

CLOUD COMPUTING

Cloud computing delivers various types of services and applications over the Internet. These services enable users to use software and hardware managed by third parties at remote locations. Some well-known cloud service providers are Google, Amazon, and Microsoft.

Details of Cloud Computing

- Cloud computing refers to on-demand delivery of IT capabilities, in which IT infrastructure and applications are provided to subscribers as metered services over a network. Cloud services are classified into three categories, namely infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS), which offer different techniques for developing cloud.

Enumerate S3 Buckets using lazys3

- lazys3 is a Ruby script tool that is used to brute-force AWS S3 buckets using different permutations. This tool obtains the publicly accessible S3 buckets and also allows you to search the S3 buckets of a specific company by entering the company name.
- `ruby lazys3.rb [Company]`

Enumerate S3 Buckets using S3Scanner

- S3Scanner is a tool that finds the open S3 buckets and dumps their contents. It takes a list of bucket names to check as its input. The S3 buckets that are found are output to a file. The tool also dumps or lists the contents of "open" buckets locally.
- `python3 ./s3scanner.py sites.txt`
- Dump all open buckets and log both open and closed buckets in found.txt:
 - `python3 ./s3scanner.py --include-closed --out-file found.txt --dump names.txt`
- Just log open buckets in the default output file (buckets.txt):
 - `python3 ./s3scanner.py names.txt`

- Save the file listings of all open buckets to a file:

- `python ./s3scanner.py --list names.txt`

S3 buckets are used by customers and end users to store text documents, PDFs, videos, images, etc. To store all these data, the user needs to create a bucket with a unique name.

Listed below are several techniques that can be adopted to identify AWS S3 Buckets:

- Inspecting HTML: Analyze the source code of HTML web pages in the background to find URLs to the target S3 buckets
- Brute-Forcing URL: Use Burp Suite to perform a brute-force attack on the target bucket's URL to identify its correct URL
- Finding subdomains: Use tools such as Findsubdomains and Robtex to identify subdomains related to the target bucket
- Reverse IP Search: Use search engines such as Bing to perform reverse IP search to identify the domains of the target S3 buckets
- Advanced Google hacking: Use advanced Google search operators such as "inurl" to search for URLs related to the target S3 buckets

Exploit Open S3 Buckets using AWS CLI

- The AWS command line interface (CLI) is a unified tool for managing AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.
- `aws configure`
 - It will ask for the following details:
 - AWS Access Key ID
 - AWS Secret Access Key
 - Default region name
 - Default output format
- `aws s3 ls s3://[Bucket Name]`
- `aws s3 mv Hack.txt s3://certifiedhacker1`
- `aws s3 rm s3://certifiedhacker1/Hack.txt`

Overview of Privilege Escalation

- Privileges are security roles assigned to users for using specific programs, features, OSes, functions, files, code, etc. to limit access depending on the type of user. Privilege escalation is required when you want to access system resources that you are not authorized to access. It takes place in two forms: vertical and horizontal.
- Horizontal Privilege Escalation: An unauthorized user tries to access the resources, functions, and other privileges of an authorized user who has similar access permissions
- Vertical Privilege Escalation: An unauthorized user tries to access the resources and functions of a user with higher privileges such as application or site administrators

Escalate IAM User Privileges by Exploiting Misconfigured User Policy

- A policy is an entity that, when attached to an identity or resource, defines its permissions. You can use the AWS Management Console, AWS CLI, or AWS API to create customer-managed policies in IAM. Customer-managed policies are standalone policies that you administer in your AWS account. You can then attach the policies to the identities (users, groups, and roles) in your AWS account. If the user policies are not configured properly, they can be exploited by attackers to gain full administrator access to the target user's AWS account.

```
aws iam create-policy --policy-name user-policy --policy-document file://user-policy.json
```

- user-policy.json

```
{  
  
  "Version": "2012-10-17",  
  
  "Statement": [  
    {  
  
      "Effect": "Allow",  
  
      "Action": "*",  
  
      "Resource": "*"
```

```
}  
]  
}
```

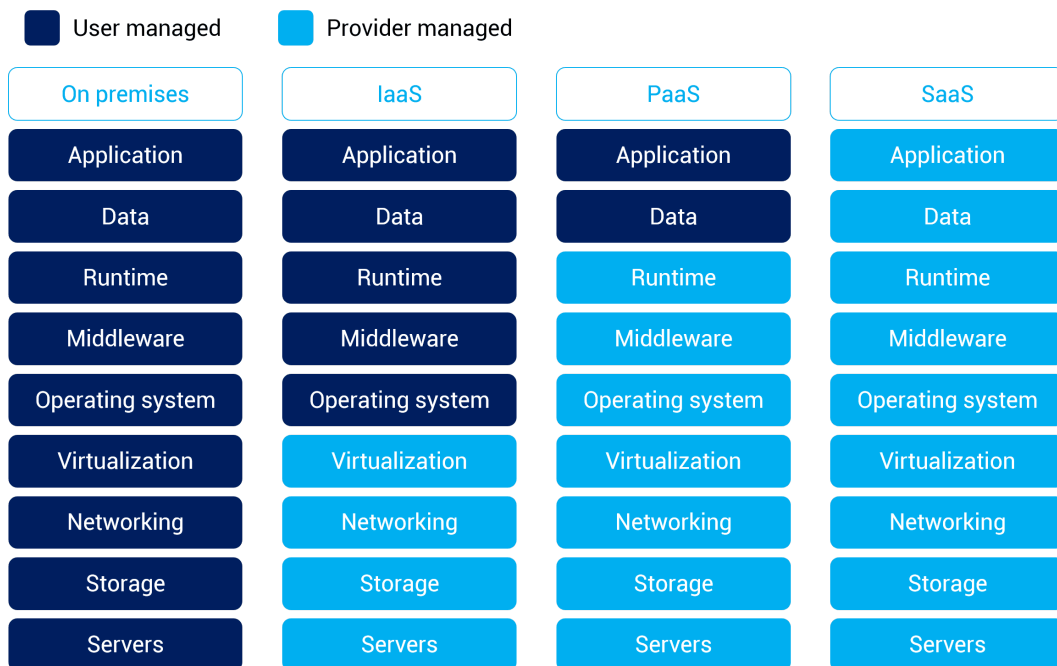
- `aws iam attach-user-policy --user-name [Target Username] --policy-arn arn:aws:iam::[Account ID]:policy/user-policy`
- `aws iam list-attached-user-policies --user-name [Target Username]`
- `aws iam list-users`
- List of S3 buckets: `aws s3api list-buckets --query "Buckets[].Name"`
- User Policies: `aws iam list-user-policies`
- Role Policies: `aws iam list-role-policies`
- Group policies: `aws iam list-group-policies`
- Create user: `aws iam create-user`

Cloud Computing Basics

- **Three Types of Service Models:**
 - **Infrastructure as a Service (IaaS)**
 - Provides virtualized computing resources
 - Third party hosts the servers with hypervisor running the VMs as guests
 - Subscribers usually pay on a per-use basis
 - e.g: AWS, Microsoft Azure, Digital Ocean, Google Cloud
 - **Platform as a Service (PaaS)**
 - Geared towards software development
 - Hardware and software hosted by provider
 - Provides ability to develop without having to worry about hardware or software
 - e.g: Heroku, Salesforce
 - **Software as a Service (SaaS)**

- Provider supplies on-demand applications to subscribers
- Offloads the need for patch management, compatibility and version control
 - e.g: Microsoft Office 365, Dropbox storage, Google Docs.

Tech stack	Type
Software	SaaS
Apps	PaaS
OS	IaaS
Virtualization	managed by provider
Storage/Networking	managed by provider



Cloud Deployment Models

- **Private Cloud** - Cloud solely for use by one tenant; usually done in larger organizations.
- **Community Cloud** - Is made up of infrastructure from several different entities which may be cloud providers, business partners, and so on. (members only type of thing)

- **Public Cloud** - Services provided over a network that is open for public to use; Amazon S3, Microsoft Azure - Open for business.
- **Hybrid Cloud** - A composition of two or more cloud deployment models.

NIST Cloud Architecture

The NIST cloud computing reference architecture (NIST SP 500-292) define five major actors; Each actor is an entity (a person or an organization) that participates in a transaction or process and/or perform tasks in cloud computing.

- **Cloud Consumer** - A person or org. that maintains a business relationship with, and use services from Cloud Providers; acquires and uses cloud products and services.
- **Cloud Provider** - A person, org. or entity responsible for making a service available; Purveyor of products and services.
- **Cloud Auditor** - Independent assessor of cloud service and security controls.
- **Cloud Broker** - Manages use, performance and delivery of services as well as relationships between Cloud Providers to Cloud consumers.
- **Cloud Carrier** - Organization with responsibility of transferring data; Intermediary that provides connectivity and transport of Cloud services from Cloud providers to Cloud consumers. (e.g: Telecom's)

! - FedRAMP - regulatory effort regarding cloud computing

! - PCI DSS - deals with debit and credit cards, but also has a cloud SIG

Five characteristics of cloud computing

The National Institute of Standards and Technology (NIST) defines cloud computing as it is known today through five particular characteristics.

1. **On-demand self-service**
2. **Broad network access**
3. **Multi-tenancy and resource pooling**
4. **Rapid elasticity and scalability**
5. **Measured service**

Threats:

- **Data Breach or Loss** - Biggest threat; includes malicious theft, erasure or modification
- **Shadow IT** - IT systems or solutions that are developed to handle an issue but aren't taken through proper approval chain
- **Abuse of Cloud Resources** - Another high threat (usually applies to IaaS and PaaS)
- **Insecure Interfaces and APIs** - Cloud services can't function without them, but need to make sure they are secure
- **Service Oriented Architecture** - API that makes it easier for application components to cooperate and exchange information
- **Insufficient due diligence** - Moving an application without knowing the security differences
- **Shared technology issues** - Multitenant environments that don't provide proper isolation
- **Unknown risk profiles** - Subscribers simply don't know what security provisions are made in the background
- **Wrapping Attack** - SOAP message intercepted and data in envelope is changed and sent/replayed
- **Session riding** - CSRF under a different name; deals with cloud services instead of traditional data centers
- **Others include malicious insiders, inadequate design and DDoS**
 - Other threats:
 - Loss/compromise of encryption keys
 - Isolation failure
 - Compliance risk
 - VM vulnerabilities
 - Vendor lock-on
 - Jurisdictional issues based on changing geographic boundaries
 - E-discovery/subpoena

- Cloud service termination/failure
- Improper/incomplete data handling & disposal
- Management network failure/interface compromise

Attacks:

1. Service hijacking via Social engineering & network sniffing
2. Session hijacking using XSS
3. DNS attacks
4. Side channel attacks - (e.g.: Using an existing VM on the same physical host to attack another)
5. Cross VM attacks
6. SQL injection
7. Cryptanalysis attacks
8. Wrapping attacks - performed during the translation of SOAP messages in the TLS layer; attackers duplicate the body of the message and send it to the targeted server impersonating the legitimate user.
9. DoS/DDoS attack
10. Main-in-the-Cloud attacks - abuse of cloud file synchronization services by tracking the user into installing malicious software that places the attacker's synchronization token for the service on their machine, allowing the attacker to steal the user's token and gain access to their files.

OWASP Top 10 Application Security Risks

1. **Injection** - Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
 - Input validation
 - Limit account privileges
2. **Broken Authentication** - Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other

implementation flaws to assume other users' identities temporarily or permanently.

3. **Sensitive Data Exposure** - Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
4. **XML External Entities (XXE)** - Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
 - If your application uses SAML for identify processing with federated security or Single Sign on (SSO). SAML uses XML.
 - If applications accepts XML directly or XML uploads from untrusted sources, or inserts untrusted data into XML documents.
 - Any of XML processors in the application or SOAP based web services that have (DTDs) enabled.
5. **Broken Access Control** - Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
6. **Security Misconfiguration** - is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
7. **Cross-Site Scripting XSS** - occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts

in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

- Reflected XSS
- Stored XSS
- DOM XSS

8. **Insecure Deserialization** - often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
9. **Using Components with Known Vulnerabilities** - Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
10. **Insufficient Logging & Monitoring** - Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Additional Attacks

1. **Directory Traversal (../)** - An attacker can get sensitive information like the contents of the /etc/passwd file that contains a list of users on the server; Log files, source code, access.log and so on
2. **Cross-site Request Forgery (CSRF)** - Forces an end user to execute unwanted actions on an app they're already authenticated on
 - Inherits identity and privileges of victim to perform an undesired function on victim's behalf
 - Captures the session and sends a request based off the logged in user's credentials
 - Can be mitigated by sending **random challenge tokens**

Cloud Security Control Layers

Problem with cloud security is what you are allowed to test and what should you test; Another concern is with a hypervisor, if the hypervisor is compromised, all hosts on that hypervisor are as well.

1. **Applications** - SDCL (Software development cycle), WAF (web application firewall)
 2. **Information** - DLP, encryption
 3. **Management** - GRC, IAM , Patch & Configuration
 4. **Network** - NIDS/NIPS, DNSSEC, QoS
 5. **Trusted Computing Model** - attempts to resolve computer security problems through hardware enhancements
 - **Roots of Trust (RoT)** - set of functions within TCM that are always trusted by the OS
1. **Computer & Network Storage** - Encryption, Host-based firewall, HIDS/HIPS
 2. **Physical** - Guards, Gates, Fences etc.

Tools

- **CloudInspect** - pen-testing application for AWS EC2 users
- **CloudPassage Halo** - instant visibility and continuous protection for servers in any cloud
- **Dell Cloud Manager**
- **Qualys Cloud Suite**
- **Trend Micro's Instant-On Cloud Security**
- **Panda Cloud Office Protection**













Cloud Hacking Labs + Hands-On Tools

Supplement: "Cloud Computing for Ethical Hackers"



TOOLS YOU'LL BE USING

Tool	Use	Platform
 AWS CLI	Command-line access to AWS	Linux/Mac/Windows

Tool	Use	Platform
 ScoutSuite	Multi-cloud security auditing	AWS, GCP, Azure
 Pacu	AWS exploitation framework	AWS
 S3Scanner	Discover public AWS S3 buckets	AWS
 Cloudsplaining	Analyze IAM security risks	AWS
 Burp Suite	Test APIs and SSRF flaws	All
 Amass	Cloud asset and subdomain discovery	All
 AWS CloudTrail	Event logging and tracking	AWS
 CloudFox	Enumeration in cloud environments	AWS
 GitLeaks	Detect secrets in repos	All

Lab 1: Cloud Subdomain & Asset Discovery

Objective: Discover cloud-based services of a target.

Tools:

- `amass`
- `whois`
- `crt.sh`

Steps:

```
bash
CopyEdit
amass enum -d example.com -o subdomains.txt
```

Then manually verify each subdomain to see if it's hosted on AWS/Azure/GCP.

✅ Output: List of cloud endpoints and possible entry points.

Lab 2: Enumerating Public S3 Buckets

Objective: Find and access misconfigured AWS S3 buckets.

Tools:

- S3Scanner
- AWS CLI

Steps:

```
bash
CopyEdit
s3scanner --bucket-list buckets.txt
```

```
bash
CopyEdit
aws s3 ls s3://vulnerable-bucket-name --no-sign-request
```

✅ Output: View contents of an open S3 bucket.

📸 Screenshot Task: List files like `passwords.txt`, `config.env`.

🔥 Lab 3: IAM Misconfiguration Hunting

Objective: Enumerate and analyze IAM roles, users, and policies.

Tools:

- Pacu
- Cloudsplaining

Steps in Pacu:

```
bash
CopyEdit
pacu
> use iam__enum_users_roles_policies_groups
```

Analyze with Cloudsplaining:


```
bash
CopyEdit
cloudsplaining scan --input-file policies.json
```

✓ Output: Privilege escalation opportunities and over-permissioned roles.

Lab 4: Attacking EC2 Metadata with SSRF

Objective: Exploit SSRF to steal IAM tokens via EC2 metadata.

Tools:

- Vulnerable EC2 web app
- 

Target:

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

✓ Output: Capture IAM access keys via SSRF.

⚠ Note: Do this **only in a test environment** (e.g., with a deliberately vulnerable EC2 instance).

Lab 5: Scanning and Auditing with ScoutSuite

Objective: Perform a multi-service security audit.

Tools:

- 

Steps:

```
bash
CopyEdit
```

```
scout aws --profile xsploit-profile
```

✓ Output: HTML report showing exposed services, IAM issues, open ports, etc.

📸 Screenshot Task: Highlight high-risk findings in IAM, EC2, S3.

🔥 Lab 6: Privilege Escalation in AWS (Pacu)

Objective: Use Pacu to escalate privileges via policy loopholes.

Steps:

```
bash
CopyEdit
> use iam__privesc_scan
```

✓ Output: Paths to escalate low-privilege roles to Admin access.

🔥 Lab 7: Enable Logging & Simulate Attacks

Objective: Enable CloudTrail, simulate malicious actions, and detect them.

Tools:

- AWS CLI
- AWS Console

Steps:

1. Enable CloudTrail from Console or CLI.
2. Create/delete an S3 bucket.
3. Analyze logs in CloudTrail dashboard.

✓ Output: Evidence of attacker actions in logs.

📸 Screenshot Task: Log entry showing `s3:DeleteBucket` action.

🔥 Lab 8: Secrets Detection in Code Repos

Objective: Discover exposed cloud credentials in public/private repos.

Tools:

- `GitLeaks`

```
bash
CopyEdit
gitleaks detect --source=.
```

✓ Output: Found keys like `AWS_SECRET_ACCESS_KEY` .

🔥 Lab 9: Serverless Vulnerabilities (AWS Lambda)

Objective: Discover insecure serverless functions.

Tools:

- AWS Lambda
- Burp Suite (to send inputs)

Attack:

- Attempt command injection in input to Lambda function.
- Try accessing metadata endpoint.

✓ Output: Execution of arbitrary commands or token leaks.

🔥 Lab 10: Full Cloud Hacking Simulation (CTF Lab)

| Create a vulnerable AWS environment for students to attack.

Environment:

- Public S3 bucket with sensitive data
- Weak IAM roles
- Disabled CloudTrail
- Lambda with SSRF
- EC2 open to the internet

🎯 Student Objectives:

- Enumerate assets
- Find the S3 bucket and download sensitive data
- Exploit IAM or Lambda to gain Admin access
- Enable CloudTrail and secure the environment

🏆 Optional: Grading based on flags captured or logs submitted

🎓 BONUS: Cloud Quiz Examples

Q1: Which AWS service logs all API actions?

- A. S3
- B. IAM
- C. CloudTrail ☒
- D. Lambda

Q2: Which endpoint is used to access EC2 metadata?

- A. `localhost:8080/meta`
- B. `http://metadata.aws`
- C. `http://169.254.169.254/latest/meta-data/` ☒

Cloud security tools

- CloudInspect
 - Penetration-testing as a service from Amazon Web Services for EC2 users
- CloudPassage Halo
 - Automates cloud computing security and compliance controls
- privacy.sexy
 - Open-source solution to increase privacy by reducing third party cloud-based data collection
 - Can also be used to harden virtual machine images and OSes that are talking to cloud services