Intrusion Detection in the Cloud

Greg Roth, AWS Identity & Access Management
Don Bailey, AWS Security

November 14th, 2013
Why should you care about this?

• Change management / monitoring is a good thing™
• Traditional intrusion detection may not detect AWS-specific environment changes
• Figure it out now, rather than later
• News flash: The bad guys know about the cloud
So let’s geek for a bit

- Intrusion detection in your AWS environment
- Universal adversary tactics to focus on
- AWS-specific security features to build with
- AWS-specific intrusion detection mechanisms w/ demos!
- Other tips, resources, Q&A
Can you have your IDS in AWS?

- Short answer: YES!
- What IS an intrusion detection system?
  - System that monitors environment; alerting to detected intrusions.
- On premise, your IDS takes advantage of that environment’s features.
- Within your AWS environment, you should do the same: Your AWS-specific IDS will likely NOT look like your traditional IDS. That’s OK!
- Operating System
- Processes
- Files

- Packets
- Flows

Server

Network

Cloud

AWS
Configuration

Amazon S3
Amazon EC2
Amazon VPC
Amazon RDS
Elastic Beanstalk
IAM

Security
Group
VPC
Subnet

Traditional IDS

Customer

This Talk

AWS

Objects
Internet Gateways
Groups, Users, Credentials

Amazon S3 Bucket
Security Group
VPC Subnet
Amazon RDS DB Instances
Applications

Amazon S3
Amazon EC2
Amazon VPC
IAM
Amazon RDS
Elastic Beanstalk
Wait! I want MY IDS in AWS!

- I.e., “traditional” IDS – rackable, stackable, network-sniffing box that streams alerts night & day
- You ♥ traditional IDS, for a number of reasons, not all of them your own, eg., compliance.
- No worries! You can still do that too in AWS
Traditional IDS in AWS

- On premises, VPC endpoint
Traditional IDS in AWS

• In cloud, as VPC NAT gateway or on-instance
Traditional IDS in AWS

- On premises, VPC endpoint
- In cloud, as VPC NAT gateway or on-instance
- Numerous AWS technology partners here
- Visit their booths … or the AWS Marketplace!
Software Infrastructure

Security (152 results) showing 1 - 10

1. NGINX Plus - Amazon Linux AMI
   - Rating: ★★★★★ (5)
   - Version: 1.2
   - Sold by: nginx, Inc.
   - Price: $0.04 to $0.68/hr for software + AWS usage fees
   - NGINX Plus AMI for AWS is provided by the original creators of NGINX web server. Run by over 40% of web sites hosted on AWS (ref: Netcraft's October 2012 Web Server Survey), ...
   - Linux, Unix, Amazon Linux 2013.03 | 64-bit Amazon Machine Image (AMI)

2. Alert Logic Threat Manager for EC2
   - Rating: ★★★★★ (5)
   - Version: 1.2
   - Sold by: Alert Logic
   - Price: $0.48 to $1.64/hr for software + AWS usage fees
   - Threat Manager for EC2 is the first Network Intrusion Detection (IDS) service specifically designed for AWS. Using this service you can now cost effectively protect security ...
   - Linux, Unix, Debian 6 | 32-bit Amazon Machine Image (AMI)

3. Vyatta Virtual Router/Firewall/VPN
   - Rating: ★★★★★ (5)
   - Version: 5.6.2 R62
   - Sold by: Vyatta, Inc.
   - Price: $0.30 to $1.20/hr for software + AWS usage fees
   - The Brocade Vyatta vRouter delivers advanced routing, firewall and VPN in a cloud-ready, software appliance. Much more than a simple gateway or firewall solution, Vyatta ...
   - Linux, Unix, Other 6.4R1 | 64-bit Amazon Machine Image (AMI)

4. Check Point Virtual Appliance for AWS
   - Rating: ★★★★★ (5)
   - Version: RT75
   - Sold by: Check Point Software Technologies
   - Price: Bring Your Own License + AWS usage fees
   - Check Point Virtual Appliance for Amazon Web Services enables customers to extend their security to the cloud with the full range of protections using Check Point Software...

https://aws.amazon.com/marketplace/
School of r00t

• Gain access
• Maintain access
• Steal stuff
Prerequisites

- AWS Identity and Access Management (IAM)
- Multi-Factor Authentication (MFA)
- Amazon S3 Bucket Logging
  http://docs.aws.amazon.com/AmazonS3/latest/UG/ManagingBucketLogging.html
- And THREE more …
Security Role

• You need insight when managing the security of many AWS accounts
• Create a “security audit role” with “read” access to policies and configurations you want to monitor.
• For more info or getting started, check out http://docs.aws.amazon.com/IAM/latest/UserGuide/WorkingWithRoles.html
What’s a Role

• Named IAM entity  (name isn’t a secret)
• Set of permissions
• No credentials: Policy specifies who can assume
Security Role (Example Policy)
Security Role (Snippet of Example Policy)

```json
{
    "Sid": "Stmt1382474270211",
    "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "s3:GetBucketLogging",
        "s3:GetBucketNotification",
        "s3:GetBucketPolicy",
        "s3:GetLifecycleConfiguration",
        "s3:GetObjectAcl",
        "s3:GetObjectVersionAcl",
        "s3:ListAllMyBuckets"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
```

https://s3.amazonaws.com/reinvent2013-sec402/secaudit.json
Demonstration: Creating Security Role

https://s3.amazonaws.com/reinvent2013-sec402/secaudit.json
Create Role

- Specify a role name. You cannot edit the role name after the role is created.

- Role Name: secaudit

Maximum 64 characters. Use alphanumeric and '+' characters.
Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in Using IAM.

Policy Name

secaudit

Policy Document

```json
{
    "Statement": [
        {
            "Sid": "Stmt1382473313140",
            "Action": [
                "CloudFormation:DescribeStackEvents",
                "CloudFormation:DescribeStackResource",
                "CloudFormation:DescribeStackResources",
                "CloudFormation:DescribeStacks",
                "CloudFormation:GetTemplate",
                "CloudFormation:ListStacks",
                ...
            ],
        }
    ]
}
```
Write-Once Storage

- What is it good for
  - Tripwire
  - Configuration audits
  - Logs
- Integrity for records of activity, historical configurations
- Further enhanced by moving off-system or limiting availability to a VERY select few
Configuring Write-Once Storage

- Bucket versioning
  http://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html

- MFA delete
  http://docs.aws.amazon.com/AmazonS3/latest/dev/MultiFactorAuthenticationDelete.html

- Go for the gusto! Create a SECOND account
  - Bucket policy
  - Role
Policy for Bucket: "writeonce"

Add a new policy or edit an existing bucket policy in the text area below.

```json
{
    "Version": "2008-10-17",
    "Id": "Policy1382581126724",
    "Statement": [
        {
            "Sid": "Stmt1382581121416",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::923022406781:root"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::writeonce/**"
        }
    ]
}
```
Versioning allows you to preserve, retrieve, and restore every version of every object stored in this bucket. This provides an additional level of protection by providing a means of recovery for accidental overwrites or deletions.

Once enabled, Versioning cannot be disabled.

- [ ] Enabled
- [ ] Suspended

[Save] [Cancel]
Audit Logs via AWS CloudTrail

- AWS CloudTrail records API calls in your account and delivers logs to your S3 bucket.
- Typically, delivers an event within 15 minutes of the API call.
- Log files are delivered approximately every 5 minutes.
- Currently in us-east-1 and us-west-2
AWS Services Supported by AWS CloudTrail

- Currently, records API call made to these AWS services.

- Includes API calls made by higher-level AWS services such as AWS CloudFormation, AWS Elastic Beanstalk, and AWS OpsWorks.
• Have a centralized write-only store? Use it!
What is in the logs?

- **Who** made the API call?
- **When** was the API call made?
- **What** was the API call?
- **What** were the resources that were acted up on in the API call?
- **Where** was the API call made from?
Who? Example 1: API Call by IAM User Bob

"userIdentity": {
    "accessKeyId": "AKEXAMPLE123EJVA",
    "accountId": "123456789012",
    "arn": "arn:aws:iam::123456789012:user/Bob",
    "principalId": "AIEXAMPLE987ZKLALD3HS",
    "type": "IAMUser",
    "userName": "Bob"
}
Who? Example 2: API Call by Federated User Alice

"userIdentity": {
  "type": "FederatedUser",
  "principalId": "123456789012:Alice",
  "arn": "arn:aws:sts::123456789012:federated-user/Alice",
  "accountId": "123456789012",
  "accessKeyId": "ASEXAMPLE1234WTROX8F",
  "sessionIssuer": {
    "type": "IAMUser",
    "accountId": "123456789012",
    "userName": "Bob"
  }
}

- Elastic Beanstalk creating AWS resources on behalf of IAM user Bob

```
"userIdentity": {
  "accountId": "123456789012",
  "arn": "arn:aws:iam::123456789012:user/Bob",
  "invokedBy": "elasticbeanstalk.amazonaws.com",
  "principalId": "ASEXAMPLE123XWTROX8F",
  "type": "IAMUser",
  "userName": "Bob"
}
```
When was the API call made?

- Start time and date of the event in ISO 8601 format.
- Unambiguous and well-defined method of representing date and time.
- AWS services sync all system clocks with centralized Network Time Protocol (NTP) servers

"eventTime": "2013-10-23T23:30:42Z"
What was the API call?
What resources were acted up on?

- API call and the service the API call belongs to.
  
  "eventName": "RunInstances"

  "eventSource": "EC2"

- Request parameters provided by the requester and Response elements returned by the AWS service.

- Response elements for read only API calls (Describe*, Get*, List*) are not recorded to prevent event size inflation.
School of r00t

- Gain access
- Maintain access
- Steal stuff
Detecting Unauthorized Access

• Types of access
  – Credentials
  – Publicly accessible resources
  – Cross account access
Detecting Unauthorized Access – Credentials

- Types of credentials
  - Login profile
  - Access key
  - X509
  - Cloudfront
  - Temporary Security Credentials

- Attachment points
  - Root account
  - IAM users

- You want to know what credentials are out there with access to your account.
Demonstration: Checking Credentials
Detecting Unauthorized Access – Public

- Publically accessible resources (NOT by default, but could be configured as such)
  - Amazon S3 Bucket
  - Amazon S3 Anonymous Objects
  - Amazon SQS Open / Public Queues

- You want to keep track of which resources are readable (or writable even) to the world
Detecting Unauthorized Access – Cross Account

• Resources that support resource policies
  – Amazon S3 Buckets
  – Amazon SQS queues
  – Amazon SNS topics

• You want to pay particular attention to any resources that have resource policies allowing cross account access.
Demonstration: Checking for Cross-Account Access to Resources
Detecting Unauthorized Access – Roles

• What is a role
  – Name
  – AssumeRole Policy
  – Capabilities

• You want to look at what roles are present in the account and who can assume them
Demonstration: Checking for Roles
Detecting Unauthorized Access – Effective Access

• Ways of expressing * (IMPLICIT *)
  – PutUserPolicy
  – Credential creation
  – PassRole *

• You want to look out for policies that could be used to GAIN all access (IAM APIs)

• IAM Policy Simulator …

https://policysim.aws.amazon.com/
YES!
{ "Statement": [ {
   "Sid": "Stmt1383555181147",
   "Action": "sns:*",
   "Effect": "Allow",
   "Resource": "*"
}, {
   "Sid": "Stmt1383555193395",
   "Action": ["s3:*","*"],
   "Effect": "Allow",
   "Resource": "*"
} ]
},

{ "Statement": [ {
   "Sid": "Stmt1383555181147",
   "NotAction": "*",
   "Effect": "Allow",
   "Resource": "*"
}, {
   "Sid": "Stmt1383555193395",
   "Action": ["iam:PutUserPolicy"],
   "Effect": "Allow",
   "Resource": "*"
} ]
},

{ "Statement": [ {
   "Sid": "Stmt1383555181147",
   "Action": "ec2:*",
   "Effect": "Allow",
   "Resource": "*"
}, {
   "Sid": "Stmt1383555193395",
   "Action": ["s3:*","iam:PassRole"],
   "Effect": "Allow",
   "Resource": "*"
} ]
}
Detecting Unauthorized Access – Effective Access

- Dump the output of various configuration APIs into write-once storage
- Pay attention to changes
- Some examples for grabbing this data …

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
Using Security Role for Amazon S3 Audit (Bucket Policies)

```python
s3 = boto.connect_s3(access_key_id, secret_access_key)
bucket_info = []
buckets = s3.get_all_buckets()
for bucket in buckets:
    try:
        policy = bucket.get_policy()
        bucket_info.append(config_line_policy("s3:bucketpolicy", bucket.name, "", policy))
    except boto.exception.S3ResponseError as e:
        bucket_info.append(config_line("s3:bucketpolicy", bucket.name, "", e.code))
output_lines(bucket_info)

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
```
user_info = []
users = iam.get_all_users().list_users_response.list_users_result.users
debug(users)
for user in users:
    policies = iam.get_all_user_policies(user.user_name)
policies = policies.list_user_policies_response.list_user_policies_result.policy_names
    for policy_name in policies:
        policy = iam.get_user_policy(user.user_name, policy_name)
            .get_user_policy_response.get_user_policy_result.policy_document
        policy = urllib.unquote(policy)
        user_info.append(config_line_policy("iam:userpolicy", user.user_name, policy_name, policy))
    output_lines(user_info)

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
Account Configuration Change Security Alerts

• Dump all the users, groups, roles, attached permissions, creds for all users
• Amazon S3 bucket, Amazon SQS queue, Amazon SNS topic policies
• Amazon EC2 security group configuration
• All goes to flat file, write-once Amazon S3 object
• Diff and detect changes

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
Demonstration: Intrusion Detection Script

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
Example Usage

SecConfig.py [-h] -a ACCESS_KEY_ID -k SECRET_ACCESS_KEY \ [-t SECURITY_TOKEN] [-r ROLE] [-v] [-d]

- **-h, --help** show this help message and exit
- **-a ACCESS_KEY_ID, --access_key_id ACCESS_KEY_ID** access key id
- **-k SECRET_ACCESS_KEY, --secret_access_key SECRET_ACCESS_KEY** secret access key
- **-t SECURITY_TOKEN, --security_token SECURITY_TOKEN** security token (for use with temporary security credentials)
- **-r ROLE, --role ROLE** role to assume
- **-v, --verbose** enable verbose mode
- **-d, --debug** enable debug mode
Example Output (Snippet)

iam:accountsummary, AccountMFAEnabled, , 1
iam:accesskey, ClassicRTTUser, Active, AKIAJQF4G2ZOZBL3FYKQ
iam:accesskey, ClassicRTTUser, Active, AKIAJVVZ456L2HVERGIQ
iam:accesskey, audit, Active, AKIAJJ7D5VQ2KAC4RX6Q
iam:accesskey, ec2test, Active, AKIAIWFQHOLKE3ARKOQ
iam:accesskey, ec2test, Active, AKIAISNKP5NBWJRQTBWA
iam:accesskey, mbp-r-managed, Active, AKIAJKVVGI7G7L5UC5OGQ
iam:accesskey, quux, Active, AKIAJ7ZICS26032EPBQ
iam:accesskey, test, Active, AKIAINTUMS4ITD5CJVSA
iam:useringroup, ClassicRTTUser, , ClassicRTTGrp
Example Output (Snippet)

s3:bucketpolicy, dcslides, , NoSuchBucketPolicy
s3:bucketpolicy, elasticbeanstalk-us-east-1-923022406781, , NoSuchBucketPolicy
s3:bucketpolicy, gbr-billreport, ,
- NoSuchBucketPolicy

sqs:queuepolicy, https://queue.amazonaws.com/923022406781/deleteme, ,
- NoSuchQueuePolicy

sns:topicpolicy, arn:aws:sns:us-east-1:923022406781:test, ,
- NoSuchTopicPolicy
sns:topicpolicy, arn:aws:sns:us-east-1:923022406781:test2, ,
- NoSuchTopicPolicy
Example Diff, Something to Look Into

< iam:userpolicy, mbp-r-managed, one, e3e0211e865b5cac2a57241edcb8aeb9d546764abba2f325b694ec840985c2ff
---
> iam:userpolicy, mbp-r-managed, ReadOnlyAccess-mbp-r-managed-
  201311111559,
b675543c022ca9bce21414468a7b62e207116f11f77e722ae2f65fed7e69ffbb
> iam:userpolicy, mbp-r-managed, one,
  1cc602178f7e876c6d38cbaa8c4adde19b1c3e5a89e6f13c29df5688eb73f50f

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
School of r00t

- Gain access
- Maintain access
- Steal stuff
DATA ALERT: Your acct used abt 75% of its data allowance for the bill ending the 25th. Monitor at vzw.com. As of 10/04 01:00 PM EDT. FREE MSG
Example OK vs UH-OH Billing Trend / Graph

[Graph showing two trends over time, one increasing (OK) and one staying flat (UH-OH).]
Billing Alerts!

• No need to wait until end of month to become aware of unexpected utilization
• Establish a baseline of known good billing over time; set your thresholds (overall or service specific)
• Investigate alerts to determine r00t (?) cause
• Simplest cloud IDS mechanism, and FREE*

* Setup of 10 alarms and receipt of 1 K notifications
You are eligible for the AWS Free Usage Tier. See the Getting Started Guide AWS Free Usage Tier to learn how to get started with the free usage tier.

Monitor your estimated charges. Enable Now to begin setting billing alerts that automatically e-mail you when charges reach a threshold you define. Learn More
## CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: **1,014**

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Billing Metrics</strong></td>
<td>35</td>
</tr>
<tr>
<td>Total Estimated Charge</td>
<td>1</td>
</tr>
<tr>
<td>By Service</td>
<td>13</td>
</tr>
<tr>
<td>By Linked Account</td>
<td>3</td>
</tr>
<tr>
<td>By Linked Account and Service</td>
<td>18</td>
</tr>
<tr>
<td><strong>EC2 Metrics</strong></td>
<td>272</td>
</tr>
<tr>
<td>Per-Instance Metrics</td>
<td>181</td>
</tr>
<tr>
<td>By Auto Scaling Group</td>
<td>56</td>
</tr>
<tr>
<td>By Image (AMI) Id</td>
<td>14</td>
</tr>
<tr>
<td>Aggregated by Instance Type</td>
<td>14</td>
</tr>
<tr>
<td>Across All Instances</td>
<td>7</td>
</tr>
<tr>
<td><strong>DynamoDB Metrics</strong></td>
<td>4</td>
</tr>
<tr>
<td>Table Metrics</td>
<td>4</td>
</tr>
<tr>
<td><strong>ELB Metrics</strong></td>
<td>95</td>
</tr>
<tr>
<td>Per-LB Metrics</td>
<td>29</td>
</tr>
<tr>
<td>Per LB, per AZ Metrics</td>
<td>37</td>
</tr>
<tr>
<td>By Availability Zone</td>
<td>20</td>
</tr>
<tr>
<td>Across All LBs</td>
<td>9</td>
</tr>
</tbody>
</table>
To create an alarm, first select a metric by browsing or searching on the right. Once you find the metric you want, select it and then click Next.
**Alarm Threshold**

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

- **Name**: My Estimated Charges
- **Description**: Estimated Monthly Charges

**Whenever charges for**: EstimatedCharges

- **is**: $ \geq 200 USD
- **for**: 1 consecutive period(s)

**Alarm Preview**

This alarm will trigger when the blue line goes up to or above the red line for a duration of 6 hours.

**Actions**

Define what actions are taken when your alarm changes state.

- **Whenever this alarm**: State is ALARM
- **Send notification to**: Please select an SNS topic

**Create topic**
Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: My Estimated Charges
Description: Estimated Monthly Charges

Whenever charges for: EstimatedCharges
is: \( \geq \) USD $ 200
for: 1 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Whenever this alarm: State is ALARM
Send notification to: 
Email list: john.stiles@example.com

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 6 hours.

Namespace: AWS/Billing
Currency: USD
Metric Name: EstimatedCharges
Period: 6 Hours
Statistic: Maximum
Example Billing Alert via CLI

Assuming You Anticipate ~ $1 K / Month ...

Week 1  Week 2  Week 3  Week 4

$1000  $750  $500  $250

= OK!
“Early” Alerts Are “Interesting” …

Week 1: $250
Week 2: $500
Week 3: $750
Week 4: $1000

= OK!  = Hmm …
More than One “Early” Alert …?

$1000
$750
$500
$250

Week 1
Week 2
Week 3
Week 4

= OK!
= Hmm ...
= Uh-Oh!
More Resources on Billing Alerts Setup …

• Monitoring your AWS charges
  http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/monitor_estimated_charges_with_cloudwatch.html

• Amazon CloudWatch Command Line Interface Reference
  http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/CLIRefERENCE.html
Create Your Own Meter-based Alerts?

- Use: programmatic access to billing data
- You have more info about the types and locations of charges
- Allows for looking for unexpected usage per region

http://docs.aws.amazon.com/awsaccountbilling/latest/about/prograccess.html
Another Tactic? Rebuild Frequently

- Breaking in is noisy and the holes tend to get patched [intrusion lifecycle]
- Auditing a system is easiest after creation
- Rebuild everything every day
Premium Support / Trusted Advisor

• Inspects AWS environment; can identify and help close security gaps, enable security features, examine permissions
  – Open security groups
  – Bucket policy
  – IAM, passwords, MFA

https://aws.amazon.com/premiumsupport/trustedadvisor/
### Summary

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Optimizing</td>
<td>3</td>
<td>3 warnings</td>
</tr>
<tr>
<td>Security</td>
<td>4</td>
<td>4 warnings</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td>4</td>
<td>4 warnings</td>
</tr>
<tr>
<td>Performance</td>
<td>4</td>
<td>4 warnings</td>
</tr>
</tbody>
</table>

**Total Potential Annual Savings:** $1,821,868

**Opportunities to Enhance Security:** 116

**Recommendations to Improve Availability:** 30

**Opportunities to Improve Performance:** 138

### Recently Launched Checks

- **New** Amazon Route 53 High TTL Resource Record Sets
  - 0 of 1461 resource record sets have TTL values that are too large

- **New** Amazon Route 53 Name Server Delegations
  - 0 of 5 hosted zones do not have all four name server delegations configured.

- **New** Amazon Route 53 Alias Resource Record Sets
  - 18 of 1491 resource record sets can be changed to alias resource record sets.
Security Group - Open Ports

- 215 of 753 Security Group port rules create potential security vulnerabilities by granting global access

IAM Use

- IAM is configured for this account

S3 Bucket Permissions

- 6 of 42 S3 Buckets have permission properties that grant global access

MFA On Root

- Root user does not have MFA enabled.

IAM Password Policy

- Password policy is not configured
Support for Security

- AWS support is the one-stop shop for AWS customers, for ANY concerns, including security-related.
- If support can not immediately address your concern, they will escalate internally to the appropriate technical team, AWS security included.

https://aws.amazon.com/support
Other Resources

- AWS Security Blog
  http://blogs.aws.amazon.com/security/

- AWS Security Center
  https://aws.amazon.com/security/

- Contact the AWS security team
  aws-security@amazon.com
NEW! Security Best Practices Whitepaper

• Help for designing security infrastructure and configuration for your AWS environment
• High-level guidance for …
  – Managing accounts, users, groups roles
  – Managing OS-level access to instances
  – Securing your data, OS, apps, infrastructure
  – Managing security monitoring, auditing, alerting, incident response

Key Takeaways

• Beyond traditional host- or network-based intrusion detection, there is intrusion detection for the cloud

• AWS provides a variety of mechanisms and support that you can and should leverage to monitor key security controls

• Tinker, give us feedback, and approach our partners about incorporating some ideas here
Downloads

https://s3.amazonaws.com/reinvent2013-sec402/secaudit.json

https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py
Please give us your feedback on this presentation

SEC402

As a thank you, we will select prize winners daily for completed surveys!