Enterprise API Security Choices:
One API, Total Security

Blake Dournaee
Senior Product Manager
Intel Data Center Software Division
Securing All Channels to the Enterprise

- Mobile/HTML5
- Mobile/Native
- Web Browser
- Internal Devs
- External Devs/3rd Party Apps
- Kiosk/Thick client

One API set, any channel, with Enterprise Security

“Hodgepodge” of legacy infrastructure and programming languages...

Difficult to apply consistent security, auditability, risk and compliance for API calls...

.. worse API data spans deployment models
Enterprise API Security Layers

- How to handle sensitive data in API requests & responses?
- How to audit API calls?
- Which partner applications can access APIs?
- Which employee applications can access APIs?
- How do I trust incoming API calls?
- How do I handle SSO for API calls?
- How do I keep API threats out?
- How do I scan incoming API calls for malware and virus payloads?
- How do I protect against app-level distributed Denial of Service (DoS)
- How do I protect clients from malware and content threats?

Remember! Security should be a declarative policy, not “coded in”
API Threats are Real and Now...

But... are these any different than what we saw in the web world?
APIs are a Semantic Tunnel

APIs: Unmediated Function Calls

Web App: Mediated Function Calls
Semantic Tunnel: Countermeasures

Countermeasures:
- Obfuscate, hide, or “turn off” certain methods
- Input validation
- Data-type checking*
- Data range checking
- Word scanning
- Pattern Scanning
- Avoid defaulting all data-types to ‘string’
- Schema validation (JSON/XML)

✓ Validation/Rejection by regex match
✓ Message size limitations
✓ Message structure limitations (XML)
✓ Attachment scanning/limitations

Remember! Security should be a declarative policy, not “coded in”
## Multi-Policy Service Level Management (SLM)

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Control Scope</th>
<th>Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Plans</td>
<td>Coarse Grained Business Enablement</td>
<td>Single API</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Messages per day, quota - enforced per calling application</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>Fine Grained Infrastructure Tuning</td>
<td>Single API, set of APIs, set of services</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transaction rate, data rate, latency, utilization (plus more) - enforced per API, service, or IP</td>
</tr>
<tr>
<td>Adaptive DoS</td>
<td>Fine Grained Infrastructure Protection</td>
<td>Connection &amp; IP Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rate shaping, alerting and blocking - Enforced per connection &amp; IP</td>
</tr>
</tbody>
</table>

- **Adaptive DoS**
  - Fine Grained Infrastructure Protection
  - Connection & IP Address
  - Rate shaping, alerting and blocking - Enforced per connection & IP

- **Quality of Service**
  - Fine Grained Infrastructure Tuning
  - Single API, set of APIs, set of services
  - Transaction rate, data rate, latency, utilization (plus more) - enforced per API, service, or IP

- **API Plans**
  - Coarse Grained Business Enablement
  - Single API
  - Messages per day, quota - enforced per calling application

---

*REST* (Representational State Transfer) is used for communication between Developers & Apps and the Adaptive DoS and Quality of Service Policy (QoS) systems. Data Store manages the policy enforcement mechanisms.
## New Developer AuthN Requirements

### API & Mobile Authentication Mechanisms

<table>
<thead>
<tr>
<th>Authenticating Credential</th>
<th>Secret</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Key</td>
<td>API Key</td>
</tr>
<tr>
<td>API Key</td>
<td>Shared Secret</td>
</tr>
<tr>
<td>OAuth Consumer Key</td>
<td>OAuth Consumer Secret</td>
</tr>
<tr>
<td>Username</td>
<td>Password</td>
</tr>
<tr>
<td>Username</td>
<td>One-time Password</td>
</tr>
</tbody>
</table>

### Enterprise Authentication Mechanisms

<table>
<thead>
<tr>
<th>Authenticating Credential</th>
<th>Secret</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Password</td>
</tr>
<tr>
<td>Certificate</td>
<td>Private Key</td>
</tr>
<tr>
<td>Kerberos Ticket</td>
<td>Password</td>
</tr>
<tr>
<td>SAML Assertion</td>
<td>Password or Private Key</td>
</tr>
<tr>
<td>Username</td>
<td>One-time Password</td>
</tr>
</tbody>
</table>

---

**Enterprises can’t afford another identity silo**

Consumer & BYOD

Existing Enterprise IDM systems
API Key Security

API Key Security Concern

• HTML5 apps are pushed to the client \textit{keys distributed to all clients}

• Clients can view source to obtain API keys

• **Solution #1:** Obfuscation
• **Solution #2:** Step-Up Authentication
Common Enterprise OAuth 2 Flows

**2-Legged OAuth**
- Client Credentials
- Resource Owner Password Credentials

- Requires out-of-band shared secret
- Client apps must be completely trusted
- No refresh tokens
- No user concept
- Mitigates exposure of shared secrets

**3-Legged OAuth**
- Authorization Code
- Implicit Grant

- Requires out-of-band shared secret
- Solves impersonation anti-pattern
- Client apps can be untrusted
- Supports refresh tokens
- Requires user concept
- Mitigates exposure of shared secrets
0. Client details are registered in a persistent store, including client name, shared secret, redirection URI - done with a custom web app.

1. Authorization Request received by server (HTTP GET)

2. Server Validates the request, including scope.

3. HTTP 200 OK back to the client with HTML Page.

4. HTTP POST with username/password and authorization.

5. Server validates user authentication (w/ LDAP) and authorization decision; generates authz code.

6. HTTP 302 Redirect with authorization code.

7. HTTP POST for access code request done over SSL w/ HTTP Basic Authentication, uses authz code.

8. Server validates authorization code, generates access code.

9. HTTP 200 OK response with JSON containing access code.

10. HTTP GET to Location API using access code in HTTP header.

Remember! Security should be a declarative policy, not “coded in.”
API Compliance

Regulatory Drivers
- Credit card data (PCI)
- Personally identifiable information (PII)
- Medical records (HIPAA)
- Financial data (SOX, GLBA)

Reduce assessment costs, avoid fines, protect customers & shareholders

Intellectual Property
- Source code, blue prints
- Roadmap data
- Business plans & models
- Data from M&A, marketing

Protect organization IP, assets, resources and improve competitive posture

Remember! Security should be a declarative policy, not “coded in”
<table>
<thead>
<tr>
<th>3rd Party Apps</th>
<th>Trust</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ RESTful Service</td>
<td>✓ DDoS Protection</td>
<td></td>
</tr>
<tr>
<td>✓ OAuth 2.0 – Auth Code Flow</td>
<td>✓ Content Attack Protection</td>
<td></td>
</tr>
<tr>
<td>✓ API Keys</td>
<td>✓ API Plans</td>
<td></td>
</tr>
<tr>
<td>✓ Server side SSL</td>
<td>✓ API Resource Protection</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enterprise Mobile Apps</th>
<th>Trust</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ RESTful Service</td>
<td>✓ DDoS Protection</td>
<td></td>
</tr>
<tr>
<td>✓ 2-Way SSL</td>
<td>✓ Content Attack Protection</td>
<td></td>
</tr>
<tr>
<td>✓ API Keys with Step-Up Auth</td>
<td>✓ API Plans</td>
<td></td>
</tr>
<tr>
<td>✓ 2-Legged OAuth – Client Credentials Flow</td>
<td>✓ API Resource Protection</td>
<td></td>
</tr>
<tr>
<td>✓ SSO – Use SAML 2.0 or OpenID Connect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ JSON Web Tokens</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Internal Client / Server</th>
<th>Trust</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ SOAP or REST</td>
<td>✓ API Plans</td>
<td></td>
</tr>
<tr>
<td>✓ SSL w/ Enterprise authentication</td>
<td>✓ API Resource Protection</td>
<td></td>
</tr>
<tr>
<td>✓ SSO – Use SAML 2.0 or OpenID Connect</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partner Web Services</th>
<th>Trust</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ SOAP WS-Security (X.509 or HMAC)</td>
<td>✓ DDoS Protection</td>
<td></td>
</tr>
<tr>
<td>✓ 2-Way SSL</td>
<td>✓ Content Attack Protection</td>
<td></td>
</tr>
<tr>
<td>✓ API Plans</td>
<td>✓ API Resource Protection</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Internal Apps/Developers</th>
<th>Trust</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ SOAP or REST</td>
<td>✓ API Plans</td>
<td></td>
</tr>
<tr>
<td>✓ API Keys</td>
<td>✓ API Resource Protection</td>
<td></td>
</tr>
<tr>
<td>✓ No Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ SSO – Use SAML 2.0 or OpenID Connect</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
“New” 3-Tier Architecture

HTML5 & Native Apps

• Clients from any channel
• Synchronous or socket communication
• Transport Level Security with session authentication

API Governance, Management and Security

• API Governance Layer – handles scalability, security, and security decoupling
  • Single point of audit, security control and compliance

Delivery & Governance Tier

• API – The backbone of the architecture
• Surface data from any system

Data Service Layer

• APIs – The backbone of the architecture

Persistence Tier

• Logical persistence tier, data may be stored in RDMS or NOSQL stores

Enterprise IDM

The API Governance Layer helps mitigate API trust and threat concerns in the application architecture
Intel® Expressway API Manager

- API Sharing Portal
- Mobile Middleware
- Apps

Security & Integration

- Best of Breed API Management
- On-prem control
- Enterprise Security & Integration
- Multi-Community API Sharing

Cloud API Management

- API packaging as products
- Reporting and analytics for usage and latency
- Share meta data via portal

App Service Gov & Integration

- Lightweight ESB
- SOA & Mobile integration
- Orchestrate & transform
- Protocol translation
- Eclipse workflow design

Security, Access, Compliance

- App & mobile firewall
- Data Loss Protection
- Federated ID Brokering
- PCI PII Data Tokenization
- Mobile friendly OAuth

Developer Community

- Developer facing services catalog, developer enablement
- Developer on-boarding
- Discovery of aggregated services from IT
API Resource Center: cloudsecurity.intel.com

Solution Brief: Internal APIs

Use Case Video

Forrester SOA to API Webinar