

---

# *Passwords LAB*

## LAB 9

---

Josiah Smythe

April 9, 2019

Dr. Lehrfeld -Information Security and Assurance  
East Tennessee State University

---

---

## 1. Purpose

This Lab was completed to show the importance of password complexity by utilizing the password cracking software John The Ripper

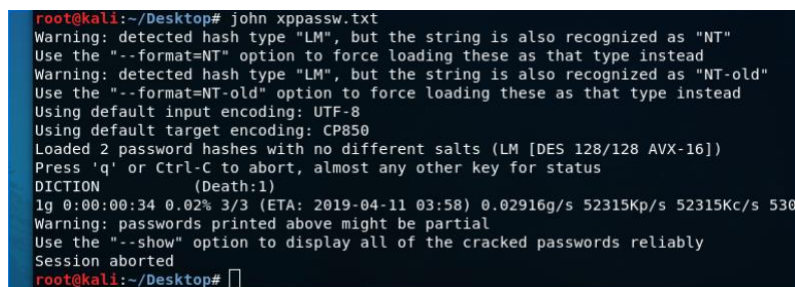
## 2. Materials

This lab was completed within the two virtual machine environments Kali Linux Rolling and Windows XP. Both of these systems were running in the virtual machine application VMware Fusion Professional, Version 11.0.2. VMware Fusion was being supported by the operation system Mac OS Mojave on a MacBook Pro 2015 15" with 16 GB of DDR3 Ram and an Intel i7-4770HQ running 2.2GHz. The research for this project was completed according to the John - Lab.dox instructions provided by the instructor and was also supported by personal class notes.

## 3. Procedures and Results

To begin evaluating passwords by hacking them a user account must be created on the Windows XP virtual Machine. This step was taken by first spinning up the Windows XP virtual machine hosted by the MacBook Pro and navigating to the user accounts area. A user was created with a username of "Death" and a password of "Diction". This password was chosen for simplicity and is not intended to be in any way secure. After the additional account was created the application PwDump7.exe was executed from the Windows XP command line. This gathered the usernames of all users on the system along with their corresponding password hashes. This information was then copied into a new text file named xppassw.txt and placed on an external USB drive.

After storing the information on the flash drive the Windows XP virtual machine was shut down and the Kali Linux Virtual Machine was booted up. The switch from Windows to Linux was made because the terminal user interface on Linux is superior to Windows for running simple and complex hacking processes via John The Ripper. Nevertheless, image #1 shows how the Linux terminal was utilized to hack the password for the windows Account named Death.



```
root@kali:~/Desktop# john xppassw.txt
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "NT-old"
Use the "--format=NT-old" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 2 password hashes with no different salts (LM [DES 128/128 AVX-16])
Press 'q' or Ctrl-C to abort, almost any other key for status
DICTION (Death:1)
lg 0:00:00:34 0.02% 3/3 (ETA: 2019-04-11 03:58) 0.02916g/s 52315Kp/s 52315Kc/s 530
Warning: passwords printed above might be partial
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
root@kali:~/Desktop#
```

### Hacking Windows credentials with John, Image #1

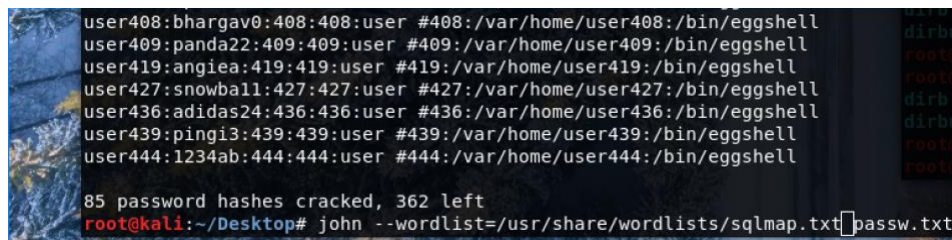
It is clear by the results of the John the Ripper command displayed in Image #1 that the password created on the Windows XP system was cracked with ease by John the Ripper. Now moving on to the second section of this lab where a large log file of Unix password hashes was provided that needed to be hacked. This section was completed in entirety with in the operation system Kali Linux.

---

---

The first step was to download the file “sample unix password list.txt” and then it was renamed passw.txt. The password file was moved to the Desktop for convenience while the hashes were being cracked. The first attempt to crack the passwords was to run John the ripper against the passwords by simply entering “John passw.txt” into the terminal. This resulted in a meager 30 passwords being hacked still leaving over 400 passwords untouched. The terminal was utilized to see a brief selection of the commands that John possesses and a simple explanation of the functionality of each. This research showed that John the ripper could pull in wordlists text files and could have run them against the password hashes. Some simple research into John the ripper provided the discovery that Kali Linux has a folder called “Wordlists” that contains numerous wordlists for hacking.

A quick look into the folder from the command line showed the files “rockyou.txt.gz” and “sqlmap.txt”. Both of these wordlists were implemented within John the Ripper against the password hashes. When rockyou.txt.gz was run against the passwords it was able to bring the number of cracked hashes from 30 to 52. The job was not done and the wordlist “sqlmap.txt” was also run against in John and it resulted in cracking 21 more passwords. The results of running the wordlist “sqlmap.txt” are shown in image #2.



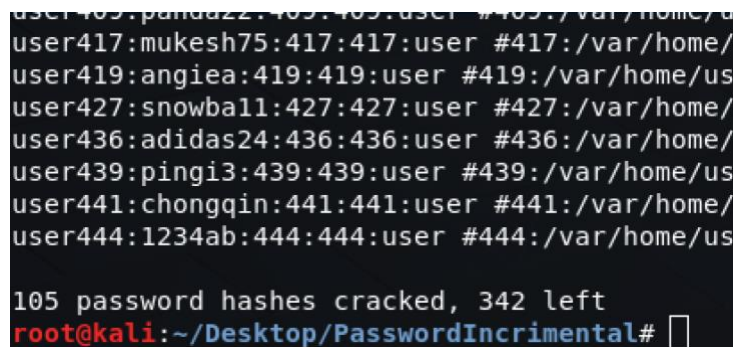
```
user408:bhargav0:408:408:user #408:/var/home/user408:/bin/eggshell
user409:panda22:409:409:user #409:/var/home/user409:/bin/eggshell
user419:angiea:419:419:user #419:/var/home/user419:/bin/eggshell
user427:snowball:427:427:user #427:/var/home/user427:/bin/eggshell
user436:adidas24:436:436:user #436:/var/home/user436:/bin/eggshell
user439:pingi3:439:439:user #439:/var/home/user439:/bin/eggshell
user444:1234ab:444:444:user #444:/var/home/user444:/bin/eggshell

85 password hashes cracked, 362 left
root@kali:~/Desktop# john --wordlist=/usr/share/wordlists/sqlmap.txt passw.txt
```

**Wordlist sqlmap.txt run in John, image #2**

After running those two files the remainder of the preset wordlists in the Wordlists folder were implemented with John, but other than the first two attempts no more passwords hacked.

When the built-in Wordlists were no longer working to crack the password hashes the website <https://wordlists.capsop.com/> was utilized to collect the larger wordlists, “Custom-WPA.txt”, “Wordlist843.txt” and “xaf2”. All three of these lists were run against the password file and each brought back several more results. The “custom-WPA.txt” was able to crack about fifteen more passwords. After running each of these wordlists against the password file it resulted in a total of one hundred and five passwords cracked by wordlists and thirty that were brute forced in the first use of John the Ripper, the results are shown below in Image #3.



```
user417:mukesh75:417:417:user #417:/var/home/
user419:angiea:419:419:user #419:/var/home/
user427:snowball:427:427:user #427:/var/home/
user436:adidas24:436:436:user #436:/var/home/
user439:pingi3:439:439:user #439:/var/home/
user441:chongqin:441:441:user #441:/var/home/
user444:1234ab:444:444:user #444:/var/home/

105 password hashes cracked, 342 left
root@kali:~/Desktop/PasswordIncremental#
```

**Total cracked by Wordlists and 30 brute-forced, Image #3**

---

---

Wordlists were not the last stand in decrypting these password hashes and the new method implemented utilized a unique brute force method. This new process utilized a condition that parallelizes the processes which allows for a significantly faster checking time. The command utilized was "John fork = null passw.txt" and this command cracked one hundred and sixty-six more passwords for a total number of two hundred and seventy-one passwords cracked.

#### 4. Observations

This process of hacking password hashes provided knowledge in understanding the necessity of password strength by length. This lab was successful for two major reasons, firstly that wordlists were utilized to tackle a bulk of the passwords, and secondly the vast number of passwords that were discovered by utilizing the efficient brute force method.

The utilization of the wordlists was significant because it allows for a very dynamic strategy for attacking the passwords. This is because when hacking passwords, it is vital to gather as much situational information about the passwords to help consider what password options may be. By utilizing word lists it allowed John the ripper to evaluate the password hashes based on common words and combinations to provide a faster process of finding the passwords. Another way that wordlist can be used to provide significant benefits to the attack will be through gathering the already hacked passwords and running them against the still hashed passwords to crack them.

The ability to expedite the process of hacking by parallelizing the processes was significant because it allowed many more combinations to be tried against the hashed passwords per minute. This process of efficiently evaluating the passwords enabled the number of hacked passwords to jump from one hundred and five to two hundred and seventy-one in only about thirteen hours of hacking. It is worthy of note that in just the first twenty-five minutes of execution of this new hacking strategy that over thirty-five passwords were cracked. This means that in a very short amount of time this way of hacking was able to overcome an obstacle that the chosen wordlists were not able to.

---