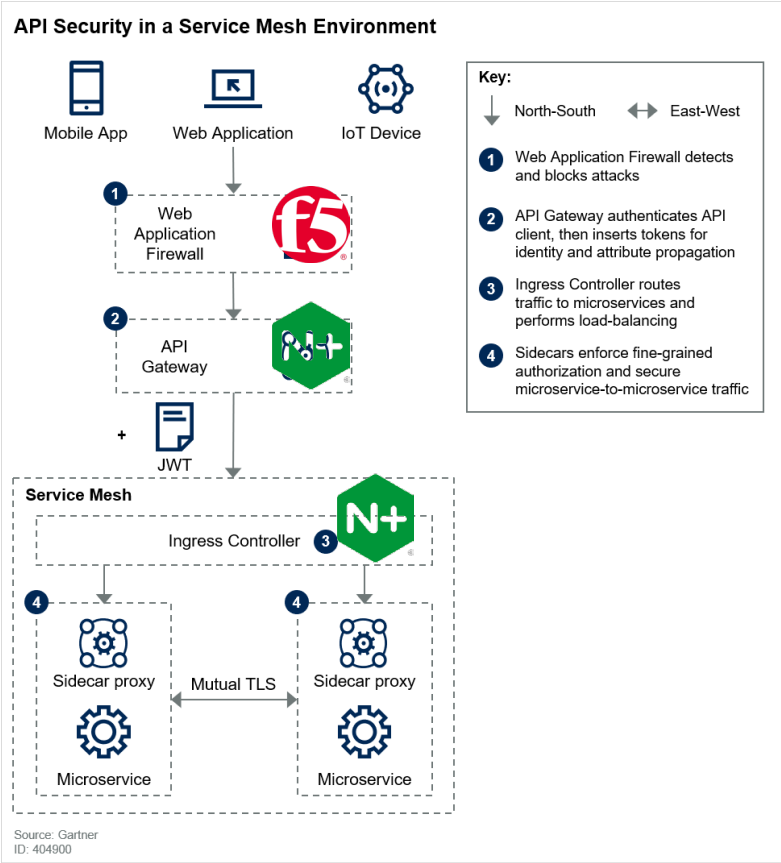
Modern App API

Figure 2. API Security Consists of API Threat Protection and API Access Control

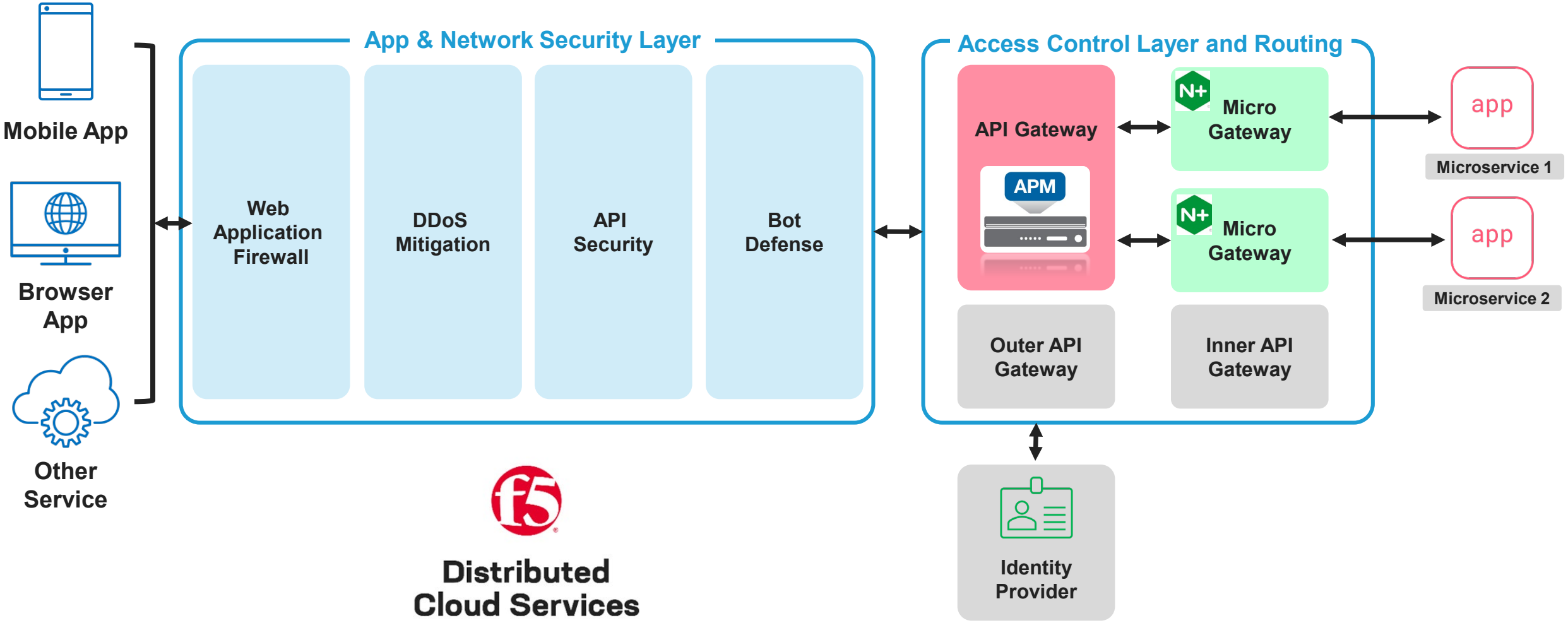
API Security Consists of API Protection and API Access Control

	 API Threat Protection	 API Access Control
Key functionality	Content validation, threat detection, traffic throttling	Authentication, authorization, identity propagation
Key technologies used	Attack signature, reputation-based control, anomaly detection, OAS message validation	OAuth 2.0, OpenID Connect, JSON Web Tokens
Product categories	Web application firewalls, API management, application delivery controllers	API management, access management software, IDaaS.

Source: Gartner
ID: 404900

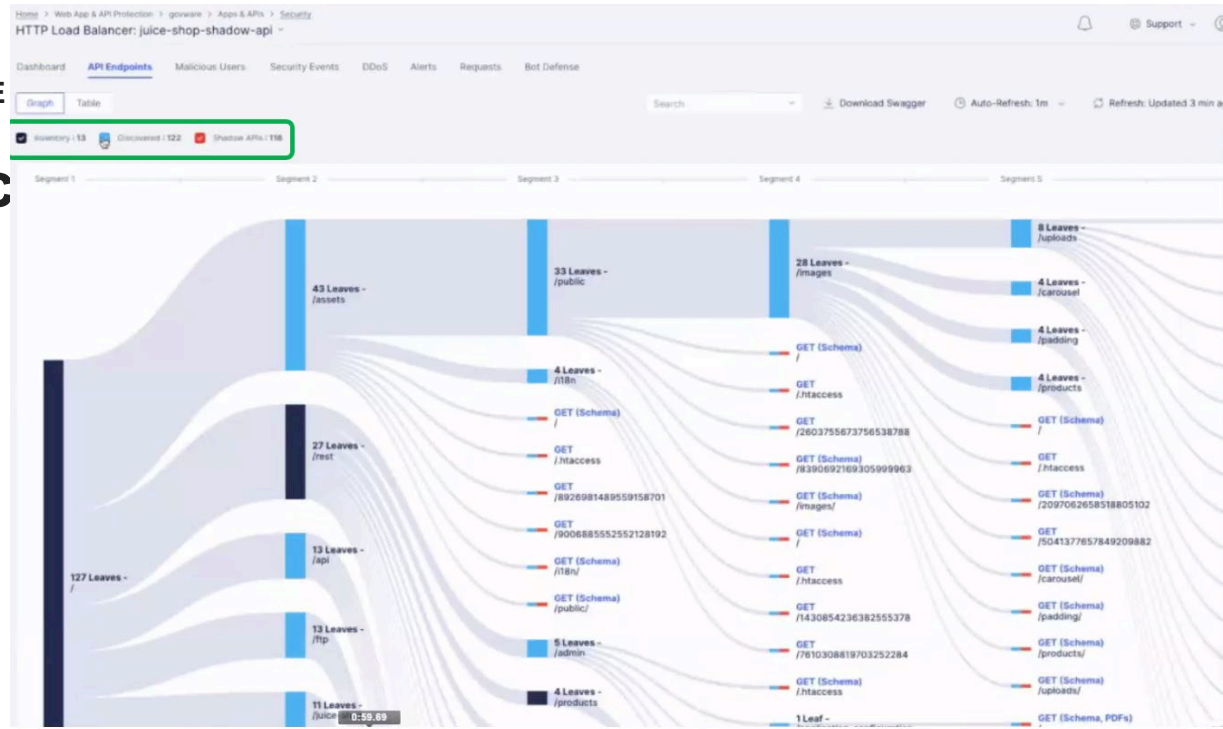


F5 API Protection Reference Architecture



F5 RECOMMENDED

Best Practice



API Gateway Architectures (G/S/A Architecture)

API Gateway Architectures can be divided into two broad aspects: API threat protection and API access control. API threat protection means detecting and blocking attacks on APIs, while API access control means controlling which applications and users can access APIs. Organizations need both. These

- 1) Real-time API Gateway
- 2) Advanced API Routing
- 3) API Security for N/S and E/W API Traffic

- JASP**
- 1) API1. Broken Object Level Authorization
 - OWASP API top 10
 - JWT Validation
 - JSON Schema Validation
 - 2) API2. Broken User Authentication
 - mTLS
 - Rate Limit and ACL
 - 3) API3. Excessive Data Exposure
 - Enable OAuth2/OIDC / Secure token management
 - 4) API4. Lack of Resource & Rate Limiting
 - Sensitive data masking / PII detection / OAS validation
 - 5) API5. Security Misconfiguration
 - Sensitive data masking / PII data detection
 - 6) API6. Mass Assignment
 - OAS validation
 - 7) API7. Injection
 - Blocking it with attack signatures / Threat campaigns
 - 8) API8. Improper Assets Management
 - Shadow API discovery / Monitoring / Blocking
 - 9) API9. Insufficient Logging & Monitoring
 - F5 XC dashboard / Log export / Alerts detail



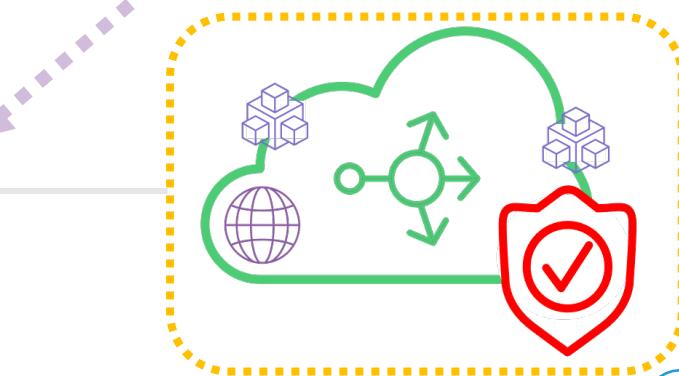
Mobile App



Browser App



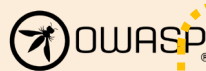
Other Service



Authorization header: access_token,
SQL Injection Attack
GET /f5api/v1/userinfo/users?id=1001
GET /api/v1.1/posts?id=12358; DROP TABLE users

BOLA Attacks:

Hmm, let me try to
change id = 1002



F5 XC WAAP

1) API3. Excessive Data Exposure

- Sensitive data masking / PII detection / OAS validation

2) API4. Lack of Resource & Rate Limiting

- API rate-limiting / Bot protection

3) API6. Mass Assignment

- OAS validation

4) API7. Security Misconfiguration

- Sensitive data masking / PII data detection

5) API8. Injection

- Blocking it with attack signatures / Threat campaigns

6) API9. Improper Assets Management

- Shadow API discovery / Monitoring / Blocking

7) API10. Insufficient Logging & Monitoring

- F5 XC dashboard / Log export / Alerts detail

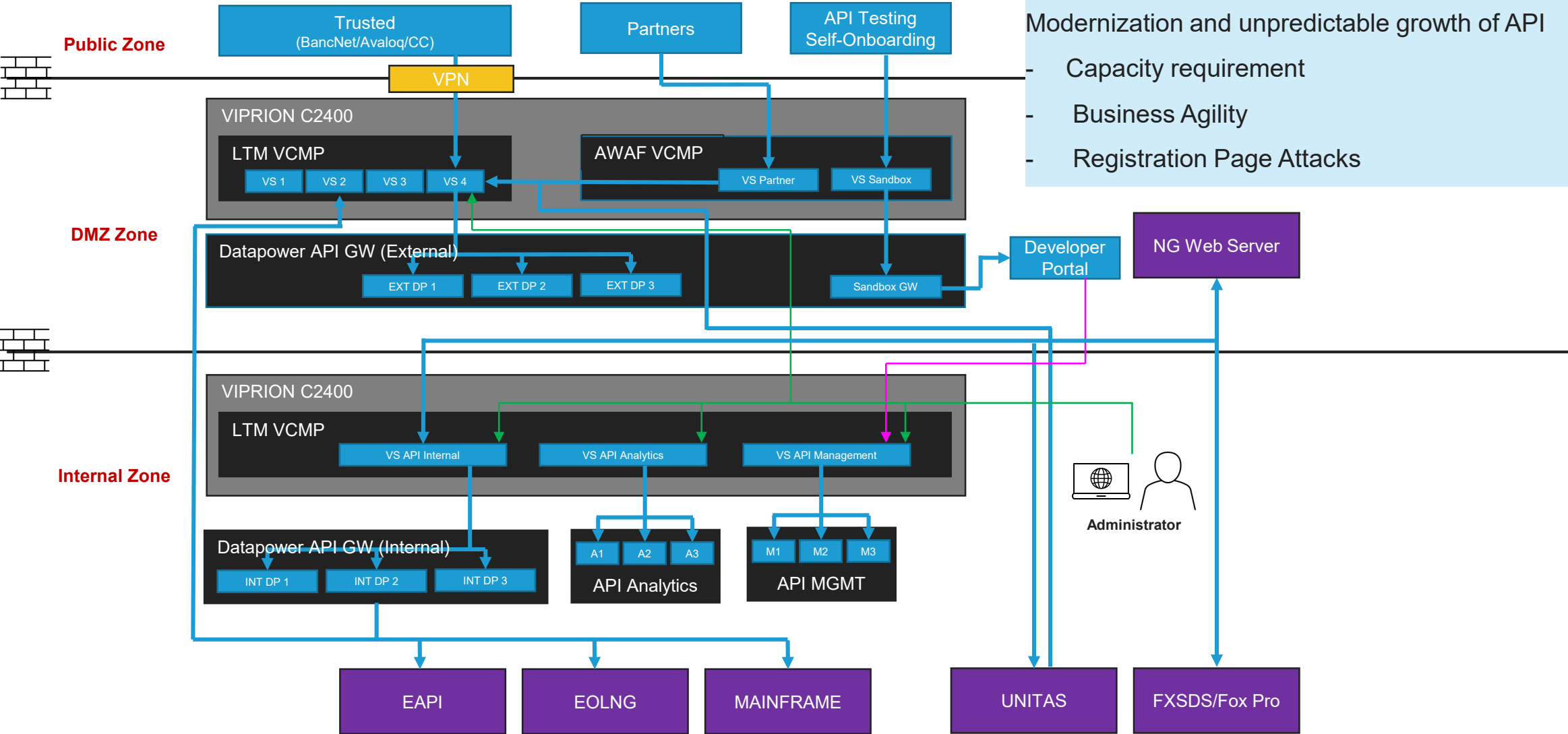


{ api }



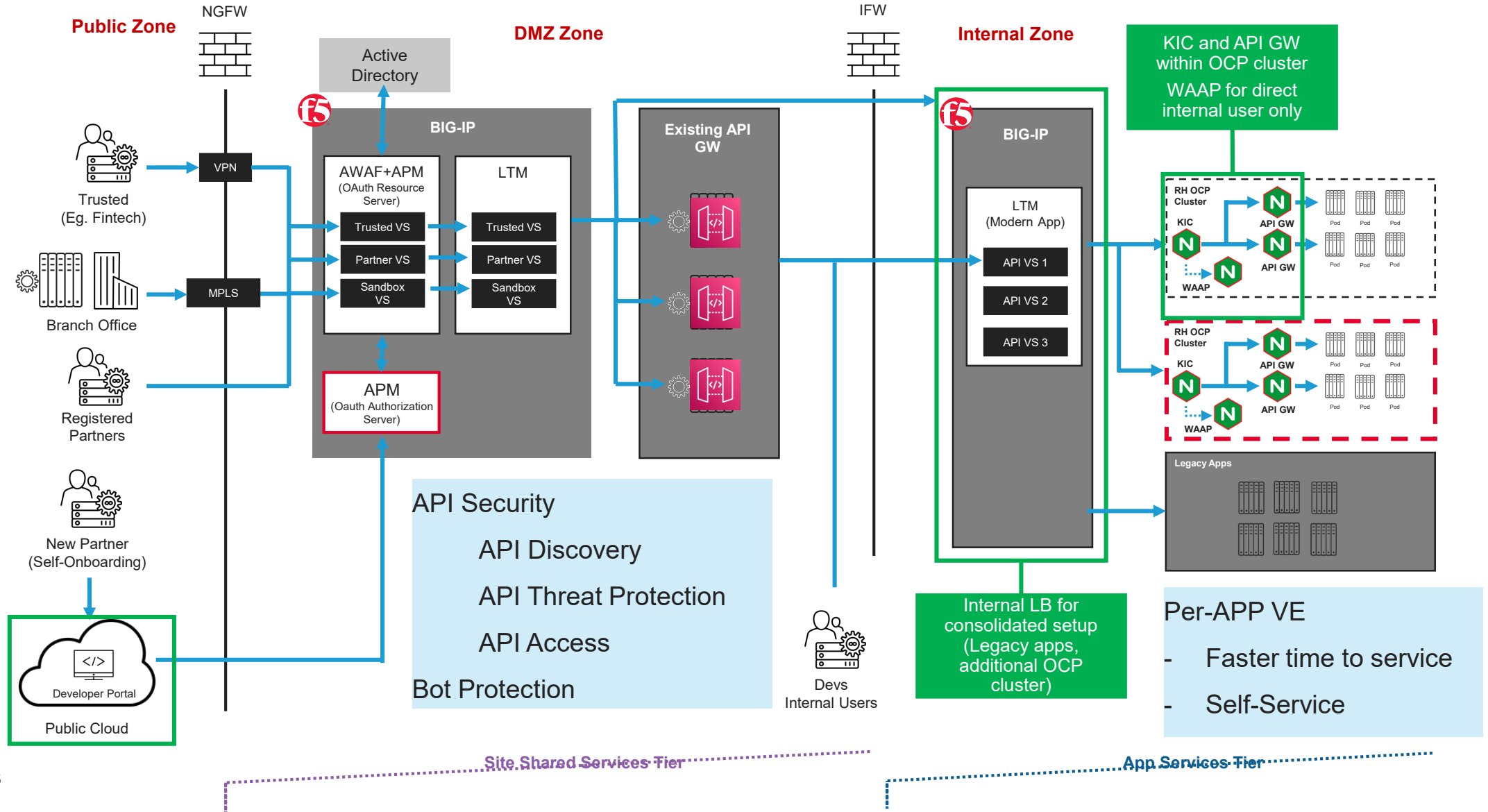
Customer Use Case

Current API Architecture for On-prem Site



Modern App + API Security Architecture

On-Premises Architecture



Modern App + API Security Architecture

Multi-Cloud Architecture

