# CloudGoat Scenario

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy# tree
.
├── assets
│   ├── db.txt
│   ├── elasticloadbalancing.log
│   └── rce_app
│       ├── app.zip
│       ├── file.txt
│       ├── index.js
│       ├── lib.js
│       ├── package.json
│       ├── package-lock.json
│       ├── README.md
│       └── static
│           ├── bootstrap.css
│           ├── index.html
│           └── mkja1xijqf0abo1h9glg.html
├── cheat_sheet_lara.md
├── cheat_sheet_mcduck.md
├── cloudgoat
├── cloudgoat.pub
├── manifest.yml
├── README.md
├── start.sh
├── start.txt
└── terraform
```

**After installing the provided scenario, we first checked the structure of the problem through the tree command to understand the structure.**

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy/assets# cat db.txt
Dear Tomas - For the LAST TIME, here are the database credentials. Save them to your
 in breach of our security policies!!!!

DB name: cloudgoat
Username: cgadmin
Password: Purplepwny2029
```

**Next, I found out about a file called db.txt. I can tell you about db's name and user and password. However, with the current information, I can't see where that db is.**

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy/assets# cat elasticloadbalancing.log
http 2019-06-18T21:36:23.594569Z app/${cgid}/d36d4f13b73c2fe7 10.10.10.23:5132 10.0.10.254:9000 0.001 0.001 0.000 200 200 485 1287 "GET http://${load_balancer_dns}
:80/mkja1xijqf0abo1h9glg.html HTTP/1.1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/537.36" - - ${
target_group_arn} "Root=1-5d095963-e2b838a764ed31d017b74cce" "-" "-" 0 2019-06-18T21:36:35.592000Z "forward" "-" "-"
http 2019-06-18T21:36:24.358063Z app/${cgid}/d36d4f13b73c2fe7 10.10.10.23:5132 10.0.10.254:9000 0.001 0.001 0.000 200 200 460 1123 "GET http://${load_balancer_dns}
:80/ HTTP/1.1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/537.36" - - ${target_group_arn} "Root=1
-5d095958-b7623797a9d1a161ae7aa7a4" "-" "-" 0 2019-06-18T21:36:24.356000Z "forward" "-" "-"
http 2019-06-18T21:36:24.667135Z app/${cgid}/d36d4f13b73c2fe7 10.10.10.23:5132 10.0.10.254:9000 0.000 0.001 0.000 200 200 421 192476 "GET http://${load_balancer_dn
s}:80/bootstrap.css HTTP/1.1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/537.36" - - ${target_gro
up_arn} "Root=1-5d095958-9945877357b50e39544fc5b7" "-" "-" 0 2019-06-18T21:36:24.443000Z "forward" "-" "-"
http 2019-06-18T21:36:24.771360Z app/${cgid}/d36d4f13b73c2fe7 10.10.10.23:5132 10.0.10.254:9000 0.000 0.001 0.000 200 200 440 1123 "GET http://${load_balancer_dns}
:80/favicon.ico HTTP/1.1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/537.36" - - ${target_group_a
```

**Checked for elasticloadbalancing.log.**

| 필드 번호 | 필드 내용 | 설명 |
|---|---|---|
| 1 | `http` | 로그 유형 (HTTP 요청) |
| 2 | `2019-06-18T21:36:23.594569Z` | 요청이 처리된 시각 (UTC) |
| 3 | `app/${cgid}/d36d4f13b73c2fe7` | Load Balancer 및 Target Group 식별자 |
| 4 | `10.10.10.23:5132` | 클라이언트 IP 주소와 포트 번호 |
| 5 | `10.0.10.254:9000` | Target IP 주소와 포트 번호 |
| 6 | `0.001 0.001 0.000` | 처리 시간 (Request → Target 전달, 응답 시간, 처리 시간) |
| 7 | `200 200` | HTTP 상태 코드 (Load Balancer 및 Target에서 반환된 상태) |
| 8 | `485 1287` | 요청 및 응답 바이트 수 |
| 9 | `"GET http://${load_balancer_dns}:80/mkja1xijqf0abo1h9glg.html HTTP/1.1"` | 클라이언트가 보낸 HTTP 요청 라인 |
| 10 | `"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/537.36"` | 클라이언트의 User-Agent 헤더 |
| 11 | `-` | 리퍼러 URL (존재하지 않음) |
| 12 | `${target_group_arn}` | Target Group ARN |
| 13 | `"Root=1-5d095963-e2b838a764ed31d017b74cce"` | X-Amzn-Trace-Id (요청 추적 ID) |
| 14 | `"-" "-" 0` | 인증 정보 및 요청 ID (인증된 정보가 없음) |
| 15 | `2019-06-18T21:36:35.592000Z` | 요청 처리 완료 시각 (UTC) |

| | | |
|---|---|---|
| 16 | `"forward" "-" "-"` | 요청 처리 방식 (Forwarded) 및 관련 정보 없음 |

**The table above will be used to obtain additional information.**

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy# cat start.txt
cloudgoat_output_aws_account_id = 296239021157
cloudgoat_output_lara_access_key_id = AKIAUJ6JYXRSRMJUPPWV
cloudgoat_output_lara_secret_key = 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
cloudgoat_output_mcduck_access_key_id = AKIAUJ6JYXRST74RYKDZ
cloudgoat_output_mcduck_secret_key = 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
```

**However, the information that can be known even if the structure is understood is finite, so we first looked at start.txt.**

**In that txt, I was able to see the profile information of the person named lala and mcduck. Now I can see the permissions of the people.**

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy# aws configure --profile Lara
AWS Access Key ID [None]: aws configure --profile Lara
AWS Secret Access Key [None]: AKIAUJ6JYXRSRMJUPPWV
Default region name [None]: us-east-1
Default output format [None]:
```

**First, let's check Lala's permissions. Based on the information I got from txt above, I learned about Lala's permissible permissions.**

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy# aws iam list-user-policies --user-name lara --profile Lara

An error occurred (IncompleteSignature) when calling the ListUserPolicies operation: Invalid key=value pair (missing equal-sign) in Authorization header (hashed wi
th SHA-256 and encoded with Base64): 'JUKls7AaziHBlDwB8xfIIXoO+PtcIueAnGBQ885nzmw='.
```

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy# aws iam list-attached-user-policies --user-name lara --profile Lara

An error occurred (AccessDenied) when calling the ListAttachedUserPolicies operation: User: arn:aws:iam::296239021157:user/lara is not authorized to perform: iam:L
istAttachedUserPolicies on resource: user lara because no identity-based policy allows the iam:ListAttachedUserPolicies action
```

```
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy# aws iam list-roles --profile Lara

An error occurred (AccessDenied) when calling the ListRoles operation: User: arn:aws:iam::296239021157:user/lara is not authorized to perform: iam:ListRoles on res
ource: arn:aws:iam::296239021157:role/ because no identity-based policy allows the iam:ListRoles action
(.venv) root@t:/home/kali/cloudgoat/rce_web_app_cgid0nkp596tuy#
```

**Unfortunately, the user named lala did not have access to role in the policy, so we did not get any additional information.**