Advance Bug Bounty

Hunting For Sensitive Data

Introduction to Sensitive Data Exposure

Definition: Sensitive data exposure is the unintentional release of private or confidential information.

Examples: API keys, passwords, tokens, private configurations.

Impact: Data breaches, unauthorized access, and serious security threats.

What is Google Dorking?

Definition: Using advanced search operators in Google to find sensitive information that has been accidentally made public.

Common Targets: GitHub repositories, websites with misconfigured files, and exposed JS files.

Why It's Useful: Finds hidden or overlooked data without directly attacking a server.

Common Google Dorking Operators

Basic Operators:

- **`site:`** limits searches to specific sites.
- **`filetype:`** finds specific file types (e.g., `.js`, `.json`).
- **`inurl:`** searches for specific terms in the URL.
- **`intitle:`** searches for terms in the page title.
- **`ext:`** finds specific file extensions (e.g., `.env`, `.php`).

Targeting GitHub for Sensitive Data

Manual Approach:

"Company" password	"Company" key
"Company" secret	"Company" pass
"Company" credentials	"Company" login
"Company" token	"Company" ftp
"Company" config	"Company" pwd

Cont...

Search Term	Results
"Company" security_credentials	LADP (Active Directory)
"Company" connectionstring	Database Credentials
"Company" JDBC	Database Credentials
"Company" ss2_auth_password	Unauthorized Access to Server
"Company" send_keys and send,keys	Keywords related to pasword failed

Cont...

```
env.put(javax.naming.Context.PROVIDER_URL, "ldap://172.16.0.197/cn=pdm.admin,ou=admin user,ou=DG tpe,DC=test,DC=com");
env.put(javax.naming.Context.SECURITY_AUTHENTICATION,"Simple");
//env.put(javax.naming.Context.SECURITY_PRINCIPAL,"cn=piduser,ou=DGUsers,ou=DG,DC=cn,DC=test,DC=com");
env.put(javax.naming.Context.SECURITY_PRINCIPAL,"cn=test,OU=PLM,OU=CA,OU=CIT,OU=TPV TPE,DC=test,DC=com");
//env.put(javax.naming.Context.SECURITY_CREDENTIALS , "piduser" );
env.put(javax.naming.Context.SECURITY_CREDENTIALS , "cscxx112129" );
//env.put(javax.naming.Context.SECURITY_CREDENTIALS , "Pdm@d20$0421" );
```

```
{
  "ConnectionStrings": {
    "Default": "Server=10.0.75.1; Database=PhoneBookDb; User=sa; Password=123qwe;"
```

```
import pypyodbc as pyodbc # you could alias
db_host = 'mspitsql15.test.th3g3nt3l.org'
db_name = 'BISE'
db_user = 'empDS'
db_password = 'Seattle@98121'
```

```
                'bundles',
                r'',
                )
ftp_th3g3nt3l = ftp('ftp.th3g3nt3lman.com',
                 'ASbd5FD',
                 'AB$1B#6mAk1HH',
    info('->Start to upload to th3g3nt3l.ftp<-')
ftpcli = ftp(ftp_th3g3nt3l)
```

Mostly Google Dorks Used by me

- Filetype:env "DB_PASSWORD" site:*.target.com
- Filetype:txt "Passwd" site:*.target.com
- Filetype:pdf "Confidential" site:*.target.com
- Filetype:sql "DUMP" site:*.target.com
- Inurl:"index of /" "Parent Directory" site:*.target.com
- Inurl:"config.php" site:*.target.com
- Filetype:xls "Password" site:*.target.com
- Filetype:bak "backup" site:*.target.com
- Inurl:"file=" inurl:"index.php" site:*.target.com
- Inurl:/.git site:*.target.com

Hunting with Wayback Machine

Use Case:

- Old versions of websites can still have sensitive data like API keys or old endpoints.
- Look for configurations, hardcoded credentials, or exposed tokens in earlier versions of JS or HTML files.

How to Use:

Go to archive.org and enter the target site URL.

Browse historical snapshots and inspect the page source

Hunting with Waybackurls

What is Waybackurls?

- A tool used to fetch archived URLs for a given domain from the Wayback Machine.
- Helps find old URLs that are no longer accessible but still archived.
- **Command Usage**:

- Install Waybackurls: `go install github.com/tomnomnom/waybackurls@latest`
- Command: `echo "target.com" | waybackurls`

Example:

- `echo "example.com" | waybackurls | grep ".js"` to find archived JavaScript files.

Analyzing Archived JS Files for Sensitive Data

Why Archived JS Files?

- JavaScript files in archives can contain hardcoded secrets that were later removed from the live site but are still accessible in the Wayback Machine.

End