Threat Modeling Cloud Applications
What You Don’t Know Will Hurt You

Scott Matsumoto
Principal Consultant
smatsumoto@cigital.com

www.cigital.com
info@cigital.com
+1.703.404.9293
Agenda

- Cloud Terminology and Background
- Threat Modeling Basics
- Threat Modeling a Hybrid, IaaS Application
  - Canonical use case for S3
  - AWS Security Credentials
  - EC2 Security Groups
  - S3 Security Controls
  - Cloud Doomsday Scenarios and Attackers
Terminology and Concepts

CLOUD COMPUTING
NIST Cloud Definition Framework

Visual Model Of NIST Working Definition Of Cloud Computing
http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

- Broad Network Access
- Rapid Elasticity
- Measured Service
- On-Demand Self-Service

Resource Pooling

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

Essential Characteristics

Service Models

Deployment Models

Public
Private
Hybrid
Community
Cloud Applications Are Subtly Different

- Cloud platforms (PaaS and IaaS) change application design as designers leverage platform strengths.

- Security for applications written on these platforms requires understanding the application architectures emerging from these designs and identifying their inherent weakness.

- Threat Modeling is an effective method for understanding how/where/what security implications arise from cloud-based applications.
Security Design Analysis

THREAT MODELING BASICS
What is a Threat Model

- A model of the a software system that depicts
  - The system structure: its components and the flow of control relationships
  - The assets (data and function) in the system
  - The security controls protecting the assets
- This model of the system is juxtaposed against
  - A list of potential "Doomsday Scenarios"
  - A list of potential attackers
Use Threat Modeling to Identify...

- Where potential attackers exist relative to the architecture
  - How attackers escalate privilege
    - ...become more formidable
  - Specific vectors of attack
- Components and assets needing additional protection
  - Ties technical risk & business assets to application design;
  - Ties attacks to role, privilege, and capability;
  - Drives security analysis, testing.
Elements of a Threat Model

- System Structure
- Assets
- Security Controls
- Doomsday scenarios
- Attackers
Threat Modeling – High-level process

1. Diagram the System Structure
2. Identify Assets and Security Controls
3. Enumerate Doomsday Scenarios
4. Identify Attackers
5. Derive misuse/abuse cases
6. Integrate with Risk Management
7. Iterate
Use Case: Leveraging S3 Storage

A HYBRID, IAAS THREAT MODEL
Using S3 Storage Use Case

- **Use Case:**
  - Use S3 Storage for long term storage rather than self hosted storage
  - Data items are large and unstructured
  - Require immediate access

- **S3 Advantages:**
  - No up-front capital expenditure
  - Disaster Recovery is built into the S3 service

- **Examples:**
  - Medical Images
  - Large media files
The traditional solution is having a Disaster Recovery site that is a mirror of the primary site.

Data replication is needed for persistent data.

- Pay for un-used capacity even for DR.
Cloud Architecture: Augment DR with AWS

- Provide immediate, limited DR capabilities
- Maintain data needing 99.99 availability in S3
- Pay only for the storage that's needed
It's Really a New Application

Original App
- Traditional enterprise application (assume n-Tier for this example)

New App
- Multiple applications interacting across two network zones
Threat Modeling – High-level process

1. Diagram the System Structure
2. Identify Assets and Security Controls

3. Enumerate Doomsday Scenarios
4. Identify Attackers

5. Derive Misuse/Abuse Cases
6. Integrate with Risk Management
7. Iterate
For an n-Tier application, the canonical OWASP-ish threat model applies
System structure reflects the AWS framework
### Threat Modeling Process

- The Threat Modeling Process Builds a sparse matrix
- Start with the obvious and derive the interesting
  - Postulate what bad things can happen without knowing "How".
  - Postulate “Hows” without knowing “Whats”

<table>
<thead>
<tr>
<th>Who</th>
<th>What</th>
<th>How</th>
<th>Impact</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;external&gt;</td>
<td></td>
<td>Web-application …</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;external&gt;</td>
<td></td>
<td>Multi-tenant res,…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;internal&gt; &amp; &lt;external&gt;</td>
<td>Disclosure of PCI data from the database</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;external&gt;</td>
<td>Gaining access to administrative functions</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Threat Modeling – High-level process

1. Diagram the System Structure
2. Identify Assets and Security Controls
3. Enumerate Doomsday Scenarios
4. Identify Attackers
5. Derive Misuse/Abuse Cases
6. Integrate with Risk Management
7. Iterate
Identify the Assets and Security Controls

- Data assets move with the new design
- Additional functional assets exist with new features
- The AWS Security Controls are different
- AWS and the Internet are equivalent network zones; user AWS Security Groups
- Enterprise infrastructure, e.g. SiteMinder, probably won't extend into AWS. What is the replacement?
EC2 Security Groups

- An EC2 Security Group is a set of ACCEPT firewall rules
  - Protocol: tcp, udp, icmp
  - Port Range
  - From:
    - Set of IP addresses (generally external hosts)
    - Security Group

- An EC2 instance can reside in one or more Security Groups
  - Use a Security Group is a "role"
  - Associate permissions with the Security Group ("role")
Integration with Enterprise Authentication

- Stand alone application mechanism means that the user store must be provisioned

- Integration with the enterprise user store implies
  - Connection from AWS back into the data center
  - Federated Identity mechanism

- The Threat Model depends on the actual control

- For this particular example, assume a SAML assertion passed through the browser
Elasticity Drives Change

- Will the During DR site be up 100% of the time and costing the company for CPU time? No.
- An EC2 Key Pair is required to launch an instance
- AWS Access Keys are required to access S3
# Most Common AWS Security Credentials

<table>
<thead>
<tr>
<th>Type</th>
<th>Usage</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign-In Credentials</td>
<td>Enter email-address and password to access secure pages</td>
<td>Access AWS Security Credentials Page</td>
</tr>
<tr>
<td>User</td>
<td>Use AWS IAM API or interface</td>
<td>Authentication and Authorization for AWS Management Console and AWS Credentials</td>
</tr>
<tr>
<td>Access Keys</td>
<td><strong>Access Key ID</strong> identifies your AWS Account&lt;br&gt;<strong>Secret Access Key</strong> is used to digitally sign the request</td>
<td>AWS SOAP and REST API requests</td>
</tr>
<tr>
<td>Key Pairs</td>
<td>The <strong>Key pair name</strong> is specified when an instance is launched. &lt;br&gt;The Public-Private key is used for SSH root access.</td>
<td>Admin access to the running instance</td>
</tr>
</tbody>
</table>

- **Authorization** is handled through the **Access Policy Language**
**S3 ACLs and Bucket Policies**

- Buckets and Objects have separate ACLs or Policies
- User identity is an Amazon S3 user/account
- Policies are more flexible and expressive
  - Define access rules for sets of object
  - Restrict by IP address, date, etc.
Using S3 Drives Design Changes

- Deliver content directly from S3 to the user
  - More efficient bandwidth usage
  - How do you handle S3 ACL or Bucket Policy?
- S3 provides for "query string authentication"
  - A time limited URL signed with your Access Key
Threat Modeling – High-level process

1. Diagram the System Structure
2. Identify Assets and Security Controls

3. Enumerate Doomsday Scenarios
4. Identify Attackers

5. Derive Misuse/Abuse Cases
6. Integrate with Risk Management
7. Iterate
Cloud "Doomsday" Scenarios to Consider

Reprioritized or Changed by Cloud
- Malicious Insider
- Data In Transit
- Management interface compromise
- Infrastructure supply chain stability
- DDoS - direct attacks and attacks against other tenants

Unique to Cloud
- Cloud termination
- Changes in jurisdiction
- Subpoena and e-Discovery of another tenant
- Multi-tenant violation of isolation
Threat Modeling – High-level process

1. Diagram the System Structure
2. Identify Assets and Security Controls
3. Enumerate Doomsday Scenarios
4. Identify Attackers
5. Derive Misuse/Abuse Cases
6. Integrate with Risk Management
7. Iterate
Additional Attackers

- Additional attackers are network, AWS Admin and malicious instances
- The multi-document shape indicates multi-tenant

Is the AWS Admin someone to worry about?
Threat Modeling – High-level process

1. Diagram the System Structure
2. Identify Assets and Security Controls
3. Enumerate Doomsday Scenarios
4. Identify Attackers
5. Derive Misuse/Abuse Cases
6. Integrate with Risk Management
7. Iterate
# Enumeration and Risk Management

<table>
<thead>
<tr>
<th>Who</th>
<th>What</th>
<th>How</th>
<th>Impact</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hacker</td>
<td>Read all stored data</td>
<td>Web-application</td>
<td>Failure to certify with HIPAA audit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access a patient's data</td>
<td></td>
<td>Failure to certify with PCI audit</td>
<td></td>
</tr>
<tr>
<td>Hacker</td>
<td>Cause system to ..</td>
<td>Known Tomcat,..</td>
<td>Failure to comply with customer SLA</td>
<td></td>
</tr>
<tr>
<td>Admin &amp; Hacker</td>
<td>Disclosure of PCI ...</td>
<td>Access DB cred</td>
<td>Failure to certify with PCI audit</td>
<td></td>
</tr>
<tr>
<td>Hacker</td>
<td>Gaining access to...</td>
<td>Intercept AWS cred</td>
<td>Breach of all application assets</td>
<td></td>
</tr>
<tr>
<td>Admin, Staff &amp; Hacker</td>
<td>Viewing patient inf...</td>
<td>Direct access to...</td>
<td>Failure to certify with HIPAA audit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Failure to comply with customer SLA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Failure to certify with PCI audit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Failure to certify with HIPAA audit</td>
<td></td>
</tr>
</tbody>
</table>

- Risk management must be done in conjunction with the business
Conclusion

- Cloud application security is platform specific
  - Application design will exploit platform features and constraints
  - Platform security controls are an important consideration in the threat model

- Threat Modeling is an effective way to move from cloud security FUD to a specific set of technical security requirements for applications
Thank you for your time

Questions?

Scott Matsumoto
smatsumoto@cigital.com