Cybersecurity Minor Project: GitHub OSINT

# 1. Introduction

This report documents the process and results of the Minor Project focused on GitHub Open Source Intelligence (OSINT) for identifying leaked secrets in public repositories. The project uses Gitleaks to automatically detect sensitive information such as API keys, passwords, and private keys present in GitHub repositories.

# 2. Objective

The main objective of this project is to identify, analyze, and document instances of exposed sensitive information in public GitHub repositories using open-source tools.

# 3. Tools & Environment

1. Gitleaks – Secret scanning tool for detecting API keys, passwords, and other sensitive information.  
2. Git – For cloning public repositories.  
3. Linux environment (Ubuntu / Kali / WSL) – For running scans and commands.  
4. Bash – Used for automation scripting.

# 4. Methodology

Step 1: Installed Gitleaks using the official GitHub release.  
Step 2: Collected public GitHub repository URLs using GitHub dorking techniques such as searching for `filename:.env "API\_KEY"` and `extension:json "password"`.  
Step 3: Created a text file (repos\_list.txt) with one repository URL per line.  
Step 4: Wrote and executed a Bash loop to clone each repository and scan it with Gitleaks.  
Step 5: Saved JSON reports of findings in the 'reports' directory.

# 5. Commands Used

sudo apt update && sudo apt upgrade -y  
wget -qO gitleaks.tar.gz https://github.com/gitleaks/gitleaks/releases/latest/download/gitleaks\_linux\_x64.tar.gz  
sudo tar xf gitleaks.tar.gz -C /usr/local/bin gitleaks  
sudo chmod +x /usr/local/bin/gitleaks  
gitleaks version  
  
mkdir repos reports  
while read -r repo; do  
 name=$(basename "$repo" .git)  
 git clone --depth 1 "$repo" repos/"$name" || continue  
 gitleaks git --repo-path repos/"$name" --report-path reports/"$name"-gitleaks.json --verbose  
done < repos\_list.txt

# 6. Findings

The scan detected sensitive information such as API keys and passwords in test repositories. Each finding includes the file path, line number, and type of secret detected. Findings were saved as JSON files in the reports directory.

# 7. Sample Output

{  
 "Description": "Hardcoded AWS Secret Key",  
 "File": "config/aws\_keys.env",  
 "StartLine": 3,  
 "EndLine": 3,  
 "Secret": "AKIA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",  
 "RuleID": "aws-secret-key"  
}

# 8. Risk & Remediation

Risks:  
- Unauthorized access to cloud resources.  
- Data breaches and financial loss.  
- Compromise of sensitive systems.  
  
Remediation Steps:  
- Immediately rotate exposed credentials.  
- Remove sensitive data from repositories.  
- Use environment variables or secret managers for storing sensitive values.

# 9. Conclusion

The GitHub OSINT project successfully demonstrated how sensitive information can be exposed in public repositories and how tools like Gitleaks can help in detecting and mitigating such risks.