Defending the Cloud from the Full Stack Hack

Erik Peterson
Director of Technology Strategy, Office of the CTO
Veracode, Inc.
@silvexis
CLOUD IS THE NEW OPERATING SYSTEM
CLOUD INFRASTRUCTURE IS CODE
WHO’S SECURING THESE APPS?
A Fully Realized Cloud Application (AWS)
Traditional App Represents Small % of System

Java, .NET, Python, etc... including 3rd Party code in red
Majority comes from the Cloud Provider
You may be tempted to Ignore all that and...

- Treat your cloud provider as the digital equivalent of infinite rack space with a nice web interface
- Rely on your existing security controls
- Assume your cloud providers mile long list of compliance certifications protects you
...after a few months this might be you
Forklifting is Dangerous

- Forklifting is the process of taking legacy data center applications and loading them into a cloud provider with little or no changes
  - Often this is the most expensive way to go Cloud

- It is also dangerous
  - IDS and network access controls are unaware of AWS API activity
  - Previously ignored or de-prioritized vulnerabilities can become critical
  - The meaning of availability changes
Ok, just embrace “DevOps” to fix it!

- DevOps is a powerful approach to building and maintaining complex, fast moving systems but…
  - It requires a culture change throughout your engineering org
  - It does not mean go fire all your Ops people and let development figure it out now
  - What about Security….
DevOps Culture tends to Fail Open

- Developers building systems who are new to the Cloud might now be the ones configuring the firewall rules and API Access Policies as well.
- Developers are often pre-disposed (and compensated) to “Just make it work”.
- Good DevOps should mean your engineering teams *care* about operations *a lot*, not that they are now responsible for it all by themselves.
EMERGENT INSECURITY

The individual components of your system may be secure but when deployed into the Cloud, the system becomes insecure.

AND IT GETS WORSE AT SCALE
4 HORSESMEN OF EMERGENT INSECURITY

- "Internet Weather" – Cloud systems and API calls are subject to unpredictable, non-persistent, network latency, system performance and connection interruption.

- Guaranteed Failure - System availability is a factor of redundancy and automation, not the stability and performance of monolithic systems.

- Software defined everything (SDe) - Virtualized networks, network interfaces, file systems, computing power and more can change independent of the underlying system.

- Out-of-band management – Cloud API’s operate outside of traditional security controls, can make all existing controls irrelevant.
What’s the Real Attack Surface?

- “If your security sucks now, you’ll be pleasantly surprised by the lack of change when you move to Cloud.”*

- **In reality is it gets worse**
  - AWS API endpoints are open by design, trumps all existing controls
  - Unexpected disclosures from provisioning systems and metadata
  - Private might be public, IP’s, data leaks through 3rd party services
  - Only thing standing between total compromise of your _entire_ datacenter is the secrecy of your API keys

*Chris Hoff, @beaker
API ACCESS == PHYSICAL ACCESS

IN THE CLOUD
THE API BYPASSES TRADITIONAL CONTROLS

- **For example, start with an basic system**
  - Block all network traffic using host FW, throw away SSH keys/passwords, install a network, log all network traffic to/from system
  - In a traditional data center, system is now inaccessible, any attempt to access would be impossible and also detected

- **In the cloud however**
  - Use APIs to snapshot the disk, mount snapshots on different system, and extract everything without touching the network or system
  - Zero indication from traditional controls that any access has taken place
  - Same is true of cloud databases, I don’t need your passwords or even SQL Injection, I just need to snapshot your DB
The attacker can:

- Sets up bitcoin mining operations within your cloud environment
- Alter your applications to spread malware
- Use your environment as a means to launch additional attacks
- Download or manipulate all of your customers data
Or nuke your entire cloud environment

(from orbit, it's the only way to be sure)
Getting API keys

- **API keys checked into source control**
  - Many cloud providers are scanning public repos for API keys to protect you from yourself
  - If you using API keys in production apps however you are likely

- **Exploiting vulnerabilities to extract keys from Metadata or Bootstrap systems**
  - Harder to deal with, requires you to think differently when testing your apps for security issues

VERACODE

RSAC Conference 2015
GitHub Example
Cloud Metadata

- Fundamentally a critical part of cloud infrastructure
- Simplifies provisioning of cloud systems, in many cases makes things more secure
- Something you need to be aware of and protect however
What is Cloud Metadata?

- Based on RFC 3927 - Dynamic Configuration of IPv4 Link-Local Addresses
- Metadata can contain all kinds of awesome things, like startup scripts and your API access credentials
- All cloud providers have it
- On Azure, metadata is not dynamic but is copied to /var/lib/waagent (linux) or %SYSTEMDRIVE%\AzureData\CustomData.bin (Windows)
Example: Accessing AWS Metadata

- AWS command line tool `ec2-metadata` will extract some (but not all!) of the metadata
- To get everything use `wget` or `curl`
Metadata Credential Extraction (AWS)

```bash
{
  "Code": "Success",
  "LastUpdated": "2014-08-13T16:43:54Z",
  "Type": "AWS-HMAC",
  "AccessKeyId": "ASIAJ4YSTDWONSUPPHIA",
  "SecretAccessKey": "iqFuJWC19AUaBxe0tbuc6MC70oQW2wehWufZ9cQV"
}
```

**Token**: "AGQODYxdzE4IaUAMsk8UJ11ng9DHLT1qMD
+vDimxTCnAbzNGcGPbV9jEPLY05DMMLABjVdk1Fo7vS8HmEDrh3eaOOT7f8aXW9BGMSdc/iF94PIt8+ko5sXgboy4XPB
+Bh44zBSKFV4WirMKfMDwAfcieER7z6Cakeg0oee6q/H0PsK9Gps1p06g+iylzLw8mT5ADz9GQQTf
+P3anq3dAl32SNYEiJR0fTQCuQeE8/dpLbnmdhOn3WyW8eF3TJFPd8/LOMQak3EMgolpAxm+eWAMj1B5Crewy4sbvBzf
+GcmFiMClzY9gFzxCx0eV09j9nPos/d9VRpFakm1tWAS+s9Rkz1zLidWNJeufuhyLSxcR5xOeXJ6/
Pte6QjIf21lep6FJExEGE3Ho0A10z4tv9Yo5c2tPafEhWsaCB0ia
+kQPExftmuIulmkRk9NugNuKcd0OzDkoftkpIFAJ0j9op2TgsDuImc0R3LSijbmghLZzG1uUePnTnr
+7DmbvmrJWihftTD2hJzFAzhsvsNVytt
+mWksSGaeBhASUosCgj2vJB1kHG6kkTcl9FnEU7XPoKvsiUSRqwZYOoeITSL4dE38pcfHhmbPAci6H5Tbj04FuRQC
+iFqAFvCD3q66fBQ="

"Expiration": "2014-08-13T23:05:19Z"
Old Vulnerabilities, New Life

- These vulnerabilities can result in a total Cloud environment compromise
  - CWE-918: SSRF
  - CWE-611: XXE
  - CWE-441: Unintended Proxy or Intermediary
  - CWE-77: Command Injection
  - CWE-200: Information Exposure
  - CWE-268: Privilege Chaining
- Why?
  - All of these can lead to unintended exposure of metadata
Real world Examples

- Prezi experienced this first hand

- Andres Riancho demonstrated this in his BlackHat talk “Pivoting in Amazon Clouds”
  - Exploited SSRF vuln to inject malicious AWS SQS messages to then exploit a celery/python vuln and take control
Excuse me, your Cloud is leaking

- Tagging of Cloud assets is critical to keeping track of things
- Tags however often can contain private data that does not have strong access protections
- If you use any 3rd party Cloud management system your tag data is replicated to those systems
- I’ve seen customer data, API keys and passwords in tags!!
Chained Privilege Escalation

- Cloud API permissions are extremely rich
  - For example there are almost 1000 policy/service permissions in AWS
  - Locking down only user admin permissions is not good enough
The Full Stack Hack

1. Cloud DNS Reveals Private IP of web server
2. Private IP of web server reveals detailed errors or admin interface
3. SSRF or XXE vulnerability exposes Metadata, revealing API Keys
4. API Key allow you to escalate privileges clone system root partition
5. Cloned system gets you SSH keys to app servers and API key with full access
6. With new credentials create trust relationship with external account and clone DB for quiet extraction
Protecting Yourself
FOR NEW PROJECTS:
Consider an eventually consistent security model

Instead of trying to enforce change control which creates brittle systems that are insecure and not survivable, design your systems to be eventually consistent with your security and operational goals.

If your system requires strict change control to maintain order, in the Cloud, you will eventually have chaos.
Eventually consistent security is a big shift

Instead of this:
- Control change
- Create Stable Environments
- Disaster Recovery Plan

Do this:
- React to Change
- Create Hostile Environments
- Disaster Survivability

Warning: If your application requires that you never drop a single transaction or can account for every single change, this approach might not be for you.
FOR ALL PROJECTS: Re-think your Threat Model

- Low priority or “mostly harmless” software vulnerabilities might now be deadly
- API Access circumvents traditional controls

- Any one of these could mean game over
  - CWE-918: SSRF
  - CWE-611: XXE
  - CWE-441: Unintended Proxy or Intermediary
  - CWE-77: Command Injection
  - CWE-200: Information Exposure
  - CWE-268: Privilege Chaining
Control and Audit API Access

- IP Restrictions are not just for firewalls
- Keys locked down to specific IP’s have less chance of damage
- Require 2 Factor Auth on Risky permissions
- Highly unlikely your app needs to be able to create or edit users
- Regularly audit your API permissions

```
{ 
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": ["192.168.1.1/32"]
      }
    }
  }
}
```
Don’t use your bill as an IDS

- Many cloud customers use their bill as their cloud IDS
- So far, most of these people have been very, very lucky

- Instead:
  - Turn on billing alerts to get near real time notifications
  - Watch your API logs for suspicious activity
  - Don’t have API logging turned on you say?!
Log all the things, especially API activity

- If you do nothing else, do this now!
- Logging is off by default in most cloud providers and is the most important thing you can log
- In AWS, turn on CloudTrail for all regions
- Use LogStash, develop your own, or buy a commercial solution to make API activity logs accessible and monitored
Free Cloud Security Analysis Tools

- Amazon (AWS)
  - Netflix Security Monkey
  - Trusted Advisor
    - [https://aws.amazon.com/premiumsupport/trustedadvisor/](https://aws.amazon.com/premiumsupport/trustedadvisor/)

- Microsoft Azure
  - Azure Operational Insights (in preview)

- Google
  - Google Cloud Security Scanner (for AppEngine)
    - [https://cloud.google.com/tools/security-scanner/](https://cloud.google.com/tools/security-scanner/)
Application Focused Cloud Security Research

- Veracode R&D project “Cloudatlas”
  - I am looking for beta participants, please contact me if interested
  - Investigating how cloud changes application security
  - Currently uncharted waters
Apply What You Have Learned Today

- Next week you should:
  - Turn on API Logging, do it NOW

- In the first three months following this presentation you should:
  - Analyze your applications for vulnerabilities that can expose metadata

- Within six months you should:
  - Rethink your threat model, consider an eventual consistency security model that doesn’t rely on strict controls to be effective
Thank You!

May all your clouds come with a Luck Dragon

epeterson@veracode.com

@silvexis