Attacking and Defending AWS

Karan Raheja

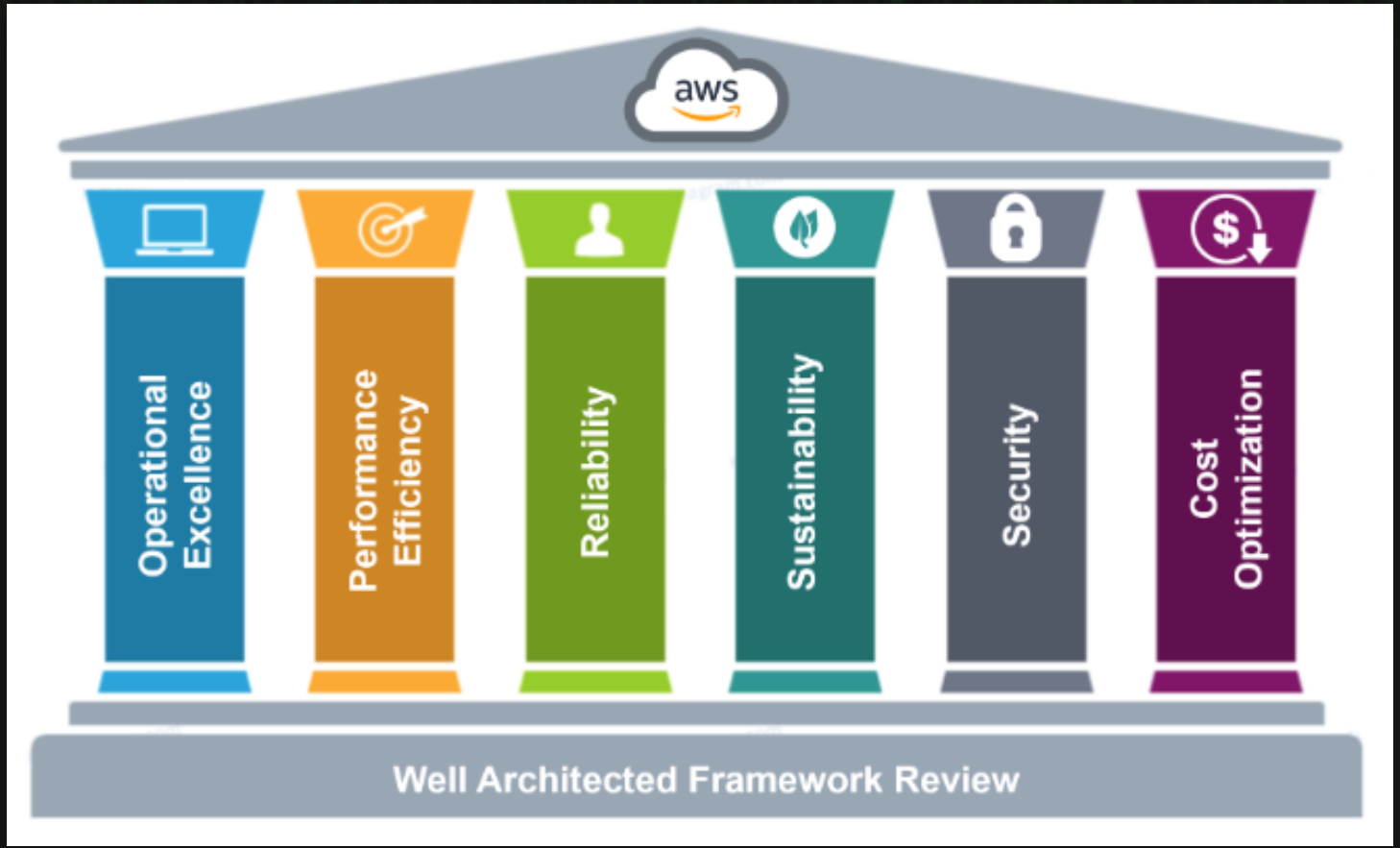
./About Me

- Working as a Security Engineer in a Fortune 100 Company
- Student of Web Application Security and Network Security
- Trying to delve into Cloud Security and DevSecOps space
- Delivered talks at Blackhat MEA’23 and The Hack Summit’23 conferences

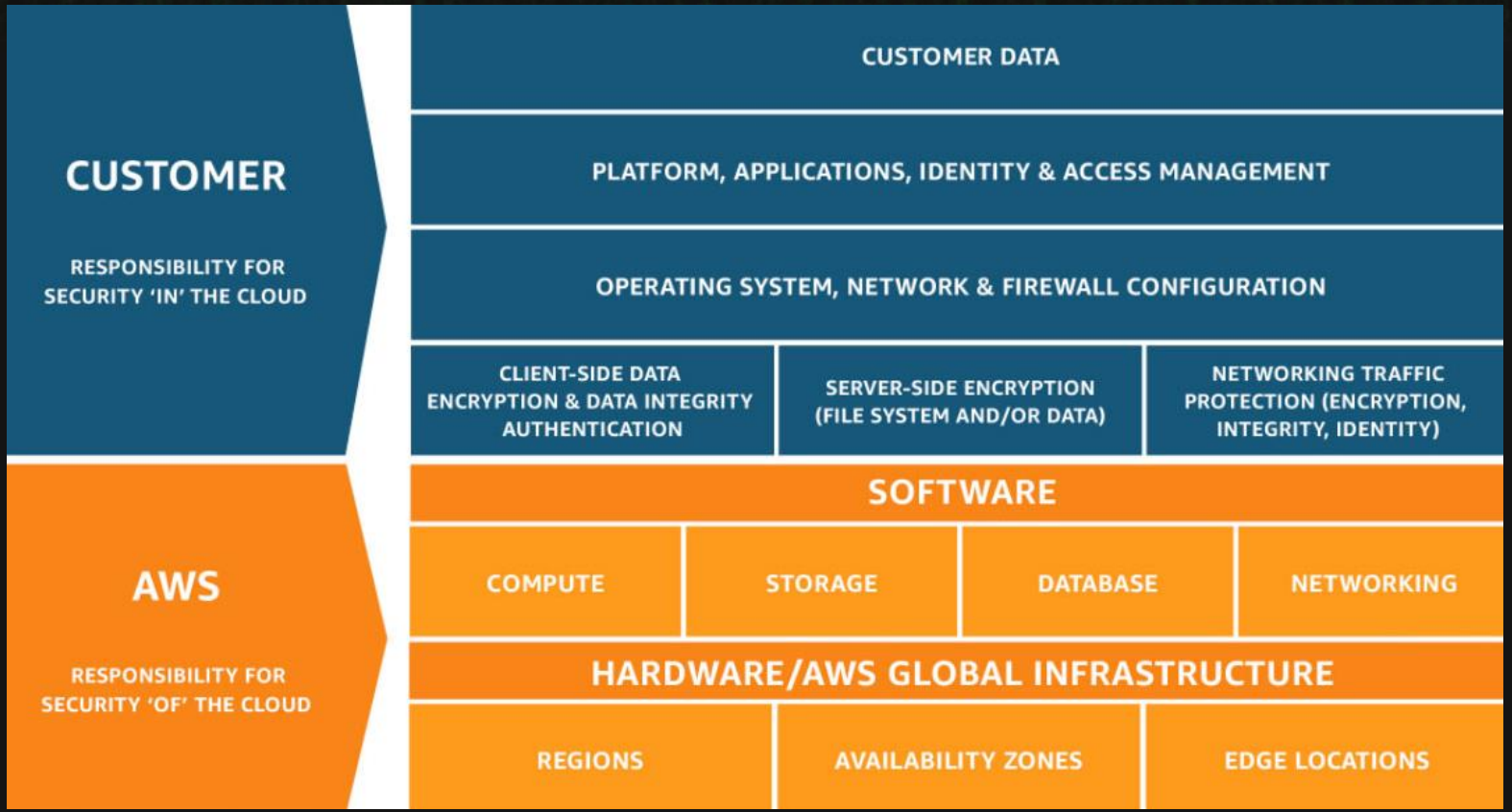
Agenda

- Well-architected framework
- Mindset that makes the difference
- AWS components
- Phishing in AWS environments
- Top 10 AWS security risks
- Attacking AWS through webapps
- Common AWS vulnerabilities
- Defending the enterprise cloud environment
- AWS MITRE ATT&CK framework

AWS Well Architected Framework



AWS Shared Responsibility Model



Attacker mindset for attacking the cloud

- If you know the enemy better, you are much more likely to win.
- Think outside in – what is exposed at the network layer
- Finding cloud provider weaknesses is more difficult than finding misconfigured resources belonging to an account.
- Look for reuse of stuff – users, roles, tokens, keys, secrets, passwords etc.
- Automate as much as possible

How do I get in?

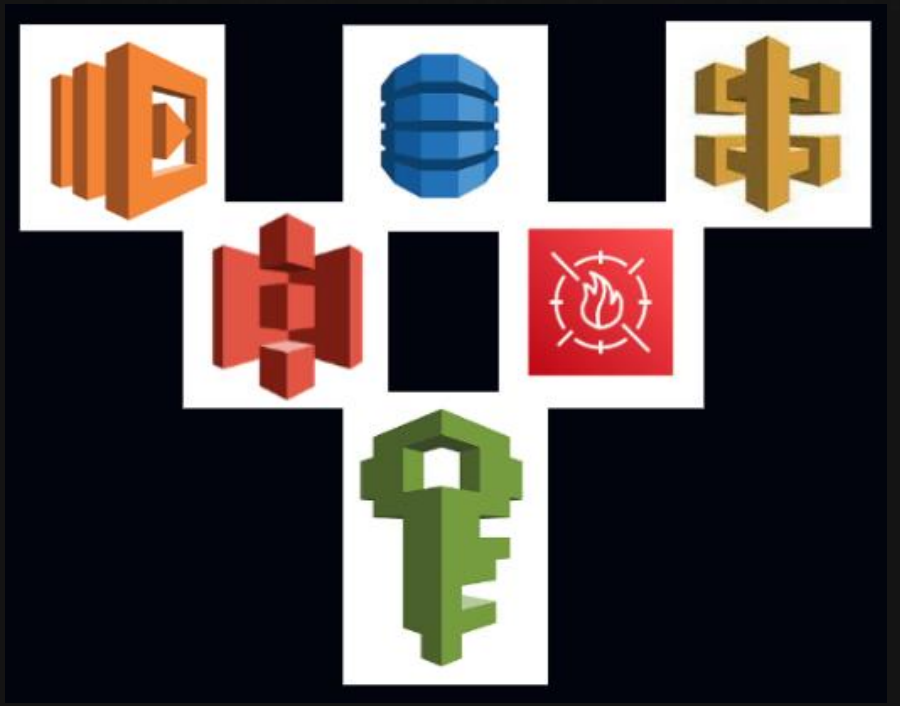
- Publicly exposed leaks(Access Keys/Passwords/Tokens) – OSINT
- Social Engineering
- Vulnerabilities in webapps hosted on AWS
- Supply Chain – 3rd party breach
- Internal employee threat –backdoors
- Compromising any exposed service

How do I get in - part II

- Almost always, the access is through leak/stealing access credentials and then moving laterally and horizontally in the network
- EC2, if exposed, can be scanned for any existing vulnerability.
- Any web application hosted on AWS infra can serve as initial foothold. Entry from web – SSRF, RCE, SSTI, LFI, RFI etc.
- Security Group/VPC misconfiguration can help attackers
- Once you're in, Pacu FTW :3 :3

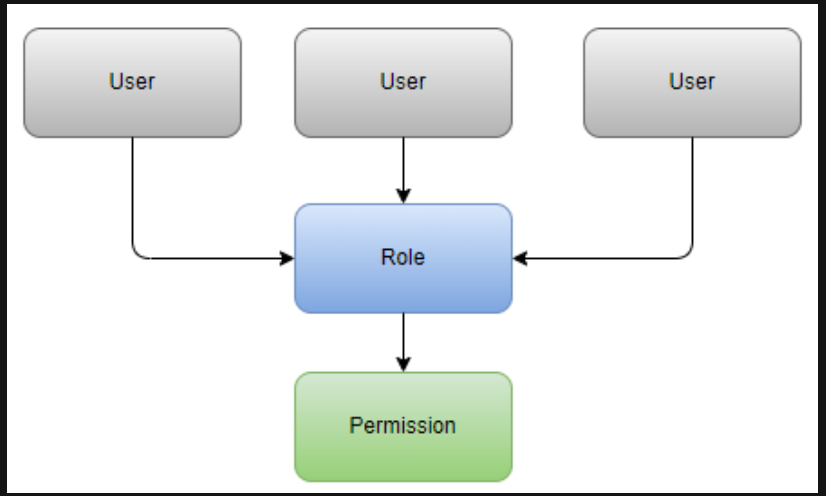
Resources to talk about

- IAM
- API Gateway
- EC2
- Lambda
- S3
- Cognito



Introduction to IAM

- Service that helps to securely control access to AWS resources
- Who, What, Which
- Primary purpose is role assignment and AuthN/AuthZ management
- Concept of IAM - RBAC
- IAM Principals, IAM Roles, IAM Users, IAM Groups



IAM Permissions

- Types of policies: AWS Managed Policies, Customer Managed Policies, Inline Policies
- Role Based Access Control, Attribute Based Access Control
- Granularity of permissions
- E.g NotAction, NotResource and NotPrincipal

IAM Credentials

- Used to authenticate & authorize access to AWS resources.
- Access Key ID, Secret Access Key
- IAM Roles
- Temporary Tokens
- AKIA/ASIA

Attacking IAM Misconfigurations

- Overly permissive IAM role can provide access to privileged resources
- “Resource”: “*” == Most likely misconfigured
- User has privileges to create/update policies
- Long-living Access Credentials
- IAM Users without MFA

Defending IAM

- Principle of Least Privileges
- Build the resource policy as fine-grained as possible
- Enable auto-rotation of credentials
- Enforce MFA for each account

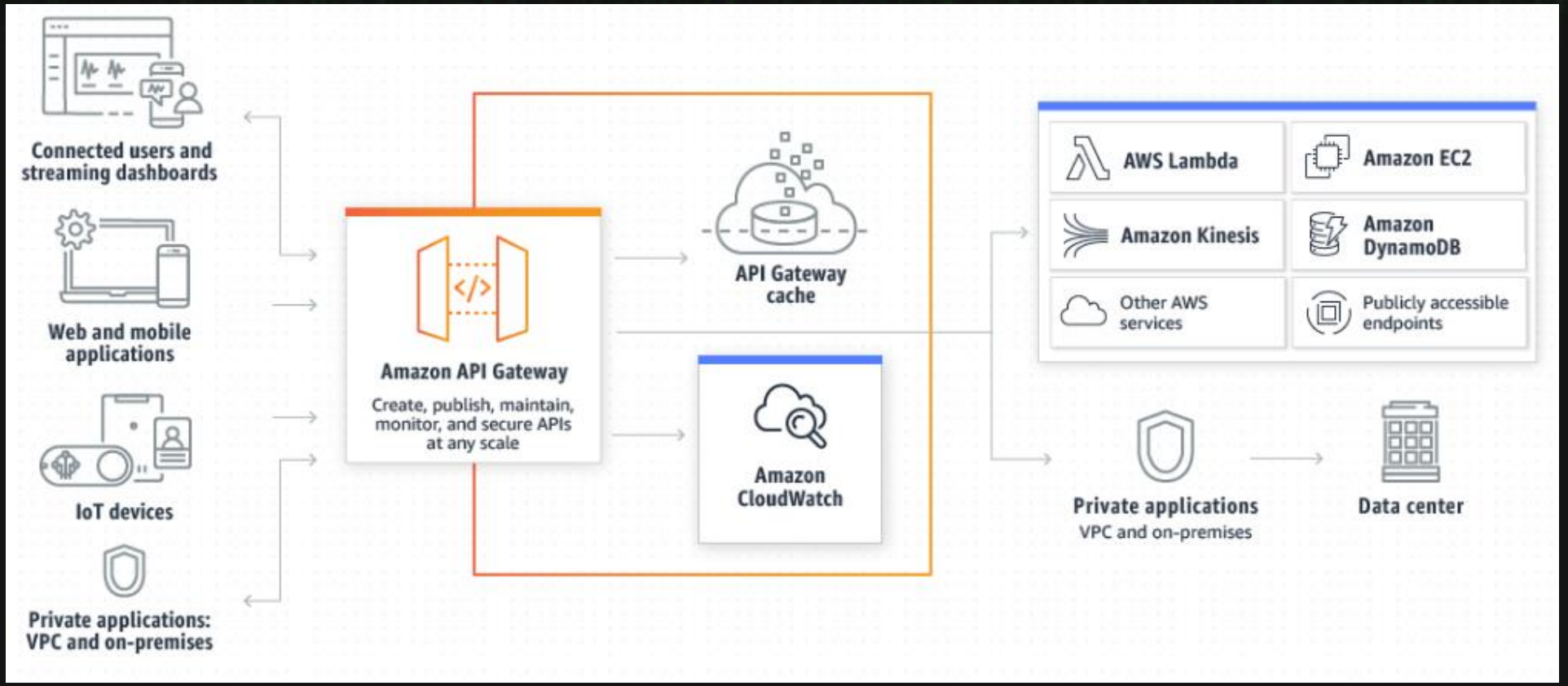
Principle of Least Privileges

- Granting minimum possible level of access required for completing a task
- Why to implement:
  - Reducing the attack surface
  - Managing access
  - Easing out compliance and audits

AWS API Gateways

- HTTP based
- Multi-Dimensional - Stateless client-server communication + websockets
- Service Endpoints
  - protocol://service-code.region-code.amazonaws.com
  - https://dynamodb.us-west-2.amazonaws.com
- https://{restapi_id}.execute-api.{region}.amazonaws.com/{stage_name}

AWS API Gateways

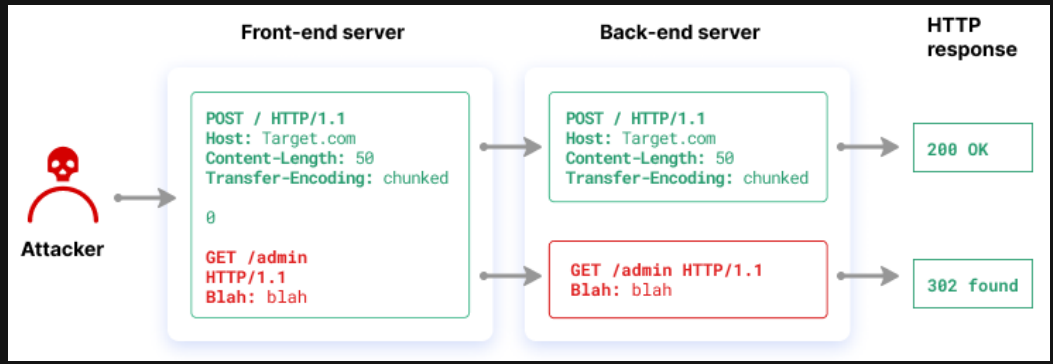


Attacking API Gateways

API Gateway Misconfigurations

- Lack of authentication on endpoints
- Misconfigured private API gateway
- Poorly configured API Gateway WAF
- Denial of Service

HTTP Header Smuggling attack ➔

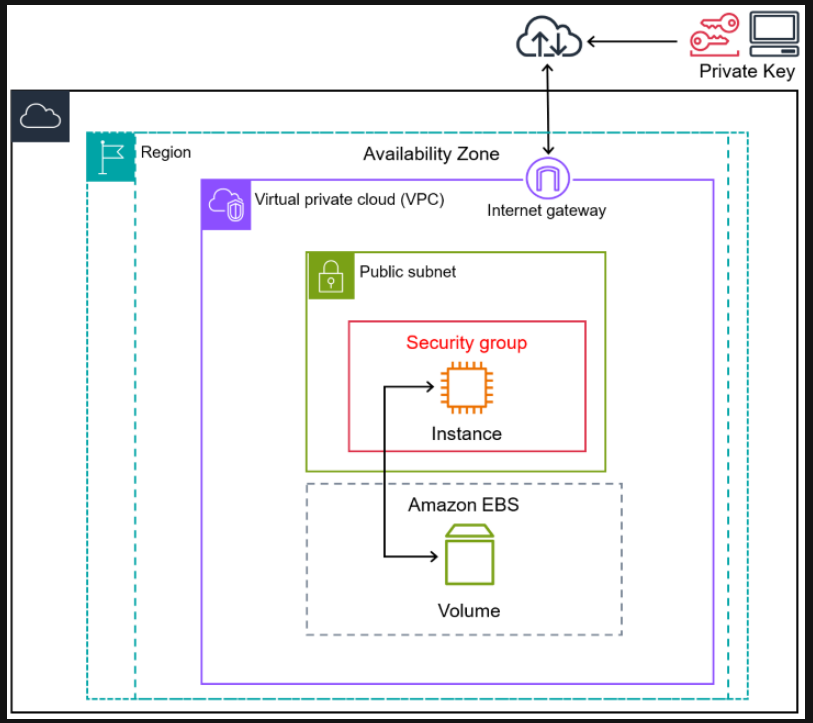


Defending API Gateways

- Enable Rate Limiting
- Implement logging and monitoring
- Implement Cloudwatch alarms along with Cloudtrail
- Monitor your security best practices using AWS Security Hub

EC2

- Compute in the Cloud
- Multiple advantages over on-prem alternatives (costing, reliability, uptime, maintenance, SeCuRiTy)
- Scale based on need



Attacking EC2 misconfigurations

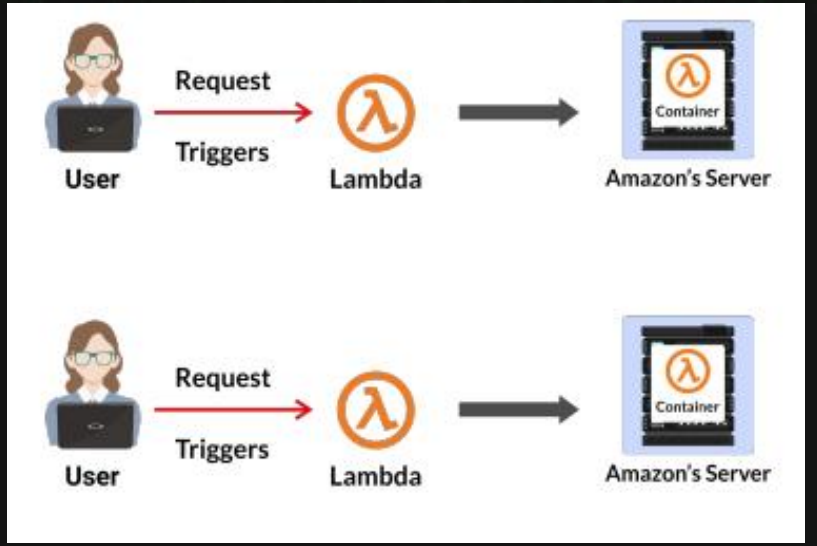
- Using public/unencrypted AMIs
- Deploying private EC2 instances in public subnets
- Stolen/Leaked credentials
- SSH brute-force (story-time)

Defending EC2

- Create separate instances for separate workloads
- Enforce least privileges in AWS IAM
- Secure the network at VPC level
- Monitor EC2 workloads for security issues
- Regularly patch and update existing EC2 workloads

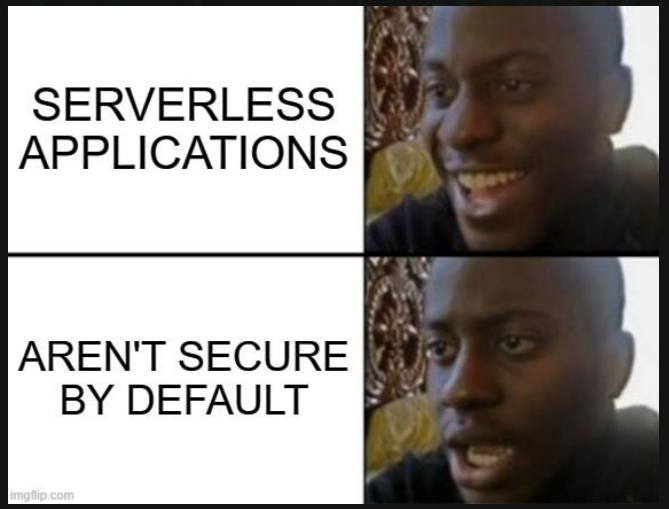
Understanding Lambda

- Function Code
- Highly Scalable
- Underlying servers are managed by AWS
- Some events which can be used to trigger lambda functions:
  - When a table is updated
  - When messages arrive
  - When objects in S3 are modified



Attacking Lambda

- Predictable API endpoint URL
- Web related vulnerabilities
- iam:PassRole, lambda:CreateFunction, lambda:InvokeFunction



Defending Lambda

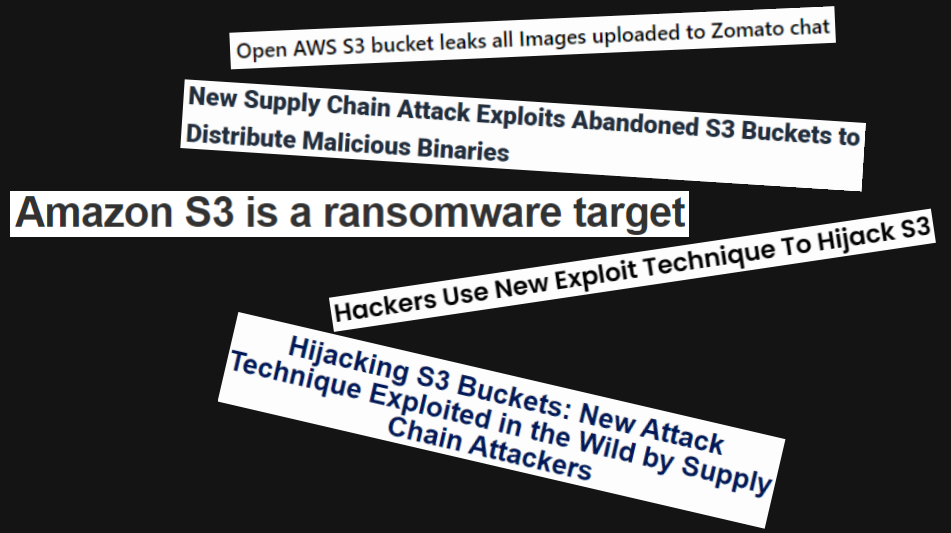
- One IAM role per function
- Protect the secrets
- API authorization
- WAF implementation
- Attack surface mapping

S3 – (Not-so)Simple Storage Service

- Globally Unique
- Object Versioning
- Most misconfigured
- Can be used for phishing
- Can be a goldmine of information if you are on the right place at the right time

Attacking S3 Misconfigurations

- Publicly exposed buckets/Public objects inside buckets
- Information present in plain text
- Access policy misconfiguration
- Absence of AuthN/AuthZ controls

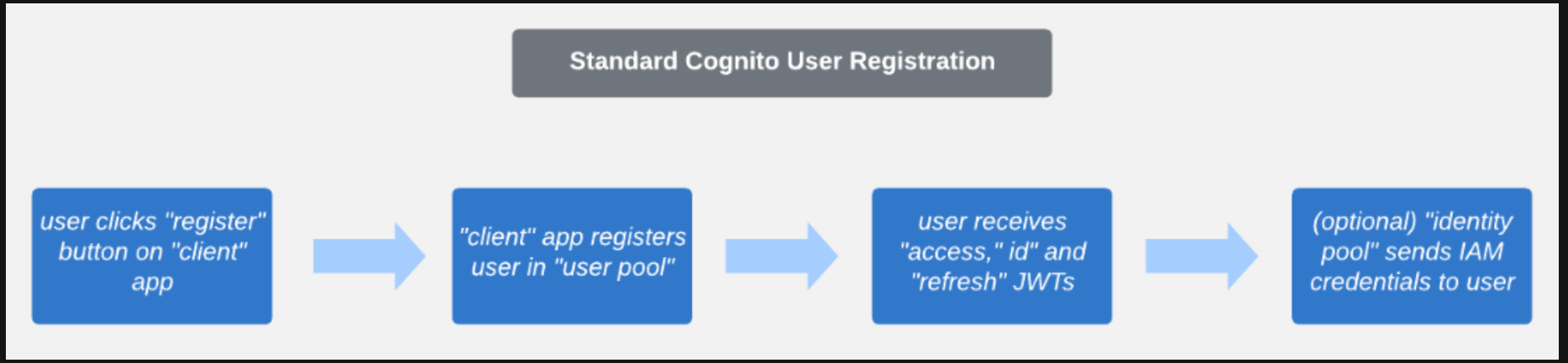


Defending S3

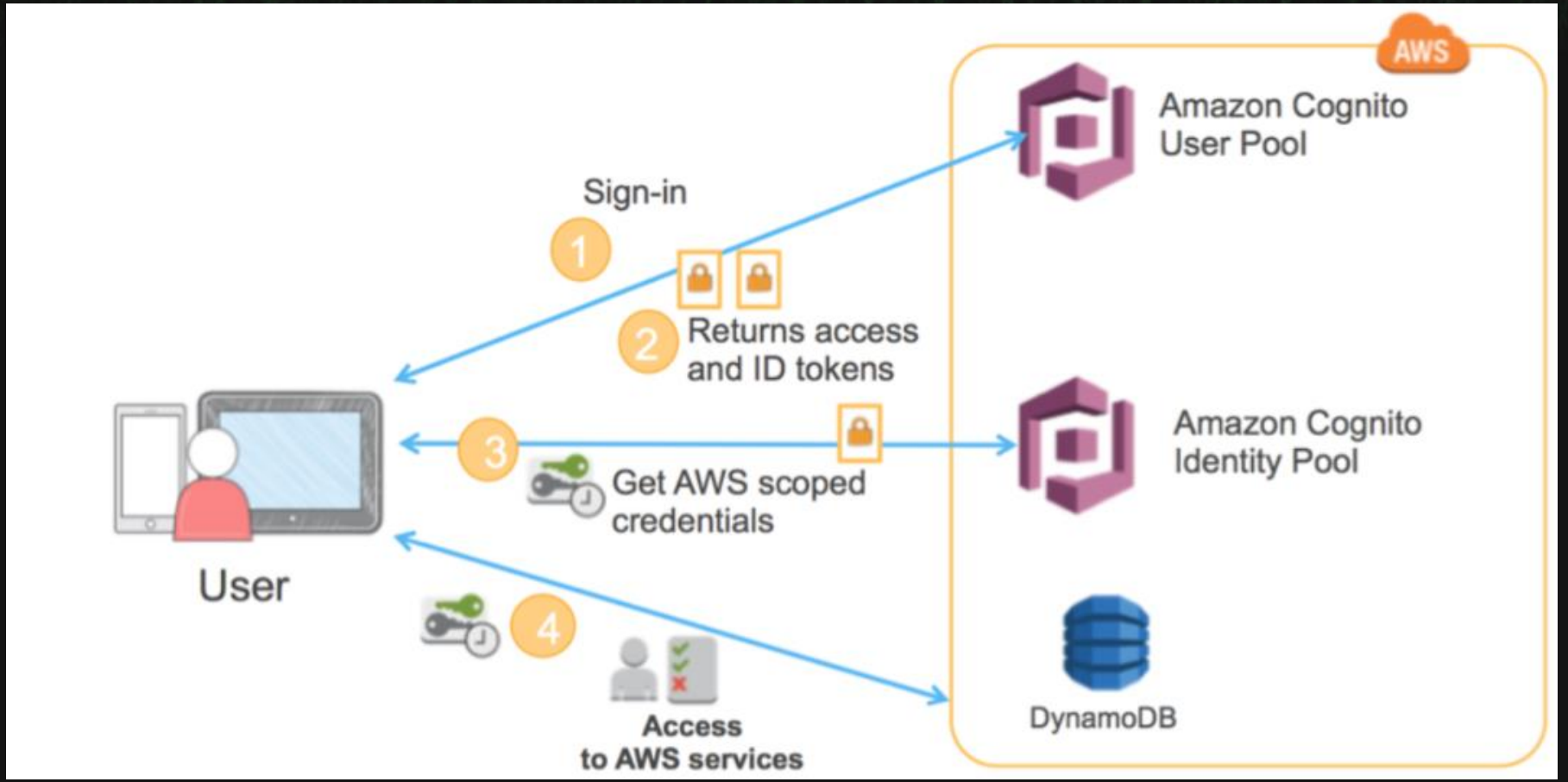
- Disable public access
- Attach relevant roles and policies to ensure least privileges
- Use assumed IAM roles which has read/write access to bucket
- Use pre-signed URLs for sharing

AWS Cognito

- Manages user authN/authZ for client applications
- Saves encrypted information as key/value pair in Amazon Cognito Sync Store
- User Pool vs Identity Pool



AWS Cognito


AWS
Amazon Cognito
User Pool
Sign-in
1
2
Returns access
and ID tokens
3
Get AWS scoped
credentials
Amazon Cognito
Identity Pool
User
4
Access
to AWS services
DynamoDB

Attacking AWS Cognito

- Cognito's "Identity pool" ID shown in HTTP responses ➔ obtain IAM credentials
- Cognito "client" & "user pool" IDs shown in HTTP responses + Cognito User registration enabled(default) ➔ register using the valid IDs
- Secrets leakage through JS files
- User enumeration

Defending AWS Cognito

- Ensure that no unintended functionalities are exposed to the users
- Implement MFA for each user
- Remove sensitive details from server responses
- Review IAM policies attached to the unauthenticated role to ensure least privileges

Subdomain Takeovers

- Dangling DNS Record → Resource (unclaimed) → Claimed → Subdomain Takeover
- EC2+DNS Record belongs to Org's AWS infra → Org deletes EC2 → Claim DNS Record
- Repeated efforts for getting IP/bucket name. If DNS Record points to that IP/bucket, you got a takeover. :3 :3 <3 <3

Phishing in AWS environments

- Traditional Phishing
  - Action Phishing: tricking a target to extract valuable info
  - Exploit Phishing: Delivering an exploit
  - Credential Phishing: Harvesting credentials
- Device authentication phishing
- CloudFormation stack phishing - https://console.aws.amazon.com/cloudformation/home?region=region#/stacks/new?stackName=stack_name&templateURL=template_location
- ACM Email validation phishing
- AWS SES verification phishing

Credits: https://ramimac.me/aws-phishing

MFA in AWS and Best Practices for Credentials Management

- Something you know + something you have/something you are
- MFA can be enabled at AWS account level for root and IAM users
- Hardware Security Key, Hardware TOTP token, Virtual authenticator
- Use temporary security credentials instead of permanent/long-term creds
- Remove AWS account root user access keys
- Promote usage of AWS Secrets Manager and AWS KMS

Attacking AWS metadata service

- Can access it through the application – could give total control
- Sensitive information in user startup scripts
- Where to look – functionality of fetching a page data and returning the info to an end user.
- http://169.254.169.254/latest/meta-data/iam/security-credentials/IAM_USER_ROLE ==> AccessKeyID, SecretAccessKey and Token

Differences between IMDSv1 and IMDSv2

IMDSv1	IMDSv2
Requires making HTTP GET requests	Session initiation through PUT requests
Uses a token-based approach for metadata access and retrieval	Eliminates the need for tokens by implementing secure, signed requests
Requests are not authenticated, hence vulnerable to SSRF	Requests are protected by a session token which is obtained via instance credentials
Request/Response method	Session-Oriented method

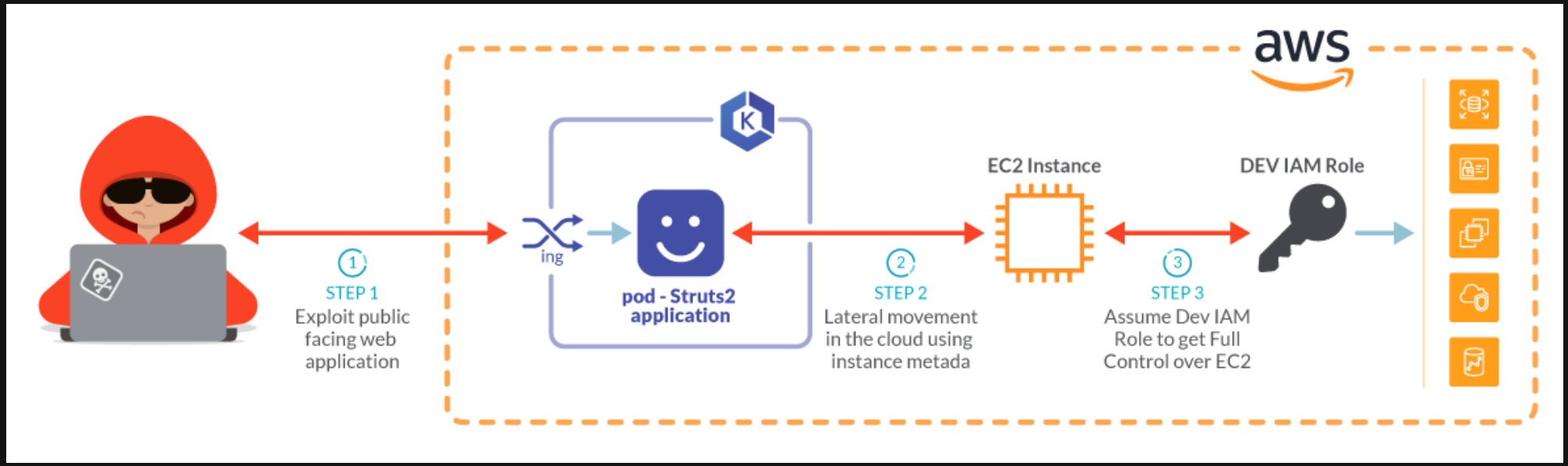
Top 10 AWS Security Risks

- Insecure S3 buckets
- IAM permissions
- Publicly exposed AMIs
- Lack of cloud security visibility
- Lack of defined roles and liability
- Unsecured sensitive data stored in the cloud
- Misconfiguration related vulnerabilities
- Vulnerabilities in source control and function repos
- Container vulnerabilities in Amazon Elastic Container Registry (ECR)
- Open-Source vulnerabilities

Attacking AWS through WebApps

- XSS
- SQL Injection
- Insecure Direct Object References
- Server-Side Request Forgery
- Sensitive Data Exposure
- Privilege Escalation

Sample Scenario



Common AWS Vulnerabilities

- Use of VPC’s default subnet
- IAM issues
- Misconfigured S3 buckets
- Exposed database origin servers
- Hanging DNS records
- Exposed credentials, keys and tokens

Top 10 Attacks in AWS -2023

- Hunting leaked IAM keys and gaining persistence with federation token
- Hunting for public S3 buckets
- Privilege Escalation through IAM permissions
- Stealing EC2 credentials via SSRF
- Hijacking public EBS Snapshots
- Compromising EC2 via instance user data
- Privilege Escalation via IAM:PassRole misconfiguration
- Discovering and stealing data from public SQS and SNS queues
- Exploiting lambda execution roles
- Subdomain takeover using S3

Source: https://www.immersivelabs.com/blog/top-10-aws-attacker-techniques-2023/

Lesser-known AWS attacks+defenses

- **Initial access**: Backdoor community resources (e.g. AMIs, CloudFormation templates, Lambda Layers, etc.) or phish with Stack Sets.
  - **Defense**: Consider using Infrastructure as Code scanning tools to enforce secure defaults and resources that are allowed to be used.
- **Recon** Abuse naming patterns to guess resource IDs (like S3 bucket names) or fingerprint existing roles or services or vendors in use.
  - **Defense**: Follow least privilege so that even resources with known names cannot be accessed unless needed, and consider randomizing resource names.
- **Lateral movement**: Abusing trust and privileges across accounts (IAM, network-level, etc.).
  - **Defense**: Follow least privilege for cross account trust, assess if your cloud security posture has a "soft center," that if an attacker gets inside it's game over.
- **Exfiltration**: Share compromised resources to an account you control to speed exfiltration, or use DNS for stealthy exfiltration.
  - **Defense**: Set up auto-remediation that will automatically unshare resources shared with unknown accounts, and turn on logging for any VPC DNS resolvers. If you want to have an isolated network, consider running your own DNS resolver and disabling the one run by AWS.

Credits: https://tldrsec.com/p/blog-lesser-known-aws-attacks

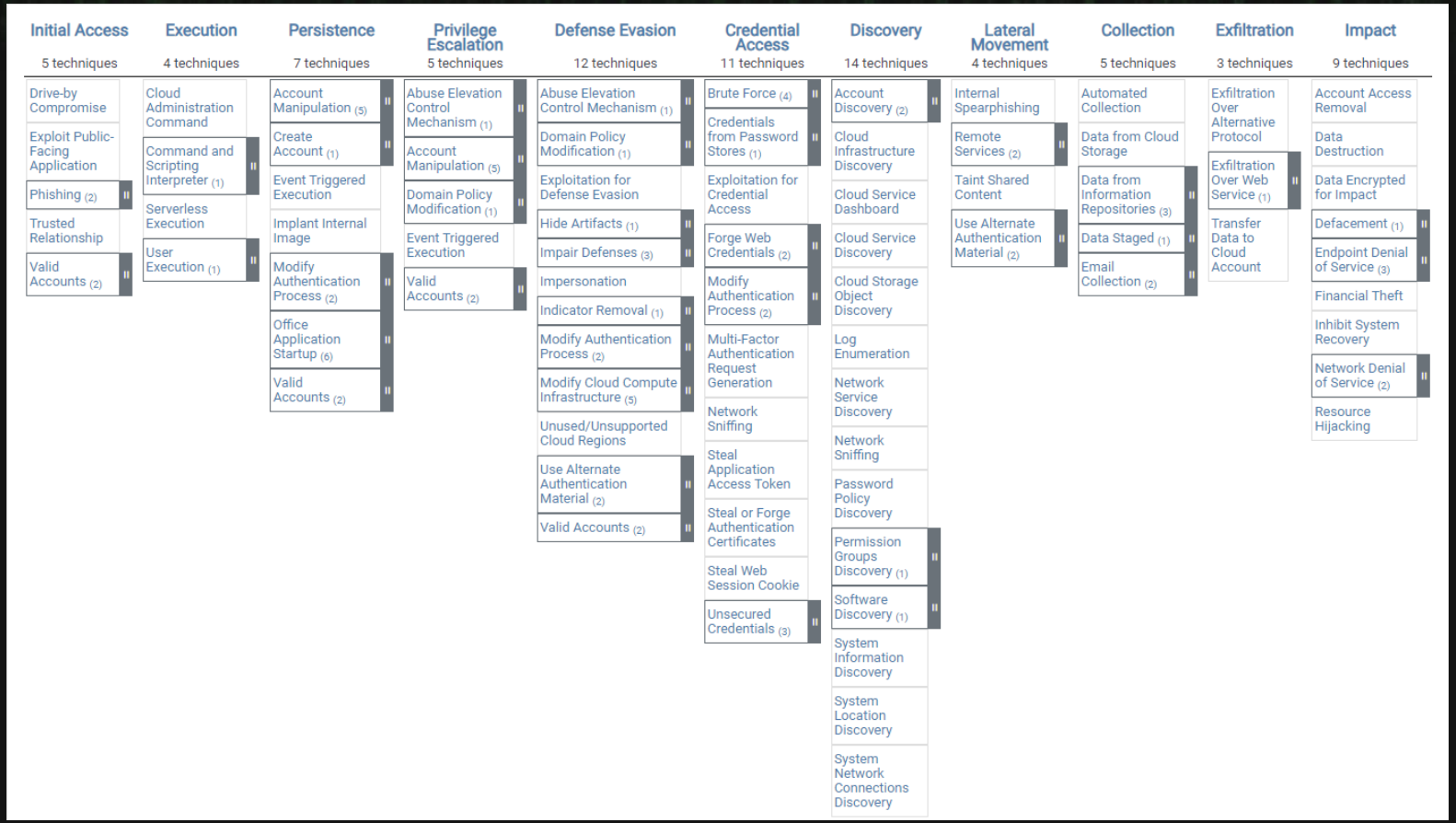
Defending Enterprise AWS

- Implement strong access control measures (P.o.L.P.)
- Regularly update and patch AWS resources
- Enable MFA for added security
- Implement proper network segmentation and robust network security controls
- Protect data in transit and at rest
- Implement continuous logging and monitoring
- Regularly backup your data and test DR/BCP procedures

Defending Enterprise AWS

- Conduct regular security testing, audits and penetration testing activities
- Establish a robust incident response plan
- Stay informed with the latest threats and industry best practices
- Minimize the attack surface by maintaining an asset inventory
- Automate security best practices
- Thoroughly understand the AWS Security and Compliance Shared Responsibility model

Cloud MITRE ATT&CK



Shoutouts

- Jatin Sethi Sir
- Ashwath Sir
- Manish Sir
- Hacktricks
- hackingthe.cloud
- Cloud Security community doing YouTube videos, medium posts and other such informative blogs and articles

thanks for your attention

-

time for questions