
The sky is falling
Nephological tales of security woe


Ben Toews

Snakeoil as a Service

- People are concerned about security these days
- People aren't sure about the security impact of the cloud
- Scared people are good customers
- Lots of people are exploiting this fear to sell bullshit snake oil

don’t panic

- Don’t buy snakeoil
- The cloud has a lot of security benefits

- We'll walk through some examples of cloud security incidents and talk about what went wrong.

- October 2013
- Adobe is a desktop software company.
- They manage downloads through a web app.
- “attackers illegally entered our network”
- Wasn’t cloud related

http://helpx.adobe.com/x-productkb/policy-pricing/ecc.html
http://nakedsecurity.sophos.com/2013/11/04/anatomy-of-a-password-disaster-adobes-giant-sized-cryptographic-blunder/
https://github.com/blog/1698-weak-passwords-brute-forced

- Compromise led to 38 million stolen account passwords

crypto is ***hard***

- Encrypted, not hashed
- ECB Block cipher (64 bit blocks)
- Password hints helped too

MongoHQ

- October 2013
- Internal support system account with same password as on Adobe
    - Adobe ->
    - Internal support system (w/ impersonation) ->
    - Customer data (passwords were bcrypted) ->
    - Buffer mongodb access -> social media auth tokens

http://techcrunch.com/2013/10/29/hosting-service-mongohq-suffers-major-security-breach-that-explains-buffers-hack-over-the-weekend/
http://arstechnica.com/security/2013/10/hack-of-mongohq-exposes-passwords-user-databases-to-intruders/
http://open.bufferapp.com/buffer-has-been-hacked-here-is-whats-going-on/

GitHub

- November 2013
- “Brute force” attack using Adobe passwords
- Already had strong rate limiting
- Rate limiting didn’t help much
- 40,000 unique IP addresses
- ~5 login attempts per account
- Used stolen accounts to get Ripple currency

account security

- shared passwords
- 2FA

- He’s just one random example
- Open source repo w/ AWS creds
- >$3000 AWS bill
- Thousands of AWS creds in public repos
- Working with AWS to scan repos

http://vertis.io/2013/12/16/unauthorised-litecoin-mining.html

Bitly

- May 2014
- Link shortener
- AWS key for backup database stored in source code
- Employee account compromised
- GitHub contacted them (they never mention GitHub)

http://www.cso.com.au/article/544802/bitly_reveals_hackers_stole_secret_keys_from_hosted_code_repository/

Bonsai

- June 2014
- Elastic search hosting
- Old AWS master key hard coded in source code
- Source code leaked
- Noticed and outage due to attacker deleting random stuff
- Worked with Amazon to lock things down and restore backups

http://status.bonsai.io/incidents/qt70mqtjbf0s

credential storage

- Don’t store creds in source code

June 2014

Code spaces was a git and subversion hosting provider.

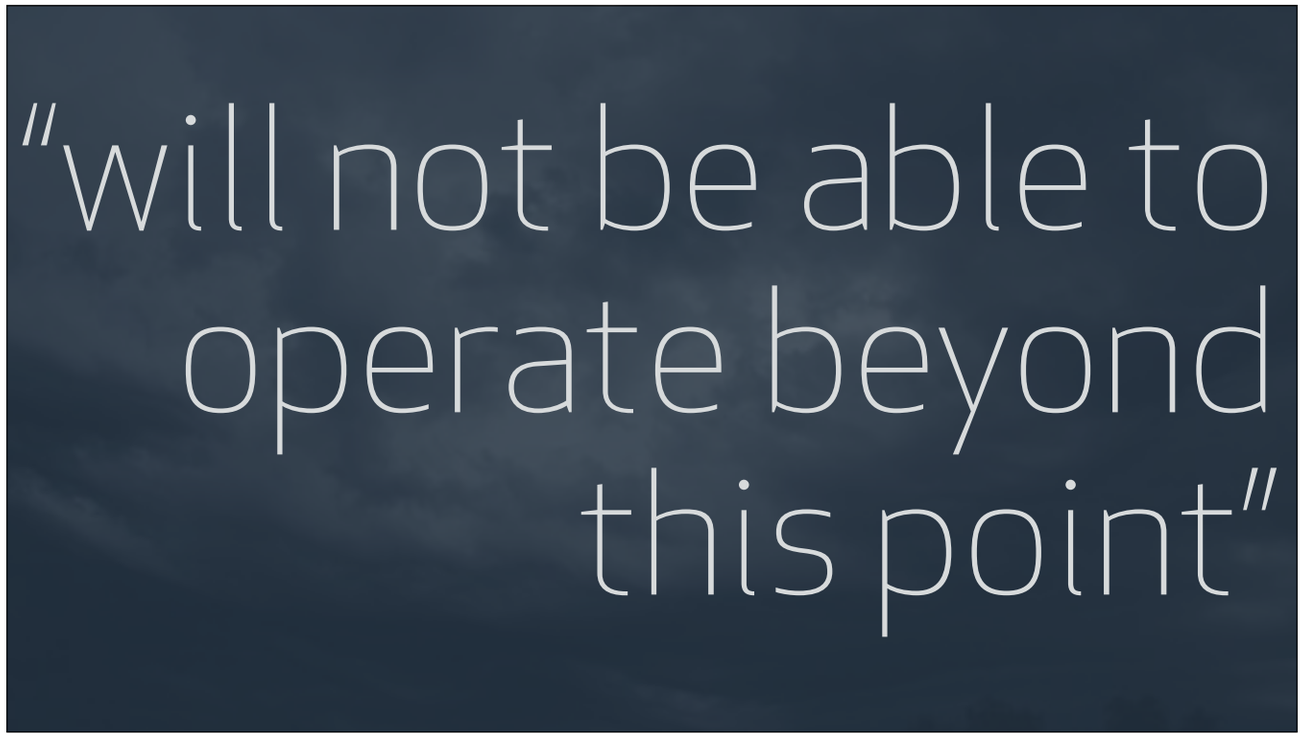
http://www.csoonline.com/article/2365062/disaster-recovery/code-spaces-forced-to-close-its-doors-after-security-incident.html
http://arstechnica.com/security/2014/06/aws-console-breach-leads-to-demise-of-service-with-proven-backup-plan/
http://threatpost.com/hacker-puts-hosting-service-code-spaces-out-of-business/106761
http://blog.trendmicro.com/the-code-spaces-nightmare/

DDoS

They noticed a DDoS attack.
The attacker left a note in their AWS console asking them for money.
WAIT, they left the note *in* the AWS console.

AWS compromised

DDoS was smokescreen.
AWS account was compromised.
They tried to regain controll of account.
Attacker noticed.
Attacker deleted everything.



They wen’t out of business 12 hours after the incident began.

account security

- shared passwords
- 2FA

disaster recovery

- I hear DR plans are good

trust

- Trustworthy providers (not Code Spaces)
- Verify trust.

- April 2013
- 0day in ColdFusion
- DB and webapp access
- Properly encrypted credit card data
- Salted/hashed passwords
- Lost deploy keys for instances
-

credential storage

- They did a pretty good job

- This isn’t just the cloud
- Alert Logic report
    - Incidents are still more common in on-prem

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

- How do you actually secure stuff?

- You can usually trust your cloud provider
    - They have people who are good at security
- Don’t get cut on the bleeding edge
- Use established providers
- Look for security docs
- Email support

verify

- Audit your logs
- FIND AWS LOG PRODUCT

- Account Security
- Application Security
- Network/Host Security
- Physical Security

- Need to trust everything up to the application
- Strong account security
    - Password manager
    - 2FA
    - Least privilege
    - Credential storage

- Need to trust everything up to the server
- Need to focus on appsec in addition to previous concerns (+ more creds to manage)
    - This is where people start putting creds in code
    - Static analysis
    - Hire appsec people
    - Hire consultants
    - Bounty program

- Need to trust everything up to the hardware
- Host/network security in addition to previous concerns (+ more creds)
    - Harden the OS
    - Patch (not always possible - eg. Heartbleed ELB)
    - Firewall (metadata API)
    - IDS

- Trust no one
- Guards with Guns