# Module 19: Cloud Computing - Practical Guide

This guide details the specialized tools and hands-on laboratory procedures for auditing and securing cloud environments (AWS, Azure, GCP).

## 1. Comprehensive Cloud Toolset

### Discovery & Enumeration

* **S3Scanner:** Automatically finds publicly accessible S3 buckets by scanning lists of common names.
* **CloudEnum:** Multi-cloud OSINT tool to find public resources (Buckets, Apps, Azure Containers) in AWS, Azure, and GCP.
* **Shodan:** Used to find exposed cloud instances by searching for specific cloud-provider IP ranges.

### Auditing & Exploitation

* **Pacu:** The "Metasploit of AWS." An exploitation framework used to test the security of AWS environments via various modules.
* **Prowler:** A security tool to perform AWS security best practices assessments, audits, and incident response.
* **CloudSploit:** An open-source tool that scans cloud accounts for security risks and misconfigurations.
* **Scout Suite:** A multi-cloud security auditing tool that reaches across APIs to find high-risk areas.

### Container Tools

* **Docker Bench for Security:** A script that checks for common best-practices around deploying Docker containers.
* **Kube-bench:** Checks whether Kubernetes is deployed securely by running checks against the CIS Kubernetes Benchmark.

## 2. Hands-On Lab Sessions

### Lab 1: Discovering Open S3 Buckets

**Goal:** Identify misconfigured storage that allows public read access.

1. **Tool:** S3Scanner.
2. **Command:** python3 s3scanner.py --include-closed names.txt
3. **Analysis:** Review the output for "Permissions: READ" or "Permissions: READ\_ACP."
4. **Verification:** Attempt to visit the bucket URL in a browser: https://[bucket-name].s3.amazonaws.com.

### Lab 2: AWS Auditing with Prowler

**Goal:** Identify security gaps in an AWS account (IAM, Networking, Logging).

1. **Prerequisite:** Configure AWS CLI with read-only credentials.
2. **Run:** prowler aws
3. **Analysis:** Review the color-coded report. Look for failures in:
   * **IAM:** "MFA not enabled for root user."
   * **S3:** "Bucket allows public access."
   * **Logging:** "CloudTrail is not enabled in all regions."

### Lab 3: Exploiting SSRF for Cloud Metadata

**Goal:** Steal IAM credentials from a compromised instance.

1. **Scenario:** You find a web app on a cloud VM vulnerable to SSRF.
2. Payload: Request the internal metadata URL:  
   http://169.254.169.254/latest/meta-data/iam/security-credentials/[role-name]
3. **Result:** The server returns an AccessKeyId, SecretAccessKey, and Token.
4. **Action:** Use these keys in your local terminal to gain the permissions of that server's role.

### Lab 4: Docker Benchmarking

**Goal:** Audit a local Docker host for vulnerabilities.

1. **Execution:** docker run --rm -it --net host --pid host --userns host --cap-add audit\_control \ -v /var/lib:/var/lib -v /var/run/docker.sock:/var/run/docker.sock \ docker/docker-bench-security
2. **Analysis:** Check for critical warnings like "Running as root user" or "Content trust for Docker not enabled."

## 3. CEH Practical Tips for Module 19

* **The Magic IP:** Always remember 169.254.169.254. This is the link-local address for Cloud Metadata services.
* **IAM Policy:** In a cloud engagement, the first thing an attacker looks for is a "Broad IAM Policy" (e.g., Effect: Allow, Action: \*, Resource: \*).
* **S3 Permissions:** Understand that "Authenticated Users" in AWS means *anyone* with an AWS account, not just users in your company.