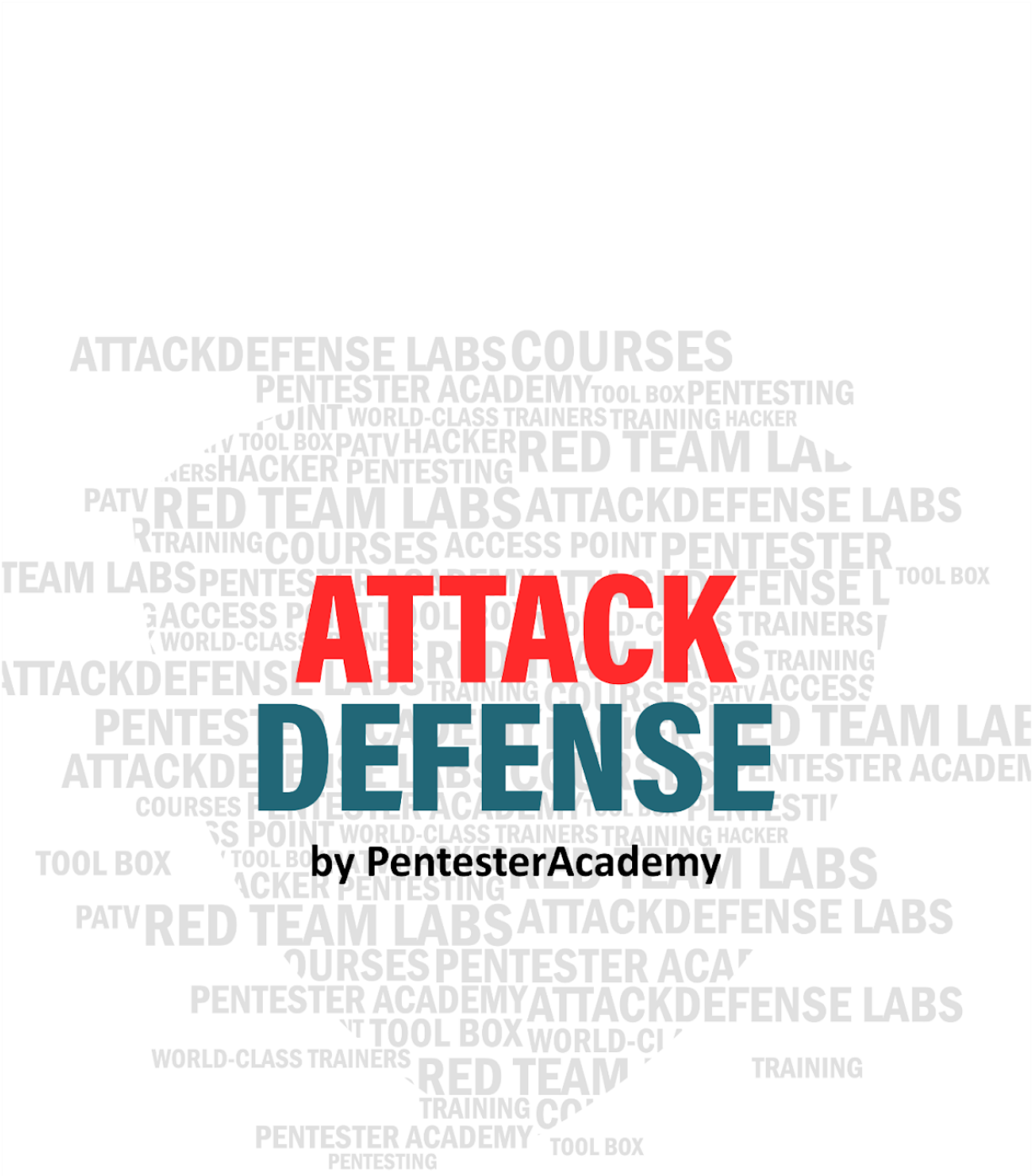
ATTACK
DEFENSE
by PentesterAcademy

Name	Detect Secrets: Hunting Sensitive Information
URL	https://www.attackdefense.com/challengedetails?cid=2152
Type	DevSecOps Basics: Sensitive Information Scan

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

The Detect-secrets is a tool used to identify the secrets in the source code of the application.

A Kali CLI machine (kali-cli) is provided to the user with detect-secrets installed on it. The source code for two sample applications is provided in the home directory of the root user.

Objective: Use the detect-secrets utility to find sensitive information in the applications!

Instructions:

- The source code of applications is provided at /root/github-repos

Solution

Step 1: Check the provided applications.

Command: ls -l github-repos/

```
root@attackdefense:~# ls -l github-repos/
total 8
drwxrwxr-x 6 root root 4096 Nov 13 11:32 flask-recipes
drwxrwxr-x 4 root root 4096 Nov 13 11:32 is-online
root@attackdefense:~#
```

We will take one example at a time and run the tool on that.

Example 1: Flask recipes

Step 1: Run the detect-secrets tool on the flask-recipes directory.

Command: detect-secrets scan ~/github-repos/flask-recipes

```
root@attackdefense:~# detect-secrets scan ~/github-repos/flask-recipes
{
  "custom_plugin_paths": [],
  "exclude": {
    "files": null,
    "lines": null
  },
  "generated_at": "2020-11-13T11:57:23Z",
  "plugins_used": [
    {
      "name": "AWSKeyDetector"
    },
    {
      "name": "ArtifactoryDetector"
    },
    {
      "base64_limit": 4.5,
      "name": "Base64HighEntropyString"
    },
    {
      "name": "BasicAuthDetector"
    },
```

```
      {
        "hashed_secret": "afc848c316af1a89d49826c5ae9d00ed769415f3",
        "is_verified": false,
        "line_number": 65,
```

```
        "type": "Basic Auth Credentials"
      }
    ],
    "github-repos/flask-recipes/app/config.py": [
      {
        "hashed_secret": "afc848c316af1a89d49826c5ae9d00ed769415f3",
        "is_verified": false,
        "line_number": 44,
        "type": "Basic Auth Credentials"
      }
    ]
  },
  "version": "0.14.3",
  "word_list": {
    "file": null,
    "hash": null
  }
}
root@attackdefense:~#
```


Issues Detected

- Sensitive information found in the source code

Example 2: is-online

Step 1: Run the detect-secrets tool on the is-online directory.

Command: detect-secrets scan ~/github-repos/is-online/


```
root@attackdefense:~# detect-secrets scan ~/github-repos/is-online/
{
  "custom_plugin_paths": [],
  "exclude": {
    "files": null,
    "lines": null
  },
  "generated_at": "2020-11-13T12:34:21Z",
  "plugins_used": [
    {
      "name": "AWSKeyDetector"
    },
    {
      "name": "ArtifactoryDetector"
    },
```

```
    {
      "base64_limit": 4.5,
      "name": "Base64HighEntropyString"
    },
```

```
  ],
  "results": {
    "github-repos/is-online/browser.js": [
      {
        "hashed_secret": "d6b66ddd9ea7dbe760114bfe9a97352a5e139134",
        "is_verified": false,
        "line_number": 4,
        "type": "JSON Web Token"
      }
    ]
  },
  "version": "0.14.3",
  "word_list": {
    "file": null,
    "hash": null
  }
}
root@attackdefense:~#
```


Issues Detected

- Hardcoded JSON Web token found.

Learnings

Perform Sensitive Information Scan using the detect-secrets tool.

References:

- Flask Recipies (https://github.com/spk/flask-recipes.git)
- Is-online (https://github.com/sindresorhus/is-online.git)