Securosis

Presents

Cloud-Sec 12 Step

Adrian Lane
Securosis, LLC

Outline

- Cloud Overview
- The '12 Steps'
- Recommendations
What ‘It’ Is

- Abstraction of Infrastructure
- Elastic and Dynamic Resource
- Resource Democratization
- Services Oriented Architecture
- Utility Model of Consumption & Allocation
What ‘It’ Is Not

- It’s not mainframe, ‘cloud in a box’ nor virtualization
Cloud Service Models

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

*Graphics courtesy Chris Hoff - Rational Survivability

---

Infrastructure as a Service

- Provider supplied network, computing and storage.
- Customer deploys & configures software
- Amazon EC2, Rackspace, GoGrid
Platform as a Service

- Provider supplied integration & middleware, abstracted resources
- Database, IAM, Messaging & Mgmt APIs
- Force.com, Azure, Database.com, Google AppEngine

Software as a Service

- Provider supplies application and abstracted resources
- Consumer supplies data
- Salesforce.com, Google Apps
Security Implications

- Variable control
- Variable visibility
- Variable resource availability
- Variable simplicity

Which Means ... Security is different than traditional IT

- Can’t test in the same way
- Vulnerabilities to data are different
- Access to logs is limited
- Security models are different
- Limited by what you can deploy
Focus

- PaaS & IaaS (This is OWASP & you're not programming SaaS environments)
- Development oriented view
- EC2: Easy access, mature API, developer friendly.,
  SOA, Cheap. Not an endorsement, it's just easier this way.
- Public or hybrid deployments

A 12 Step Program -
For Cloud Security
Application Security

What We’ll Cover

- Identity and Access
- Code Analysis
- Deployment & Configuration
- Design
- Architecture
- Testing
- Threat Modeling
Application Security & SDLC

1. Architecture

- SOA by design & functional segmentation
- IAM & federation scheme
- Threat/Risk Analysis
Trust Boundaries

Traditional

IaaS

PaaS

Design for Threats

Example Mapping Threat Model to Countermeasures

<table>
<thead>
<tr>
<th>Threat</th>
<th>Security Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
<td>Authentication</td>
</tr>
<tr>
<td>Tampering</td>
<td>Digital Signature, Hash</td>
</tr>
<tr>
<td>Repudiation</td>
<td>Audit Logging</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td>Encryption</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>Availability</td>
</tr>
<tr>
<td>Elevation of privilege</td>
<td>Authorization</td>
</tr>
</tbody>
</table>
App-Dev Lifecycle
Security Issues

- App Stack control
- What you can test
- Instance validation
- Vendor API's and abstraction
- Multi-tenancy
- Different threat models

App-Dev Lifecycle
Security Issues Cont.

- Threat models are different as your zones of control are different
- Geared for SOA (Interfaces, IAM, IPC)
- Auditing
- Elasticity~dynamism
- Your ‘provider’ and service levels
3. Identity & Authorization

- Registration
- Propagation
- User Management
- De-provisioning
- Audit

Identity
Access

- Access Control Policies
- Authentication
- Authorization
- Audit

Identity Management Concerns

- Vendor supplied system?
- In-cloud or hybrid?
- Do you really want to pass credentials?
- SSO? Then plan on federated identity
Options

- SAML - Emerging standard for federated identity
- OAuth - Token based authentication - avoid passing user/password
- WS-Federation - Integrates with WS-* stacks
- Open ID - Webapp only - not really secure

4. Code Analysis

Static Analysis

- IDE Integration
- Prioritization
- Code Coverage
- VA
- DAST Integration
- Platform Neutral
5. Testing

- Leverage private cloud (Dev vs Prod)
- Fuzzing
- Dynamic Analysis
- Pen Testing
- SLA's

6. App Protection

- WAF - Network config, deployment models, provider
- Honeypots - Stack embedded solutions
- Firewalls - Inherent to provider/deployment
<table>
<thead>
<tr>
<th>Year</th>
<th>Developers</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>CGI, PERL</td>
<td>Network firewalls, SSL</td>
</tr>
<tr>
<td>1997</td>
<td>ASP, JSP</td>
<td>Network firewalls, SSL</td>
</tr>
<tr>
<td>1998</td>
<td>EJB, J2EE, DCOM</td>
<td>Network firewalls, SSL</td>
</tr>
<tr>
<td>1999</td>
<td>SOAP, XML</td>
<td>Network firewalls, SSL</td>
</tr>
<tr>
<td>2001</td>
<td>Rest, SOA</td>
<td>Network firewalls, SSL</td>
</tr>
<tr>
<td>2003</td>
<td>Web 2.0</td>
<td>Network firewalls, SSL</td>
</tr>
<tr>
<td>2007</td>
<td>Cloud Computing</td>
<td>Network firewalls, SSL</td>
</tr>
</tbody>
</table>

This? Not so much.
You’re not worried about backup tapes and missing hard drives

You are worried about snapshots of entire system going public

7. Encryption
• Moving data into the cloud
• Enforcing rights management
• Securing stored data
• Crypto shredding

Encrypting Application Data

• Application Layer - more work but cloud independent
• TDE - Transparent Database Encryption
• Volume - simple archive security

Encryption
8. Key Management

- Keys and certificates are basic authentication - not user name and passwords
- Easy to leave key in image
- Local or hybrid key server
Monitoring Technologies

• WAF
• DAM
• FAM
• DLP
• Service Monitors

What's provided by vendor?
What deployment model?

Web / App Plug-in (WAF, DLP, DAM)

IaaS Load Balancer (WAF)

IaaS - Agent (All)

SaaS - DNS Redirect (WAF, DLP)

IaaS - Pre-built AMI (All)
Infrastructure Security

- Trusted Images
- Patching
- Configuration

10. Deployment Stack Hardening
Trusted Images

10. Deployment Stack Hardening

Advanced Instance Options

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable monitoring or enter data that will be available from your instances once they launch.

Kernel ID: [Use Default]  
RAM Disk ID: [Use Default]

Monitoring: 
Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data:
https://s3.amazonaws.com/YourCompany/YourDrive/YourSpecialInitScript.txt

as text

as file: base64 encoded

Termination Protection:
Prevention against accidental termination.

Shutdown Behavior:
Stop: Choose the behavior when the instance is shutdown from within the instance.
Patching
10. Deployment Stack Hardening

Configuration
10. Deployment Stack Hardening
- Security zones are your basic security control
- You define access points and data flow
- Provider may not allow full visibility to networks
- Provider shields you from other customers

11. Security Zones
Cost Considerations

- Inexpensive to design for scale
- Network expense zero - unless you cross physical zones.
- Shut down what you don’t use!

Provider Issues

- API’s and lock-in - abstraction more important
- Secure AMI’s
- Network addressing and availability zones
- Lack of logs in multi-tenant environments
- SOD and admin access
Cloud-Sec 12-Step

Adrian Lane
Securosis, LLC
@adrianlane
alane@securosis.com