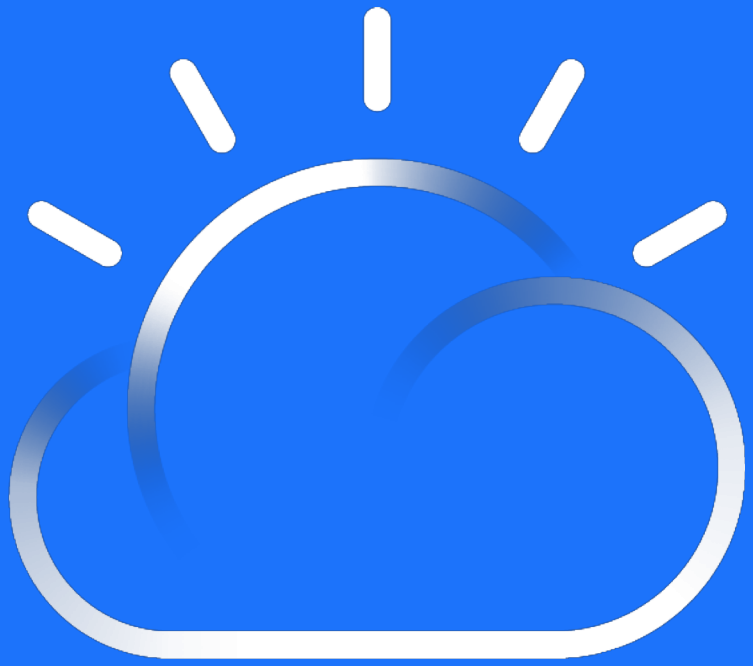
Integration Technical Conference 2019

Machine Learning Cybersecurity for IBM API Connect

Francois Lascelles
Field CTO
Ping Identity



IBM **Cloud**

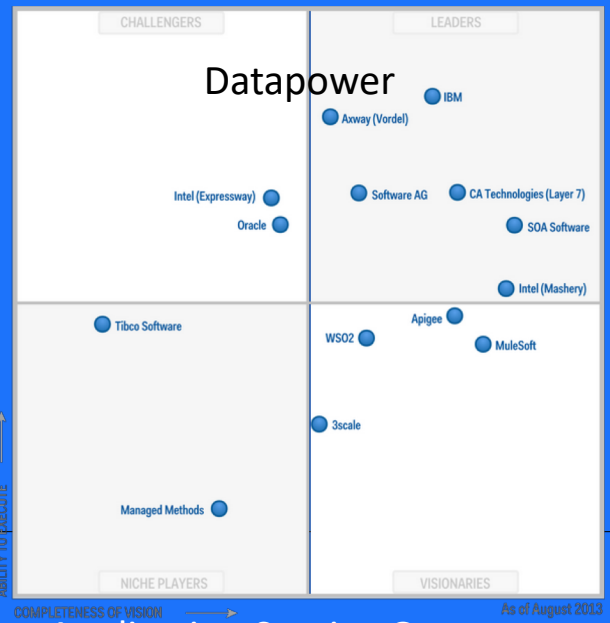
IBM

My API Security Journey

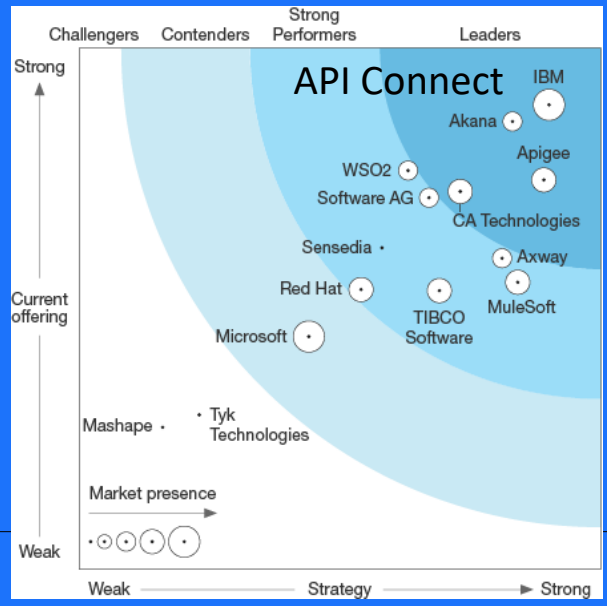
2002 Layer 7 Technologies

2013 CA API Management

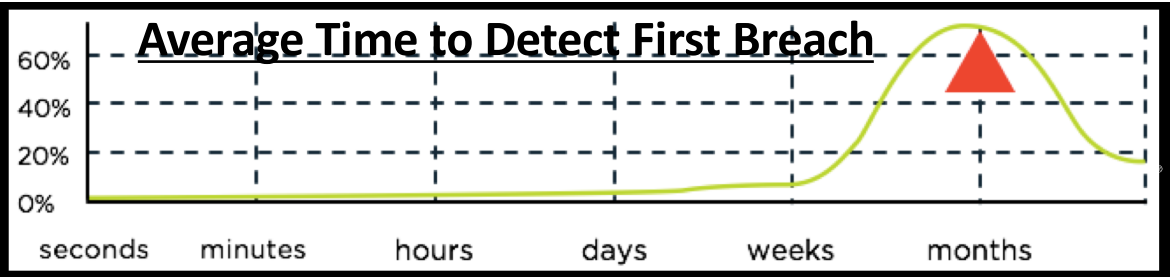
Today Ping Identity



Gartner Application Service Governance 2013



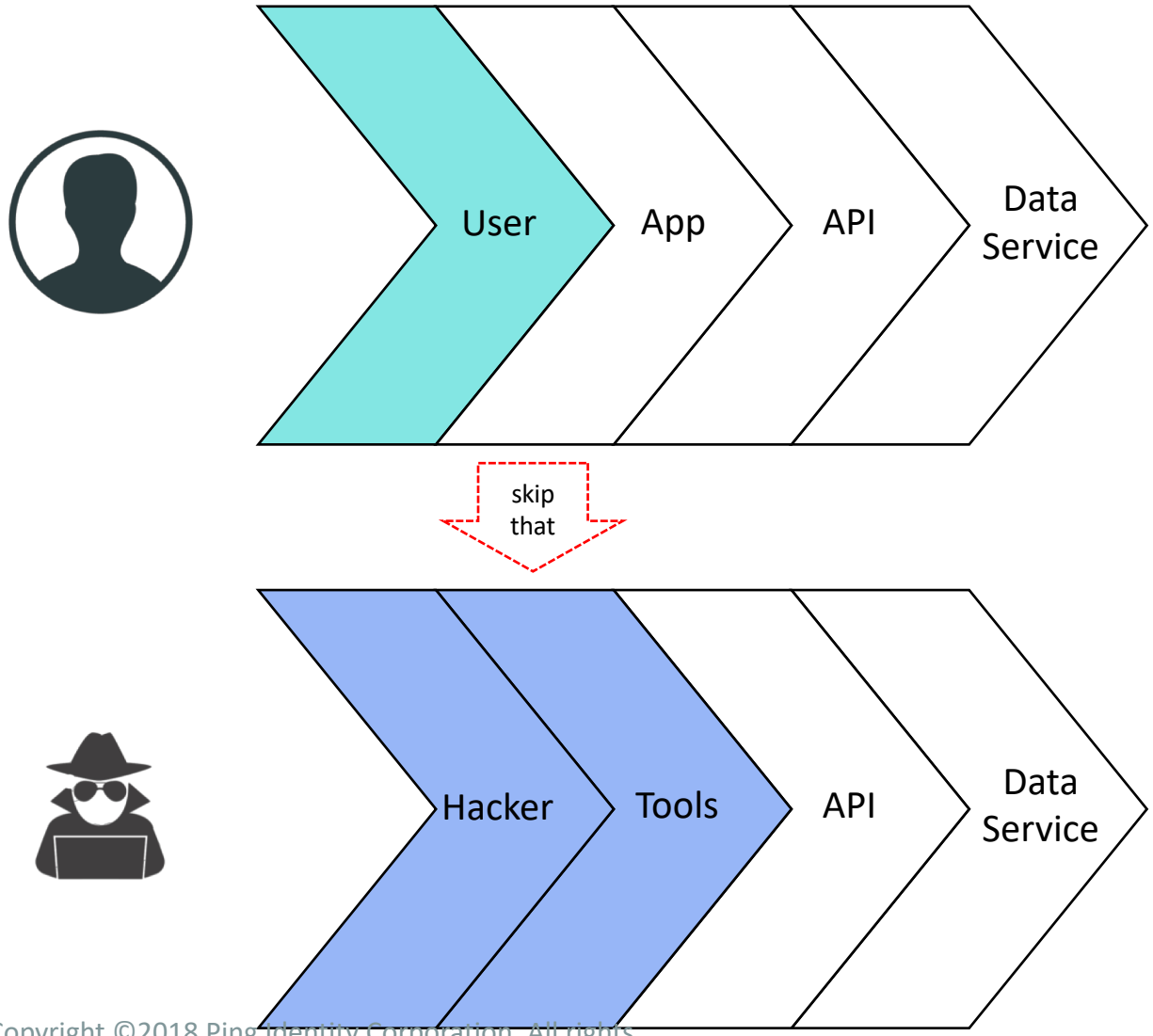
Forrester APIM Wave 2016



2018 Verizon DBIR

- API Breaches go undetected for months, years
- Enterprises need our help to improve API cybersecurity
- Gartner: “by 2020, API abuses will be the most frequent attack vector that result in breaches”
- Some attacks can’t be detected with traditional API security
- Help is on the way from IBM and Ping Identity

HACKERS USE YOUR API OUTSIDE OF YOUR APP



- Client-side rules skipped
- Unexpected and untested-for API abuse scenarios
- Freedom to poke around and find vulnerabilities

YES, HACKERS KNOW ABOUT YOUR API

Your API is either well documented, easily reverse-engineered, or both.

Developer portal

- https proxy
- JS, browser tools
- Debug tools, APKtool, etc
- GitHub

- Some documented/some hidden features
- Nothing is hidden if it can be reverse-engineered

API Breach Impact

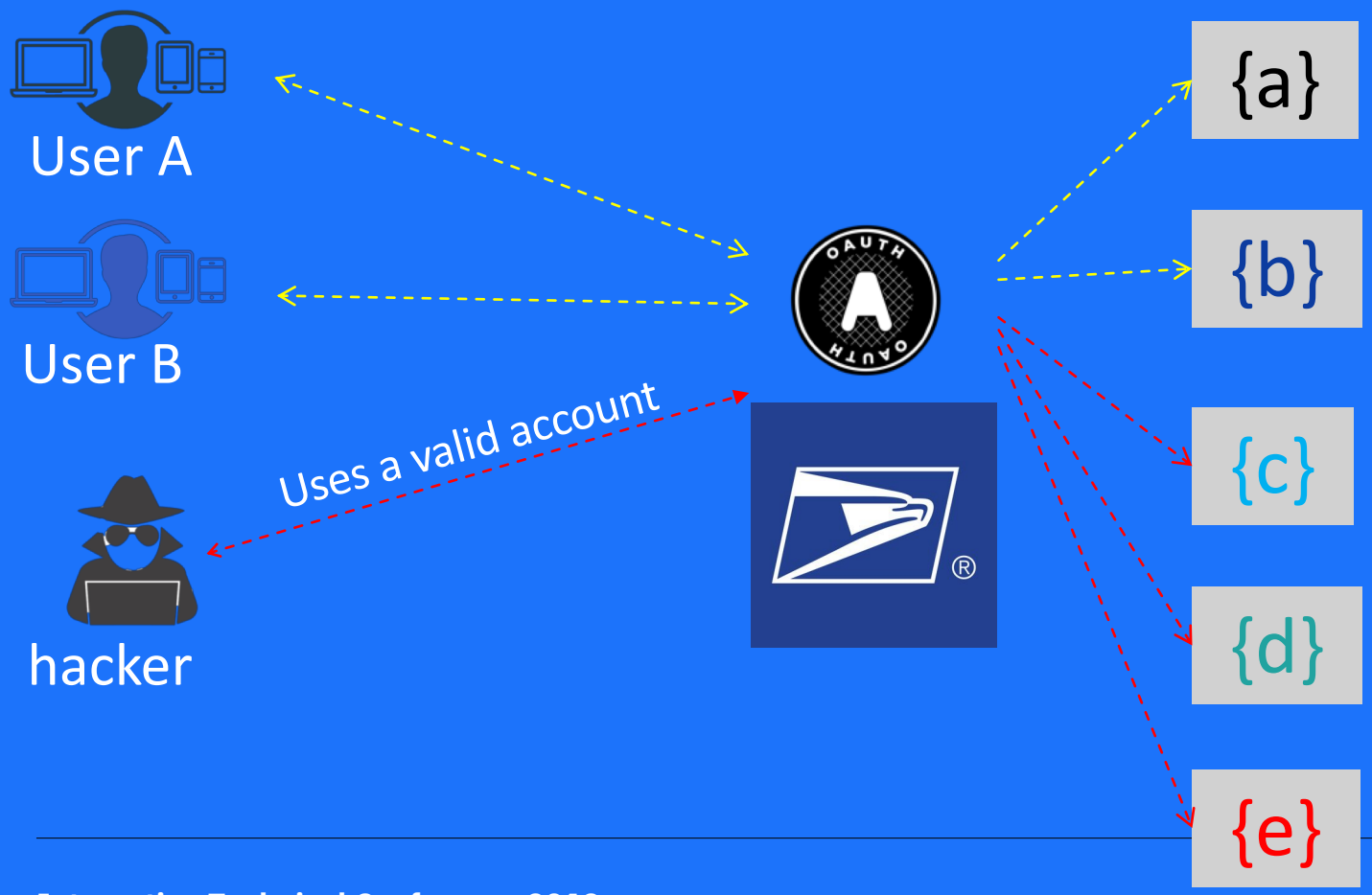
- Client and Patient accounts taken over …
- Industrial control systems taken hostage or worse …
- Services, mobile apps shut down or disrupted …
- Data breaches – theft of customer records, private data, credit cards, …
- Fraud for banks, retailers, payment processors, …
- Ransom, jobs lost, …

APIs Not secured

- An API that was designed for internal use became accessible from outside.
- Property valuation details and contact information was leaked.
- Reputation damaged
- CEO now looking for work

- “Connect with your friends” social API
- Not expected to be called outside app
- Data breach
- Ransom

API Takeover



- **Hacker uses valid account** to access API
- **Uses discovered vulnerabilities to access other accounts**
- **Takes over accounts** and steal data, photos, private information
- Continues undetected for months
- USPS, t-Mobile, Verizon,…

Demo

attack.start();

Denial of service

February 2019

json-patch
{ 10s of thousands of instructions }

{apiserver}

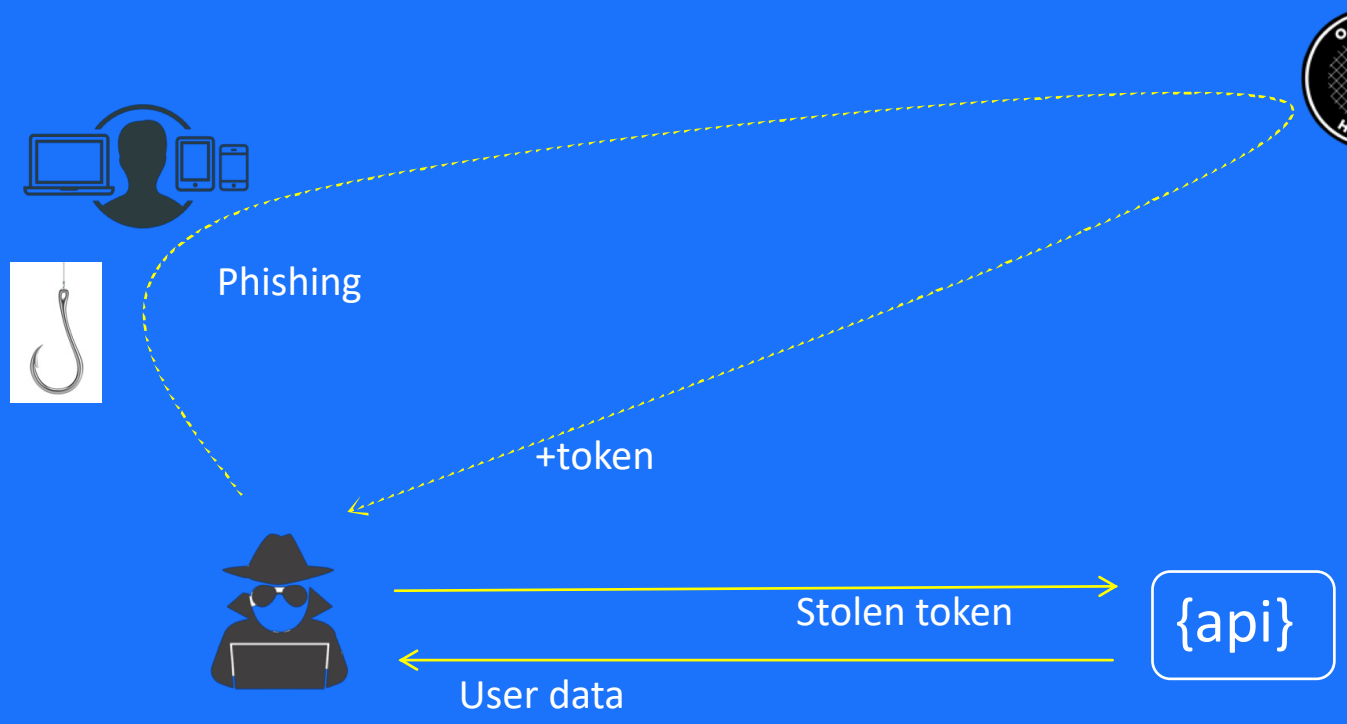
- Exhaust process
- DOS

Fix: apiserver will now return 413 RequestEntityTooLarge error if a json patch contains more than 10,000 operations

What are the chances of you missing such rules in your gateway?

Ref.:
https://github.com/kubernetes/kubernetes/pull/74000
https://github.com/kubernetes/kubernetes/issues/74534

Stolen tokens



- Exploit user and/or client weakness
- Immune to meticulous security practices (entitlement checks, mfa, content validation) because token is valid and is associated to user data being breached
- Stolen token are then used for data exfiltration, account take-over and other exploits

Indirect stolen tokens

IBM

```
>collections
_
```

GitHub leaking client secrets

Default creds

Password spraying

Password reuse

Remote Access Trojan

Mimikatz

creds

impersonate

stolen token

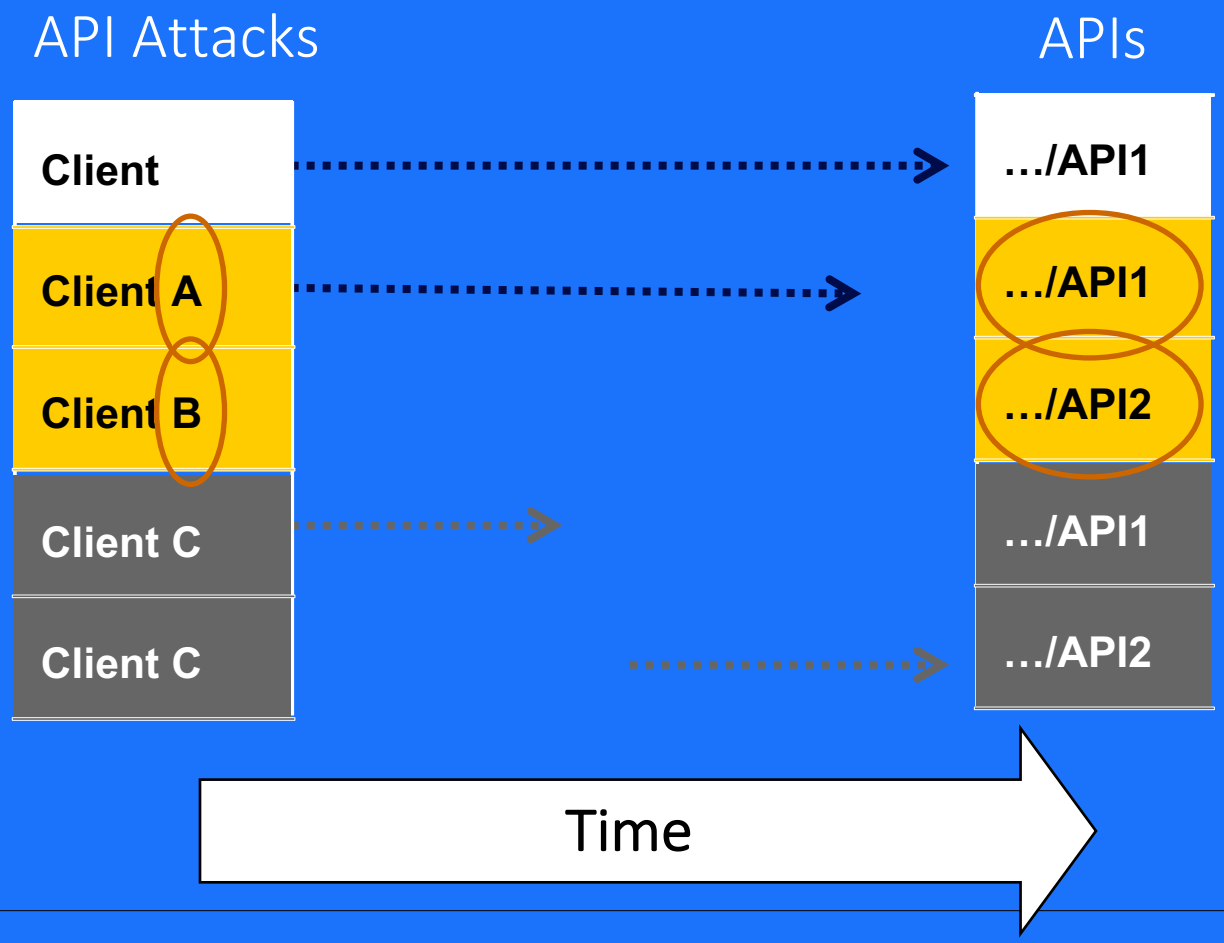
{api}

If you can't trust your own tokens, then what?

API Security vs API cybersecurity

- Traditional API Security is mostly transactional scope
- No visibility on high-level activity and behavioral patterns
- Advanced tooling, but hard to configure and test
- Relies on knowledgeable human operators
- Expects tokens to not be compromised

API traffic pattern analysis with machine learning



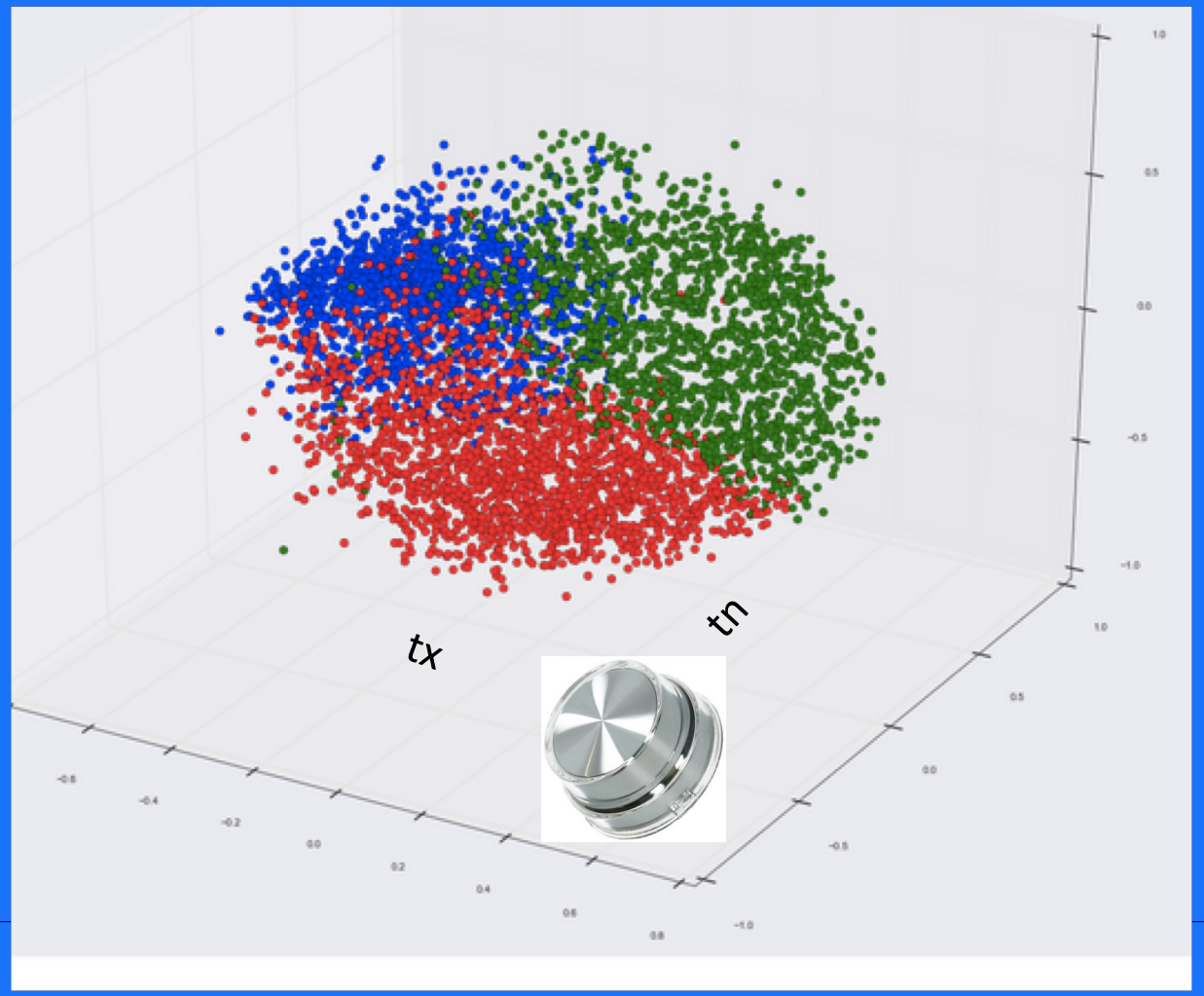
- Analyze API activity against normal use
- Detect attacks based on behavior and not just user-defined rules
- Identify probing activity
- Monitor excessive error activity
- Etc.

API Modeling Considerations

- **Per-API models**
- **Request/response correlation**
- **Allow human input**
- **Model lifecycle across environments**

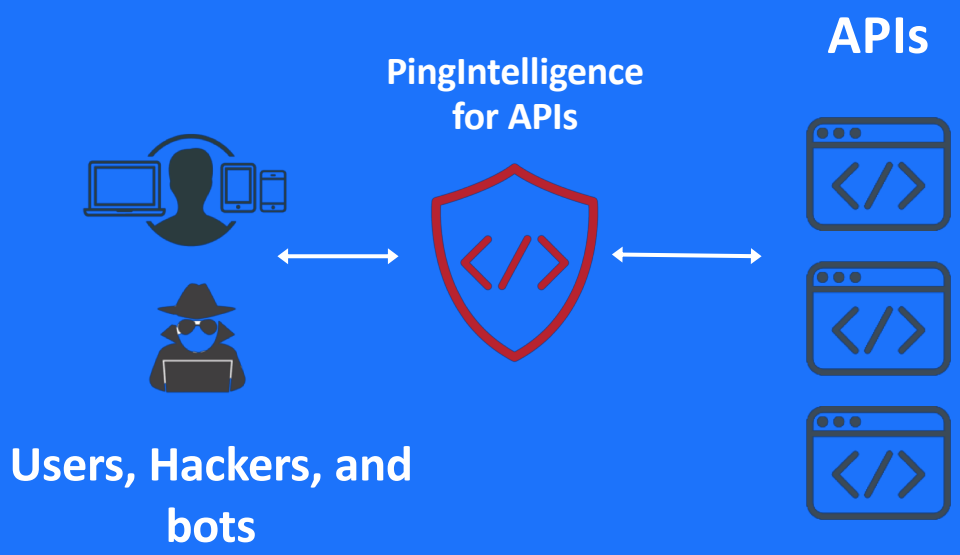
Model training and human input

1. **Training creates models**
   - Good data is key
   - Output: normal and extreme threshold characteristics
2. **Human tweaks**
   - Knobs to fine-tune



Introducing PingIntelligence for APIs

- **Advanced API security** with artificial intelligence for REST and WebSocket APIs
- **Self-learned security** – no policies or rules to write
- Automatic detection and blocking of attacks
- **Forensics reporting**



Design time

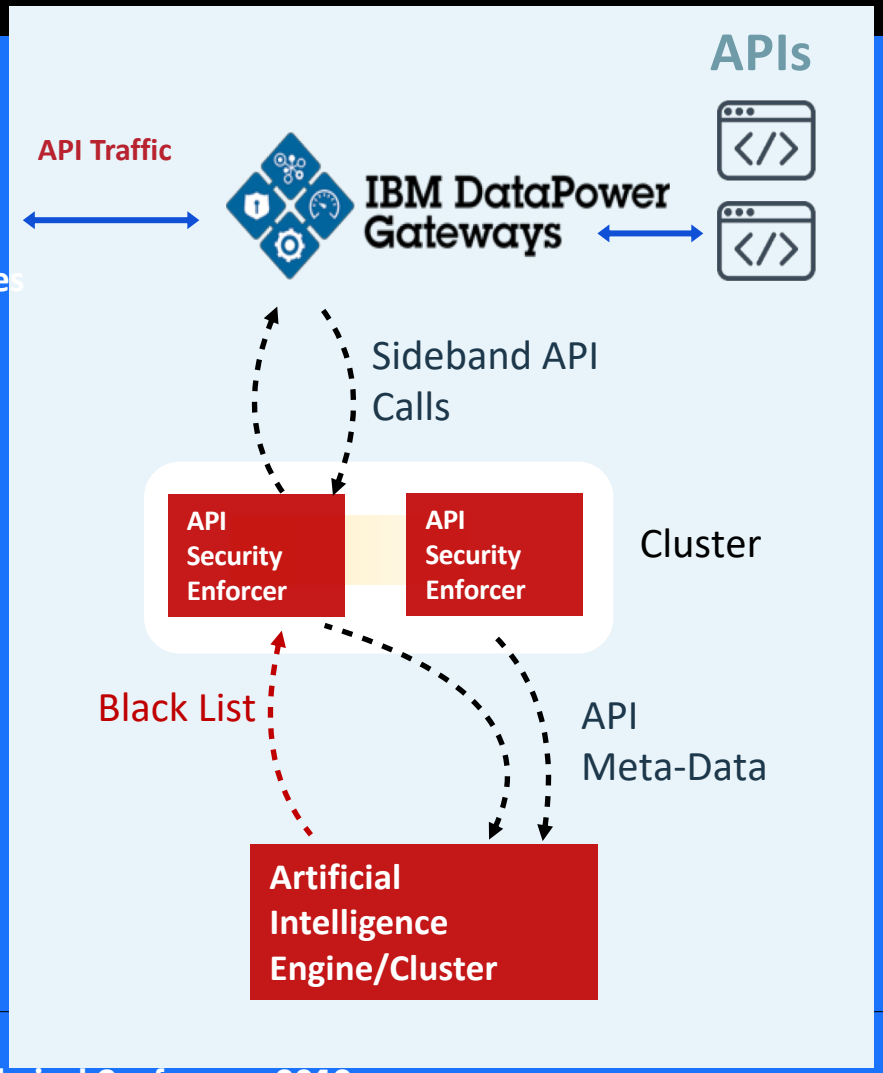
Governance implementor

1. Global policies attached to API at catalog level

- Request path
- Response path

2. API Connect provisions rules to Datapower instances

Runtime

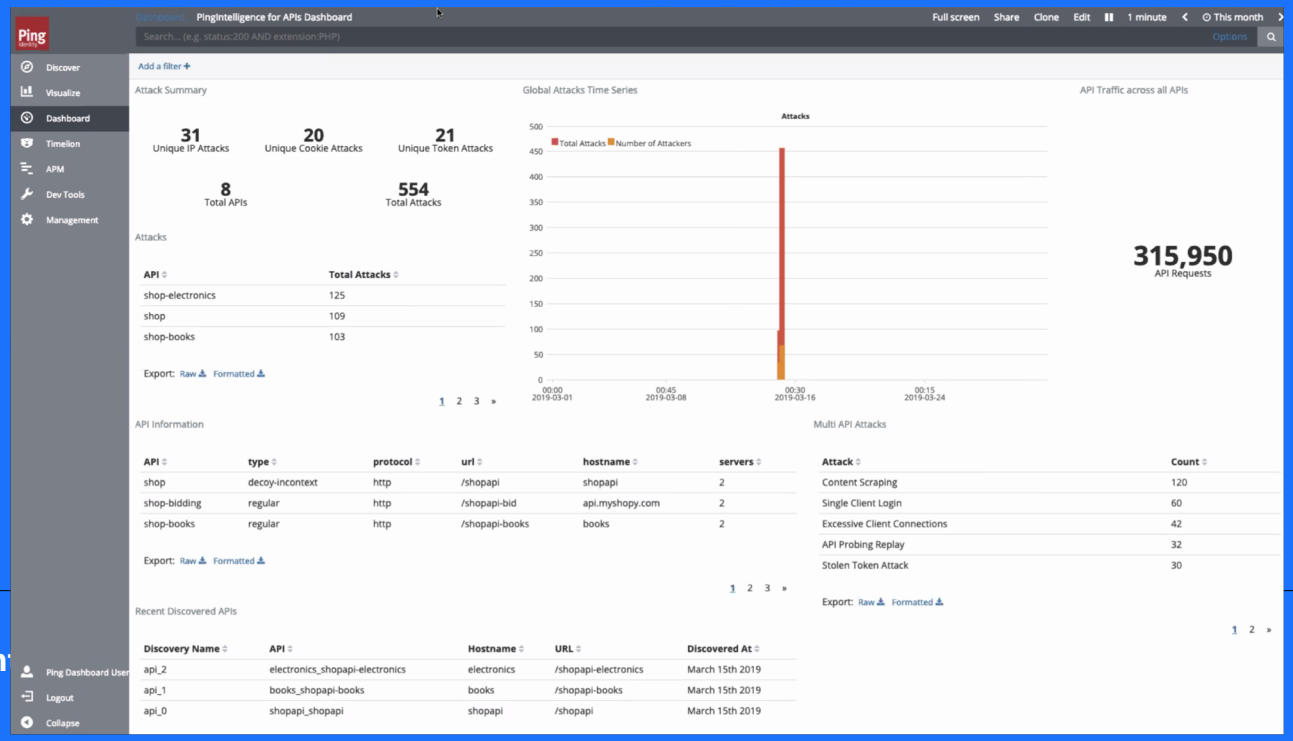


Sideband Architecture

- Datapower inline of API traffic
- Datapower makes the following calls:
  - Synchronous sideband API call to pass **Request** meta-data to **ASE** which responds with:
    - **OK** – send API request to App servers
    - **403** – block client request (or not)
  - Synch/Asynch sideband API call to pass **Response** meta-data to **ASE** which always responds with **OK**
- ASE works out-of-band with AI Engine to detect API attacks
  - **API meta-data** sent from each **ASE** node to **AI Engine** for attack detection and reporting
  - **AI Engine** updates black list when blocking is enabled

After the attack: forensics

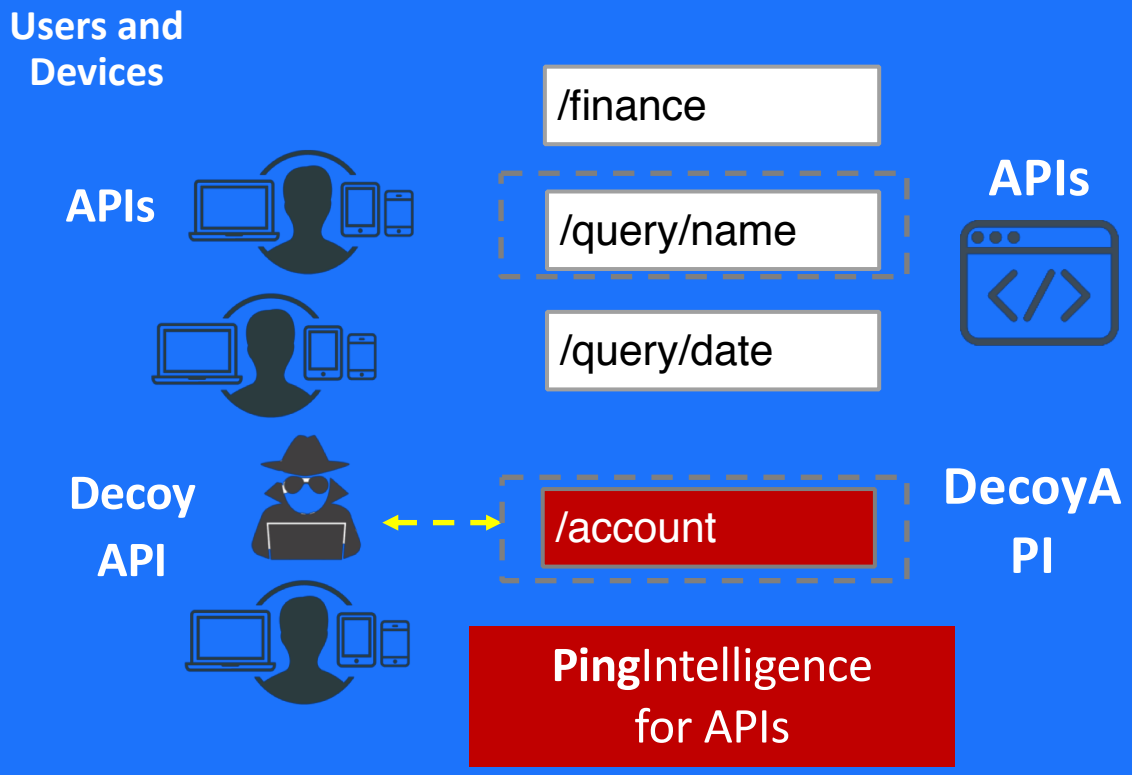
- Repair damage
- Inform affected users
- Comply with regulations




```
"details": {
    "metrics": {
      "token": "jk4278rjk98.3229.0dskjfdg6743kj",
      "total_requests": 33,
      "ip_list": [
        {
          "ip": "100.100.1.10",
          "total_requests": 27,
          ....
          },
          "methods": {
            "PUT": 27
          },
          "urls": {
            "/shopapi-bid/order": 27
          },
          "apis": {
            "shop-bidding": 27
          }
        },
        {
          "ip": "100.100.21.21",
          "total_requests": 6,
          ...
```

API Deceptions

Leverage Hacking Behaviors Against Attackers



Instant Hacking Detection

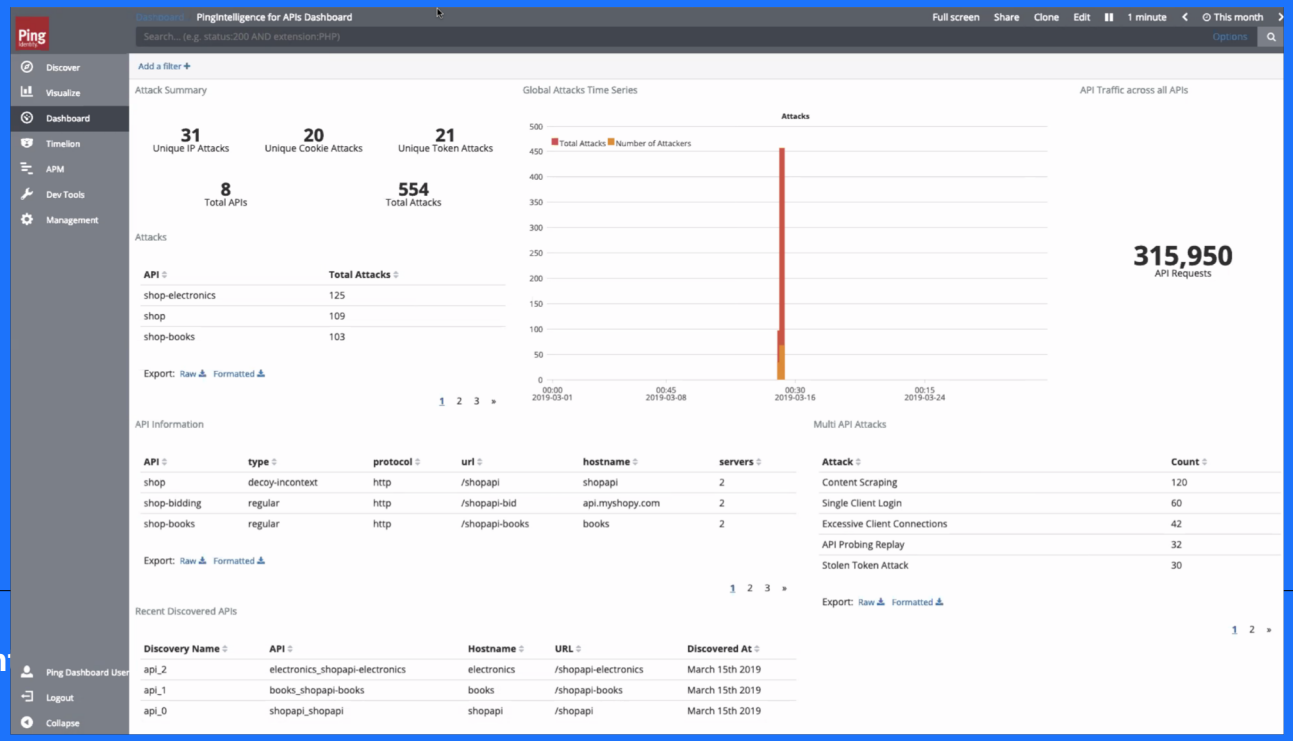
1. Decoy APIs attract probing hackers
2. Source identified instantly
3. Blocks access to production APIs

Demo

attack.stop();

After the attack: forensics

- Repair damage
- Inform affected users
- Comply with regulations




```
"details": {
    "metrics": {
      "token": "jk4278rjk98.3229.0dskjfdg6743kj",
      "total_requests": 33,
      "ip_list": [
        {
          "ip": "100.100.1.10",
          "total_requests": 27,
          ....
          },
          "methods": {
            "PUT": 27
          },
          "urls": {
            "/shopapi-bid/order": 27
          },
          "apis": {
            "shop-bidding": 27
          }
        },
        {
          "ip": "100.100.21.21",
          "total_requests": 6,
          ...
```

Comprehensive API Security with Ping and IBM

PingIntelligence for APIs

AI-powered Threat Protection for APIs

Data & Application Attacks
Advanced Persistent Threats, Data exfiltration, Deletion

DoS & DDoS Attacks
DDoS API attack, Login service DDoS attack, Botnet attacking API

Login Attacks
Stolen tokens or cookies, Credential stuffing, fuzzing,

Message Security
JSON/XML threat protection, SQL injection, XSS, Schema validation, Encryption & signature, Redaction, AV scanning

Access Control
Authentication, Authorization, Token Translation

Rate Limiting
Client throttling, Provider throttling, Quotas

Network Privacy
SSL/TLS

Kin lane, API evangelist
Evolving API Security Landscape white paper

- https://www.pingidentity.com/en/resources/client-library/white-papers/2018/evolving-api-security-landscape.html

Thank You