Deciphering Case Study

James Early
Bastian Tenbergen

November 2021

Deciphering

Background

Encrypting and cyphering messages is common practice for many applications. From encrypting internet traffic, storing payment information, or exchanging private data, the act of obfuscating plain text using a systematic method is a central aspect of modern life and telecommunications. A plethora of encryption/decryption methods exist, some are more robust than others, and some are more useful depending on the application case. Picking the right cypher for the right application case to prevent unauthorized parties to access the information is therefore a key concern.

Case Study Overview

In this case study, we will explore the different types of application cases for different kind of encryption methods and apply techniques, tools, and strategies to decipher the hidden messages. Cunning and clever thinking may be required, or in the absence of that, perhaps some googling.

Student Instructions

Task 1:

Let the following be an encrypted message you intercepted from some type of secret society, calling its members to action. You know it is a type of substitution cypher, but you don't know the substitution alphabet. Decrypt message:

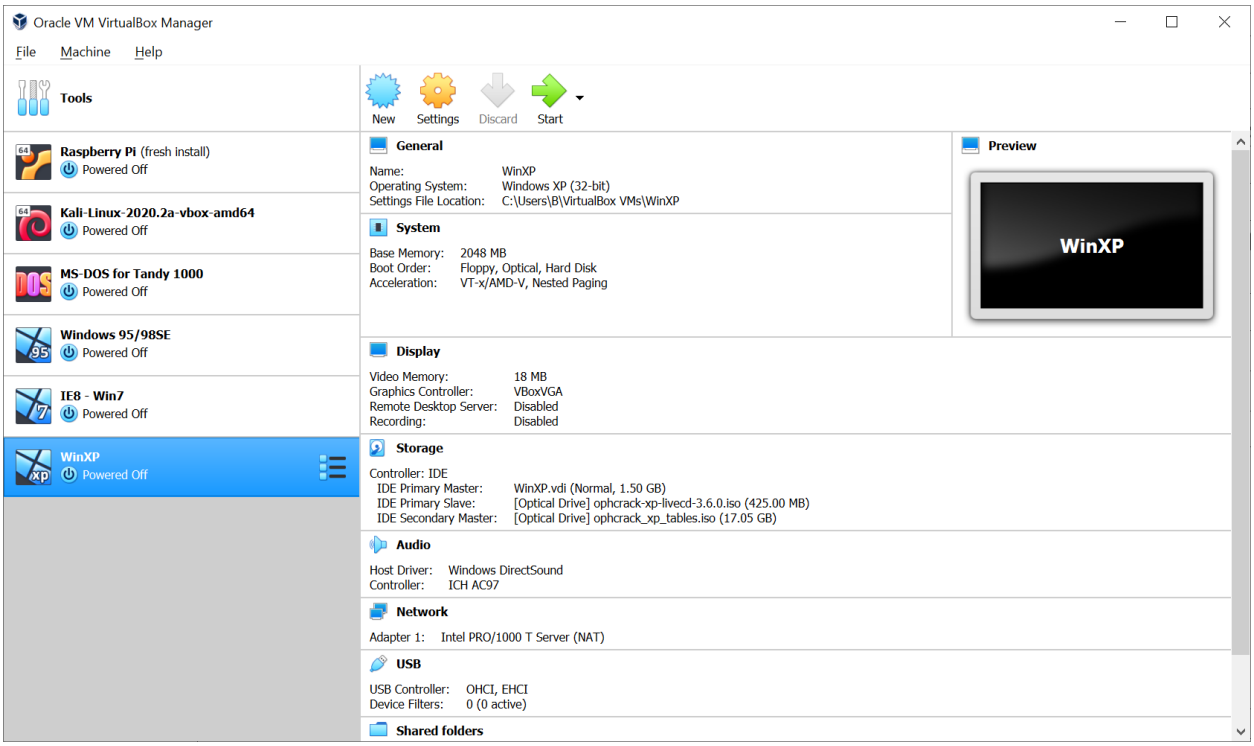
```
nq egdq fa pduzw kagd ahmxfuzq
```

Hint: Intercepted intelligence indicates that receivers were asked to set the "decoder ring to b=12".

Task 2:

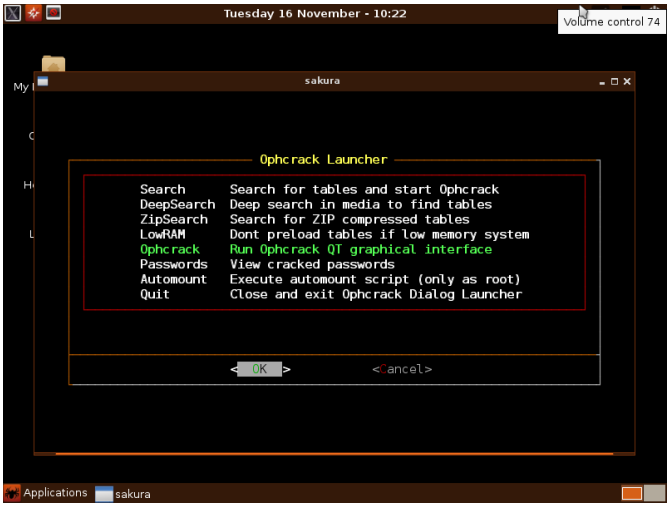
In this task, you are asked to crack the passwords of a virtual Windows XP installation. Follow the following steps:

1. Download and install VirtualBox: https://www.virtualbox.org/
2. Find the `.ovm` file supplied to you with this case study.
3. Import the appliance by double-clicking the `.ovm` file.
4. You should see a new virtual machine named "WinXP":

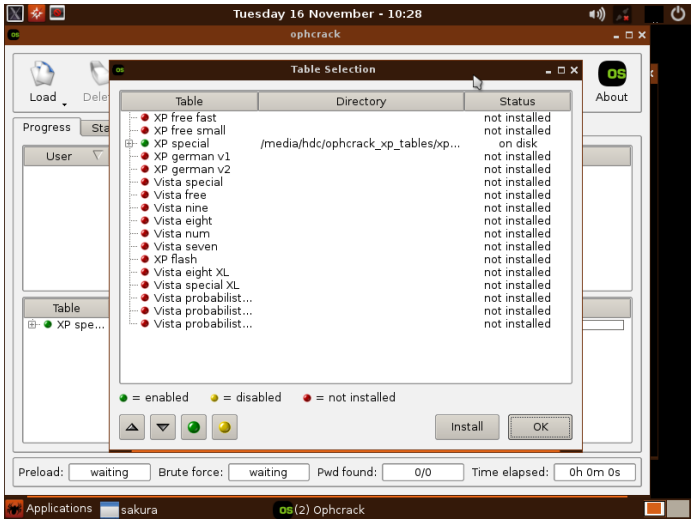


5. Download the ophcrack XP live CD: https://ophcrack.sourceforge.io/download.php
   Be sure to pick the right version of ophcrack.
6. Optional: download all Rainbow Tables for Windows XP, particularly the XP special tables: https://ophcrack.sourceforge.io/tables.php
7. In your VirtualBox WinXP appliance, select “Settings” and add the live CD to a virtual IDE slot. Also, make sure the boot order is such “optical” comes before “Hard Disk”. The above image shows the appropriate settings.
8. Start the virtual machine. Rather than Windows XP, ophcrack should load. Select “manual”:

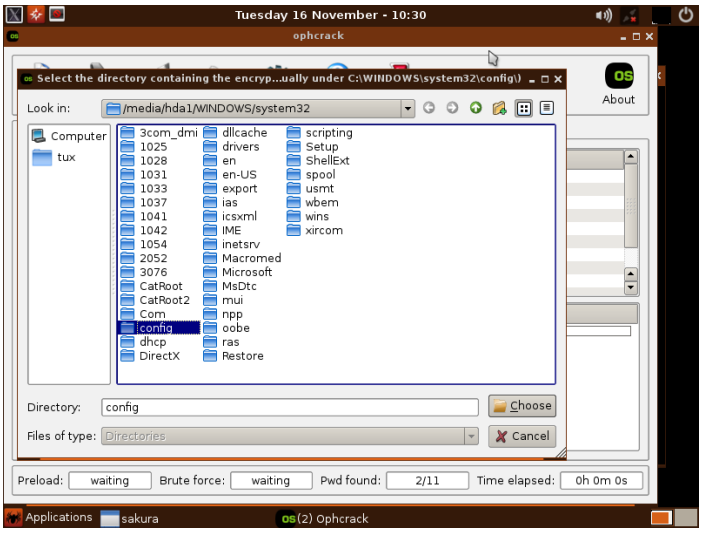
9. As ophcrack boots, select your language and keyboard locale, and wait for the system to finish booting. Ignore the warning that no tables are found, and press ENTER to continue. On the next screen, select “Run Ophcrack QT graphical interface.”



10. In the next screen, select “Tables” and “Install”. Navigate to the live CD folder where the rainbow tables are stored (usually inside `/media/cdrom/`). You may also point the installation to a USB drive onto which you copied the XP special rainbow tables you downloaded in Step 6, as shown in the next image. Click OK.



11. Back on the main screen, select “Load” and “Encrypted SAM”. Point the file picker to the following folder in the Windows installation supplied to you and click “choose”: `/media/hda1/Windows/system32/config`



12. Select “Crack” and wait to answer the following questions:
    - Which usernames are available on the Windows installation?
    - What are their passwords?

    Note, that some passwords might not be able to be decrypted with the rainbow tables you selected. This is normal. In this case, ophcrack will say “not found”

Task 3:

Find the file `secret.png`[1] supplemented together with this case study. The image contains a secret message hidden within. Decrypt the message.

Hint: The message contains only alphanumeric characters and regular punctuation.

Instructor notes

This case study may be assigned as a formative homework assignment, whole or in part, for individual students or small student teams. It is also possible to use this case study as an in-class exercise. Assessment is at the discretion of the instructor. Note that for Task 2, albeit Windows XP has reached end of life and Microsoft no longer provides support, the authors of this case study cannot supply a copy. It is recommended to obtain an installation medium of Windows XP and prepare several user accounts, such as those shown below in the example solution for Task 2.

[1] Image credit: © 2020, SUNY Oswego. Used with permission.

Example solution

Task 1:

The message can be decrypted using five methods:

1. Brute-force letter frequency analysis. One would start with shorter words and heuristically substitute letters. For example, "`nq`" could be `an`, `be`, `is`, `or`, `to`, … and "`egdq`" could be `drink`, `four`, `have`, `this`, `sure`, … . Since both "`nq`" and "`egdq`" end in "`q`", it's likely that the first words are "`be sure`". Therefore, the letter "`q`" corresponds to "`e`" and can be replaced in other words of the message.
2. The specific cypher used is a "shift cypher". Knowing that b=12 means each letter is shifted by 12 letters.
3. Using an online shift cypher decoder, e.g., https://www.xarg.org/tools/caesar-cipher/
4. Using a home-made decoder ring: https://dabblesandbabbles.com/printable-secret-decoder-wheel/
5. With sufficient knowledge of pop culture.

The decrypted message reads:

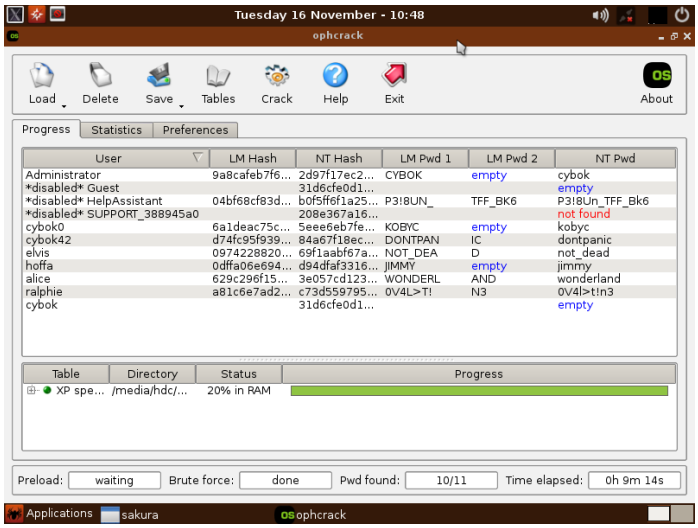
```
be sure to drink your ovaltine
```

Task 2:

username	**password**
Administrator	cybok
disabled Guest	(empty)
***disabled* HelpAssistant**	**P3!8Un_TFF_Bk6**
disabled SUPPORT_388945a0	(can't be found with standard tables)
cybok0	kobyc
cybok42	dontpanic
elvis	**not_dead**
hoffa	jimmy
alice	wonderland
ralphie	**0V4l>t!n3**
cybok	(empty)

Bold printed rows require XP tables with special characters, which are also available on the website given above, however not part of the rainbow table collection included in the live CD.

An example of the solution the students should see is:



Task 3:

The message hidden within the image is an example of a steganography cypher. The message can be extracted from the image using the Linux tool `steghide` or an online steganography decoder, such as this one: https://stylesuxx.github.io/steganography/

The resulting message looks like this:

```
----[----->+<]>++.-[-->+<]>.+[->+++<]>.-[--->+<]>----.+++.--
------..--[--->+<]>.-[---->+<]>++.+[->+++<]>.++++++++++.-
-..--------.++++++++++++.++++[->+++<]>+.++++++.--------
.++++++++++.--[->+++<]>+.
```

This is obviously not alphanumeric text, but a program in the language brainf*ck. It can be converted into ordinary text using an online interpreter of this message, such as: https://copy.sh/brainfuck/

The output of the program is the hidden message:

```
A crummy commercial?
```