SALT

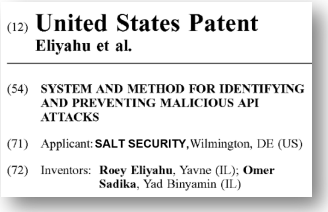
Orange Cyberdefense

# API security: context is key

Martijn Bosschaart
Solutions Engineering EMEA
martijnb@salt.security

# Salt Security defined and leads the API security category

## The only patented API attack prevention


(12) United States Patent
Eliyahu et al.

(54) SYSTEM AND METHOD FOR IDENTIFYING AND PREVENTING MALICIOUS API ATTACKS

(71) Applicant: SALT SECURITY, Wilmington, DE (US)

(72) Inventors: Roey Eliyahu, Yavne (IL); Omer Sadika, Yad Binyamin (IL)

The Home Depot · AB InBev · Goldman Sachs · MassMutual · GAP · Takeda · AON · Nationwide · MARKEL · Amway · PETSMART · ally

CapitalG · SEQUOIA · CROWDSTRIKE

Forbes 2021 NEXT BILLION-DOLLAR STARTUPS · Gartner COOL VENDOR 2020 · 2020 SINET 16 INNOVATOR · WINNER 2021 CISO CHOICE AWARDS Application Security · SCawards 2021 FINALIST Best Emerging Technology

SALT

*APIs are*

~~Software is~~
eating the world.

# Bad actors are innovating with APIs too

BREWDOG

ITPro.

**BrewDog API token flaw exposed data on 200,000 shareholders and customers, researchers claim**

by: Zach Marzouk 8 Oct 2021

proximus

telecompaper:::

Proximus reports possible leak of customer data through digital hack of API systems

NEWS | BROADBAND | BELGIUM | 16 MAR

Ledger

COINTELEGRAPH
The future of money

**Ledger data leak: A 'simple mistake' exposed 270K crypto wallet buyers**

DEC 24, 2020

SIXT SE

NATIONAL

**Sixt SE: Sixt contains cyber-attack | #cybersecurity | #cyberattack**

May 2, 2022

---

Gartner

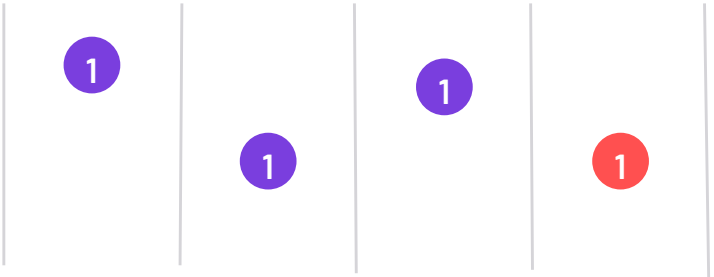
6 December 2021
A Look Back

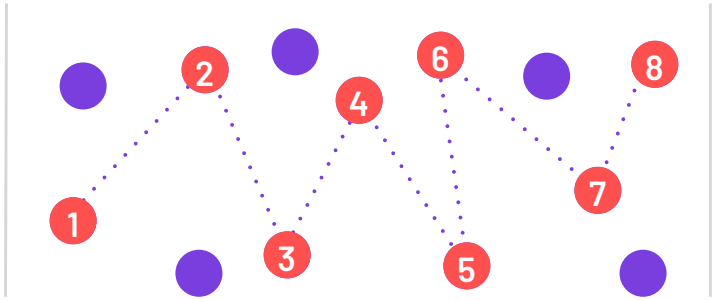
*"As 2022 approaches, this prediction could arguably be counted as "missed" — but only because we* ***underestimated the steep rise*** *in attacks on APIs."*

SALT

# These headlines keep happening because the world has changed

| | PAST | TODAY |
|---|---|---|
| Attack surface | Few APIs, static, minimal shared data | 1000s of APIs, dynamic, extensive shared data |
| Attacks | **One and done**<br>Single API call - seconds to minutes<br>Known attacks – SQLi, XSS, etc. | **Low and slow**<br>Sequence of API calls - days to weeks<br>Business logic attacks – requires context |



## The most dangerous http response code

# 200 OK

# Better context is the “how” behind all great security
# Context is hard with APIs

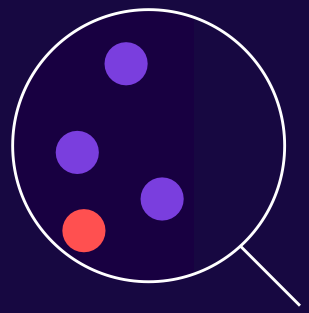
- A sea of API calls and responses
- Subtle manipulations are easy to miss

What a WAF sees

SALT

# Better context is the “how” behind all great security
# Context is hard with APIs

- A sea of API calls and responses
- Subtle manipulations are easy to miss



What other API security solutions see

SALT

Context requires a wider lens

Needs cloud-scale big data + mature ML/AI

SALT

# OWASP API Security Top 10

**A1:** Broken Object Level Authorization

**A2:** Broken Authentication

**A3:** Excessive Data Exposure

**A4:** Lack of Resources & Rate Limiting

**A5:** Broken Function Level Authorization

**A6:** Mass Assignment

**A7:** Security Misconfiguration

**A8:** Injection

**A9:** Improper Assets Management

**A10:** Insufficient Logging & Monitoring

| Threat Agents | Exploitability | Weakness Prevalence | Weakness Detectability | Technical Impact | Business Impacts |
|---|---|---|---|---|---|
| API Specific | Easy: **3** | Widespread **3** | Easy **3** | Severe **3** | Business Specific |
| | Average: **2** | Common **2** | Average **2** | Moderate **2** | |
| | Difficult: **1** | Difficult **1** | Difficult **1** | Minor **1** | |

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

# A1 – Broken Object Level Authorization (BOLA)

Legitimate – userId matches in the query parameter and request

**Request:**
GET /v1/customers/15981?**userId=207939055** HTTP/1.1

**Authorization**: Bearer gwwh1Y4epjv9Y

**Cookie**: _ga=GA1.3.630674023.1502871544;
_gid=GA1.2.1579405782.1502871544;**userId=207939055**
**Host**: payments-api.dnssf.com
**X-Forwarded-For**: 54.183.50.90

**Response:**
200 OK

```
{
    userId: 207939055,
  firstName: "John",
  lastName: "Smith",
  email: "john.smith@acme.com",
  phoneNumber: "+1650123123"
}
```

Attack - Attacker changes the userId in the query parameter

**Request:**
GET /v1/customers/15981?**userId=207938044** HTTP/1.1

**Authorization**: Bearer gwwh1Y4epjv9Y

**Cookie**: _ga=GA1.3.630674023.1502871544;
_gid=GA1.2.1579405782.1502871544;**userId=207939055**
**Host**: payments-api.dnssf.com
**X-Forwarded-For**: 54.183.50.90

**Response:**
200 OK

```
{
    userId: 207938044,
  firstName: "David",
  lastName: "Miller",
  email: "david.miller@example.com",
  phoneNumber: "+1912456456"
}
```

Unauthorized access can result in

- unauthorized data access
- data loss
- data manipulation
- can also lead to full account takeover

API protection solutions must detect when 2 identifiers should always be identical to prevent an attacker from manipulating one of them

SALT

# A1 - BOLA : Verizon

## Exposure of personal information of 2 million Verizon Wireless customers

verizon

**RETAIL INSTALLMENT CONTRACT**

**RETAIL INSTALLMENT SALE AGREEMENT / RETAIL INSTALLMENT OBLIGATION**
**SUBJECT TO STATE REGULATION**

SELLER (CREDITOR): Verizon Wireless Services, LLC ("Verizon Wireless")
One Verizon Way, Basking Ridge, NJ 07920 (908) 559-7000

| | |
|---|---|
| INSTALLMENT SALE AGREEMENT # | 1[redacted]6 |
| BUYER'S/CUSTOMER'S NAME | [redacted] |
| BUYER'S/CUSTOMER'S CONTACT MOBILE NUMBER | 8[redacted]2 |
| ACCOUNT OWNER'S ADDRESS | 957[redacted] TX [redacted] |
| DESCRIPTION OF GOODS | IPHONE 8 SPACE GRAY 64GB [redacted] **("Device")** |
| TRANSACTION DATE | 09/17/2018 |

YOUR COMPANY, meaning the Buyer/Company named above, agree to pay US, the Seller/Creditor named above as Verizon Wireless, the Total Sale Price of the goods identified above according to the Terms of this Retail Installment Sale Agreement/ Retail Installment Obligation (referred to below as "Agreement").

| **ANNUAL PERCENTAGE RATE** **The cost of Customer's credit at a yearly rate** | **FINANCE CHARGE** **The dollar amount the credit will cost Customer** | AMOUNT FINANCED The amount of credit provided to you; or on your behalf | TOTAL OF PAYMENTS The amount Customer will have paid after all payments are made as scheduled | TOTAL SALE PRICE The total cost of Customer's purchase on credit including your down payment of $0.00 |
|---|---|---|---|---|
| **0%** | **$0.00** | $599.99 | $599.99 | $599.99 |

Your Company's payment schedule will be:

Number of Payments:24; Payment 1:$25.22; Payments 2-24:$24.99

When Payments are Due:

| | | | | | | |
|---|---|---|---|---|---|---|
| **Payments 1 to 6** | 10/30/2018 | 11/29/2018 | 12/30/2018 | 01/30/2019 | 02/27/2019 | 03/30/2019 |
| **Payments 7 to 12** | 04/29/2019 | 05/30/2019 | 06/29/2019 | 07/30/2019 | 08/30/2019 | 09/29/2019 |
| **Payments 13 to 18** | 10/30/2019 | 11/29/2019 | 12/30/2019 | 01/30/2020 | 02/28/2020 | 03/30/2020 |
| **Payments 19 to 24** | 04/29/2020 | 05/30/2020 | 06/29/2020 | 07/30/2020 | 08/30/2020 | 09/29/2020 |

**PAYMENTS RECEIVED 15 OR MORE DAYS AFTER YOUR COMPANY'S DUE DATE MAY INCUR A LATE PAYMENT FEE OF UP TO 5% OR $5, WHICHEVER IS LESS. PLEASE SEE YOUR COMPANY'S AGREEMENT TERMS FOR ANY ADDITIONAL INFORMATION ABOUT NONPAYMENT, DEFAULT, ANY REQUIRED PAYMENT IN FULL BEFORE THE SCHEDULED PAYMENT DATES, AND PREPAYMENT TERMS.**

| | |
|---|---|
| ITEMIZATION OF AMOUNT FINANCED | $599.99 |
| (A) CASH PRICE (excluding tax) | $599.99 |
| (B) DOWN PAYMENT (if applicable) | $0.00 |
| (C) FINANCE CHARGE | $0.00 |
| (D) TAXES* | $49.50 |
| (E) AMOUNT FINANCED (the amount of credit provided to you on your behalf) | $599.99 |

* Not included in Amount Financed

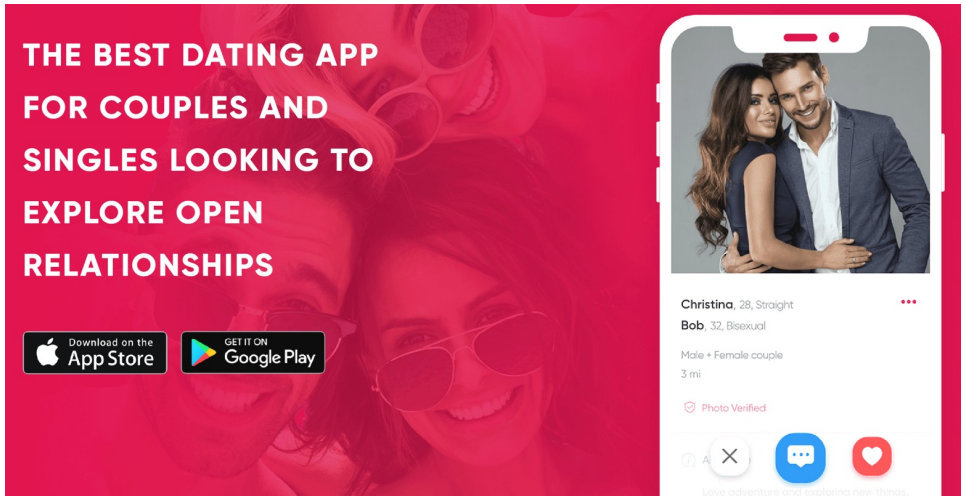
- While authentication was needed to access the files, the expert initially managed to access one contract, linked to a specific phone number and contract number, after brute-forcing the URL's GET parameters.
- The researcher then realized that modifying the value of one of these parameters would display a different contract.

```
After a quick check, I learnt that 1310000000 was the lowest contract number
that could be viewed and 1311999999 was the highest. That means that there was
information of around 2 million Verizon Pay Monthly customers exposed.
```

https://daleys.space/writeup/0day/2019/09/09/verizon-leak.html

# A3 – Excessive Data Exposure: Three Fun

## Exposing near real time location and PII




It exposes the near real time location of any user; at work, at home, on the move, wherever.

It exposes users dates of birth, sexual preferences and other data.

3fun emailed the researcher to grumble (because that's the thing you should be upset about...).

It exposes users private pictures, even if privacy is set.



https://www.pentestpartners.com/security-blog/group-sex-app-leaks-locations-pictures-and-other-personal-details-identifies-users-in-white-house-and-supreme-court/

# A3 – Excessive Data Exposure: Three Fun

## Exposing near real time location and PII

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type |
|---|---|---|---|---|---|---|---|---|
| 322 | https://www.go3fun.co | POST | /account_kit_reg | ✓ | | 200 | 447 | JSON |
| 325 | https://www.go3fun.co | POST | /user/device_token | ✓ | | 200 | 198 | JSON |
| 326 | https://www.go3fun.co | POST | /user/update | ✓ | | 200 | 265 | JSON |
| 327 | https://www.go3fun.co | POST | /reset_push_badge | | | 200 | 198 | JSON |
| 329 | https://www.go3fun.co | GET | /match_users?from=0&latitude=51.[redacted] | ✓ | | 200 | 23807 | JSON |
| 331 | https://www.go3fun.co | GET | /user/refresh | | | 200 | 788 | JSON |
| 334 | https://www.go3fun.co | POST | /user/update_location | ✓ | | 200 | 198 | JSON |
| 338 | https://www.go3fun.co | POST | /upload_photo | ✓ | | 200 | 479 | JSON |
| 339 | https://www.go3fun.co | GET | /i_like_list?from=0&offset=30 | ✓ | | 200 | 201 | JSON |
| 340 | https://www.go3fun.co | GET | /chatted_list | | | 200 | 201 | JSON |
| 341 | https://www.go3fun.co | POST | /reset_push_badge | | | 200 | 198 | JSON |
| 344 | https://www.go3fun.co | GET | /user/refresh | | | 200 | 992 | JSON |
| 348 | https://www.go3fun.co | GET | /matched_list?from=0&offset=30 | ✓ | | 200 | 201 | JSON |

Request | Response

Raw | Headers | Hex | JSON Beautifier


```
    ],
    "latitude": "51.[redacted]",
    "membership": "2",
    "birthday": "1977-[redacted]",
    "sex_orient": "4",
    "gender": "1",
    "longitude": "-0.1[redacted]",
    "photo_verified_status": "1",
    "active": "0",
    "partner_sex_orient": "0",
    "liked_me": "0",
    "settings": {
      "show_online_status": "1",
      "show_distance": "1"
    },
    "username": "[redacted]",
    "usr_id": "17[redacted]",
    "about_me": "Kinky and attractive french financier open to many things ..."
  },
  {
    "last_login": "2019-06-24 20:21:12",
    "private_photos": [
      {
        "icon": "https://s3.amazonaws.com/3fun/821/[redacted]small.jpg",
        "photo_id": "38[redacted]",
        "py": "500",
        "px": "750",
        "photo": "https://s3.amazonaws.com/3fun/821/[redacted]big.jpg",
        "descr": null
      }
```


You'll see the latitude and longitude of the user is disclosed.

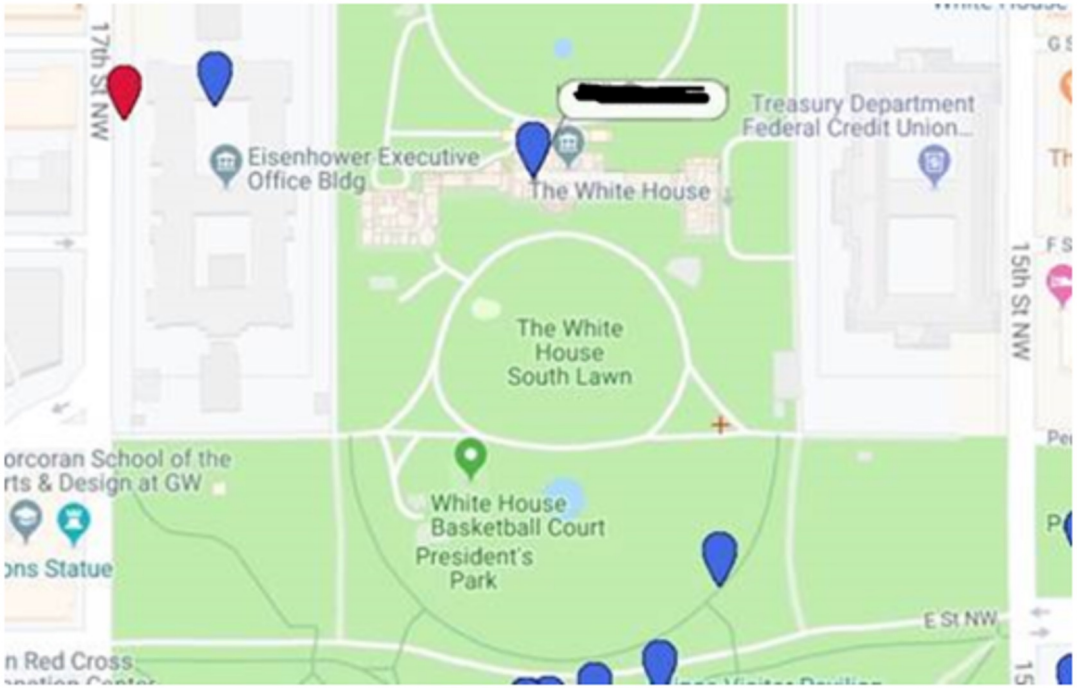
Now, the user can restrict the sending of the lat/long so as not to give away their position.

BUT, that data is only filtered in the mobile app itself, not on the server. It's just hidden in the mobile app interface if the privacy flag is set. The filtering is client-side, so the API can still be queried for the position data.

https://www.pentestpartners.com/security-blog/group-sex-app-leaks-locations-pictures-and-other-personal-details-identifies-users-in-white-house-and-supreme-court/

# A3 – Excessive Data Exposure: Three Fun

## exposing near real time location and PII




Including one in the White House, although it's technically possible to re-write ones position, so it could be a tech savvy user having fun making their position appear as if they are in the seat of power.

https://www.pentestpartners.com/security-blog/group-sex-app-leaks-locations-pictures-and-other-personal-details-identifies-users-in-white-house-and-supreme-court/

# Why is cloud-scale, big data necessary to gain full context?

## Raw Traffic

*Billions of calls per month*

RESPONSE
200 OK

Server:nginx/1.6.2
Date:Mon, 02 Apr 2018 09:10:13 GMT
Connection:keep-alive
X-Frame-Options:SAMEORIGIN
Content-Type:text/html; charset=UTF-8
Content-Length: 41

{"account_balance": 2006, "userId": 107939053, "description": null, "subscriptions": {"has_more": false, "total_count": 0, "object": "list", "data": [ ], "url": "/v1/customers/f86e2276-98d0-4ad6-81ef-68fa1bcf5281/subscriptions"}, "livemode": false, "currency": "usd", "shipping": null, "id": "f86e2276-98d0-4ad6-81ef-68fa1bcf5281", "delinquent": false, "created": 1504792830, "default_source": null, "object": "customer", "sources": {"has_more": false, "total_count": 0, "object": "list", "data": [ ], "url": "/v1/customers/f86e2276-98d0-4ad6-81ef-68fa1bcf5281/sources"}, "discount": null, "email": "john.doe@mail.com", "metadata": {}}

REQUEST
GET /v1/customers/f86e2276-98d0-4ad6-81ef-68fa1bcf5281?accountId=f86e2276-98d0-4ad6-81ef-58fc1bcf5382 HTTP/1.1

Accept: */*
Accept-Encoding: "gzip, deflate"
Authorization:
"Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhbGciOiJSUzI1NiIsImF1ZCI6IjN2MUo4WHczemVHdUdPT2Im0Wh0WFdFUkIiLCJlbWFpbCI6ImNhcm90NDEyMkBnbWFpbC5jb20iLCJleHAiOiIxNjAzNzg2Njc3IiwiaWFOIjoiMTYwMzcwMDI3NyIsImlzcyI6Imh0dHBzOi8vY2hhcmdlc25idHdyb2suYXV0aDAuY28tIiwic3ViIjoiYXV0aDB8NWY5NT0xZmU1ZDkxMWUwMDEiLCJ0eXAiOiJKViOifQ.0hNZPMk14zkX7mcFk1zfw00gzoLhLbaygvl3PNHFT2w
Authorization.alg: "RS256"
Authorization.aud: "3v1J8Xw3zeGuGOOifAhPXWERB"
Authorization.email: "carot4122@gmail.com"
Authorization.exp: "1603786677"
Authorization.iss: "https://chargesnetwrok.auth0.com"
Authorization.sub: "auth0|5f9541fe5d911e001"
Authorization.typ: "JWT"
Authorization.userId: "1603700277"
Connection: "keep-alive"
Content-Length: "0"
Content-Type: "application/json; charset=UTF-8"
Host: "payments-api.dnssf.com"
User-Agent: "secful-script"
X-Forwarded-For: "54.183.50.90"
Cookie:_ga="GA1.3.630674023.1502871544" _gid="GA1.2.1579405782.1502871544" userId=" 107939053 "

## Structural Metadata

*100s to 1000s per call*

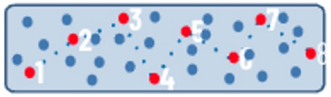
domain
protocol
method
URI
URI parameter names
URI parameter count
URI parameter length
URI parameter datatype
request.headers
request.headers count
request.headers.names
request headers.names.datatype
request headers.names.length
request headers.names.classification
request headers names.value.datatype
request.headers.names.value.length
request.headers.names.value.classification
request.size
request.body.content-type
request.body.content-type.parameters
request.body.content-type.parameters.names
request.body.content-type.parameters.names.datatype
request.body.content-type.parameters.names.length
request.body.content-type.parameters.names.classification
request.body.content-type.parameters.names.value.datatype
request.body.content-type.parameters.names.value.length
request.body.content-type.parameters.names.value.classification
response.size
response.headers
response.headers count
response.headers.names
response.headers.names.datatype
response.headers.names.length
response.headers.names.classification
response.headers names.value.datatype
response.headers.names.value.length
response.headers.names.value.classification
response.body.content-type
response.body.content-type.parameters
response.body.content-type.parameters.names
response.body.content-type.parameters.names.datatype
response.body.content-type.parameters.names.length
response.body.content-type.parameters.names.classification
response.body.content-type.parameters.names.value.datatype
response.body.content-type.parameters.names.value.length
response.body.content-type.parameters.names.value.classification
...

## Behavioral Attributes

*100s to 1000s per call*

session correlation
user identification
API characteristics (internal/external)
authentication identification
static data determination
dynamic data determination
request header data relationships
request body data relationships
response header data relationships
response body data relationships
sensitive data relationships
call sequences
call frequency
user attributes
user past behavior
...

## Analysis Windows



seconds
minutes
hours
days
weeks
and beyond ...

## AI Algorithms

The only way to effectively discern user intent in near real-time with no alert fatigue (false positives) and no missed security events (false negatives)

SALT

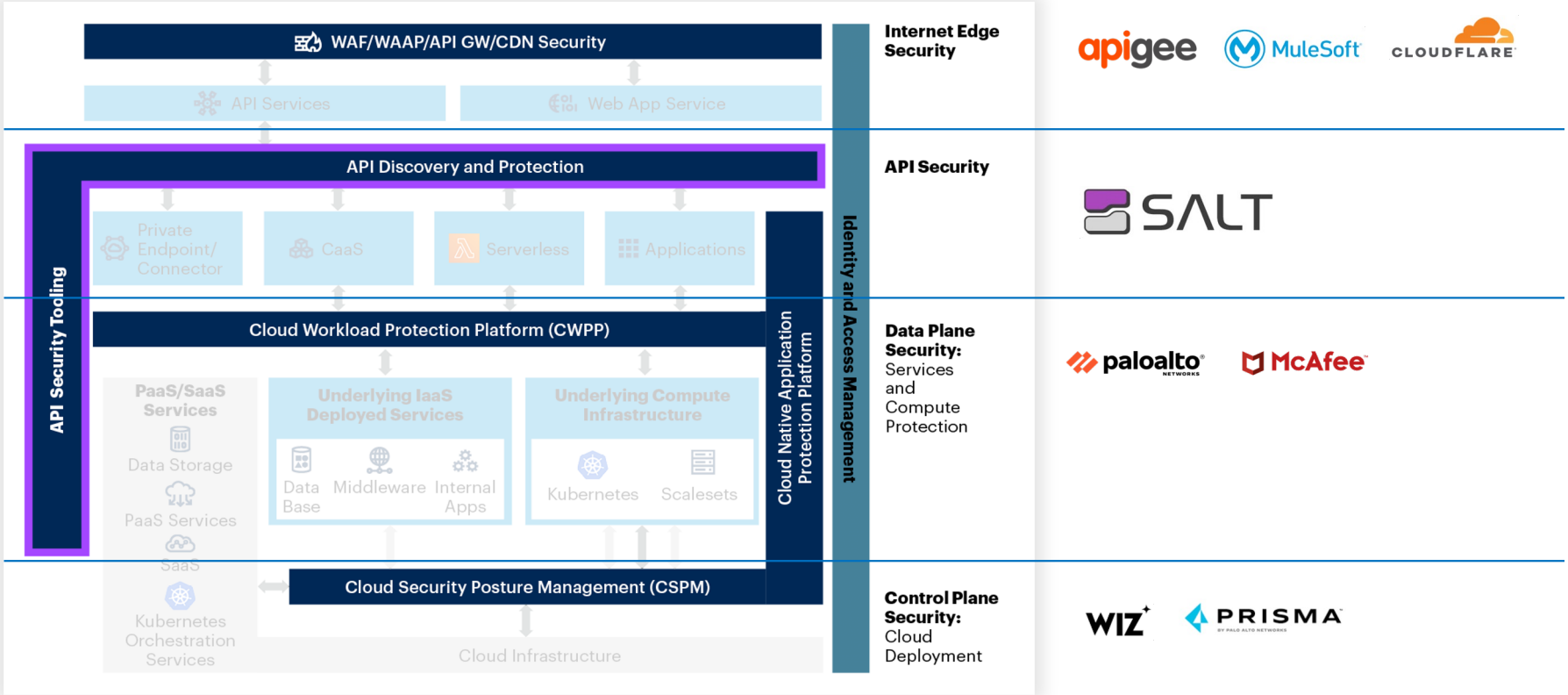
# In API security, context is the only way to accurately discern user intent

**Ask yourself...**

Could we detect a slow frequency (one request per minute or per hour) single ID BOLA attack in an API that is fielding over a billion requests a month?

# The Salt solution

# Gartner validates today's unique requirements – API Security is its own category




Gartner "Advance Your Platform-as-a-Service Security," Richard Bartley, 25 August 2021

# Successful API security must:

## Discover APIs

and sensitive data continuously

↓

Essential foundation

## Protect APIs

against "in the wild" attacks

↓

Immediate risk reduction

## Shift left

to improve API security posture

↓

Future, limited risk reduction

# How is Salt different from a WAF?

## API Security

| Discovery | Runtime protection | Improved API security posture |
|---|---|---|
| dynamic API catalog | baseline unique API traffic | OAS gap analysis |
| zombie and shadow APIs | attacker blocking during recon | attacker fingerprinting |
| sensitive data leak detection | OWASP API Top 10 ++ attacks | API vulnerabilities |
| data classification | cross-site scripting, SQL injection | auto routing remediation details |
| | L3 for DDoS | |
| | Some OWASP Top 10 attacks | |

5% overlap

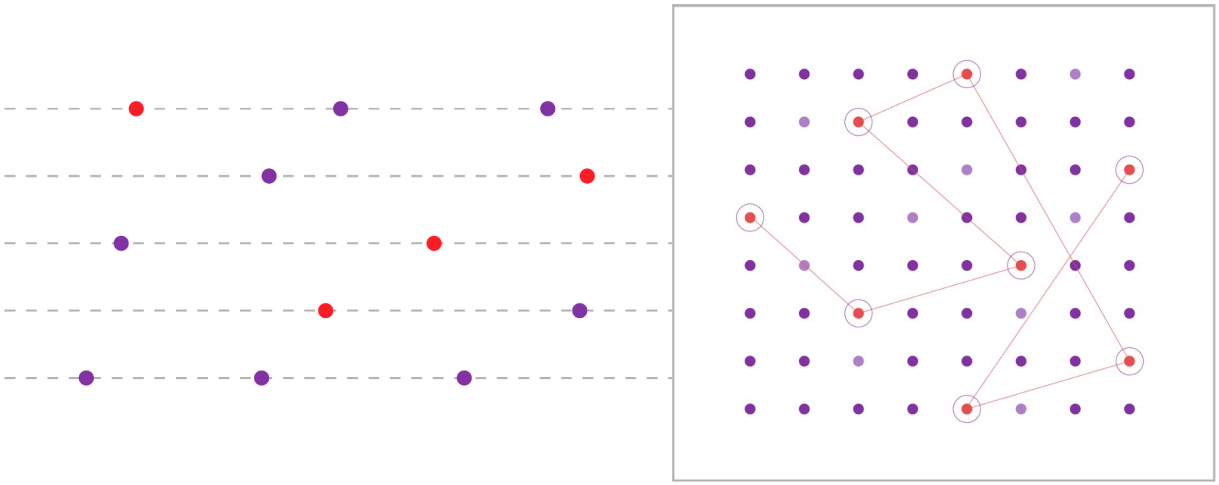
### SALT

- Out of band – no latency
- Prevent business logic manipulation
- Automatic

### WAF

- Inline
- Stop known bad
- Manual

# The Salt approach – the context to see the full picture



API Discovery

Runtime Protection

Shift Left Security

1. Deploy in minutes across all environments
   Agentless traffic monitoring

2. Apply cloud-scale big data and ML/AI to build rich context
   API Context Engine (ACE)

3. Get full coverage across all your APIs
   Salt Security API Protection Platform

# Salt in action – agentless complete protection in minutes, for any environment

1. Not inline
2. Automated, continuous
3. Environment agnostic

**Clients**

SaaS, web, mobile, IoT

API communications

**API collection options**

**Agentless**

API gateways

apigee, MuleSoft, Kong, Amazon API Gateway, Azure API Management, IBM DataPower, axway, TIBCO, akana, WSO2

microservices

kubernetes, docker

load balancers, edge, servers

f5, NGINX, vmware, Cloudflare Workers, Gigamon, envoy, servers

cloud

aws, Azure, Google Cloud

## Customer environments

Block – manual or automatic

Mirror of all API traffic

**Salt Hybrid Server** (optional)

API metadata

Alerts, remediation tickets

**Operations and management**

Jira Software, PagerDuty, slack, splunk>, sumo logic, servicenow, snowflake, Webhooks

## Salt cloud

API discovery
Data classification
Baselining for 100s of attributes
Attack detection and prevention

# Top use cases for API security

Discover shadow APIs

Prevent sensitive data exposure

Stop API attacks

Prevent account takeover

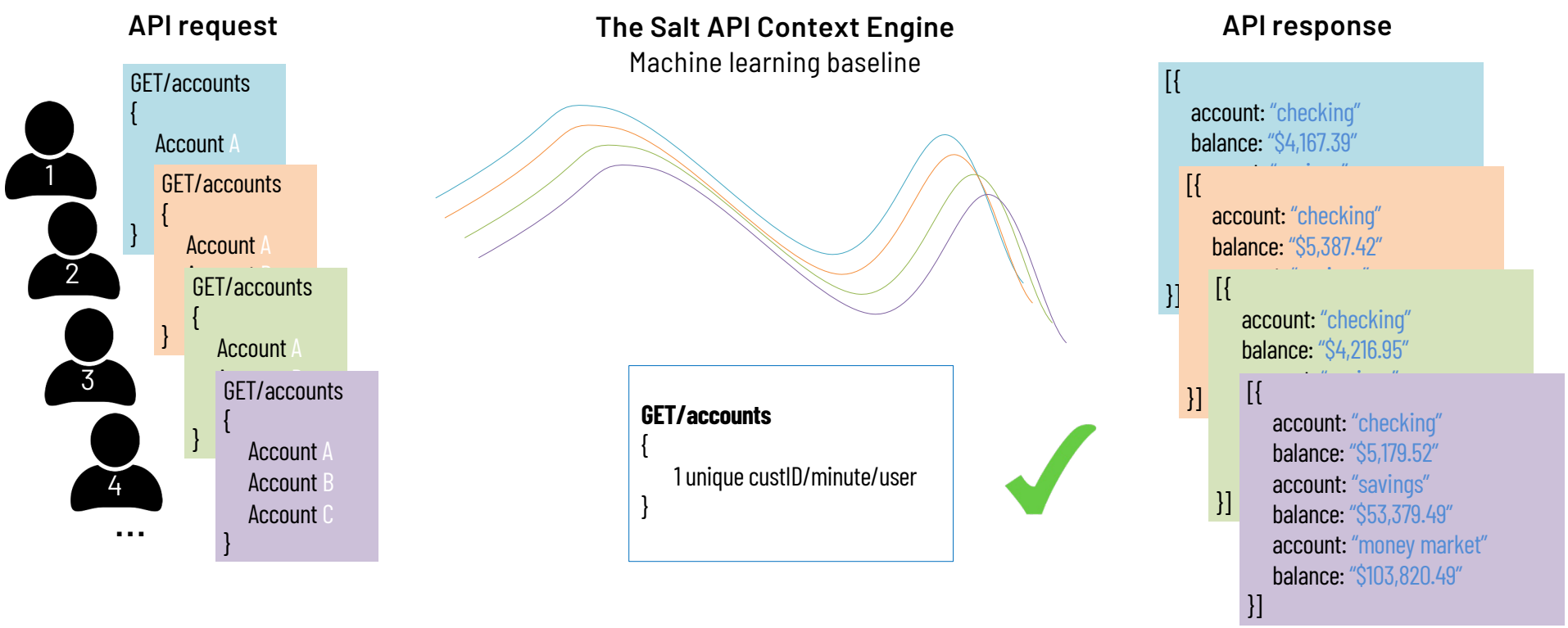
Prevent data exfiltration

Reduce investigation time

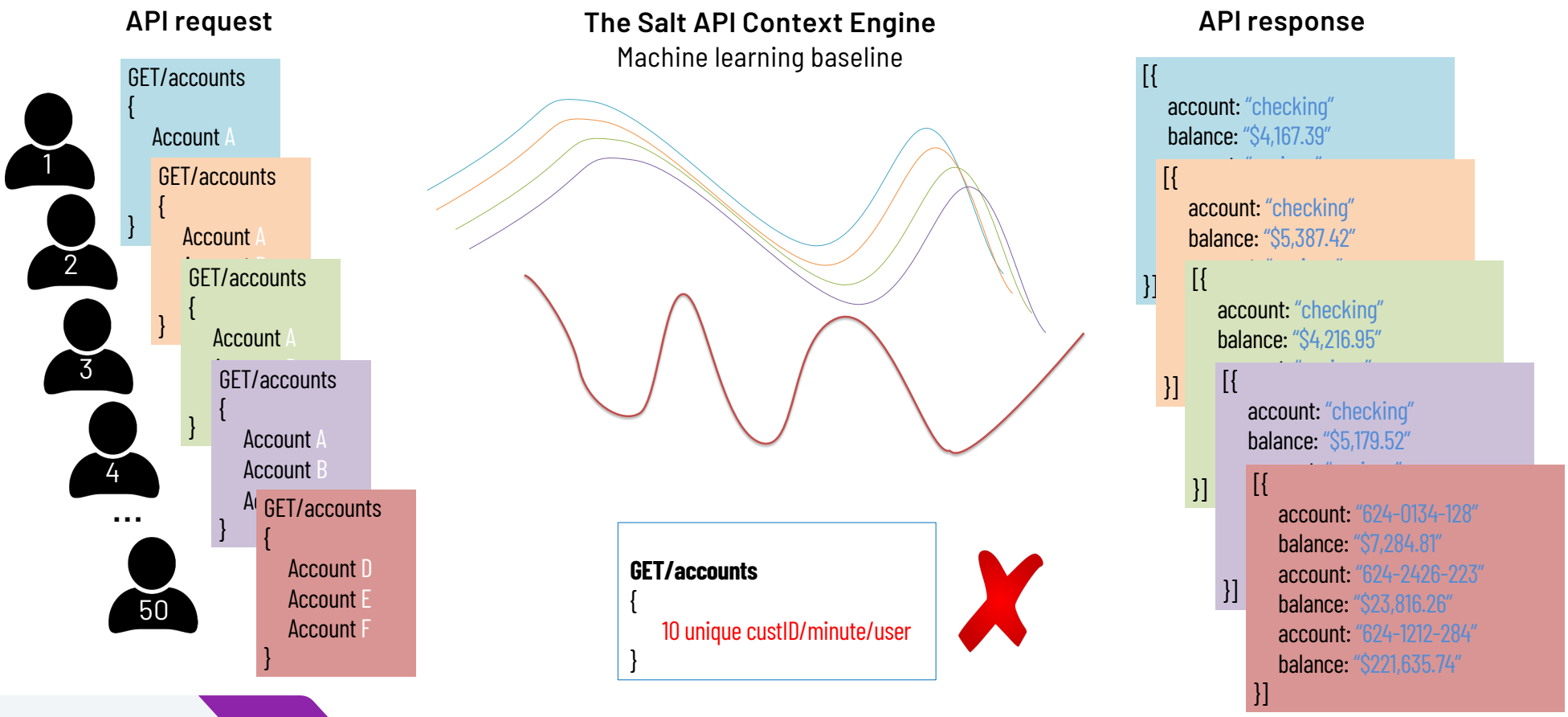
Provide remediation insights

Simplify compliance

# Detecting and blocking a single-parameter BOLA (and the attacker behind it)



# Detecting and blocking a single-parameter BOLA (and the attacker behind it)



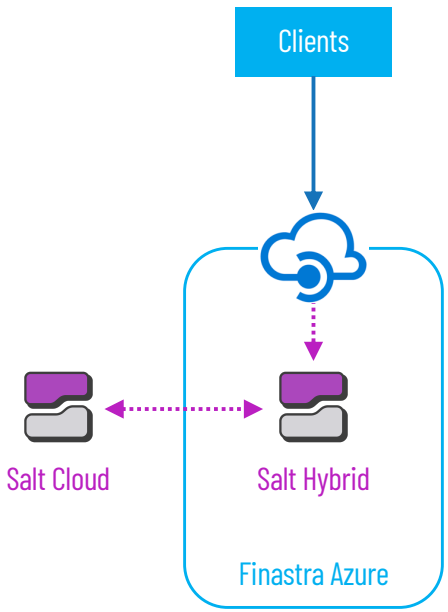

SALT

# Addressing the OWASP API Top 10 threats

| OWASP API Security Top 10 Threats | WAFs | API Gateways | OAS Schema Validation | On prem API Security | SALT |
|---|---|---|---|---|---|
| API1:2019 – Broken Object Level Authorization | ✗ | ✗ | ✗ | ✗ | ✓ |
| API2:2019 – Broken Authentication | ✗ | manual | ✗ | partial | ✓ |
| API3:2019 – Excessive Data Exposure | ✗ | ✗ | ✗ | partial | ✓ |
| API4:2019 – Lack of Resources and Rate Limiting | ✗ | manual, partial | manual, partial | partial | ✓ |
| API5:2019 – Broken Function Level Authorization | ✗ | partial | ✓ | ✓ | ✓ |
| API6:2019 – Mass Assignment | ✗ | ✗ | ✗ | partial | ✓ |
| API7:2019 – Security Misconfiguration | partial | ✗ | ✗ | ✓ | ✓ |
| API8:2019 – Injection | (signature based) | (signature based) | ✗ | ✓ | ✓ |
| API9:2019 – Improper Assets Management | ✗ | manual, partial | manual, partial | ✓ | ✓ |
| API10:2019 – Insufficient Logging and Monitoring | partial | partial | partial | ✓ | ✓ |

FINASTRA

FinTech




## Drivers

- Protect FusionFabric SaaS banking platform and marketplace
- Meet regulatory requirements

## Results

- Automatically block sophisticated, ongoing credential stuffing attacks
- Identify high-severity vulnerabilities for quick remediation
- Simplify compliance with protection and reporting

# Only Salt delivers the depth of protection you need

## Weeks of data, 4 years+ of training models, real-time analysis, network effect

### API discovery

- Intelligent parsing of APIs, endpoints
- More accurate data classification

### Runtime protection

- Stop more “in the wild” attacks
- Block attackers, not attacks

### Shift left

- Pre-prod tests tuned to your APIs
- Runtime insights for dev team

SALT

## Closing remarks

- API security is NOT a product by itself. It's a full stack strategy.
- You can't protect what you can't see!
- Context is key, on premises in-line solutions fall short!

Thank you!

Questions?