Laboratory Exercise E2 - Introduction to Password Auditing

Author: Dr. David Raymond
First Edition: August 2017
Last Updated: July 2026

Due Date:

Points Possible:

Overview

This laboratory exercise will provide some hands-on experience with password strength analysis using command-line tools in Linux.

Recommended Reading

Wikipedia article titled 'passwd', which introduces and describes the Unix/Linux passwd and shadow files: Wikipedia: passwd

Resources required

This exercise requires the latest Cyber Basics environment running in the Cyber Range.

Initial Setup

Log in to your Cyber Range account and select the latest Cyber Basics environment. Start your environment and, if necessary, log in using the username **student** and password **student**.

Tasks

Task 1: Introduction to password auditing.

On Linux systems, user accounts are stored in the **/etc/passwd** file (a world-readable text file), and passwords are hashed and stored in **/etc/shadow** (a text file only readable by root). You have administrative (root) access on your Kali virtual machine – go ahead and "cat" those files to see what they look like (see Wikipedia: passwd). You'll note that the hashed passwords are stored in the shadow file, which is only readable by users with administrative privileges. On Windows systems, hashed passwords are stored in the SAM (database) file, located in C:. This file is not a simple text file – you've got to use special tools to read it.

We'll use a password auditing tool called John the Ripper (JTR), probably the most effective and widely known password cracker. JTR is available from www.openwall.com/john. You can pay for the "pro" version or select one of the 'official free versions' for your operating system. Windows and Unix/Linux executables are available. If you want the best performance and are comfortable compiling software from the source code, download the latest community-enhanced "jumbo patch" and compile it for your specific platform.

JTR can be run in various modes, including dictionary and hybrid modes, both of which use a "dictionary" provided by the user. Dictionaries are compiled from widely used passwords, words scraped from a company's website, etc.

Exercise 1: Crack Linux passwords. Here we'll create a couple of new accounts on our Linux VM, one with a good password and one with a poor one:

1. Ensure you are at a 'root user' prompt ends with **#** instead of **$**. use the sudo command to execute the su (switch user) command to become "root" as follows:

```
$ sudo su
```

2. Create 2 accounts: (one with a bad password and one with a good one.)

```
# useradd johnsmith
```

- the lightweight method to add a Linux user account

```
# passwd johnsmith
New password: 12345
Retype new password: 12345
# useradd janedoe
# passwd janedoe
New password: <non-dictionary word with numbers, chars, etc.>
Retype new password: <same as above>
```

3. Now let's see which ones we can crack. Copy /etc/shadow to a text file, and run john against it:

```
# cp /etc/shadow ./pass.txt
```

- copy the shadow file to the john folder

```
# cat pass.txt
```

- see the usernames/encrypted passwords.

```
# john -format:crypt pass.txt
```

- let's start cracking!

JTR will attempt to decipher the passwords and display any that it 'cracks' as it goes along. It starts in "single crack" mode, mangling usernames and other account information. It then moves on to a dictionary attack using a default dictionary, then with a hybrid attack, then brute force, where it will try every possible combination of characters (letters, numbers, and special characters) until it cracks them all.

The password for the johnsmith account should be cracked in a few minutes.

- ***You generally do not want to wait for John to crack any 'good' password in brute force mode, as it could take months or years to complete. If this should ever happen, You can always press [CTRL]-[C] to stop execution.***

John uses the following files to manage execution. Most are all stored in the **/usr/share/john** folder on your Kali virtual machine (john.pot is stored elsewhere as indicated):

- **password.lst** is John's default dictionary. You can specify another wordlist on the command line using the **–wordlist=** directive

for example:

```
# john --wordlist=/usr/share/dict/american-english /etc/shadow
```

- **john.conf** is read when JTR starts up and has rules for dictionary mangling for the hybrid crack attempt
- **john.rec** is used to record the status of the current password cracking attempt. If john crashes, it will start where it left off instead of starting again from the beginning of the dictionary.
- /root/.john/**john.pot** lists passwords that have already been cracked. If you run john again on the same shadow file, it won't show these cracked passwords unless you delete this file first.

Task 2. More password audit.

John the Ripper's default dictionary (described above) is a short list of common passwords. Sometimes, a standard English dictionary is a better option. In this exercise, we will download a Linux shadow file that contains a set of user accounts and hashed passwords, and then attempt to determine the passwords.

```
# wget artifacts.virginiacyberrange.net/gencyber/shadow
# john shadow
```

Let John run for a few minutes, then stop with [CTRL]-[C]. How many passwords are revealed? Use the –show option:

```
john --show shadow
```

Now we will run john again with a different dictionary. First, we will ensure an American English dictionary is installed on our Kali Linux system. (If the dictionary is already there, the below command will update it.)

```
# sudo apt update; sudo apt install -y wamerican
```

Next, we will run John with the new dictionary by invoking the –wordlist directive at the command line.

```
# john shadow --wordlist=/usr/share/dict/american-english
```

How many of the passwords were revealed this time?

Why was this pass more effective?

References

- Wikipedia passwd article: Wikipedia: passwd
- John the Ripper (JTR): John the Ripper