IT’S NOT YOUR CLOUD WHICH IS INSECURE, IT’S MOSTLY THE OPERATING SYSTEM OR APPS!

$RANDOM Security Auditor
AGENDA

▸ AWS access security

▸ Unix Security
Have a look at the AWS security guideline:
http://aws.amazon.com/security
**ENABLE MFA TOKENS EVERYWHERE**

- Provide an additional factor to the authentication step
- MFA is assigned to root account and IAM users
- Can also be assigned to roles
- Physical (YubiKey, etc.) or virtual token (Google Authenticator) could be used
HOW MANY PEOPLE OR APPS HAVE ACCESS TO YOUR KINGDOM?

“The Queen of England”
REDUCE NUMBER OF IAM USERS

- Review IAM policies for users, groups and roles
- Consider Identity Federation
  - Delegate access keys for API usage
  - Allow you to create temporary access keys
DO YOUR EC2 INSTANCES NEED TO CONTACT OTHER SERVICES?
USE ROLES FOR EC2

- Reduce attack surface area
- Create EC2 specific roles
- Assign specific policies to the roles
- EC2 roles supported by "aws-cli"
IAM SHOULD HAVE KEY ROTATION EVERY 90 DAYS!

AWS Security Specialist
AWS ACCESS SECURITY

ROTATE ALL KEYS REGULARLY

- Require automation workflow to replace keys
  - Track age of an access key
  - create new key
  - deploy new key to automation process
  - test
- deactivate old key
DO NOT “ALLOW ALL” IN SECURITY GROUPS

- EC2 ip address range is a favourite for scanners
- Control ingress of data by port, IP and Security Group
- Monitor security groups regularly
DO NOT “ALLOW ALL” IN SECURITY GROUPS
AWS ACCESS SECURITY

BASTION HOSTS

- Server used for system management
- Access tightly controlled
- Management only enabled from these host
- Stop bastion host when not in use
AWS ACCESS SECURITY

BASTION HOST

TCP Port 22 from Bastion Host only

Web Servers
Security Group
TCP 80, 443 from ELB

App Servers
Security Group
TCP 8080 from Web

Database Servers
Security Group
TCP 3306 from App

Bastion Host
Security Group

SSH
UNIX SECURITY

Based on the draft security policy we’re working on:
https://portal.avira.org/display/CSIN/Security+Policy
AWS ACCESS SECURITY

UNIX USER AND GROUPS

- Give them the minimal amount of privileges they need
- Be aware when and where they login from
- Make sure you remove inactive accounts
- The use of the same userid on all computers and networks
- The creation of group user-id's should be absolutely prohibited
AWS ACCESS SECURITY

OPENSSH

- Limit Users SSH access by creating extra groups
- Disable root login via SSH
- Restrict the interface for the service
- Disable empty passwords
- Disable password authentication
- Use log analyser
- Use strict mode for ssh service
Services should listen only on the used IP / interface

Run local firewalls, you know your applications the best

Firewall also outgoing traffic to prevent attackers to download additional malware

Allow administration services (like SSH) only from secure locations
QUESTIONS?

THANKS!