Securosis

Presents

Cloud-Sec 12 Step

Adrian Lane
Securosis, LLC

What 'It' Is

- Abstraction of Infrastructure
- Elastic and Dynamic Resource
- Resource Democratization
- Services Oriented Architecture
- Utility Model of Consumption & Allocation

NIST Model of Cloud Computing

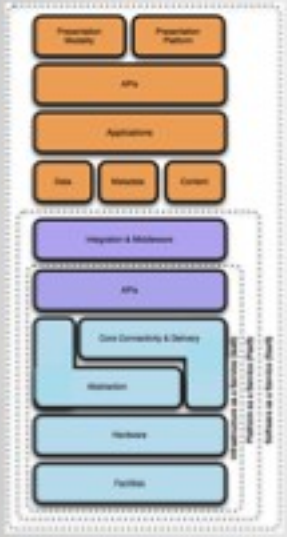
Visual Model Of NIST Working Definition Of Cloud Computing
http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

What 'It' Is Not

- It's not mainframe, 'cloud in a box' nor virtualization

Securosis

Cloud Service Models

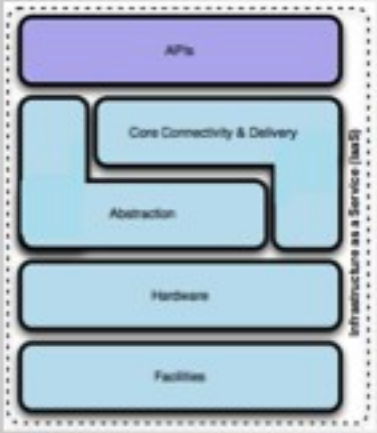


- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

*Graphics courtesy Chris Hoff - Rational Survivability

Securosis

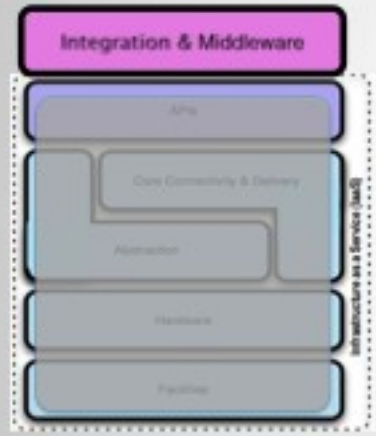
Infrastructure as a Service



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

Securosis

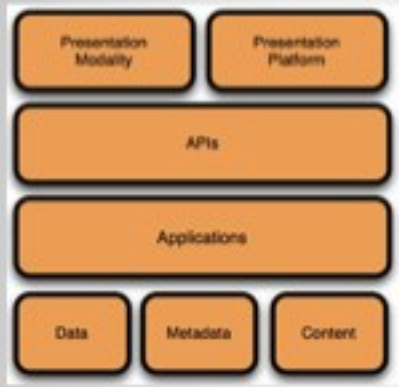
Platform as a Service



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

Securosis

Software as a Service



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

Securosis

Security Implications

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

Securosis

Which Means ...

Security is different that 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

Securosis

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

Securosis

A 12 Step Program -

For Cloud Security

Securosis


Application
Security
Securosis


What We'll Cover
Identity and Access
Code Analysis
Deployment & Configuration
Design
Architecture
Testing
Threat Modeling
Securosis

Application Security & SDLC

Securosis

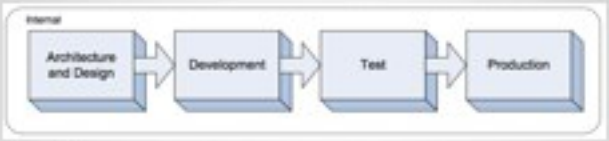
- SOA by design & functional segmentation
- IAM & federation scheme
- Threat/Risk Analysis

1. Architecture

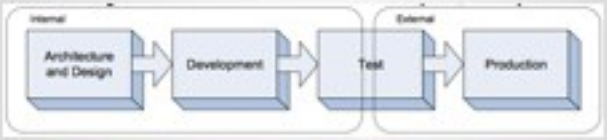
Securosis

Trust Boundaries

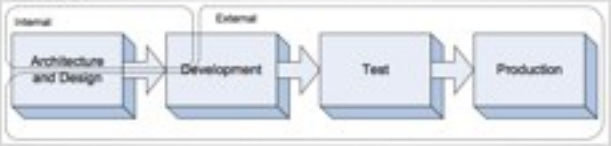
Traditional



IaaS



PaaS



Securosis

Design for Threats

Example Mapping Threat Model to Countermeasures

Threat	Security Service
Spoofing	Authentication
Tampering	Digital Signature, Hash
Repudiation	Audit Logging
Information Disclosure	Encryption
Denial of Service	Availability
Elevation of privilege	Authorization

Securosis

App-Dev Lifecycle Security Issues

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

Securosis

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

Securosis

3. Identity & Authorization

Securosis

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

Identity

Securosis

- Access Control Policies
- Authentication
- Authorization
- Audit

Access

Securosis

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

Identity Management Concerns

Securosis

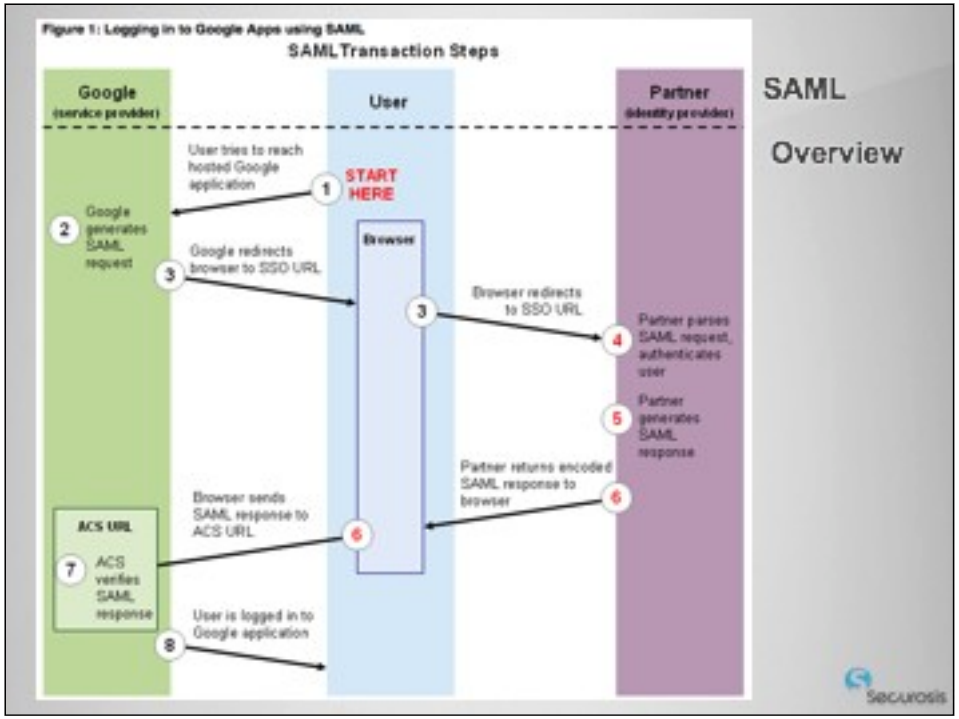
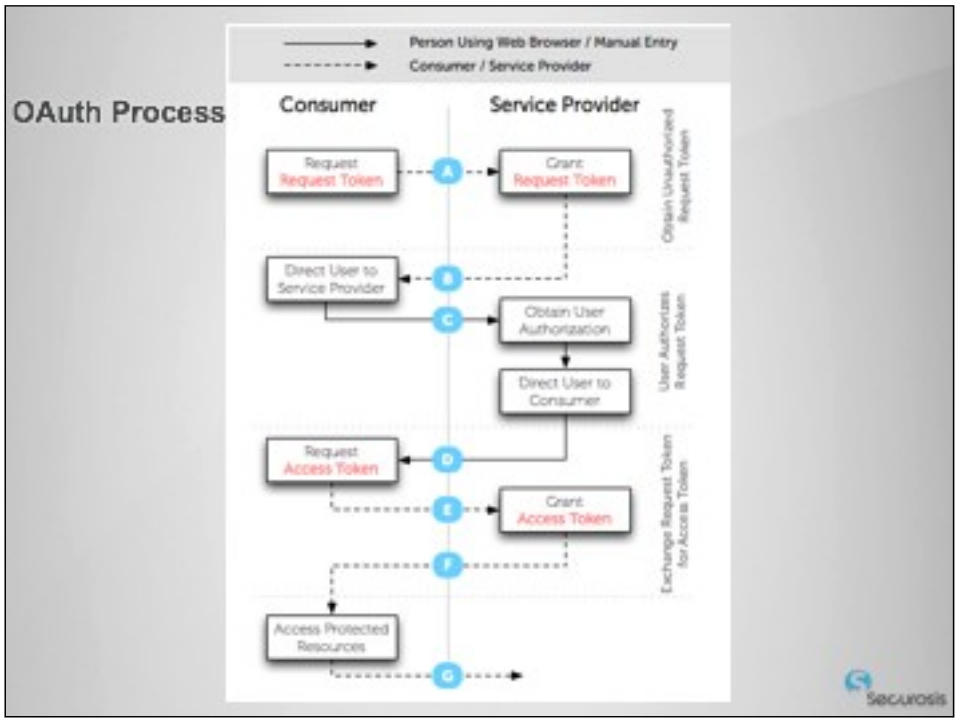


Figure 1: Logging in to Google Apps using SAML
SAML Transaction Steps
Google
(service provider)
User
Partner
(identity provider)
SAML
Overview
User tries to reach hosted Google application
1
START HERE
2
Google generates SAML request
Google redirects browser to SSO URL
3
Browser
3
Browser redirects to SSO URL
4
Partner parses SAML request, authenticates user
5
Partner generates SAML response
Partner returns encoded SAML response to browser
6
Browser sends SAML response to ACS URL
6
ACS URL
7
ACS verifies SAML response
User is logged in to Google application
8
Securosis


OAuth Process
Person Using Web Browser / Manual Entry
Consumer / Service Provider
Consumer
Service Provider
Request Request Token
A
Grant Request Token
Obtain Unauthorized Request Token
Direct User to Service Provider
B
C
Obtain User Authorization
Direct User to Consumer
User Authorizes Request Token
Request Access Token
D
E
Grant Access Token
F
Exchange Request Token for Access Token
Access Protected Resources
G
Securosis

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

Securosis

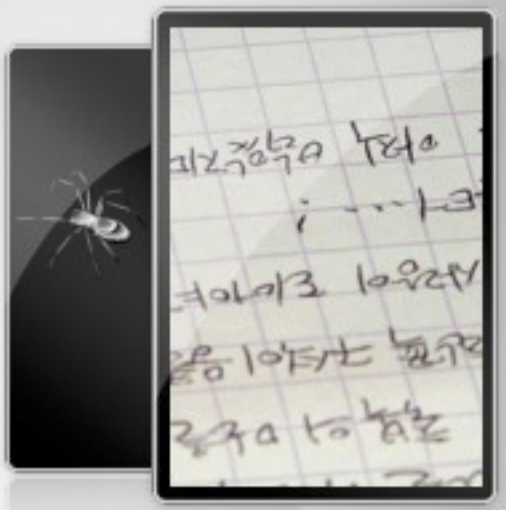
4. Code Analysis

Static Analysis

- IDE Integration
- Prioritization
- Code Coverage
- VA
- DAST Integration
- Platform Neutral

Securosis

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



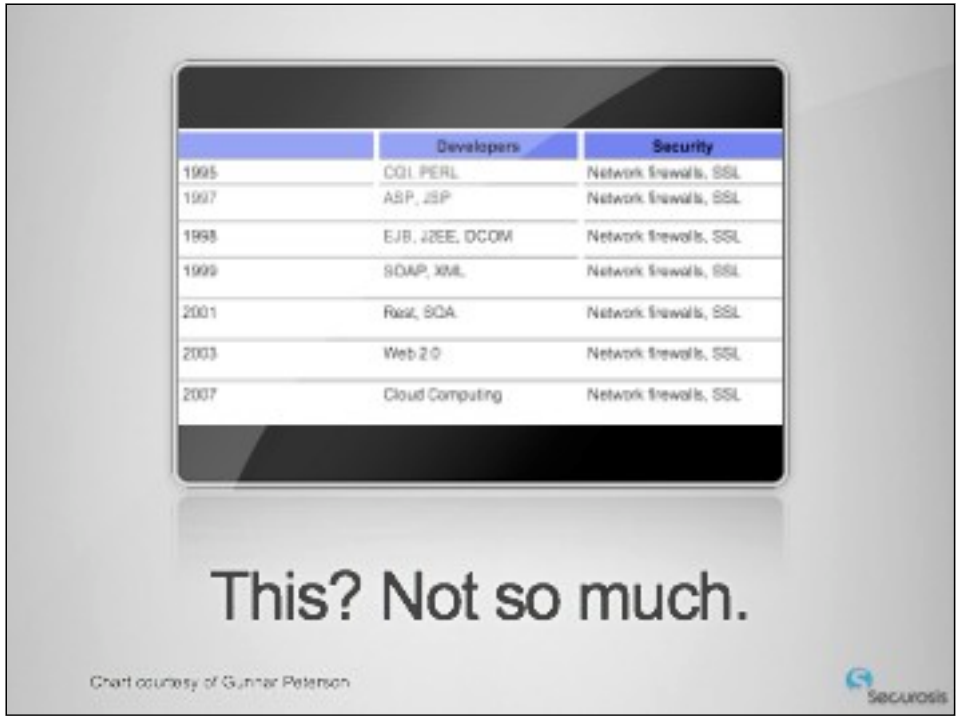
5. Testing

Securosis

6. App Protection

- WAF - Network config, deployment models, provider
- Honeypots - Stack embedded solutions
- Firewalls - Inherent to provider/deployment

Securosis


Developers
Security
1995
CGI, PERL
Network firewalls, SSL
1997
ASP, JSP
Network firewalls, SSL
1998
EJB, J2EE, DCOM
Network firewalls, SSL
1999
SOAP, XML
Network firewalls, SSL
2001
Rest, SOA
Network firewalls, SSL
2003
Web 2.0
Network firewalls, SSL
2007
Cloud Computing
Network firewalls, SSL
This? Not so much.
Chart courtesy of Gunnar Peterson
Securosis


Information Security
Securosis

Data Security Lifecycle

Securosis

- You're not worried about backup tapes and missing hard drives
- You are worried about snapshots of entire system going public

7. Encryption

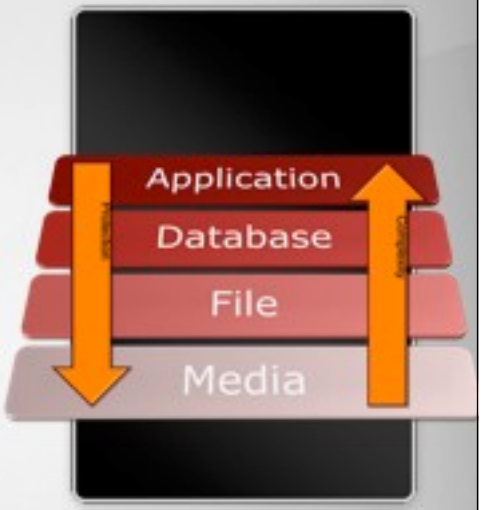
Securosis

- Moving data into the cloud
- Enforcing rights management
- Securing stored data
- Crypto shredding

Encrypting Application Data

Securosis

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



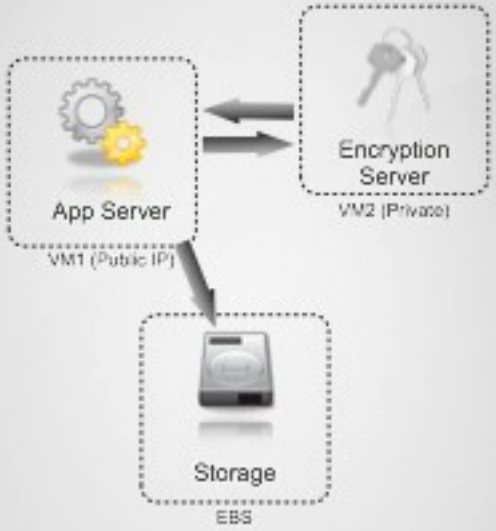
Encryption

Securosis

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

8. Key Management

Securosis



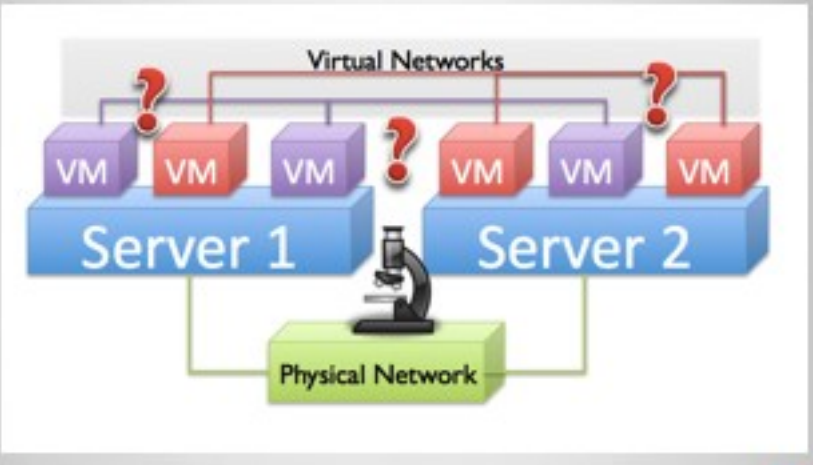
Application Layer Encryption

Securosis

Monitoring

Securosis

Loss of Network Visibility



Securosis

Monitoring Technologies

- WAF
- DAM
- FAM
- DLP
- Service Monitors

What's provided by vendor?

What deployment model?

Securosis

IaaS Load Balancer (WAF)

SaaS - DNS Redirect (WAF, DLP)

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

IaaS - Agent (All)

IaaS - Pre-built AMI (All)

Securosis

Infrastructure Security

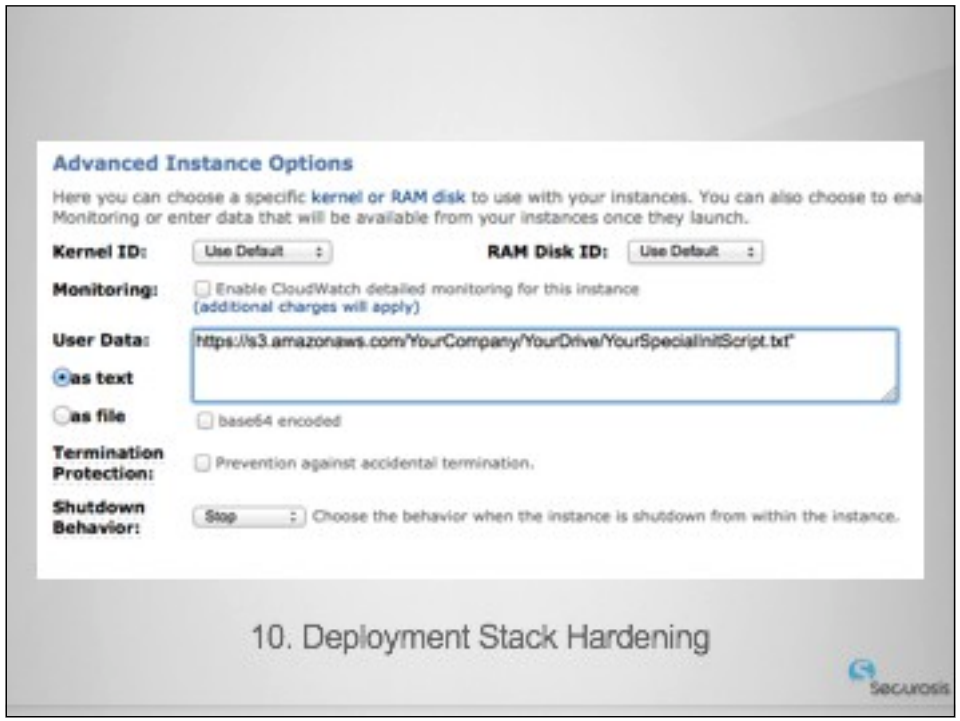
- Trusted Images
- Patching
- Configuration

10. Deployment Stack Hardening

Securosis

Trusted Images

10. Deployment Stack Hardening



10. Deployment Stack Hardening

Patching

10. Deployment Stack Hardening

Securosis

```
#Your Special cloud-config Script

fixrouting:
 - &fix_routing |
   public_ipv4=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
   ifconfig eth0:0 $public_ipv4 up

configmysql:
 - &config_mysql |
   /etc/init.d/mysql stop
   mkdir /encrypted
   /etc/init.d/apparmor teardown
   cd /etc/apparmor.d
   rm /etc/apparmor.d/usr.sbin.mysqld
   wget https://s3.amazonaws.com/YourCo/YourDrive/usr.sbin.mysqld
   cd /etc/mysql
   wget https://s3.amazonaws.com/YourCo/YourDrive/ccak-mysql-startup
   mv /etc/mysql/my.cnf /etc/mysql/my.cnf.bak
   wget https://s3.amazonaws.com/YourCo/YourDrive/my.cnf
   chmod 644 /etc/mysql/my.cnf
   /etc/init.d/apparmor start
   rm /etc/update-motd.d/*
   cd /etc/update-motd.d
   wget https://s3.amazonaws.com/YourCo/YourDrive/00-header

runcmd:
 - [ sh, -c, "fix_routing" ]
 - [ sh, -c, "config_mysql" ]
```

Configuration

10. Deployment Stack Hardening

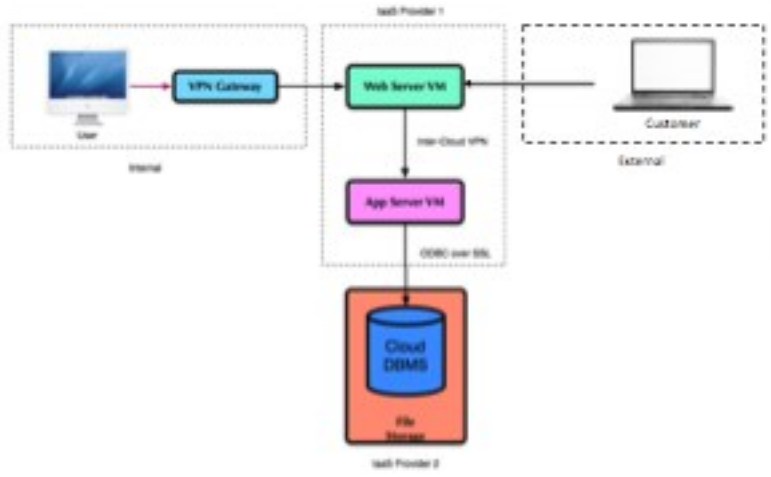
Securosis

- 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

Securosis

Security Zones



Securosis

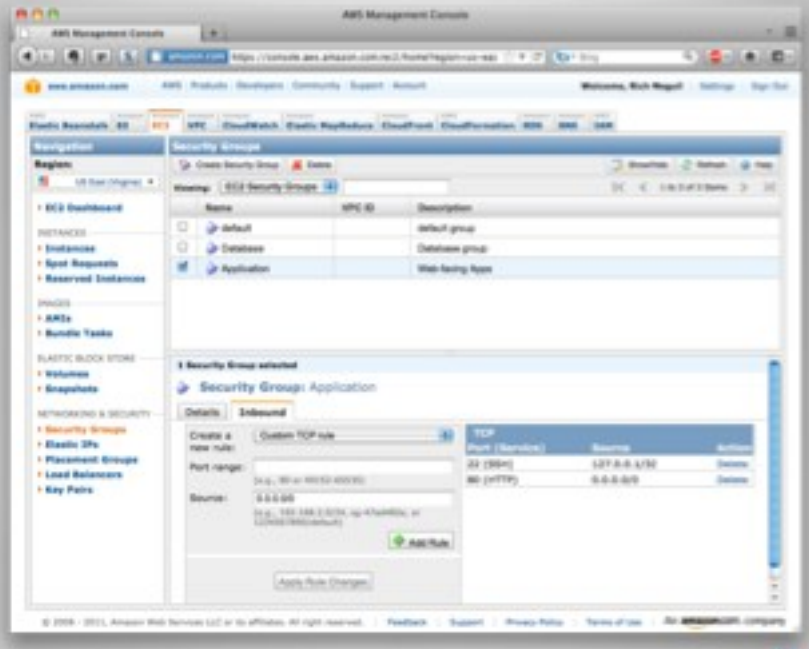
12. Management Plane

Securosis

PaaS - Centralized Management

Securosis

Geared for Development & Deployment



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

Securosis

Cloud-Sec 12-Step

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