Customer Showcase:

How Dow Jones Uses AWS to Create a Secure Perimeter Around Its Web Properties

Kamal Verma, Dow Jones
Shawn Marck, AWS

Wednesday, January 23, 2019
Dow Jones brings together world-leading data, media, membership and intelligence solutions to power the most ambitious companies and professionals

https://www.dowjones.com  https://dowjones.jobs
What you will learn today

- Find out how to automate security on AWS
- How a small team can control cloud security
- Useful operational processes and controls
Project Saturn

Secure Software Environment v1.1

Icon by Freepik from www.flaticon.com
Secure perimeter

1. Edge proxy the outside
2. Inside serves edge only
3. Edge rules on who can get in
4. Protects at Layer-3
5. Protects at Layer-7
6. Creates a green zone
7. Intrusion detection or deny access
8. Object caching on the edge
9. Early SSL termination
10. AWS Shield Advanced

CloudFront Edge Network
# WAF: Application Security

<table>
<thead>
<tr>
<th>OWASP Top 10 - 2013</th>
<th>OWASP Top 10 - 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A1 – Injection</strong></td>
<td><strong>A1:2017-Injection</strong></td>
</tr>
<tr>
<td><strong>A2 – Broken Authentication and Session Management</strong></td>
<td><strong>A2:2017-Broken Authentication</strong></td>
</tr>
<tr>
<td><strong>A3 – Cross-Site Scripting (XSS)</strong></td>
<td><strong>A3:2017-Sensitive Data Exposure</strong></td>
</tr>
<tr>
<td><strong>A4 – Insecure Direct Object References [Merged+A7]</strong></td>
<td><strong>A4:2017-XML External Entities (XXE) [NEW]</strong></td>
</tr>
<tr>
<td><strong>A5 – Security Misconfiguration</strong></td>
<td><strong>A5:2017-Broken Access Control [Merged]</strong></td>
</tr>
<tr>
<td><strong>A6 – Sensitive Data Exposure</strong></td>
<td><strong>A6:2017-Security Misconfiguration</strong></td>
</tr>
<tr>
<td><strong>A7 – Missing Function Level Access Contr [Merged+A4]</strong></td>
<td><strong>A7:2017-Cross-Site Scripting (XSS)</strong></td>
</tr>
<tr>
<td><strong>A8 – Cross-Site Request Forgery (CSRF)</strong></td>
<td><strong>A8:2017-Insecure Deserialization [NEW, Community]</strong></td>
</tr>
<tr>
<td><strong>A9 – Using Components with Known Vulnerabilities</strong></td>
<td><strong>A9:2017-Using Components with Known Vulnerabilities</strong></td>
</tr>
<tr>
<td><strong>A10 – Unvalidated Redirects and Forwards</strong></td>
<td><strong>A10:2017-Insufficient Logging&amp;Monitoring [NEW,Comm.]</strong></td>
</tr>
</tbody>
</table>

*Image by www.owasp.org*
Alarms and integration with Slack

**ALERT: www.example.com** APP 11:03 PM
@channel Previous alarm of high error rate is cleared now. OK
ok #test is notified.
Reason: Threshold Crossed: 1 datapoint [0.0 (03/02/18 04:01:00)] was not greater than or equal to the threshold (80.0).

**ALERT: www.example.com** APP 12:01 AM
@channel More than 80% of requests are returning 4XX or 5XX over period of two minutes. !

**ALERT: www.example.com** APP 7:11 AM
@channel Previous alarm of high error rate is cleared now. OK
Application pattern: legacy
Application pattern: poor communication hygiene
Application pattern: encrypted in transit

browser

1. TLS

Amazon CloudFront

2. TLS

Application Load Balancer

3. TLS

Amazon EC2
Application pattern: serverless SPA
Application pattern: serverless SPA
Recommendations

- Measure performance
- Create a pattern
- Automate
- Serverless
  - On the edge
  - Backend processes
What’s next? (version 2.0)

1. Automatic back-off
2. JWT verification on the edge
3. Honey-pot integration
4. Bot and content scraping detection
5. Integrate more alarms, like peak detection
Q&A
Thank You!
AWS Services

- Amazon CloudFront
- AWS Lambda
- Amazon Route 53
- Amazon Athena
- AWS Certificate Manager
- Amazon CloudWatch
- AWS Firewall Manager
- AWS Shield
- Application Load Balancer
- AWS WAF
- Amazon S3