10/9/2021

Lab 2 – IS.6303.001

Password Cracking with FireEye Commando

Joseph Mendez
QXO307

Joseph Mendez
Qxo307
IS-6303-001

The two main focuses for this lab are exploration and installation of the Commando VM as well as its use to crack two password files. These are a SAM pwned file and a shadow password file. We'll start the process with a short set of instructions for Commando's installation, then move on towards analyzing the tool categories along with some of the tools installed. Afterwards a brief word on how to get the password files off the computer and a tool that can be used to convert the hashes into plaintext. Lastly, we'll attempt to crack the passwords using a password cracking tool.

First off, we are provided a link to Commando's *GitHub* page so we can download the VM and install it on a Windows VM. A bit of background of the VM of interest; the Commando VM is a Windows-based virtual machine created by Mandiant Fire-Eye to serve as an alternative to the Linux-based *Kali* platform. One of the key aspects of the Commando VM, and one of my personal favorite features, is its modularity. Thanks to services such as *Chocolately*, the VM can install individual packets; allowing for increased customization and versatility to fit both red team and blue teamers. Commando's Mandiant website states that the VM comes with 140 tools right off the bat. Some of these tools, such as *Nmap*, *Wireshark*, and *Sysinternals*, being key tools in the industry [1].

Moving on to a quick overview of Commando's installation, I'll be summarizing the steps on Commando's Mandiant website as those are more thorough than the instructions on the GitHub page. Mandiant states that the user should create a Windows virtual machine with the following specs:

- Using Windows 7 Service Pack 1 or Windows 10
  - To be more specific, the *GitHub* mentions that acceptable versions of Windows 10 include *1803*, *1809*, *1903*, *1909*, *2004*, *20H2*, and *21H1* [2].
- VM created should have a minimum of *60 GB* of space
- And *2 GB* of memory.

Once the Windows VM creation is complete, users should download the Commando Repository from *GitHub* and decompress it. Then, open a *PowerShell* page as administrator and navigate to the directory where the repository is saved. Once there, users should run the "*Set-ExecutionPolicy unrestricted*" command and respond with "*Y*" when prompted. Afterwards, users should run the *install.ps1* file within the Shell and input their password when directed. After this is done, the script will run and install the necessary files on its own. According to the website, the installation itself should take around 2 to 3 hours to complete. Due to various installation requirements, the VM will also reboot various times to complete these installation requirements [1]. Once everything is done, the desktop will look as follows:

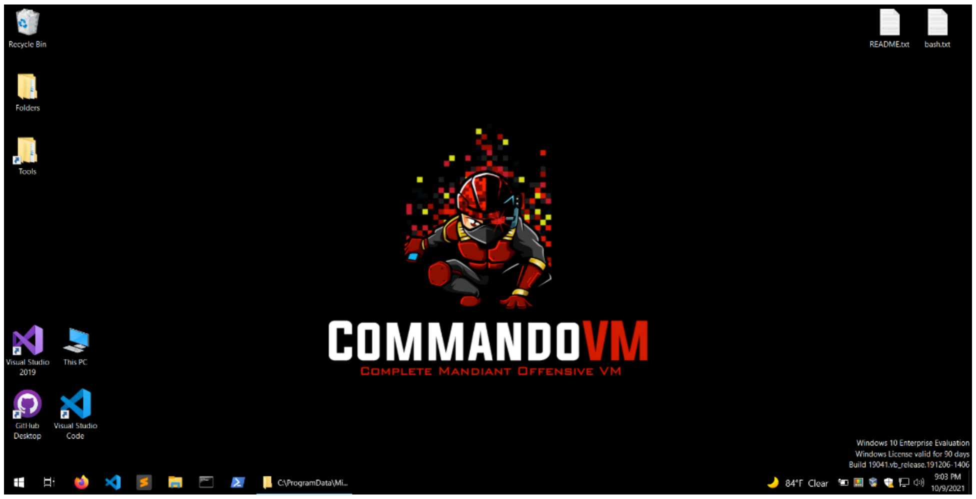


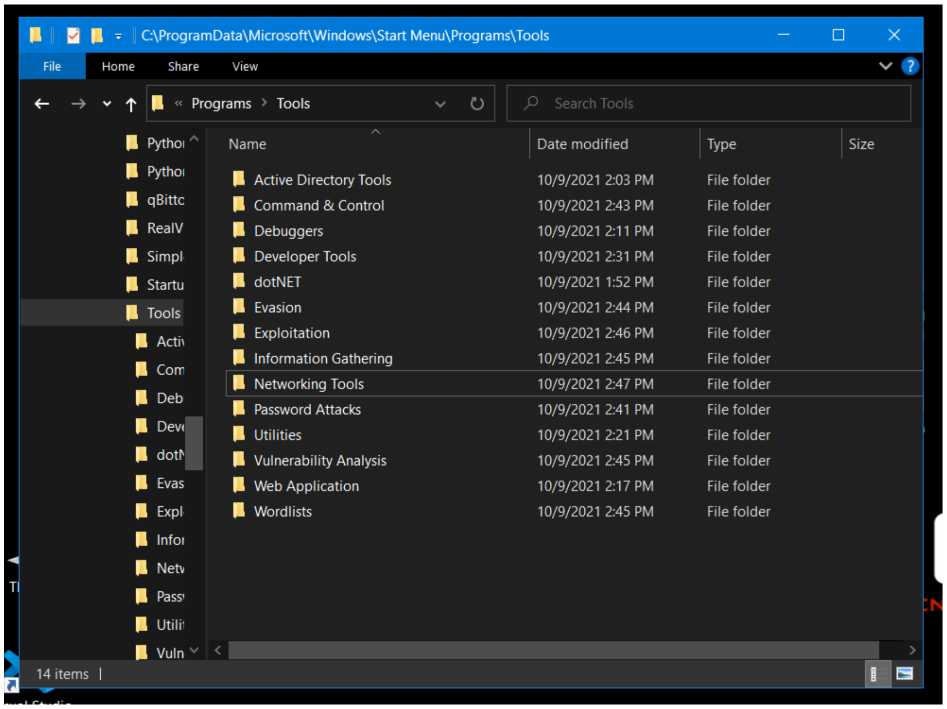
Figure 1 – Commando VM desktop

Now, moving on to our first main section of the report, let's navigate to the *Tools* shortcut shown in Figure 1. As we can see below in Figure 2, the tools are all divided up into 14 different folders. These folders are as follows:

1. **Active Directory Tools** – these are tools that help managing the active directory. One such tool is ***SysInternals***; a tool that helps "manage, troubleshoot, and diagnose your Windows systems and applications" [8].
2. **Command and Control** – C2C tools help users use their own systems to take control of vulnerable victim systems after exploiting them. ***Covenant*** is one such tool. It is a C2C framework that works to display the vulnerabilities of *.NET*. According to its *GitHub* page, the tool has a myriad of features, such as being multi-platform thanks to targeting *.NET Core* and allowing for the creation of listener profiles [4].
3. **Debuggers** – these tools, such as ***Windbg.x64***, allow users to debug their systems. The tool helps users analyze and examine both crash dumps and *CUP* registers [9].
4. **Developer Tools** – tools in this category are to allow users to develop apps, scripts, and others. In our case, we have the ***Microsoft Visual Build Tools*** which are a developer tool set for *Microsoft Visual Studio*.
5. **dotNET** – the dotNET tools are those that deal with *.NET*. For example, the ***dnSpy*** tool serves as a debugger and editor. According to its *GitHub* page, this tool can both debug and edit *.NET* and *Unity* assemblies [10].
6. **Evasion** – evasion tools allow hackers or red teamers to push an exploit or malware to their target without detection. A popular evasion tool is ***Demiguise***. This tool creates an *HTML* file that hides an encrypted *HTA* file within it. By doing so, unknowing victims access the website with the malicious *HTML* and decrypt the *HTA* file, at which point the payload is delivered [5].
7. **Exploitation** – useful to any red teamer, exploitation tools help take advantage of vulnerable or target systems' faults for one's use. ***Invoke-ACLpwn*** is a tool under this

category as it functions like the *Bloodhound* tool, which will be discussed below. Instead of just gathering information from the active directory, it also attempts to exploit any found vulnerabilities to gain access to domain admin. Another good tool within this category is ***Mimikatz***, a tool that allows users to exploit the active directory and use attacks such as ***Pass-the-Hash*** [3].

8. **Information Gathering** – tools falling under this category allow users to scope out the network and other parts of a system as to gather information on potential targets. One of the key tools in this category is the ***Nmap*** tool. This allows users to scan a network and see what hosts are available, along with any services being run and ports open on the hosts. On the earlier note of this VM also being useful to blue teamers, the ***BloodHound*** tool scans the active directory and helps users understand possible attack vectors due to configurations [3].
9. **Networking Tools** – these are tools that allow users to monitor and manipulate various aspects of a network or server network. For example, the ***ProxyCap*** tool is "capable of easily rerouting all your network traffic through a chain of proxies" [6]. Another tool in this folder is the ***Wireshark*** tool, which captures the packet on a network for users to analyze.
10. **Password Attacks** – as the name of the folder implies, the focus of these tools is to exploit and get access to password-protected accounts. The ***Invoke-TheHash*** tool performs a pass the hash attack with the goal of executing commands on remote systems [7]. The ***CredNinja*** tool is another useful one that helps users know which credentials on network are valid or invalid [8].
11. **Utilities** – these are just general tools that don't necessarily fall within the other tools. Some here include the hex editor ***HxD*** and ***DB Browser for SQL***. However, a useful tool that I personally use often is the ***CyberChef*** tool. A "jack-of-all trades" tool, it does a myriad of things such as encoding, deciphering, measuring entropy, and others [8].
12. **Vulnerability Analysis** – tools in this category are meant to help users find attack vectors or weak spots that are vulnerable to a malicious attack. As such, the tools in this category are of great use to both blue and red teamers. One such tool is ***NtdsAudit***, a tool that can display data on passwords as well as dump their hashes [15].
13. **Web Application** – the tools within this category are helpful for those who wish to test the security of their web applications. Tools such as ***Burp Suite*** can help with this by using a proxy to navigate through a user's website while scanning for vulnerabilities [3].
14. **Wordlists** – these "tools" are mostly just a large repository of dictionaries that users can use during password cracking.



C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tools

Name	Date modified	Type	Size
Active Directory Tools	10/9/2021 2:03 PM	File folder	
Command & Control	10/9/2021 2:43 PM	File folder	
Debuggers	10/9/2021 2:11 PM	File folder	
Developer Tools	10/9/2021 2:31 PM	File folder	
dotNET	10/9/2021 1:52 PM	File folder	
Evasion	10/9/2021 2:44 PM	File folder	
Exploitation	10/9/2021 2:46 PM	File folder	
Information Gathering	10/9/2021 2:45 PM	File folder	
Networking Tools	10/9/2021 2:47 PM	File folder	
Password Attacks	10/9/2021 2:41 PM	File folder	
Utilities	10/9/2021 2:21 PM	File folder	
Vulnerability Analysis	10/9/2021 2:45 PM	File folder	
Web Application	10/9/2021 2:17 PM	File folder	
Wordlists	10/9/2021 2:45 PM	File folder	

Figure 2 – Commando VM Tools folder

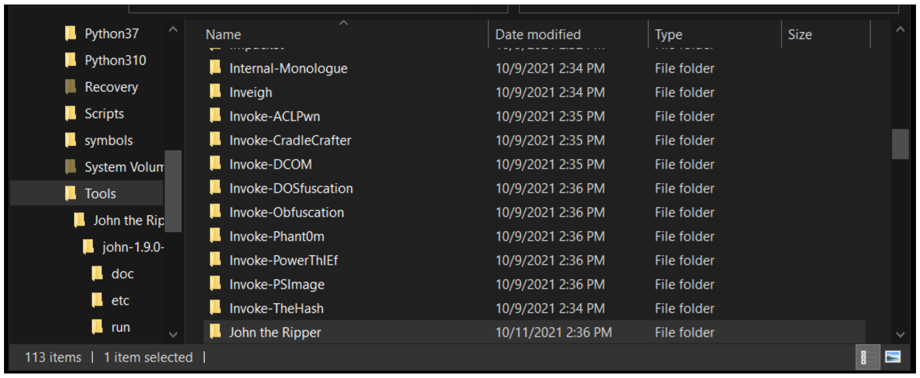
For a quick side note before moving on to the second bulk of the report, let's talk about password and password hash recovery. When it comes to the job of cyber analysts, mainly forensic specialists, the task of password recovery is one that comes up more often than expected. Reasons for this can vary, for example one situation could be a client asking for a computer or mobile device password to be recovered. A more forensic situation could be needing to access the computer of a suspect to gather evidence. To do so, analysts would have to first dump the password files then crack the hashes contained within.

When it comes to password dumping, there are many tools that can be used, some of which even come installed with the Commando VM. One such tool is the previously mentioned *Mimikatz*. In this case, by running the "*privilege::debug*", "*token::elevate*", and "*lsadump::sam*" commands, users can access and dump the *SAM* password file. Another tool that can be used is the *Impacket* tool, which will extract the *SAM* file with the following command: "*./secretsdump.py -sam /root/Desktop/sam -system /root/Desktop/system LOCAL*". *Metasplot* can also be used to extract the file through various methods such as a *HashDump*, *credential_collector*, and *load wiki* [13].

Those tools help users recover the password hashes, but not the password's plaintext itself. To get this plaintext, users need to use a password hash cracking software. In this case, a solid password cracking tool is *John the Ripper*. This software is well known in the community for being able to use multiple attacks, being able to detect the hash type of the password and working on passwords for *ZIP* and *PDF* files [13]. The process of how to crack passwords with this software will be explained in the following section of the report.

With that out of the way, let's move on the fun part of this report: password cracking. With the techniques discussed above, cyber specialists should have the hashes for user passwords at hand. The issue, however, is that these passwords are hashed and not in plaintext. To be able to get these passwords to plaintext, we'll have to use a password cracking tool. In terms of this report, we are presented with the issue of not having any such tool within our VM. The password attacking tools that do come installed by default on Commando don't need to "un-hash" the password to function. For this task, we'll be using the previously password cracking tool *John the Ripper* as well as its GUI, *Johnny*.

To start off, we'll navigate to the download page for *John the Ripper*, download the *.zip* file, and extract it into our "*Tools*" folder within our *Local Disk* directory. From there we will repeat the same process with the download for *Johnny*. After the installation for *Johnny* is complete, it'll create a shortcut to itself in the desktop; this will be moved to the "*Password Attacks"* folder we discussed earlier in the report. Before we can start cracking any passwords, we must first configure *Johnny* to have the directory address for *John.exe*. This can be done by going down to *Settings* and browsing for the executable file in the "*John the Ripper executable*" section.



Name	Date modified	Type	Size
Internal-Monologue	10/9/2021 2:34 PM	File folder	
Inveigh	10/9/2021 2:34 PM	File folder	
Invoke-ACLPwn	10/9/2021 2:35 PM	File folder	
Invoke-CradleCrafter	10/9/2021 2:35 PM	File folder	
Invoke-DCOM	10/9/2021 2:35 PM	File folder	
Invoke-DOSfuscation	10/9/2021 2:36 PM	File folder	
Invoke-Obfuscation	10/9/2021 2:36 PM	File folder	
Invoke-Phant0m	10/9/2021 2:36 PM	File folder	
Invoke-PowerThIEf	10/9/2021 2:36 PM	File folder	
Invoke-PSImage	10/9/2021 2:36 PM	File folder	
Invoke-TheHash	10/9/2021 2:34 PM	File folder	
John the Ripper	10/11/2021 2:36 PM	File folder	

113 items | 1 item selected |

Figure 3 – John the Ripper downloaded

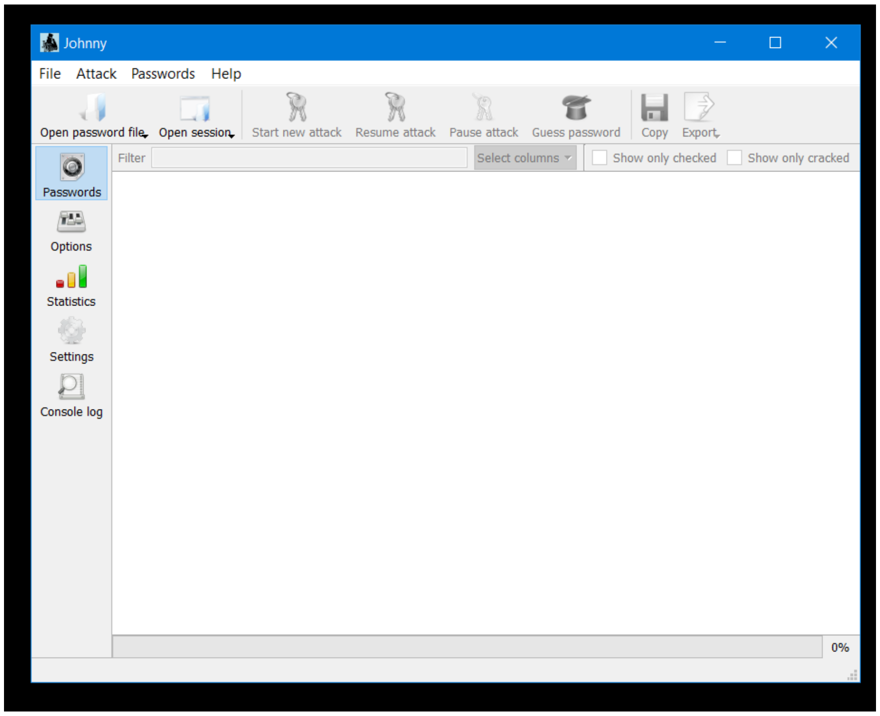
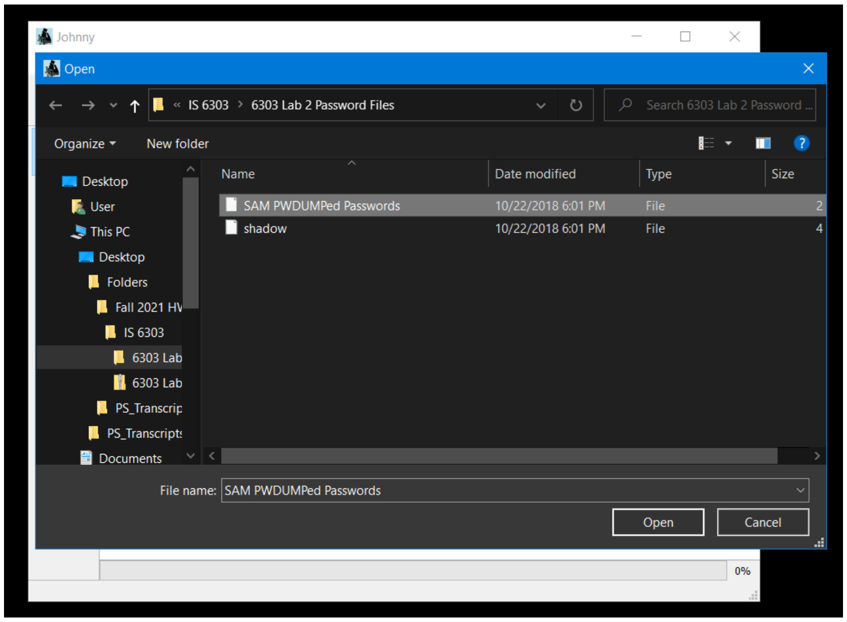


Figure 4 – Johnny GUI

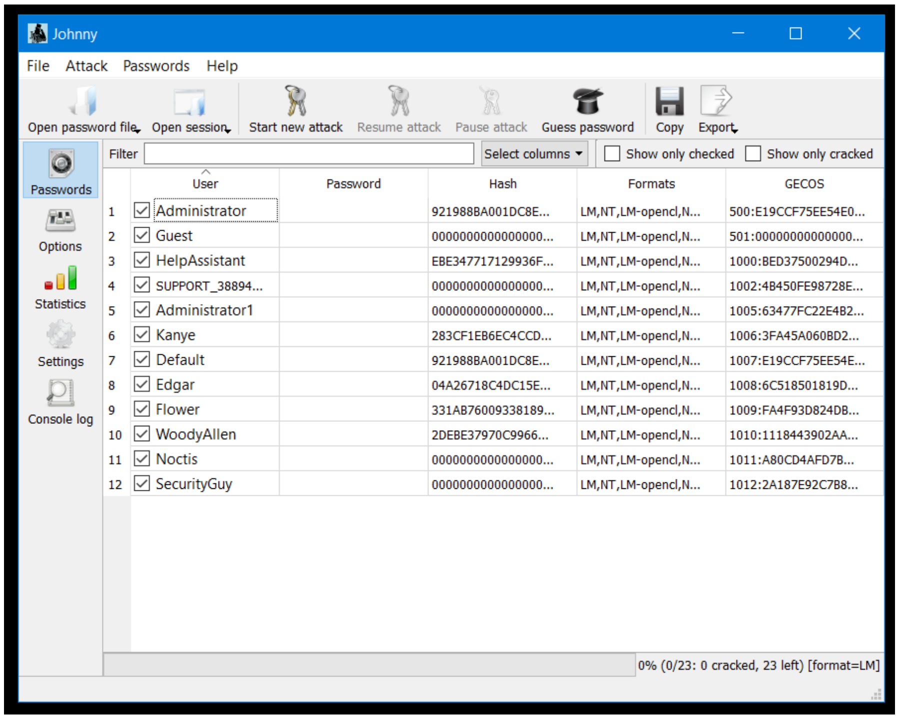
With this done, we can now start cracking the first of the two files: *SAM PWDUMPed Passwords*. The first step in the process is to click on the "*Open PASSWD File*", then navigating to and selecting the desired file. *Johnny* will then display the screen shown on Figure 6. Before trying out any other options, we will run a default attack. This can be done by selecting the "*Start a New Attack*" option.



Name	Date modified	Type	Size
SAM PWDUMPed Passwords	10/22/2018 6:01 PM	File	2
shadow	10/22/2018 6:01 PM	File	4

File name: SAM PWDUMPed Passwords

Figure 5 – Selecting the SAM PWDUMPed Passwords file


Johnny
File Attack Passwords Help
Open password file Open session Start new attack Resume attack Pause attack Guess password Copy Export
Passwords
Options
Statistics
Settings
Console log
Filter
Select columns
Show only checked
Show only cracked

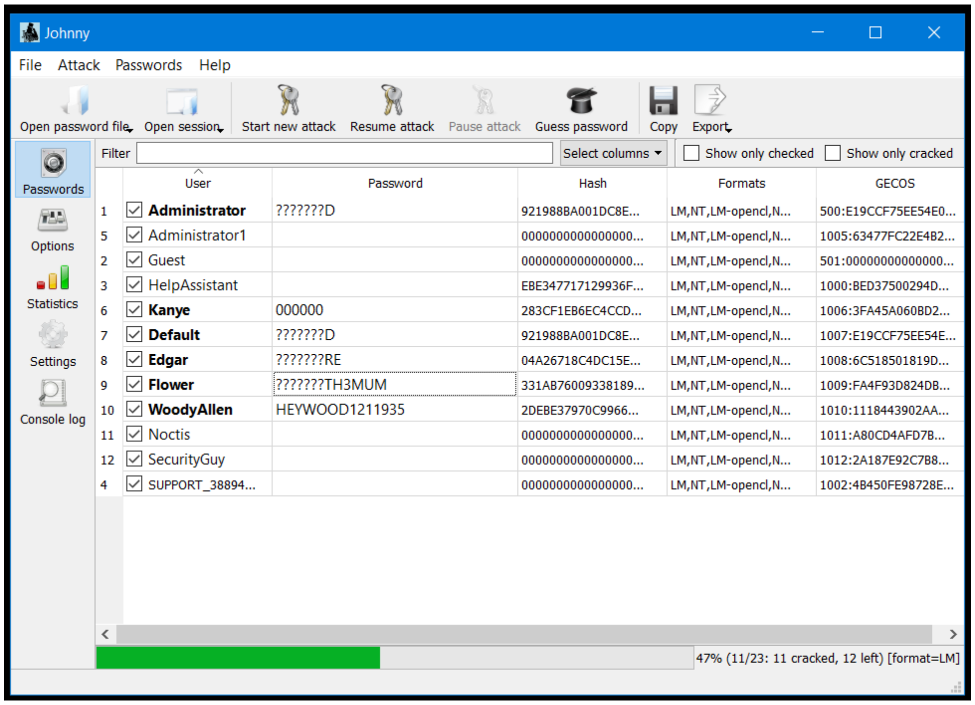
		User	Password	Hash	Formats	GECOS
1	✓	Administrator		921988BA001DC8E...	LM,NT,LM-opencl,N...	500:E19CCF75EE54E0...
2	✓	Guest		0000000000000000...	LM,NT,LM-opencl,N...	501:0000000000000...
3	✓	HelpAssistant		EBE347717129936F...	LM,NT,LM-opencl,N...	1000:BED37500294D...
4	✓	SUPPORT_38894...		0000000000000000...	LM,NT,LM-opencl,N...	1002:4B450FE98728E...
5	✓	Administrator1		0000000000000000...	LM,NT,LM-opencl,N...	1005:63477FC22E4B2...
6	✓	Kanye		283CF1EB6EC4CCD...	LM,NT,LM-opencl,N...	1006:3FA45A060BD2...
7	✓	Default		921988BA001DC8E...	LM,NT,LM-opencl,N...	1007:E19CCF75EE54E...
8	✓	Edgar		04A26718C4DC15E...	LM,NT,LM-opencl,N...	1008:6C518501819D...
9	✓	Flower		331AB76009338189...	LM,NT,LM-opencl,N...	1009:FA4F93D824DB...
10	✓	WoodyAllen		2DEBE37970C9966...	LM,NT,LM-opencl,N...	1010:1118443902AA...
11	✓	Noctis		0000000000000000...	LM,NT,LM-opencl,N...	1011:A80CD4AFD7B...
12	✓	SecurityGuy		0000000000000000...	LM,NT,LM-opencl,N...	1012:2A187E92C7B8...

0% (0/23: 0 cracked, 23 left) [format=LM]

Figure 6 - SAM PWDUMPed Passwords hashes in Johnny

Within the first couple of minutes, the basic attack has partially cracked some of the passwords; mainly those of Administrator, Kanye, Default, Edgar, and Flower. It has also managed to fully crack the password for the WoodyAllen user account. That password is "**HEYWOOD1211935**". A screenshot of the attack in that state will be shown in Figure 7. In that screenshot, we also see the password for the Kanye account: "**000000**". Here we also see that the partial cracks have multiple "?" in their plaintext. By navigating over to the "Console Log" we can see that many of these passwords are fragmented into two parts. We also see that the HelpAssistant account was partially cracked. Below are all the fully cracked accounts and passwords thus far for the *SAM PWDUMPed Passwords* file:

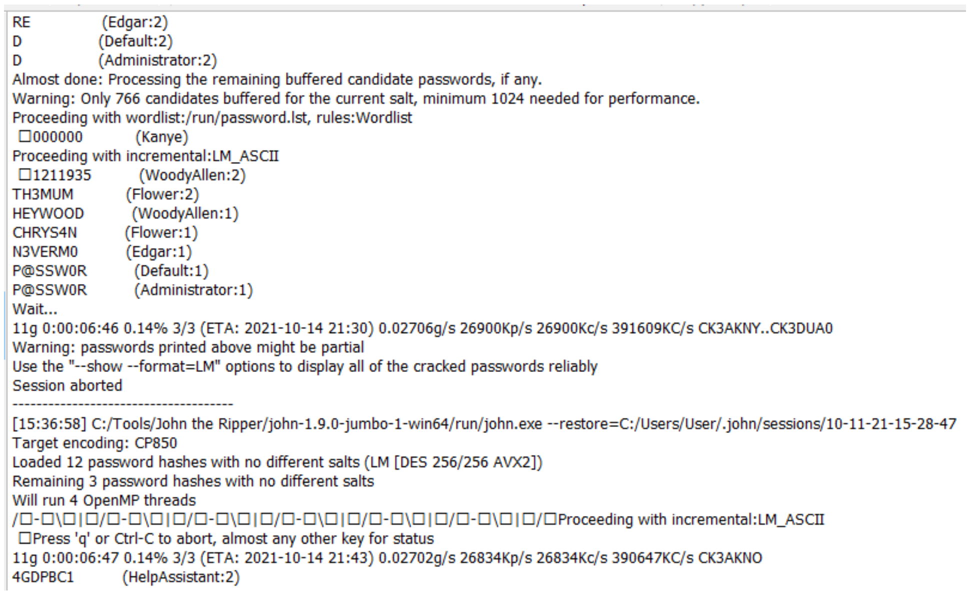
- Administrator – **P@SSW0RD**
- Default – **P@SSW0RD**
- Edgar – **N3VERM0RE**
- Flower – **CHRYS4NTH3MUM**
- Kanye – **000000**
- WoodyAllen - **HEYWOOD1211935**



	User	Password	Hash	Formats	GECOS
1	☑ **Administrator**	???????D	921988BA001DC8E...	LM,NT,LM-opencl,N...	500:E19CCF75EE54E0...
5	☑ Administrator1		0000000000000000...	LM,NT,LM-opencl,N...	1005:63477FC22E4B2...
2	☑ Guest		0000000000000000...	LM,NT,LM-opencl,N...	501:00000000000000...
3	☑ HelpAssistant		EBE347717129936F...	LM,NT,LM-opencl,N...	1000:BED37500294D...
6	☑ **Kanye**	000000	283CF1EB6EC4CCD...	LM,NT,LM-opencl,N...	1006:3FA45A060BD2...
7	☑ **Default**	???????D	921988BA001DC8E...	LM,NT,LM-opencl,N...	1007:E19CCF75EE54E...
8	☑ **Edgar**	???????RE	04A26718C4DC15E...	LM,NT,LM-opencl,N...	1008:6C518501819D...
9	☑ **Flower**	???????TH3MUM	331AB76009338189...	LM,NT,LM-opencl,N...	1009:FA4F93D824DB...
10	☑ **WoodyAllen**	HEYWOOD1211935	2DEBE37970C9966...	LM,NT,LM-opencl,N...	1010:1118443902AA...
11	☑ Noctis		0000000000000000...	LM,NT,LM-opencl,N...	1011:A80CD4AFD7B...
12	☑ SecurityGuy		0000000000000000...	LM,NT,LM-opencl,N...	1012:2A187E92C7B8...
4	☑ SUPPORT_38894...		0000000000000000...	LM,NT,LM-opencl,N...	1002:4B450FE98728E...

47% (11/23: 11 cracked, 12 left) [format=LM]

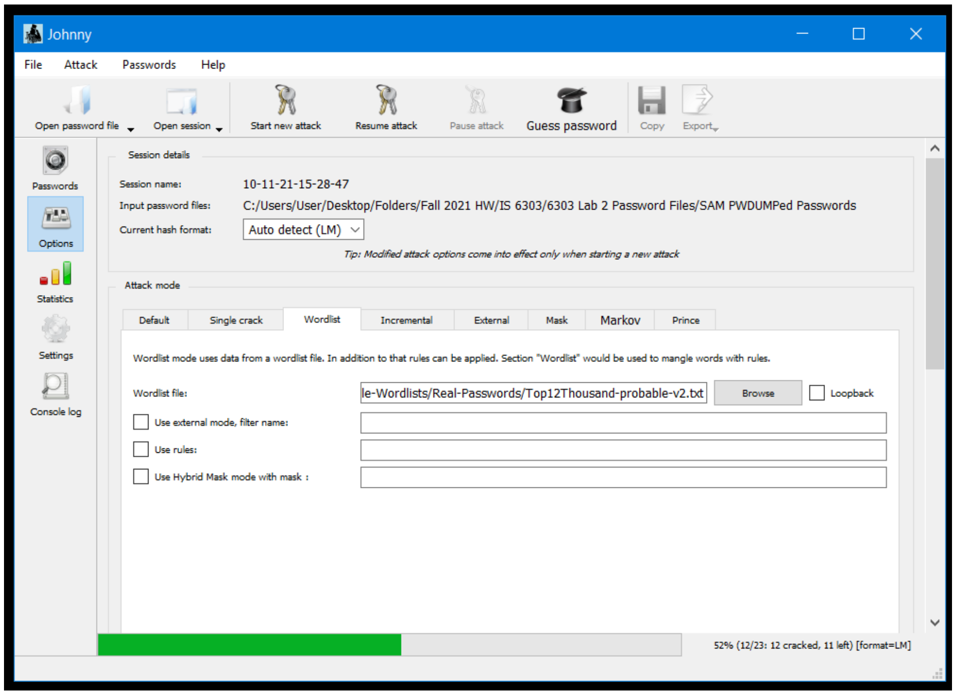
Figure 7 – Fully cracked and partially cracked passwords



```
RE              (Edgar:2)
D               (Default:2)
D               (Administrator:2)
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 766 candidates buffered for the current salt, minimum 1024 needed for performance.
Proceeding with wordlist:/run/password.lst, rules:Wordlist
 ☐000000          (Kanye)
Proceeding with incremental:LM_ASCII
 ☐1211935         (WoodyAllen:2)
TH3MUM          (Flower:2)
HEYWOOD          (WoodyAllen:1)
CHRYS4N         (Flower:1)
N3VERM0         (Edgar:1)
P@SSW0R          (Default:1)
P@SSW0R          (Administrator:1)
Wait...
11g 0:00:06:46 0.14% 3/3 (ETA: 2021-10-14 21:30) 0.02706g/s 26900Kp/s 26900Kc/s 391609KC/s CK3AKNY..CK3DUA0
Warning: passwords printed above might be partial
Use the "--show --format=LM" options to display all of the cracked passwords reliably
Session aborted
------------------------------------
[15:36:58] C:/Tools/John the Ripper/john-1.9.0-jumbo-1-win64/run/john.exe --restore=C:/Users/User/.john/sessions/10-11-21-15-28-47
Target encoding: CP850
Loaded 12 password hashes with no different salts (LM [DES 256/256 AVX2])
Remaining 3 password hashes with no different salts
Will run 4 OpenMP threads
/☐-☐\☐|☐/☐-☐\☐|☐/☐-☐\☐|☐/☐-☐\☐|☐/☐-☐\☐|☐/☐-☐\☐|☐/☐Proceeding with incremental:LM_ASCII
 ☐Press 'q' or Ctrl-C to abort, almost any other key for status
11g 0:00:06:47 0.14% 3/3 (ETA: 2021-10-14 21:43) 0.02702g/s 26834Kp/s 26834Kc/s 390647KC/s CK3AKNO
4GDPBC1          (HelpAssistant:2)
```

Figure 8 – Both halves of the partially cracked passwords

With around an hour and a half in the incremental phase of a default attack and no more un-hashing done, let's make the attack more specific to see if we get better results. By navigating to the "*Options"* section in *Johnny*, we can set which method we want to use to crack passwords. Previously set on "*Default*", let's navigate to the "*Wordlist*" option. Here, we can browse for different wordlists to use. Thankfully, Commando comes installed with quite a few wordlists for us to use. We'll use one by clicking on the "*Browse*" option, then navigating to the "*Wordlists*" folder, then "*Portable-Wordlists*", then "*Real-Passwords*", and select the "*Top12Thousand-probable-v2.txt*" file. From there we can select the "*Start New Attack*" option and see what passwords are generated. Sadly, this did not generate any passwords after having run for around an hour. On that note, the other types of attacks were tried, along with most of the provided wordlists. No results either. Another attempt was made, as the pwdumped password hashes follow the following format: *"<Username>:<User ID>:<LM hash>:<NT hash>:<Comment>:<Home Dir>:"* [12]. As such, by deleting the *User ID*, *LM Hash*, and *Comment* areas we should be left with just the *Username* and *NT hash*. Running that through *Jack the Ripper*, we can set the hash type to "*NT*" and run another more scan [12]. After an hour this also did not yield any results. One more attempt was made, for the original password file, this time by using the popular wordlist *rockyou.lst*. But no passwords were cracked with it either. Despite these efforts, there was one password that was partially cracked; the password for HelpAssistant being **???????4GDPBC1**.



Session details

Session name: 10-11-21-15-28-47
Input password files: C:/Users/User/Desktop/Folders/Fall 2021 HW/IS 6303/6303 Lab 2 Password Files/SAM PWDUMPed Passwords
Current hash format: Auto detect (LM)

Tip: Modified attack options come into effect only when starting a new attack

Attack mode

Wordlist mode uses data from a wordlist file. In addition to that rules can be applied. Section "Wordlist" would be used to mangle words with rules.

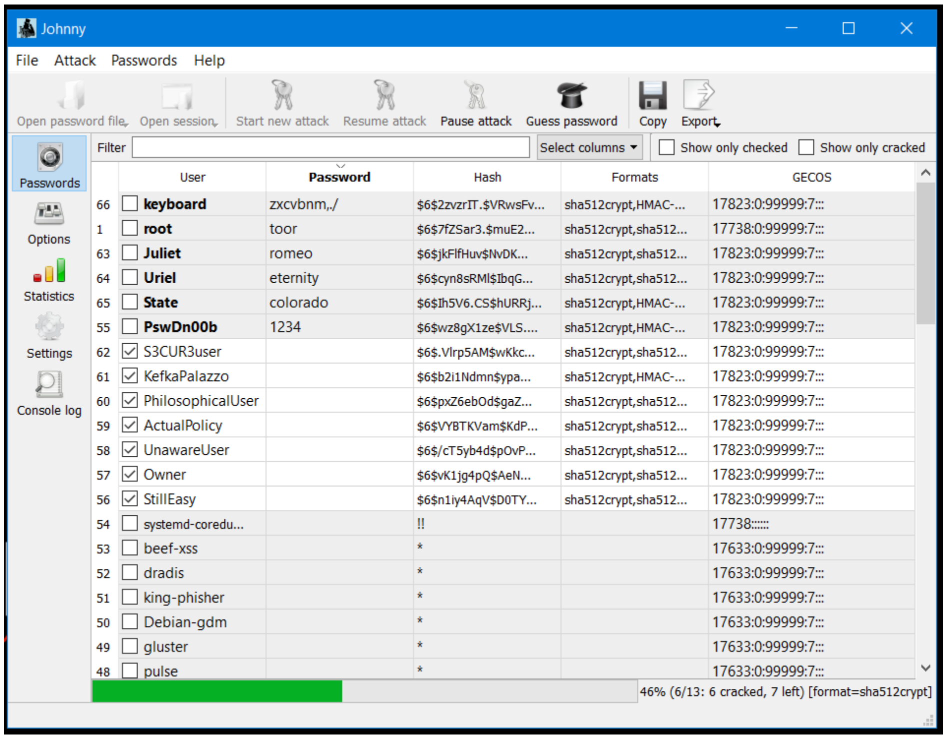
Wordlist file: le-Wordlists/Real-Passwords/Top12Thousand-probable-v2.txt
Loopback
Use external mode, filter name:
Use rules:
Use Hybrid Mask mode with mask :

52% (12/23: 12 cracked, 11 left) [format=LM]

Figure 9 – Wordlist attack with a specified wordlist

Moving on from that, we still have the *shadow* file that needs to be decrypted. We can start the process off by, much similarly to the work above, clicking on the "*Open PASSWD File*", then navigating to and selecting the "*shadow*" file. Thanks to an article on cybercity, we know that the shadow file format contains an indicator to know the hash type; the format for this is "*$id$salt$hashed*". As shown in the figure below, all the hashes contain "*$6$*" at the very beginning. This indicates that the hashes are in the *SHA-512* format [14]. Once the file is in *Johnny*, we'll run a default attack to see what passwords are gathered. Thankfully for us, we go get a few passwords almost immediately. We then ran another attack, this time using a wordlist attack with the *rockyou.lst*. Thankfully with this wordlist, more passwords were cracked, leaving around half of the remaining passwords uncracked. And similarly to the previous password file, other attacks and wordlists were used (each for around an hour or two) to see if more passwords were obtained, but none were in the end. Along with the passwords gathered in the previous attacks, these are the ones that we've been able to gather thus far:

- keyboard - **zxcvbnm,./**
- root – **toor**
- Juliet – **romeo**
- Uriel – **eternity**
- State – **colorado**
- PswDn00b – **1234**



		User	Password	Hash	Formats	GECOS
66	☐	**keyboard**	zxcvbnm,./	$6$2zvzrIT.$VRwsFv...	sha512crypt,HMAC-...	17823:0:99999:7:::
1	☐	**root**	toor	$6$7fZSar3.$muE2...	sha512crypt,sha512...	17738:0:99999:7:::
63	☐	**Juliet**	romeo	$6$jkFlfHuv$NvDK...	sha512crypt,sha512...	17823:0:99999:7:::
64	☐	**Uriel**	eternity	$6$cyn8sRMl$IbqG...	sha512crypt,sha512...	17823:0:99999:7:::
65	☐	**State**	colorado	$6$Ih5V6.CS$hURRj...	sha512crypt,HMAC-...	17823:0:99999:7:::
55	☐	**PswDn00b**	1234	$6$wz8gX1ze$VLS....	sha512crypt,HMAC-...	17823:0:99999:7:::
62	✓	S3CUR3user		$6$.Vlrp5AM$wKkc...	sha512crypt,sha512...	17823:0:99999:7:::
61	✓	KefkaPalazzo		$6$b2i1Ndmn$ypa...	sha512crypt,HMAC-...	17823:0:99999:7:::
60	✓	PhilosophicalUser		$6$pxZ6ebOd$gaZ...	sha512crypt,sha512...	17823:0:99999:7:::
59	✓	ActualPolicy		$6$VYBTKVam$KdP...	sha512crypt,sha512...	17823:0:99999:7:::
58	✓	UnawareUser		$6$/cT5yb4d$pOvP...	sha512crypt,sha512...	17823:0:99999:7:::
57	✓	Owner		$6$vK1jg4pQ$AeN...	sha512crypt,sha512...	17823:0:99999:7:::
56	✓	StillEasy		$6$n1iy4AqV$D0TY...	sha512crypt,sha512...	17823:0:99999:7:::
54	☐	systemd-coredu...		!!		17738::::::
53	☐	beef-xss		*		17633:0:99999:7:::
52	☐	dradis		*		17633:0:99999:7:::
51	☐	king-phisher		*		17633:0:99999:7:::
50	☐	Debian-gdm		*		17633:0:99999:7:::
49	☐	gluster		*		17633:0:99999:7:::
48	☐	pulse		*		17633:0:99999:7:::

46% (6/13: 6 cracked, 7 left) [format=sha512crypt]

Figure 10 – Some of the cracked shadow passwords

To start wrapping everything up; looking back at everything we've covered in this report, I would conclude that Commando is a rather useful VM. I remember when I first started my bachelor's I was completely new to the world of cyber security. One of the things I remember thinking about while I went through my classes was that it would be nice if there was a VM like *Kali* that ran on a Microsoft OS, since it was what I was familiar with the most at the time. The familiarity that came from the OS plus all the tools that come installed with it are what sell me on Commando. That, along with its customizability make it the ideal VM for red and blue teamers who are more comfortable with the Microsoft OS.

Joseph Mendez
Qxo307
IS-6303-001

In terms of the password cracking, I feel that this was a pretty good exercise as it helps users who are not familiar with the process become acquainted. I already knew how to work with password cracking software to various degrees of success, and this exercise helped either solidify my knowledge or teach me new things as I scoured various sources and tutorials. It also helps those without the experience know how difficult it is to crack a well-made password and, by extension, their importance.

Joseph Mendez
Qxo307
IS-6303-001

Bibliography

[1] Barteaux, Jacob. "Commando VM: The First of Its Kind Windows Offensive Distribution." *Mandiant*, 10 Dec. 2021, https://www.mandiant.com/resources/commando-vm-windows-offensive-distribution.

[2] Barteaux, Jake. "Complete Mandiant Offensive VM (Commando VM), a Fully Customizable Windows-Based Pentesting Virtual Machine Distribution." *GitHub*, https://github.com/mandiant/commando-vm.

[3] Joyce, Kevin. "Commando VM: Introduction." *Stealthbits Technologies*, 3 Nov. 2020, https://stealthbits.com/blog/commando-vm-introduction/.

[4] Cobbr. "Cobbr/Covenant: Covenant Is a Collaborative .NET C2 Framework for Red Teamers." *GitHub*, https://github.com/cobbr/Covenant.

[5] Nccgroup. "Nccgroup/Demiguise: HTA Encryption Tool for Redteams." *GitHub*, https://github.com/nccgroup/demiguise.

[6] "How to Work with ProxyCap: Proxycap Tutorial: PROXYCAP Manual." *ProxyCap - Manual (Tutorial)*, https://5socks.net/Manual/ProxyCap-eng.html.

[7] Alex. "Pass-the-Hash Attack (How to Use NTLM without Cracking a Password)." *Ethical Hacking and Penetration Testing*, 5 Feb. 2020, https://miloserdov.org/?p=4142.

[8] Alex. "Commando VM: Windows for Hackers." *Ethical Hacking and Penetration Testing*, 6 Aug. 2019, https://miloserdov.org/?p=3161#6.

[9] Marshall, Don, et al. "Download Debugging Tools for Windows - Windbg - Windows Drivers." *WinDbg - Windows Drivers | Microsoft Docs*, 16 June 2021, https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/debugger-download-tools.

[10] dnSpy. "DnSpy/Dnspy: .Net Debugger and Assembly Editor." *GitHub*, https://github.com/dnSpy/dnSpy.

[11] Felix. "What Is aad3b435b51404eeaad3b435b51404ee?" *You Gotta Hack That | Penetration Testing Consultancy*, 26 July 2013, https://yougottahackthat.com/blog/339/what-is-aad3b435b51404eeaad3b435b51404ee.

[12] Borges, Alexandre. “Ntroduction to Password Cracking – Part 1.” *WordPress*, https://alexandreborgesbrazil.files.wordpress.com/2013/08/introduction_to_password_cracking_part_1.pdf.

[13] Chandel, Raj. “Credential Dumping: Sam.” *Hacking Articles*, 8 Apr. 2020, https://www.hackingarticles.in/credential-dumping-sam/.

[14] Gite, Vivek. “Understanding /Etc/Shadow File Format on Linux.” *NixCraft*, 4 Mar. 2021, https://www.cyberciti.biz/faq/understanding-etcshadow-file/.

[15] Dionach. “Dionach/Ntdsaudit: An Active Directory Audit Utility.” *GitHub*, https://github.com/Dionach/NtdsAudit.