

RSAConference2018

San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CSV-R12



#RSAC

RED TEAM VS. BLUE TEAM ON AWS

Teri Radichel

CEO
2nd Sight Lab
@teriradichel

Kolby Allen

DevOps Engineer
Zipwhip
@kolbyallen

Attacker vs. Defender



Cloud Admin...Duh Duh Duh.



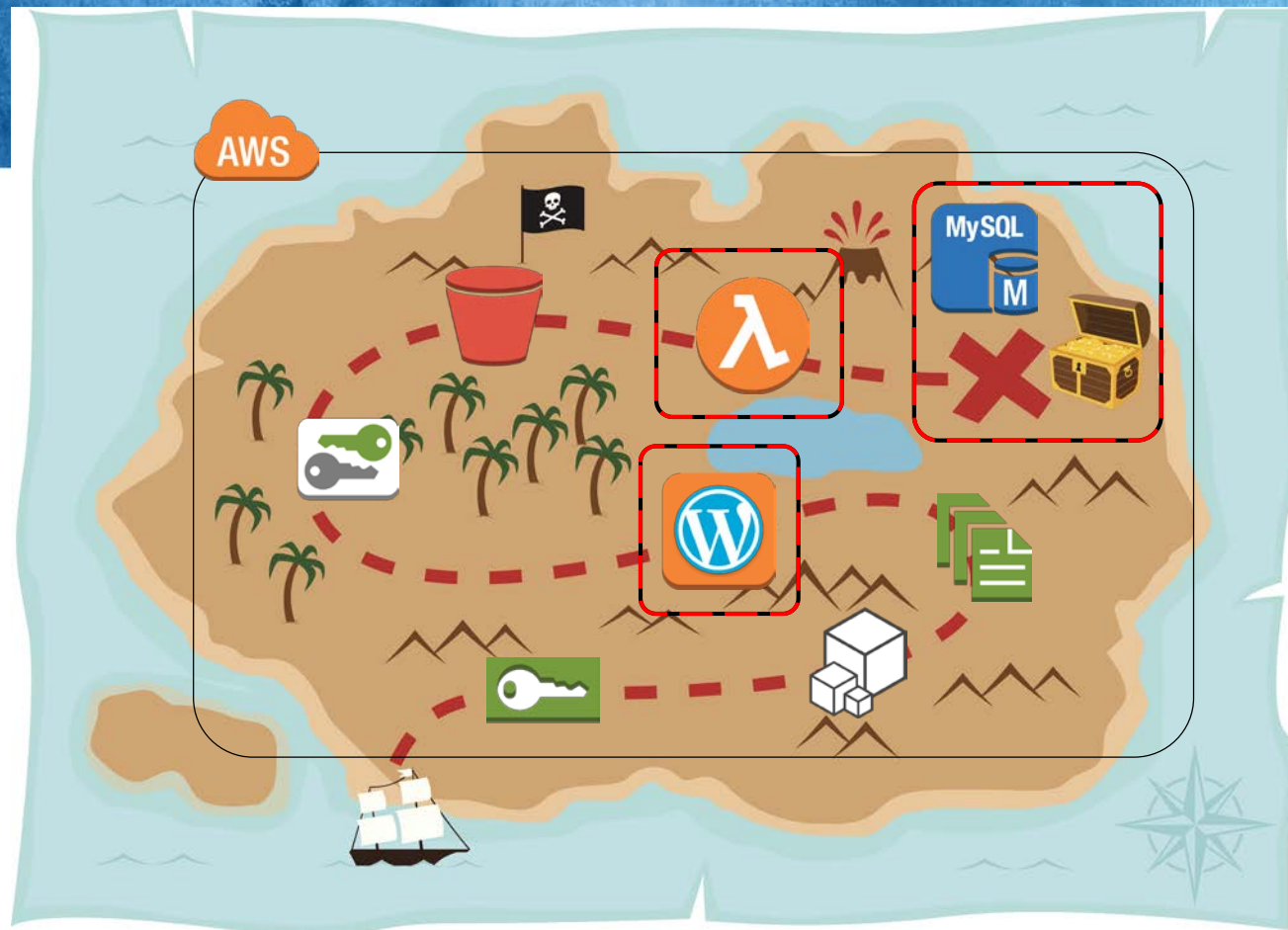
Would Be A Boring Talk...



GAME
OVER



#RSAC



Instead...

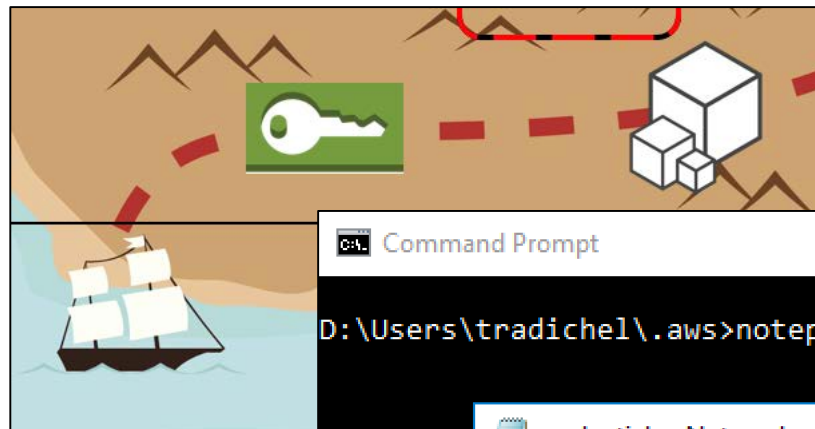
Let's search for buried treasure!

Some background



- Initial Setup
 - Vanilla Account
 - Single Admin User
 - Base VPC & defaults
 - AWS Tutorial: Elastic Beanstalk with WordPress
 - <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/php-hawordpress-tutorial.html>
 - AWS Tutorial: Lambda Accessing RDS in VPC
 - <https://docs.aws.amazon.com/lambda/latest/dg/vpc.html>

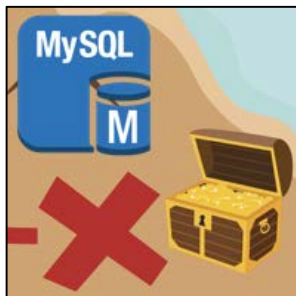
Pilfer Credentials ~ Read Only Access



```
Command Prompt
D:\Users\tradichel\.aws>notepad .credentials

credentials - Notepad
File Edit Format View Help
[default]
aws_access_key_id = AKIAJRWIVY7NJ6UYCKEQ
aws_secret_access_key = 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
|
```

Look for RDS Databases



```
aws rds describe-db-instances --filter --query DBInstances[].[DBInstanceIdentifier,MasterUsername,DBSubnetGroup.VpcId,Endpoint.Address] --output=table --color off
```

supersecretdb?! That sounds like a good target...

```
Command Prompt
D:\Users\tradichel>aws rds describe-db-instances --filter --query DBInstances[].[DBInstanceIdentifier,MasterUsername,DBSubnetGroup.VpcId,Endpoint.Address] --output=table --color off
```

DescribeDBInstances			
aa1fe08ildto0z5	wordpress	vpc-96c34cfe	aa1fe08ildto0z5.cl5fcy9momq1.us-east-2.rds.amazonaws.com
supersecretdb	kolbyadmin	vpc-96c34cfe	supersecretdb.cl5fcy9momq1.us-east-2.rds.amazonaws.com

Examine Selected Database Subnets



```
aws rds describe-db-instances --filter "Name=db-  
instance-id,Values=supersecretdb" --query  
DBInstances[].DBSubnetGroup.Subnets[].Subnet  
Identifier --output table --color off
```

Hmm... let's check out: **subnet-1ae9df57**


```
Command Prompt  
D:\Users\tradichel>aws rds describe-db-instances --filter "Name=db-  
instance-id,Values=supersecretdb" --query DB  
Instances[].DBSubnetGroup.Subnets[].SubnetIdentifier --output table --color off  
  
+-----+  
| DescribeDBInstances |  
+-----+  
| subnet-1ae9df57 |  
| subnet-d48c0fbc |  
+-----+
```

What Traffic Do NACLs Allow?



```
aws ec2 describe-network-acls --filter  
"Name=association.subnet-id,Values=subnet-1ae9df57"  
--query NetworkAcls[0].Entries --output table --color off
```

All traffic allowed ~ Sweet.



```
Command Prompt  
D:\Users\tradichel>aws ec2 describe-network-acls --filter "Name=association.subnet-id,Values=subnet-1ae9df57" --query NetworkAcls[0].Entries --output table --color off
```

DescribeNetworkAcls				
CidrBlock	Egress	Protocol	RuleAction	RuleNumber
0.0.0.0/0	True	-1	allow	100
0.0.0.0/0	True	-1	deny	32767
0.0.0.0/0	False	-1	allow	100
0.0.0.0/0	False	-1	deny	32767

What Traffic Do DB Security Groups Allow?



```
Command Prompt
```

DescribeSecurityGroups	
SecurityGroups	
Description	Created from the RDS Management Console: 2018/03/22 05:26:10
GroupId	sg-217f3e4a
GroupName	rds-launch-wizard-1
OwnerId	310610724838
VpcId	vpc-96c34cfe
IpPermissions	
FromPort	3306
IpProtocol	tcp
ToPort	3306
IpRanges	
CidrIp	172.31.0.0/16
IpPermissionsEgress	
IpProtocol	-1
IpRanges	
CidrIp	0.0.0.0/0

aws ec2 describe-security-groups --filter "Name=group-id,Values=sg-217f3e4a" --output table --color off

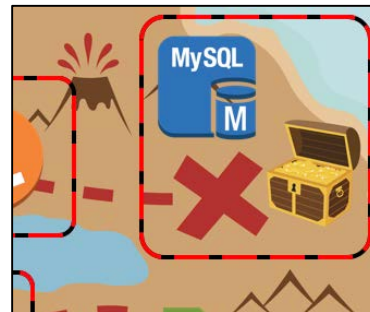
Port 3306
172.31.0.0/16

Find VPC With Access to Database



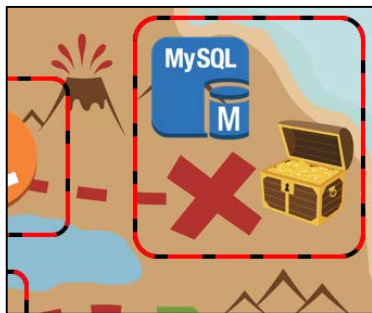
```
aws ec2 describe-vpcs --filter  
"Name=cidrBlock,Values=172.31.0.0/16" --query  
Vpcs[].VpcId --output table --color off
```

vpc-96c34cfe is assigned to CIDR 172.31.0.0/16



```
Command Prompt  
D:\Users\radichel>aws ec2 describe-vpcs --filter "Name=cidrBlock,Values=172.31.0.0/16" --query Vpcs[].VpcId --output table --color off  
-----  
| DescribeVpcs |  
+-----+  
| vpc-96c34cfe |  
+-----+
```


VPC Security Groups ~ 3306 Egress



```
aws ec2 describe-security-groups --filter  
"Name=egress.ip-permission.to-port,Values=3306  
Name=vpc-id,Values=vpc-96c34cfe" --output table --  
color off
```

None...hmm...

```
Command Prompt  
D:\Users\tradichel\.aws>aws ec2 describe-security-groups --filter "Name=egress.ip-permission.to-port,Values=3306 Name=vp  
c-id,Values=vpc-96c34cfe" --output table --color off  
-----  
|DescribeSecurityGroups|  
+-----+
```

Security Groups ~ No Outbound Restrictions

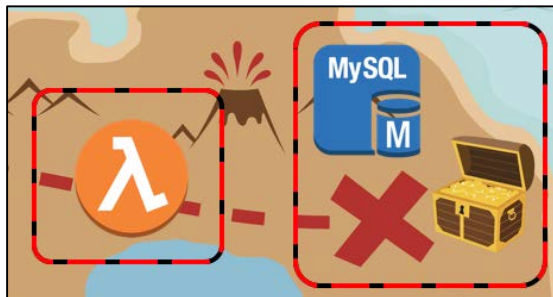


```
aws ec2 describe-security-groups --filter "Name=egress.ip-  
permission.cidr,Values='0.0.0.0/0',Name=vpc-id,Values=vpc-96c34cfe" --output  
table --color off --query SecurityGroups[].GroupId
```

```
Command Prompt  
D:\Users\tradichel\.aws>aws ec2 describe-security-groups --filter "Name=egress.ip-  
permission.cidr,Values='0.0.0.0/0',Name=vpc-id,Values=vpc-96c34cfe" --output  
table --color off --query SecurityGroups[].GroupId  
|DescribeSecurityGroups|  
+-----+  
sg-217f3e4a  
sg-3984c552  
sg-8db2c2e6  
sg-91b5c5fa  
sg-93aade8f  
sg-a0b9c9cb  
sg-a9bbcbcb  
sg-b1bacada  
sg-b97b3ad2  
+-----+
```

Cool. Wide Open Outbound.
Let's see what's using these.

Check Lambda Functions



```
aws lambda list-functions --query  
Functions[?VpcConfig.SecurityGroupIds==  
[`sg-93aade8`]].FunctionName --output  
table --color off
```

```
Command Prompt  
D:\Users\tradichel\.aws>aws lambda list-functions --query Functions[?VpcConfig.SecurityGroupIds==[`sg-93aade8`]].FunctionName --output table --color off  
-----  
| ListFunctions |  
+-----+  
| CreateTableAddRecordsAndRead |  
+-----+
```

Query Lambda Code Location



`aws lambda get-function --function-name CreateTableAddRecordsAndRead --query Code.Location`

Gives us URL to code location in S3...

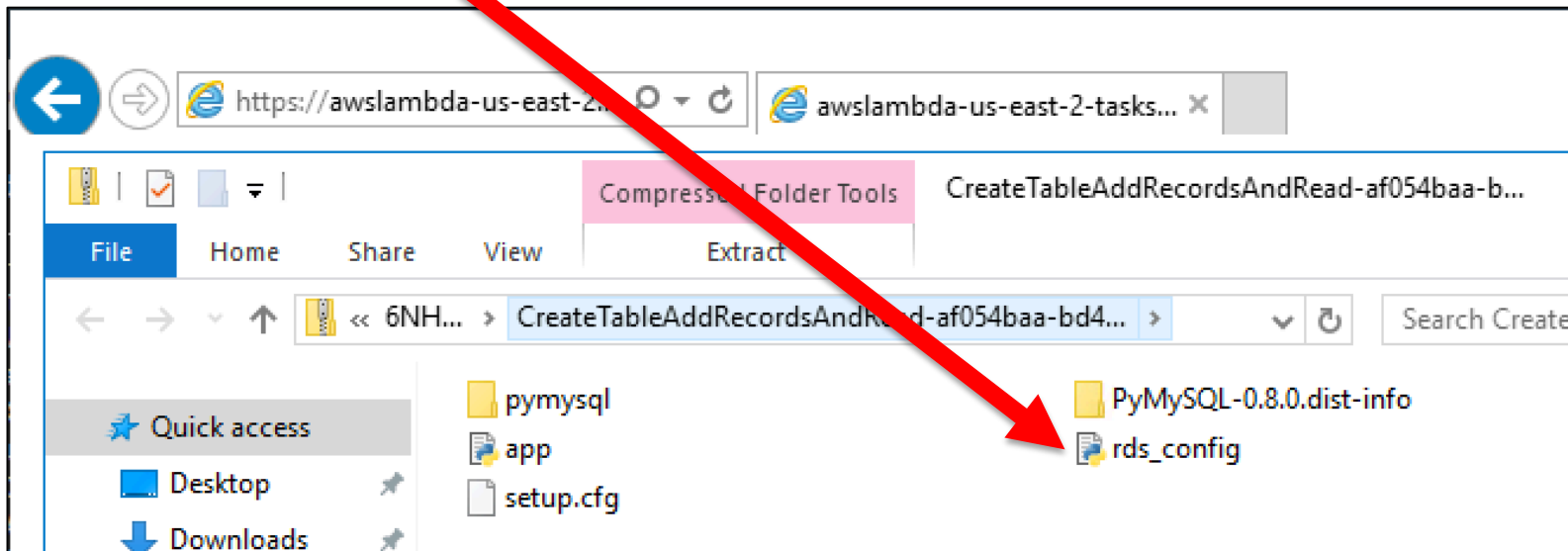
```
Command Prompt

D:\Users\tradichel\.aws>aws lambda get-function --function-name CreateTableAddRecordsAndRead --query Code.Location
"https://awslambda-us-east-2-tasks.s3.us-east-2.amazonaws.com/snapshots/310610724838/CreateTableAddRecordsAndRead-af054b
aa-bd47-414d-bd56-54ed119cd6f8?versionId=jBCcttupKmmceMPTJn_c3RgrsJnGn6r&X-Amz-Security-Token=FQoDYXdzECQaDPQOBfxx1bL3a
fb5mSK3A%2BqmpSSsGqJzxXZwQg1w9DRVrt2bY9GbgxT9D73PydpgXo4GR6uaQjdHRQmxnU%2F27fDQ9KvtjVLKoGPEGoayleVvkzGzPpVVevo40v04gZ3SS
C8exKeqLFUI5NPSfju0LID%2BdZJxvE60qoB2XWws1gM8wuyZgNP26Yb4pdHq0bXVxXxbnigZwo0G0mlrVIvQWZH%2FgymxPQN22DG%2F1sY%2FeUA3mhKOE
UqgEzor0iMMA3vkZfV2bdrWjcSutTt5XtFFmKSNwWAdn0%2Bslj28jp9Sca50D3o58o%2F2FLpRq8gvP82WxmbzYjnX9yVGwMqjfxQyK%2FDt%2BUiY3J4
VM%2B4B6PF4oXEA0Mkxjhh6SXBqZS8ma2hAmkQUI%2FCSHb21EWbbzS9MYEVLrDvDQBzyhqvwX%2F1R%2B1jZ7TaD3Bd08WMEo8JxR8i%2BDuRCjq2yTkLbL
WCULUzr6ybChufIFRHSTNd15ikmDEwXRO4CWQuSi1i61QHVVYQKhhdMEmeOEFsLakCMk0ACmf9adrG0Iz5W5L2PWbXQ7DLjcQ3Nu3xpyFxmETVvNuU1%2BrMn
1YTCgvQnQiKwOpbvzgb1%2BLypFjvwo%2F9vR1QU%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20180323T043812Z&X-Amz-SignedHea
ders=host&X-Amz-Expires=600&X-Amz-Credential=ASIAI6UZQFNPGFJYM4XA%2F20180323%2Fus-east-2%2Fs3%2Faws4_request&X-Amz-Signa
ture=a322f24da64b08be598484198eb75da5b761c3f629e909503d4b03c213aa4ba5"
```


Go To URL...Check out the code



Hmm, what's in this file?




About that rds_config file...



Oops. Database credentials.



 rds_config - Notepad

File Edit Format View Help

```
db_username = "lambdauser" db_password = "@ccess!1" db_name = "supersecretdb"
```

Look for Instances That Can Exfil



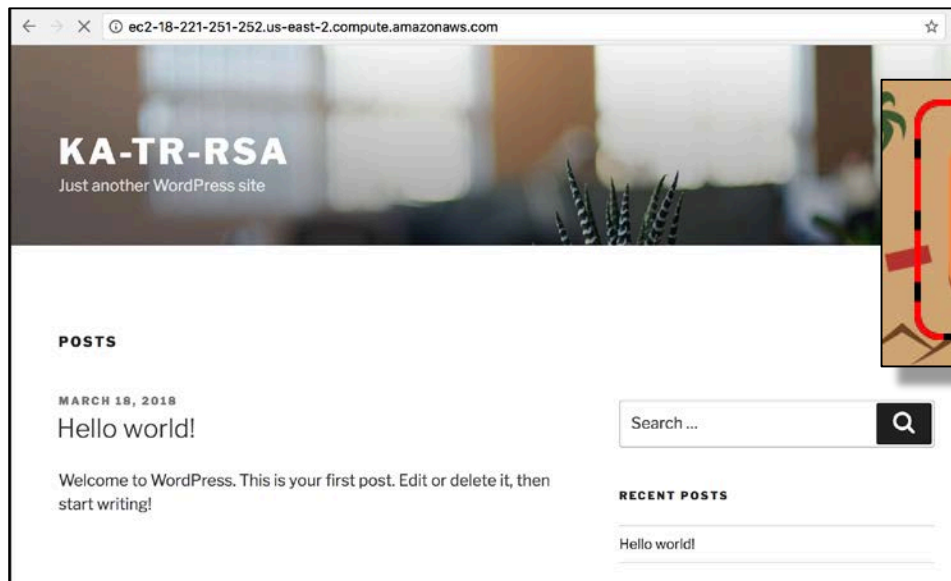
```
aws ec2 describe-instances --output text --query  
Reservations[].Instances[].NetworkInterfaces[].  
Association.[PublicIp,PublicDnsName]
```

Check the domains in a browser to find web sites.



```
Command Prompt  
D:\Users\tradichel\.aws>aws ec2 describe-instances --output text --query Reservations[].Instances[].NetworkInterfaces[].  
Association.[PublicIp,PublicDnsName]  
18.188.35.35      ec2-18-188-35-35.us-east-2.compute.amazonaws.com  
18.221.251.252   ec2-18-221-251-252.us-east-2.compute.amazonaws.com
```

Exploit Web Site and Exfil



**Scan Site. Exploit Vulnerability.
Upload code to connect to DB.
Publish to public web site.**

IAM Best Practices



- Roles
- Least Privilege
- Segregation of Duties
- IAM Top 10



Protecting Credentials



- User training ~ Phishing and handling of credentials
- Password policies and rotation
- MFA!!
- Require frequent re-auth – especially to sensitive apps
- Prevent deployment of code with embedded credentials
<https://github.com/aws-labs/git-secrets>

IAM Configuration



WOW THAT IS A LOT OF YAML!!

<https://github.com/allenk1/2018rsapresentation/blob/master/Default-IAM-Profile.yaml>

IAM Master - Initial Roles



```
...
  "Resource": "arn:aws:iam::AWS_ACCOUNT_ID:role/IAMMasterRole",
  "Action": "iam:ListAccountAliases",
  "Effect": "Allow",
  "Sid": "AllowUserstoListAccounts",
  "Resource": "*"
},
{
  "Sid": "AllowUserstoListUsers",
  "Effect": "Allow",
  "Action": "iam:ListUsers",
  "Resource": "*"
},
{
  "Sid": "AllowUserstoGetAccountPasswordPolicy",
  "Effect": "Allow",
  "Action": "iam:GetAccountPasswordPolicy",
  "Resource": "*"
},
{
  "Sid": "AllowUserstoGetAccountSummary",
  "Effect": "Allow",
  "Action": "iam:GetAccountSummary",
  "Resource": "*"
},
...

```

Sid: AllowUserstoListAccounts
Effect: Allow
Action:
- "iam:ListAccountAliases"
- "iam:ListUsers"
- "iam:GetAccountPasswordPolicy"
- "iam:GetAccountSummary"
Resource: "*"

- Allows users to view enough information to get into IAM
- Can get the PW Policy ← IMPORTANT so it can apply
- List Users – needed in order to find themselves

IAM Master - Initial Roles



```
...
  "iam:CreateAccessKey",
  "iam:CreateLoginProfile",
  "iam:DeleteAccessKey",
  "iam:DeleteLoginProfile",
  "iam:GetLoginProfile",
  "iam:ListAccessKeys",
  "iam:UpdateAccessKey",
  "iam:UpdateLoginProfile",
  "iam:ListSigningCertificates",
  "iam:DeleteSigningCertificate",
  "iam:UpdateSigningCertificate",
  "iam:UploadSigningCertificate",
  "iam:ListSSHPublicKeys",
  "iam:GetSSHPublicKey",
  "iam:DeleteSSHPublicKey",
  "iam:UpdateSSHPublicKey",
  "iam:UploadSSHPublicKey"
]
Resource: "arn:aws:iam::*:user/${aws:username}"

```



Sid: AllowUserstoManageOwnAccount
Effect: Allow
Action:
- "iam:ChangePassword"
- "iam:CreateAccessKey"
- "iam:CreateLoginProfile"
- "iam>DeleteAccessKey"
- "iam>DeleteLoginProfile"
- "iam:GetLoginProfile"
- "iam:ListAccessKeys"
- "iam:UpdateAccessKey"
- "iam:UpdateLoginProfile"
- "iam:ListSigningCertificates"
- "iam:DeleteSigningCertificate"
- "iam:UpdateSigningCertificate"
- "iam:UploadSigningCertificate"
- "iam:ListSSHPublicKeys"
- "iam:GetSSHPublicKey"
- "iam:DeleteSSHPublicKey"
- "iam:UpdateSSHPublicKey"
- "iam:UploadSSHPublicKey"
Resource: "arn:aws:iam::*:user/\${aws:username}"

Actions allow users to manage their account – BUT NOT PERMISSIONS

Resource only allows them to perform on their username – can't modify anyone else



IAM ~ User Roles



```
...
  "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
  "Sid": "AllowUserstoDeactiveTheirMFAWhenUsingMFA",
  "Effect": "Allow",
  "Action": "iam:DeactivateMFADevice",
  "Resource": "arn:aws:iam::*:mfa/${aws:username}"
},
{
  "Sid": "AllowUserstoManageThierMFA",
  "Effect": "Allow",
  "Action": "iam:CreateVirtualMFADevice",
  "iam:DeleteVirtualMFADevice",
  "iam:EnableMFADevice",
  "iam:ResyncMFADevice",
  "Resource": "arn:aws:iam::*:mfa/${aws:username}"
},
{
  "Sid": "AllowUserstoListOnlyThierMFA",
  "Effect": "Allow",
  "Action": "iam:ListVirtualMFADevies",
  "iam:ListMFADevies",
  "Resource": "arn:aws:iam::*:mfa/*"
}
]
```

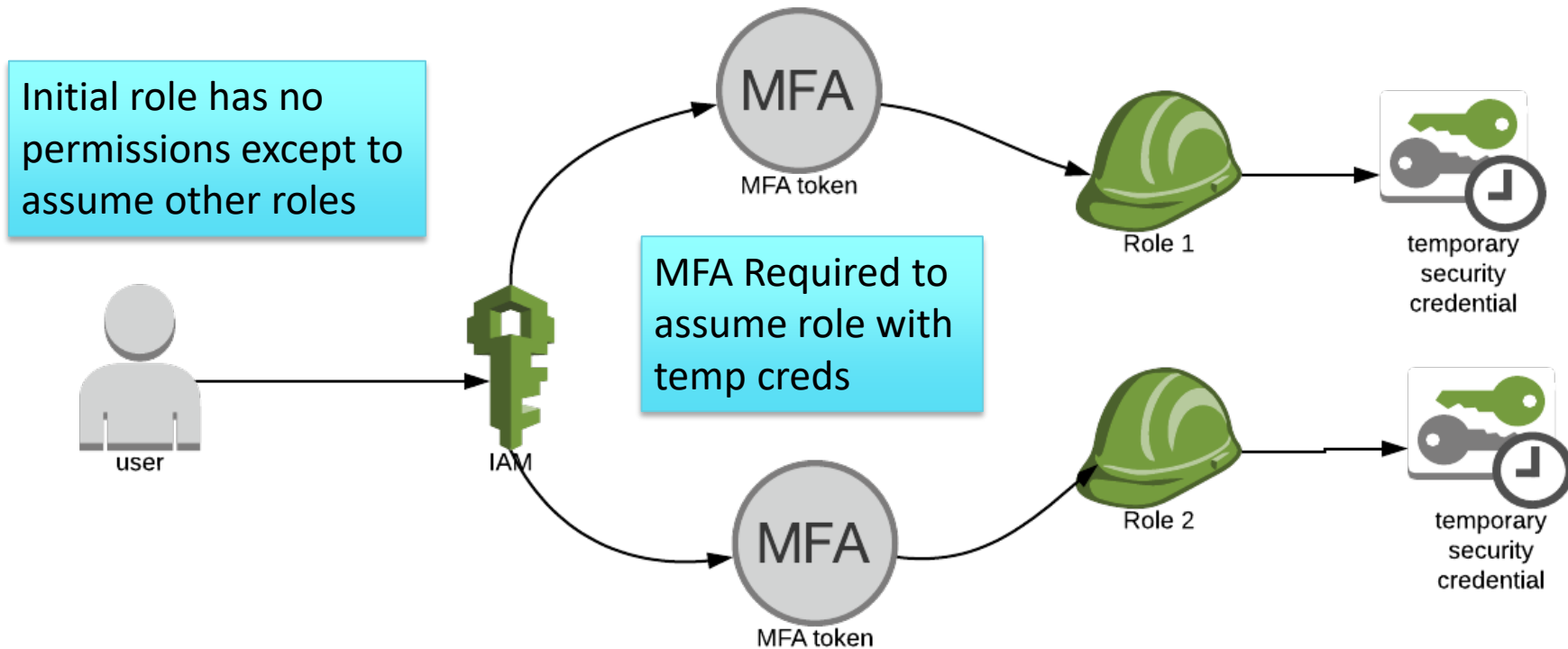
```
Sid: AllowUserstoListOnlyThierMFA
Effect: Allow
Action:
- "iam:ListVirtualMFADevies"
- "iam:ListMFADevies"
Resource:
- "arn:aws:iam::*:mfa/*"
- "arn:aws:iam::*:user/${aws:username}"
-

Sid: AllowUserstoManageThierMFA
Effect: Allow
Action:
- "iam:CreateVirtualMFADevice"
- "iam:DeleteVirtualMFADevice"
- "iam:EnableMFADevice"
- "iam:ResyncMFADevice"
Resource:
- "arn:aws:iam::*:mfa/${aws:username}"
- "arn:aws:iam::*:user/${aws:username}"
-

Sid: AllowUserstoDeactiveTheirMFAWhenUsingMFA
Effect: Allow
Action:
- "iam:DeactivateMFADevice"
Resource:
- "arn:aws:iam::*:mfa/${aws:username}"
- "arn:aws:iam::*:user/${aws:username}"
Condition:
Bool:
"aws:MultiFactorAuthPresent": "true"
```

- Allows users to manage this MFA
- Must login with MFA to remove device

IAM ~ Assumed Roles



IAM Master



#RSAC

```
1. ~ (fish)
~ aws rds describe-db-instances --query DBInstances[].DBInstanceIdentifier --output=text --region us-east-2
An error occurred (AccessDenied) when calling the DescribeDBInstances operation: User: arn:aws:iam::310610724838:user/secu
re-readonly is not authorized to perform: rds:DescribeDBInstances
! ~ aws sts assume-role --role-arn "arn:aws:iam::310610724838:role/Secure-ReadOnly-Role" --role-session-name "rsa" --
serial-number "arn:aws:iam::310610724838:mfa/secure-readonly" --token-code "095733"
{
  "Credentials": {
    "AccessKeyId": "A...",
    "SecretAccessKey": "A...",
    "SessionToken": "A...",
    "Expiration": "2018-03-23T01:51:21Z"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "A...",
    "Arn": "arn:aws:sts::310610724838:assumed-role/Secure-ReadOnly-Role/rsa"
  }
}
~ export AWS_ACCESS_KEY_ID=...; export AWS_SECRET_ACCESS_KEY=...;
export AWS_SESSION_TOKEN=...
~ aws rds describe-db-instances --query DBInstances[].DBInstanceIdentifier --output=text --region us-east-2
aalf08ildto0z5 supersecretddb
~ (docker-for-desktop)
```

MFA!

Failure due to default policy not having permissions

Temporary credential request & setting at environmental variable

Commands work!

CloudTrail



Event history

Your event history contains the create, modify, and delete activities for [supported services](#) taken by people, groups, or AWS services in your AWS account. To view a complete log of your CloudTrail events, create a trail and then go to your Amazon S3 bucket or CloudWatch Logs.

You can view the last 90 days of events. Choose an event to view more information about it. [Learn more](#)

Filter: Select attribute Enter lookup value Time range: 2018-01-14 12:00 AM — 2018-01-20 12:00 AM				
Event time	User name	Event name	Resource type	Resource name
2018-01-19, 05:00:02 PM	[REDACTED]	CreateTags		[REDACTED]
2018-01-19, 05:00:01 PM	[REDACTED]	CreateTags		[REDACTED]
2018-01-19, 04:50:21 PM	[REDACTED]	CreateTags		[REDACTED]
2018-01-19, 04:50:17 PM	[REDACTED]	ModifyInstanceAttribute	EC2 Instance	[REDACTED]
2018-01-19, 04:50:12 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]
2018-01-19, 04:50:11 PM	[REDACTED]	CreateTags		[REDACTED]
2018-01-19, 04:50:11 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]
2018-01-19, 04:50:10 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]
2018-01-19, 04:50:09 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]
2018-01-19, 04:50:08 PM	[REDACTED]	ModifyInstanceAttribute	EC2 Instance	[REDACTED]
2018-01-19, 04:50:07 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]
2018-01-19, 04:50:06 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]
2018-01-19, 04:50:06 PM	[REDACTED]	DeleteRoute	EC2 RouteTable	rtb-[REDACTED]

Feed data to events



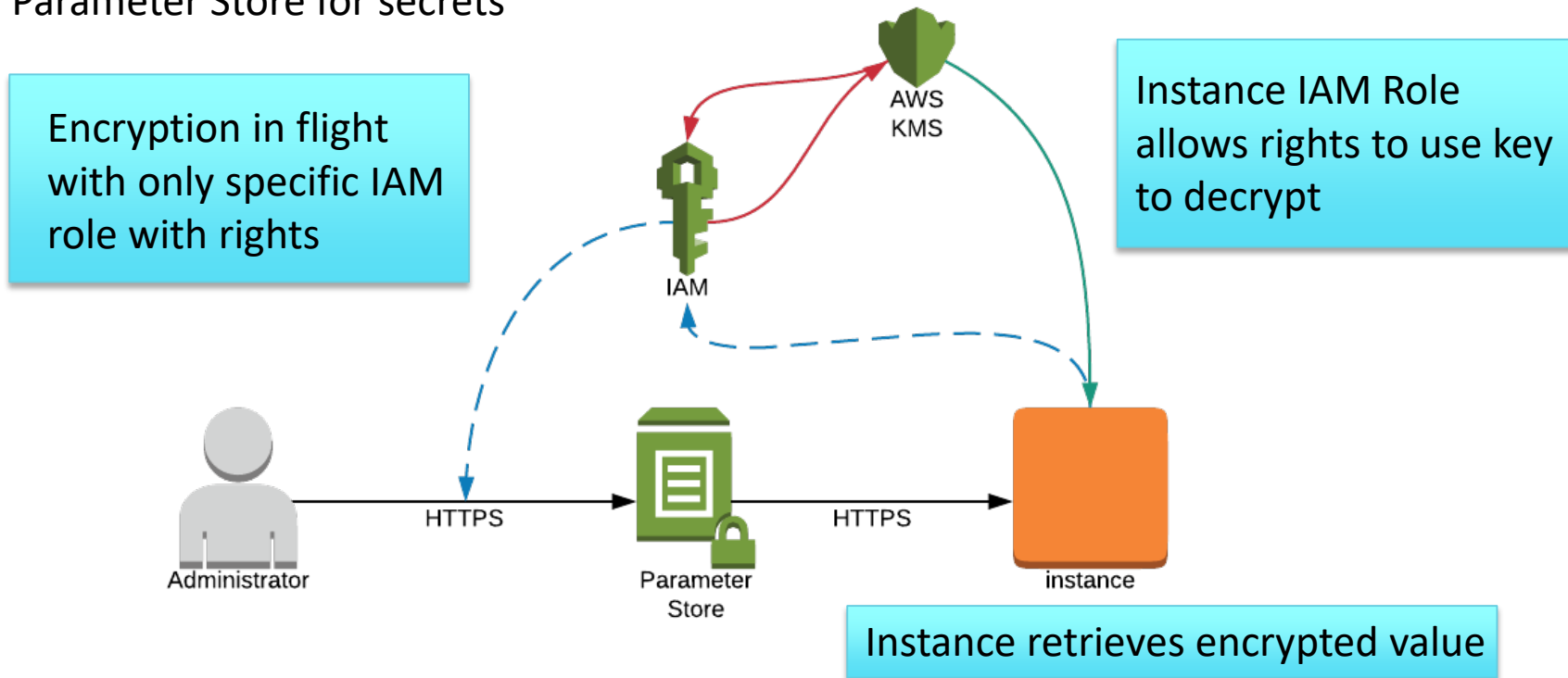
Respond

Monitor all API Actions

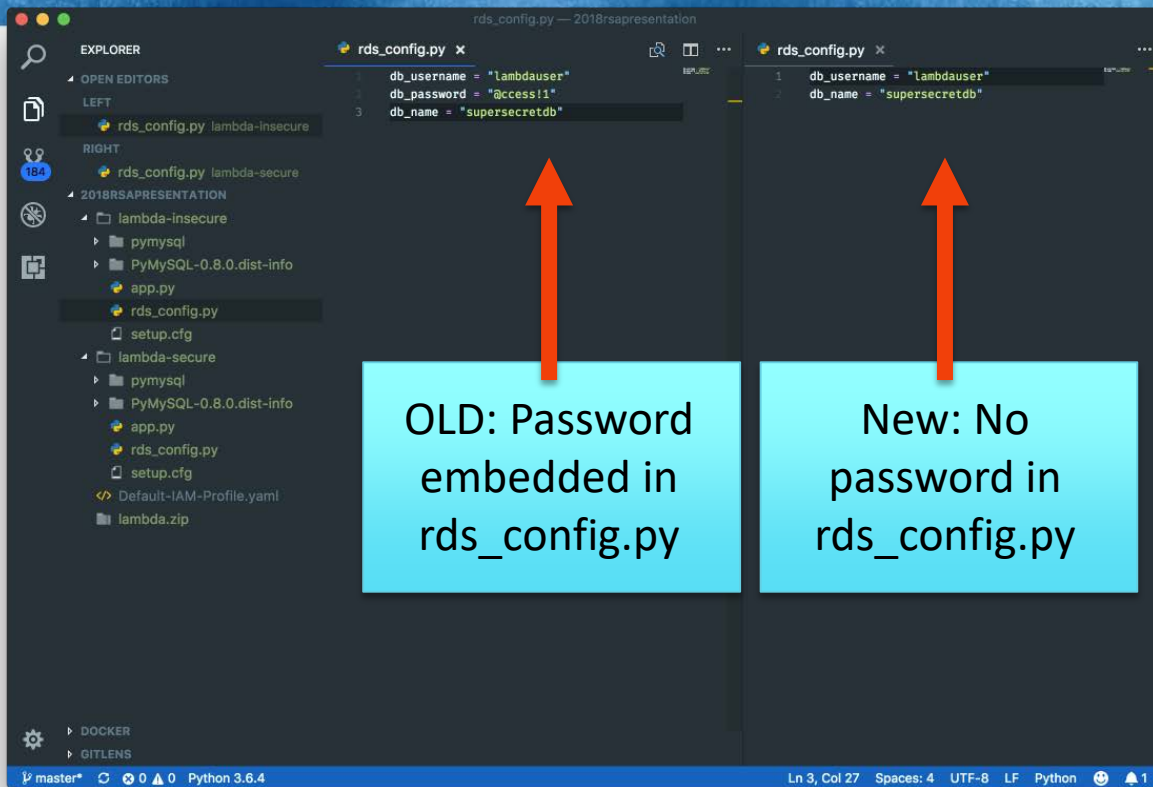
Scan and Secure



EC2 Parameter Store for secrets



EC2 Parameter Store



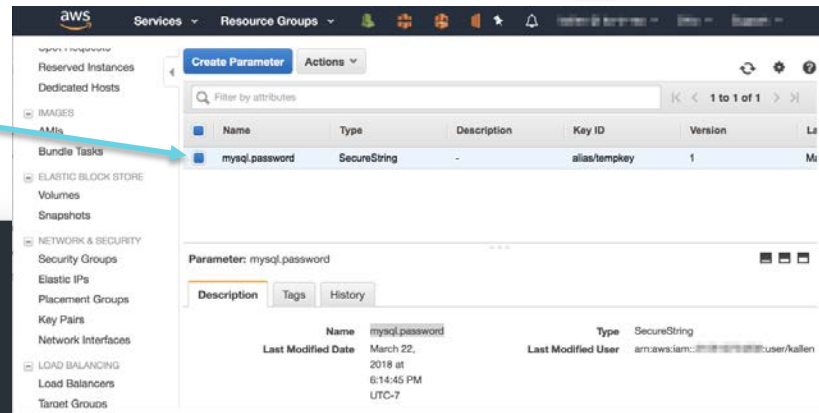
EC2 Parameter Store



Calls AWS SSM

```
app.py — 2018rsapresentation

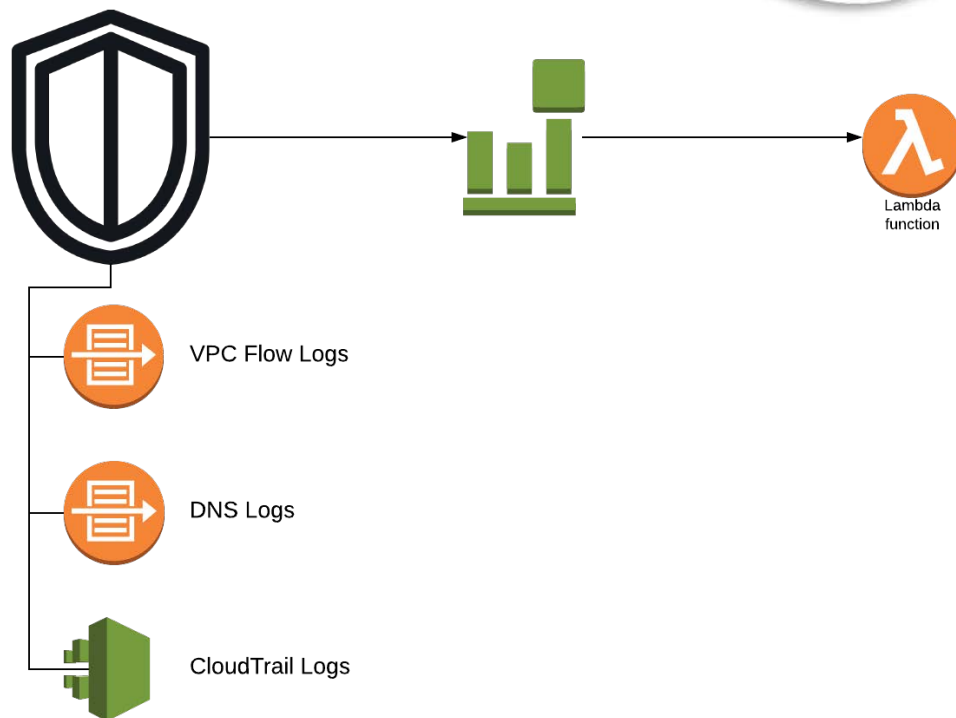
8 rds_host = "supersecretb.c15fcy9momq1.us-east-2.rds.amazonaws.com"
9 name = rds_config.db_username
10 db_name = rds_config.db_name
11
12
13 logger = logging.getLogger()
14 logger.setLevel(logging.INFO)
15
16 client = boto3.client('ssm')
17 response = client.get_parameters(Names=['mysql.password'], WithDecryption=True)
18 password = response["Parameters"][0]["Value"]
19
20
21 try:
22     conn = pymysql.connect(rds_host, user=name, passwd=password, db=db_name, connect_timeout=5)
23 except:
24     logger.error("ERROR: Unexpected error: Could not connect to MySQL instance.")
25     sys.exit()
26
27 logger.info("SUCCESS: Connection to RDS mysql instance succeeded")
28 def handler(event, context):
29     ...
```



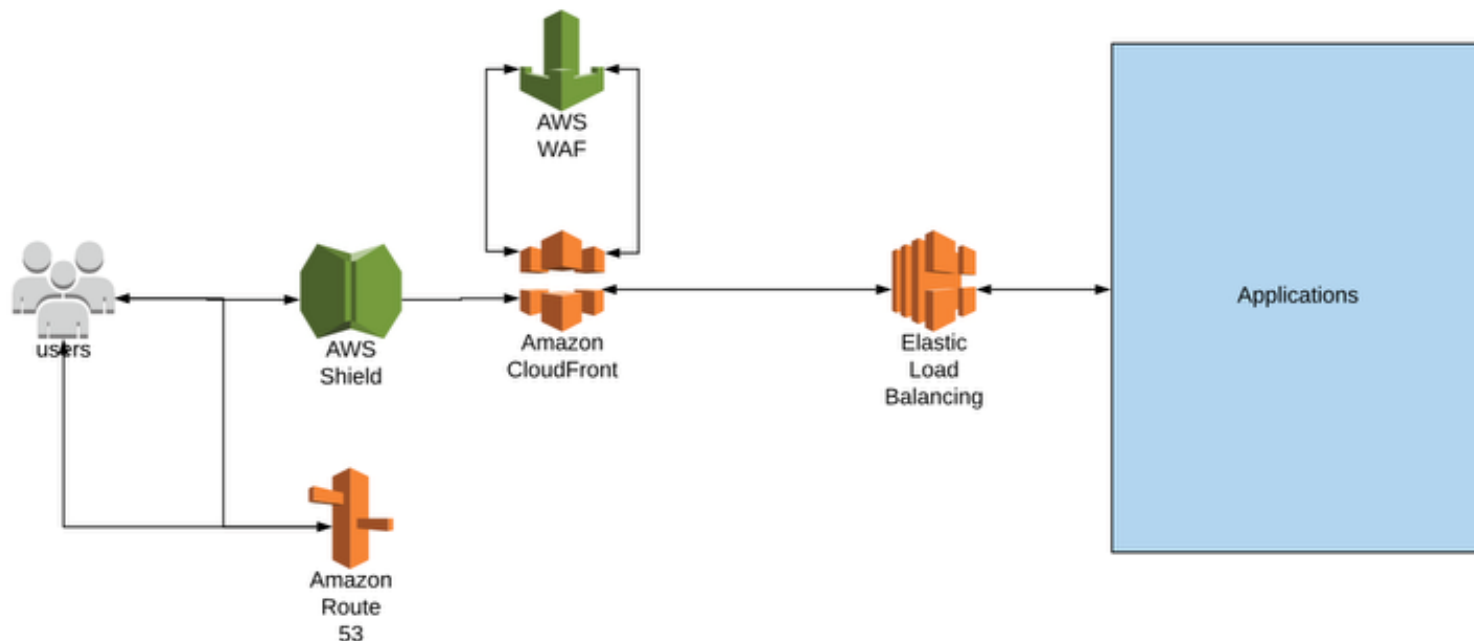
Monitoring



- AWS GuardDuty
- VPC Flow Logs
- CloudTrail
- Config
- Log shipping
- Secure log backups
- Automate Remediation



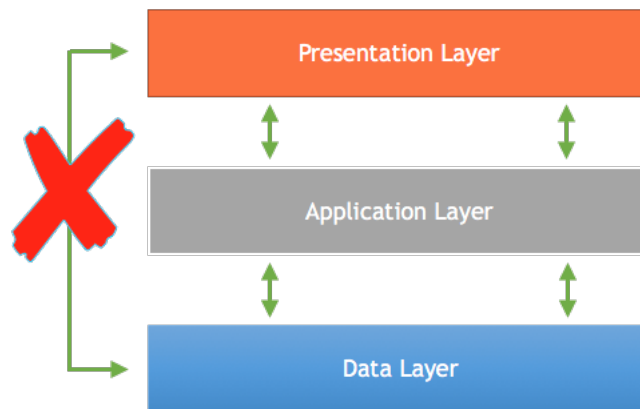
WAF Security



Network Architecture



- Presentation Layer
 - Application Layer
 - Data Layer
-
- Limited NACL & Security Groups between subnets
 - Limit all outbound traffic

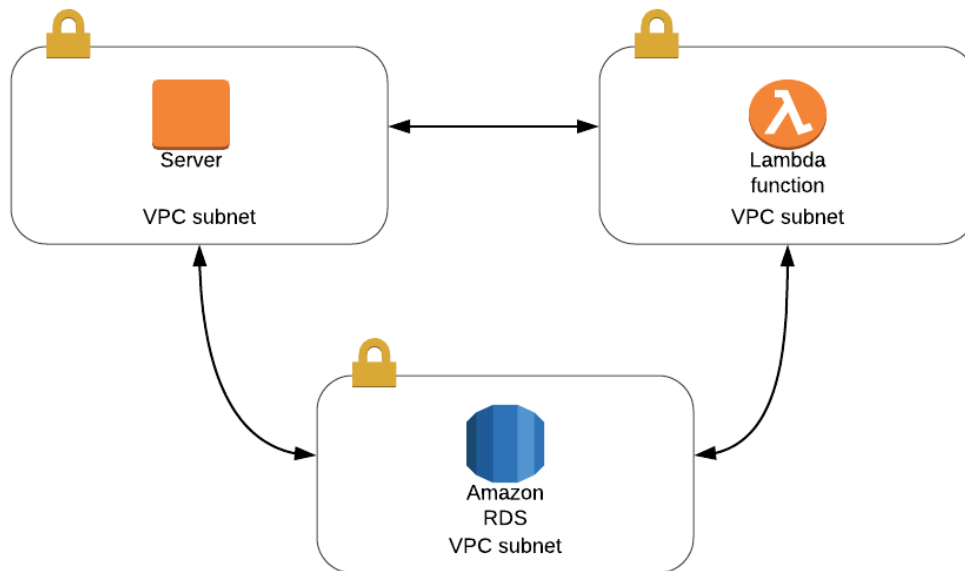


Network Architecture



BAD NETWORK

- NACLs are wide open
- Wide open inbound rules on security groups
- Security groups all everything to talk to internet

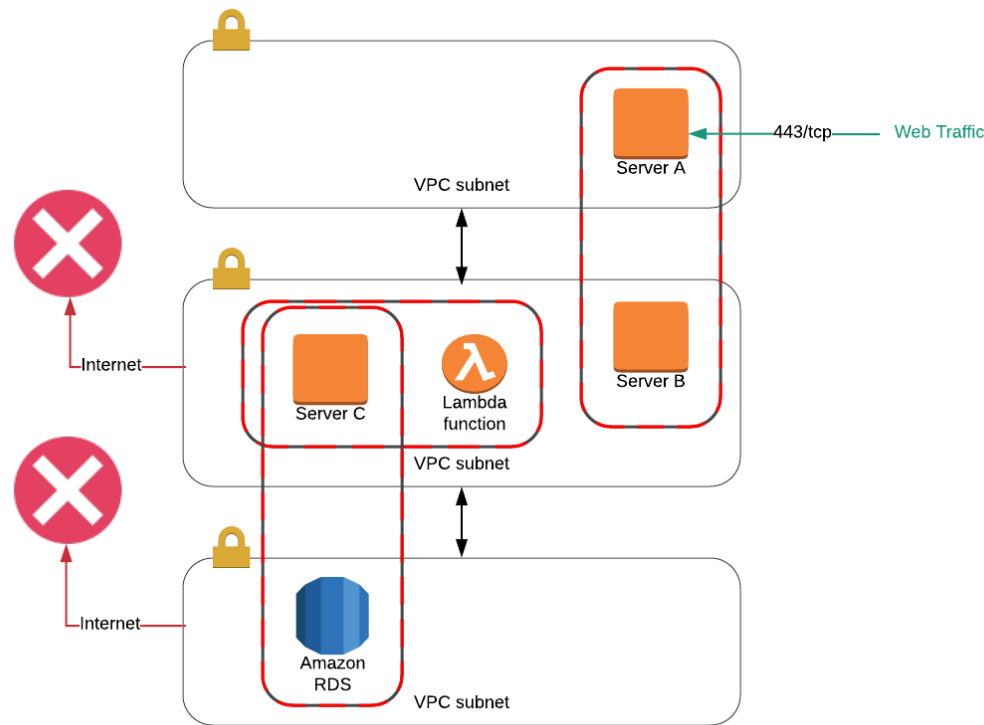


Network Architecture



BETTER NETWORK

- NACLs limit access between subnets
- Security Groups limiting access to specific servers
- Blocking internet where not needed



Conclusion



- Red Team:
 - Attackers can use the same tools used by DevOps teams.
 - Cloud APIs provide a means for mapping out an entire account.
 - Read only access can be powerful.
- Blue Team:
 - Restrict access
 - Automated deployment
 - Architect networks to minimize open ports and pivoting
 - Protect secrets - don't embed in code!
 - Monitor everything

RSA®Conference2018

San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CSV-R12

THANK YOU!



#RSAC

Teri Radichel

CEO
2nd Sight Lab
@teriradichel

Kolby Allen

DevOps Engineer
Zipwhip
@kolbyallen