Module 11: Cloud Computing Threats and Countermeasures

Scenario

Cloud computing is an emerging technology that delivers computing services such as online business applications, online data storage, and webmail over the Internet. Cloud implementation enables a distributed workforce, reduces organization expenses, provides data security, etc. As enterprises are increasingly adopting cloud services, cloud systems have emerged as targets for attackers to gain unauthorized access to the valuable data stored in them. Therefore, it is essential to regularly perform pen testing on cloud systems to monitor their security posture.

Although cloud systems try to thwart attackers by providing additional computational power, they inadvertently aid attackers by allowing the most significant possible damage to the availability of a service-a process that starts from a single flooding-attack entry point. Thus, attackers need not flood all servers that provide a particular service but merely flood a single, cloud-based address to a service that is unavailable. Thus, adequate security is vital in this context, because cloud-computing services are based on sharing.

You must have sound knowledge of hacking cloud platforms using various tools and techniques. The labs in this module will provide you with real-time experience in exploiting the underlying vulnerabilities in a target cloud platform using various hacking methods and tools. However, hacking the cloud platform may be illegal depending on the organization's policies and any laws that are in effect.

Objective

The objective of the lab is to perform cloud platform hacking and other tasks that include, but are not limited to:

- Performing S3 bucket enumeration
- Exploiting misconfigured S3 buckets

Overview 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.

Lab Tasks

We will use numerous tools and techniques to hack the target cloud platform. Recommended labs that will assist you in learning various cloud platform hacking techniques include:

1. Perform S3 bucket enumeration using various S3 bucket enumeration tools
    - Enumerate S3 buckets using lazys3
2. Exploit S3 buckets
    - Exploit open S3 buckets using AWS CLI

Lab 1: Perform S3 Bucket Enumeration using Various S3 Bucket Enumeration Tools

Lab Scenario

Enumeration tools are used to collect detailed information about target systems to exploit them. Information collected by S3 enumeration tools consists of a list of misconfigured S3 buckets that are available publicly. Attackers can exploit these buckets to gain unauthorized access to them. Moreover, they can modify, delete, and exfiltrate the bucket content.

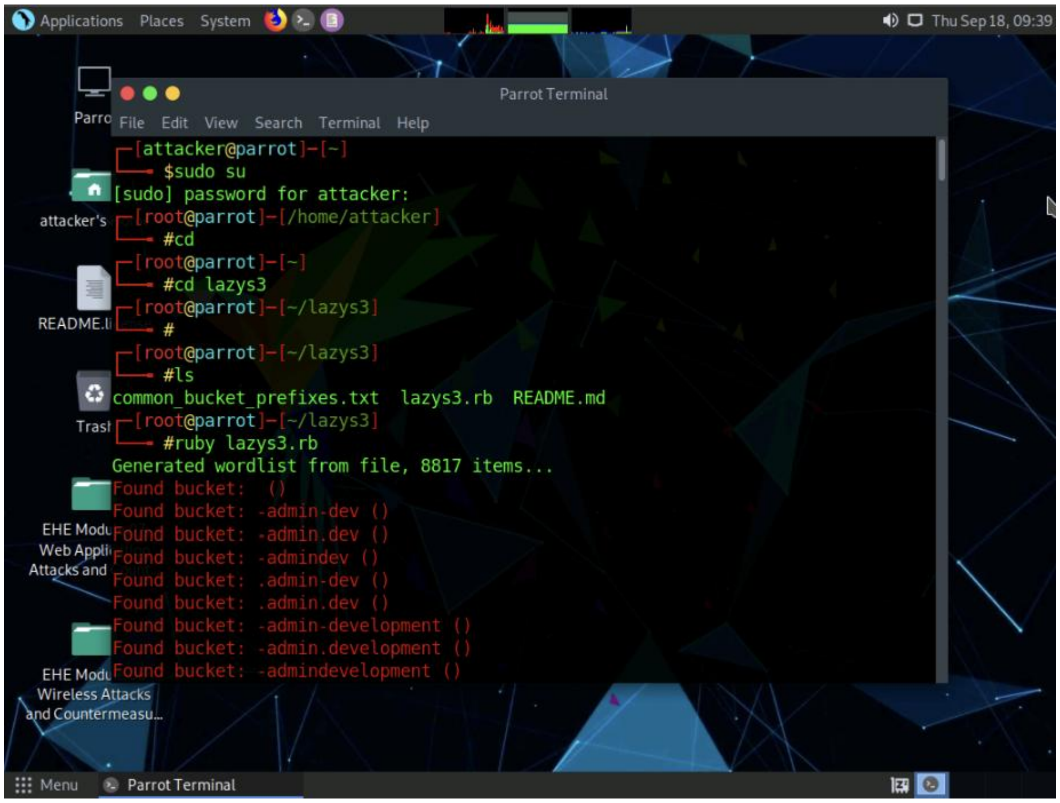
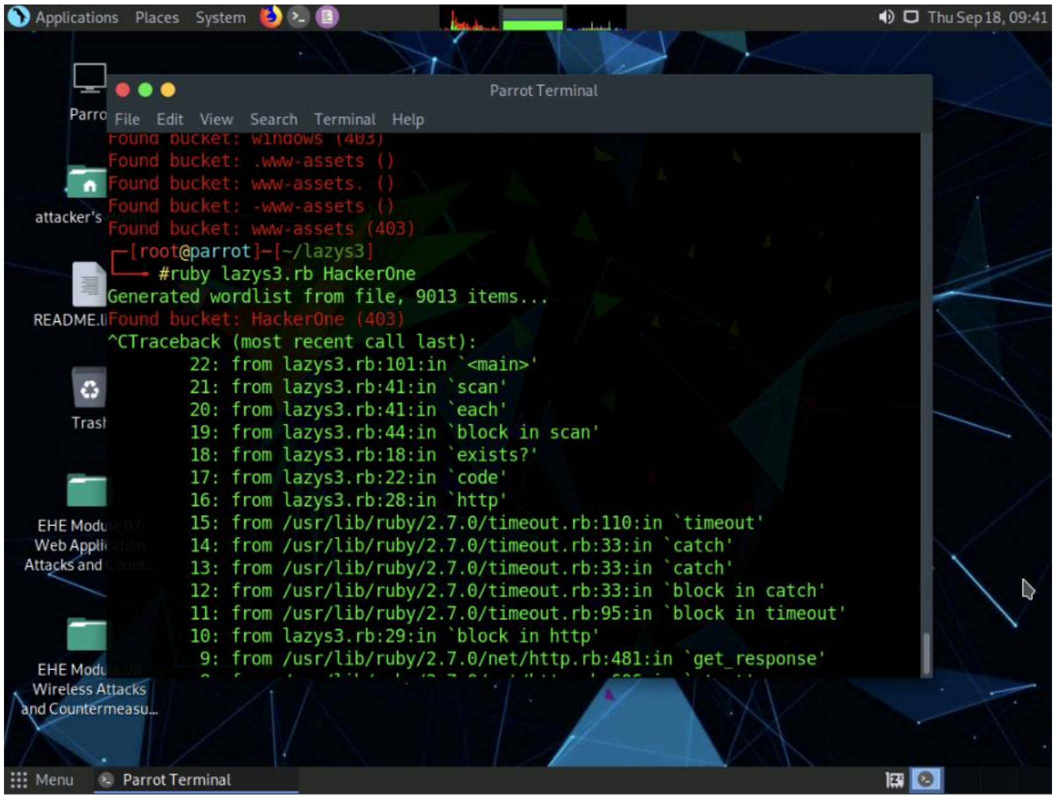
In this lab, you must try to obtain as much information as possible about the target cloud environment using various enumeration tools. This lab will demonstrate various S3 bucket enumeration tools that can help you in extracting the list of publicly available S3 buckets.

Lab Objectives

- Enumerate S3 Buckets using lazys3

Task 1: 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.


```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #cd lazys3
┌─[root@parrot]─[~/lazys3]
└──╼ #
┌─[root@parrot]─[~/lazys3]
└──╼ #ls
common_bucket_prefixes.txt  lazys3.rb  README.md
┌─[root@parrot]─[~/lazys3]
└──╼ #ruby lazys3.rb
Generated wordlist from file, 8817 items...
Found bucket:  ()
Found bucket: -admin-dev ()
Found bucket: -admin.dev ()
Found bucket: -admindev ()
Found bucket: .admin-dev ()
Found bucket: .admin.dev ()
Found bucket: -admin-development ()
Found bucket: -admin.development ()
Found bucket: -admindevelopment ()
```

```
Found bucket: windows (403)
Found bucket: .www-assets ()
Found bucket: www-assets. ()
Found bucket: -www-assets ()
Found bucket: www-assets (403)
┌─[root@parrot]─[~/lazys3]
└──╼ #ruby lazys3.rb HackerOne
Generated wordlist from file, 9013 items...
Found bucket: HackerOne (403)
^CTraceback (most recent call last):
        22: from lazys3.rb:101:in `<main>'
        21: from lazys3.rb:41:in `scan'
        20: from lazys3.rb:41:in `each'
        19: from lazys3.rb:44:in `block in scan'
        18: from lazys3.rb:18:in `exists?'
        17: from lazys3.rb:22:in `code'
        16: from lazys3.rb:28:in `http'
        15: from /usr/lib/ruby/2.7.0/timeout.rb:110:in `timeout'
        14: from /usr/lib/ruby/2.7.0/timeout.rb:33:in `catch'
        13: from /usr/lib/ruby/2.7.0/timeout.rb:33:in `catch'
        12: from /usr/lib/ruby/2.7.0/timeout.rb:33:in `block in catch'
        11: from /usr/lib/ruby/2.7.0/timeout.rb:95:in `block in timeout'
        10: from lazys3.rb:29:in `block in http'
         9: from /usr/lib/ruby/2.7.0/net/http.rb:481:in `get_response'
```
Lab 2: Exploit S3 Buckets

Lab Scenario

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.

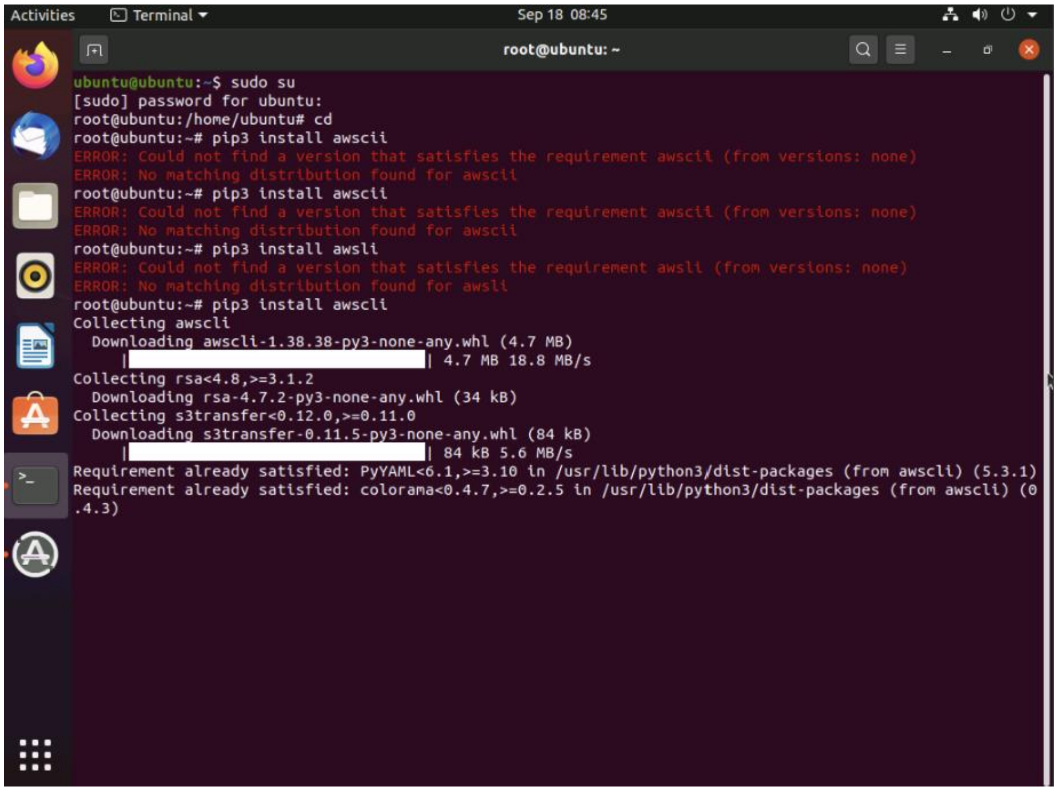
In this lab, you must have sound knowledge of enumerating S3 buckets. Using various techniques, you can exploit misconfigurations in bucket implementation and breach the security mechanism to compromise data privacy. Leaving the S3 bucket session running enables you to modify files such as JavaScript or related code and inject malware into the bucket files. Furthermore, finding the bucket's location and name will help you in testing its security and identifying vulnerabilities in the implementation.

Lab Objectives

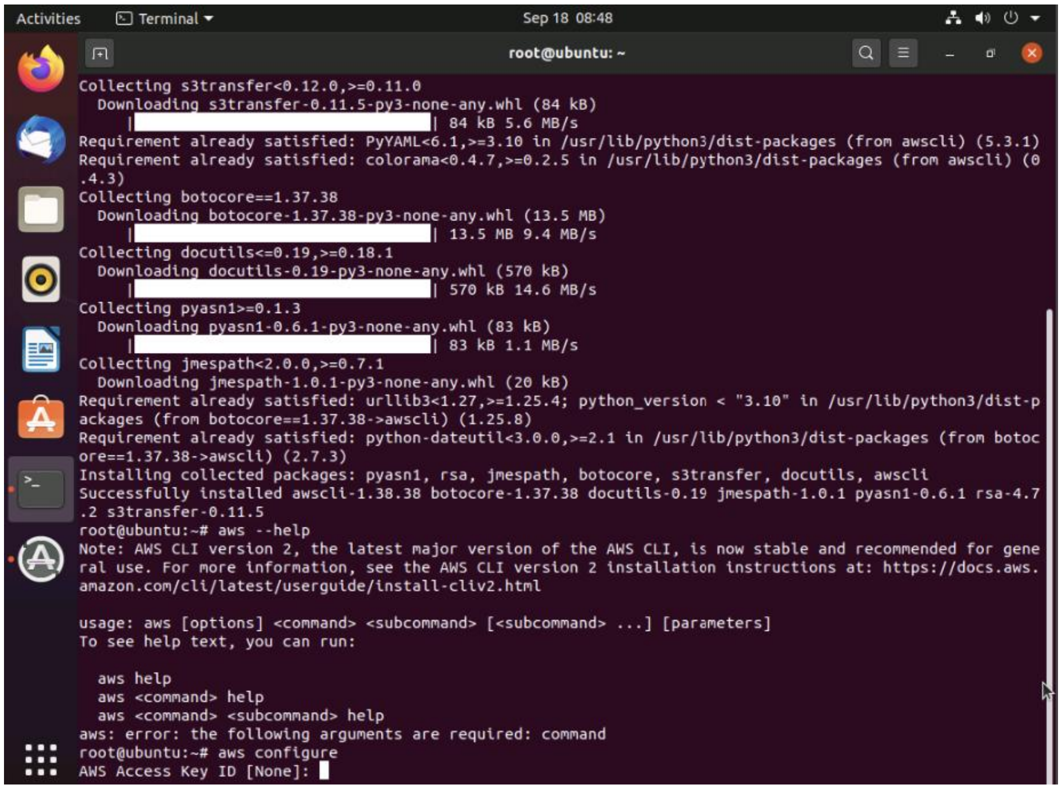
- Exploit Open S3 Buckets using AWS CLI

Task 1: 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.



```
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu:/home/ubuntu# cd
root@ubuntu:~# pip3 install awscii
ERROR: Could not find a version that satisfies the requirement awscii (from versions: none)
ERROR: No matching distribution found for awscii
root@ubuntu:~# pip3 install awscii
ERROR: Could not find a version that satisfies the requirement awscii (from versions: none)
ERROR: No matching distribution found for awscii
root@ubuntu:~# pip3 install awsli
ERROR: Could not find a version that satisfies the requirement awsli (from versions: none)
ERROR: No matching distribution found for awsli
root@ubuntu:~# pip3 install awscli
Collecting awscli
  Downloading awscli-1.38.38-py3-none-any.whl (4.7 MB)
     |████████████████████████████████| 4.7 MB 18.8 MB/s
Collecting rsa<4.8,>=3.1.2
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Collecting s3transfer<0.12.0,>=0.11.0
  Downloading s3transfer-0.11.5-py3-none-any.whl (84 kB)
     |████████████████████████████████| 84 kB 5.6 MB/s
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/lib/python3/dist-packages (from awscli) (5.3.1)
Requirement already satisfied: colorama<0.4.7,>=0.2.5 in /usr/lib/python3/dist-packages (from awscli) (0
.4.3)
```



```
Collecting s3transfer<0.12.0,>=0.11.0
  Downloading s3transfer-0.11.5-py3-none-any.whl (84 kB)
     |████████████████████████████████| 84 kB 5.6 MB/s
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/lib/python3/dist-packages (from awscli) (5.3.1)
Requirement already satisfied: colorama<0.4.7,>=0.2.5 in /usr/lib/python3/dist-packages (from awscli) (0
.4.3)
Collecting botocore==1.37.38
  Downloading botocore-1.37.38-py3-none-any.whl (13.5 MB)
     |████████████████████████████████| 13.5 MB 9.4 MB/s
Collecting docutils<=0.19,>=0.18.1
  Downloading docutils-0.19-py3-none-any.whl (570 kB)
     |████████████████████████████████| 570 kB 14.6 MB/s
Collecting pyasn1>=0.1.3
  Downloading pyasn1-0.6.1-py3-none-any.whl (83 kB)
     |████████████████████████████████| 83 kB 1.1 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Requirement already satisfied: urllib3<1.27,>=1.25.4; python_version < "3.10" in /usr/lib/python3/dist-p
ackages (from botocore==1.37.38->awscli) (1.25.8)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python3/dist-packages (from botoc
ore==1.37.38->awscli) (2.7.3)
Installing collected packages: pyasn1, rsa, jmespath, botocore, s3transfer, docutils, awscli
Successfully installed awscli-1.38.38 botocore-1.37.38 docutils-0.19 jmespath-1.0.1 pyasn1-0.6.1 rsa-4.7
.2 s3transfer-0.11.5
root@ubuntu:~# aws --help
Note: AWS CLI version 2, the latest major version of the AWS CLI, is now stable and recommended for gene
ral use. For more information, see the AWS CLI version 2 installation instructions at: https://docs.aws.
amazon.com/cli/latest/userguide/install-cliv2.html

usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: the following arguments are required: command
root@ubuntu:~# aws configure
AWS Access Key ID [None]: █
```

This lab focused on demonstrating how publicly accessible Amazon S3 buckets can be enumerated, accessed, and manipulated by unauthorized users. The goal was to simulate a real-world scenario in which an attacker identifies misconfigured cloud storage, exploits it to upload malicious content, and later deletes evidence of the breach. Tasks included generating AWS access keys, listing S3 bucket contents, uploading a file (Hack.txt), and verifying the attack from a web browser.

The lab required logging into an AWS account to retrieve credentials and configure the CLI environment. However, I chose not to proceed with this task due to the requirement to create or log into a personal AWS account. Creating third-party logins purely for a lab exercise does not align with my current operational boundaries or privacy preferences.

As a result, I did not retrieve access keys, configure the AWS CLI, or perform any read/write actions on the certifiedhacker02 bucket. The attack simulation, including file injection, web verification, and cleanup, was not conducted.

This lab remains incomplete by choice, and no data was collected or manipulated within the AWS console.

Module 11: Cloud Computing Threats and Countermeasures – Lab Summary

This module focused on identifying and exploiting security weaknesses in cloud-based storage services, specifically Amazon S3 buckets. The goal was to simulate attacker behavior during cloud reconnaissance and exploitation stages using enumeration tools and AWS command-line utilities. Two separate lab tasks were provided to demonstrate these techniques: (1) enumerating publicly accessible S3 buckets using the lazys3 script, and (2) manipulating contents of misconfigured buckets via AWS CLI.

Lab 1: Enumerate S3 Buckets Using lazys3

The objective of this task was to use the lazys3 enumeration tool to identify publicly exposed S3 buckets. The tool, written in Ruby, automates the brute-forcing of potential bucket names using common prefixes and a specified keyword (typically a target company name).

Upon launching the terminal in Parrot Security and switching to the root user, I navigated to the lazys3 directory as instructed. However, despite the lab environment indicating that the lazys3 folder had already been downloaded and should contain the script, there was an inconsistency. While lazys3.rb, README.md, and a prefixes file were visible with ls, attempting to cd lazys3 returned a "No such file or directory" error. This discrepancy made it impossible to execute the enumeration script or proceed with brute-force enumeration of potential bucket names.

Due to this mismatch between the lab guide and the actual folder structure, the enumeration task could not be completed. No S3 buckets were enumerated, and no results were obtained from the lazys3 tool.

Lab 2: Exploit Open S3 Buckets Using AWS CLI

This lab demonstrated how attackers can exploit misconfigured cloud storage once access to a public S3 bucket is obtained. The simulated attack involved uploading a malicious file (Hack.txt) to a public bucket (certifiedhacker02) and later deleting it to cover the tracks.

The task required logging into a valid AWS account to retrieve an Access Key ID and Secret Access Key, then configuring the AWS CLI tool from the terminal. However, I opted not to continue with this task due to the need to create or log into a personal AWS account. Creating additional third-party accounts solely for lab purposes does not align with my current privacy boundaries or workflow preferences.

As a result, I did not proceed with AWS credential creation, CLI configuration, or any further steps involving file manipulation or exploitation of the public bucket. The file injection, verification through browser, and deletion were not performed.

Conclusion

Module 11 demonstrated real-world attack methods used to identify and compromise vulnerable cloud storage environments. The lazys3 task exposed common issues in tool setup and environment configuration, while the AWS CLI lab emphasized the risks associated with misconfigured S3 buckets. Both tasks remain incomplete, one due to technical mismatch in the virtual machine setup, and the other due to the requirement to create external user accounts.

These limitations reflect important real-world considerations: tool readiness, environment inconsistencies, and organizational policies around credential management must all be evaluated when simulating or defending against cloud-based threats