POLITÉCNICO DE LISBOA

ISEL - SRC – 2023/24

Practical exercises – Computer Network Security

Note: Remember that you should only test this kind of practical exercises in YOUR network: Home network or using machines connected in a VirtualBox/Docker containers. You shouldn't do the experiences in your work or school's networks.

Groups from 1 to 3 students; Delivery date: See Moodle.

Introduction

The objective of these practical exercises is to consolidate some of the concepts about network security taught in the theorical classes. It also aims to stimulate students' curiosity on a very important topic today, a theme that includes so many areas of computer engineering, computer networks, and where it is not very frequent to have someone who can dominate them all. The idea is to allow the student to choose exercises in which he/she feels more comfortable and in which he also can have some enjoyment in the execution of the practical exercises.

Note: You do not need to read all this document, you can start to choose from the index (Contents).

Legal advice

All exercises with attacks should be executed on local virtual machines whenever possible. **You must not test any security concepts where that can affect other users, namely in a production network, including in the ISEL/IPL's network. You can test a few experiments at home, if you have a domestic network, but do not let the tests go out of your local network! You may have serious legal problems!**

You should not do this kind of work connected to an enterprise's network and their computers, especially if you are connected through a VPN of your enterprise. If you do it, it is possible that many links will be blocked by the Content Filtering of the web proxies.

POLITÉCNICO DE LISBOA

Contents

Methodology

- Each working group can have from one to three students.
- Every exercise has a number between brackets [n] after its name to give an idea of its complexity.
- **Each group of students must do at least the practical exercises so that the sum of their complexities be equal or greater than 40 (forty).** If a group of students does more exercises than necessary to achieve this value, only the first **40** in the final report will count for the final practical grade of the course unit (SRC).
- Each group must do, **at least, one exercise from each topic group**. The groups are numbered.
- During the semester the students will receive several exercises to do at the practical classes or/and at home.
- Some of these exercises are to be done in practical classes. The professor will choose which ones.

Each student in a group must be able to demonstrate all the practical exercises done by the group during the oral presentation.

The professor may use some of the exercises in this document in practical classes. The exercises done in the practical classes also count to the total value necessary to be approved in the practical part of the course.

Each group must send an email to the professor with information about which practical exercises have decided to do. This email must be sent by the end of the third week after the date of the publication of this document. **Note:** If the exercises chosen by the group and the ones that are done in the practical class aren't the same the group can change the ones it chose, in the same group of exercises (topic), by the one done in the practical classes.

Each practical exercise done in a practical class must have its own individual report. This report must be delivered two weeks after the practical class. All the individual reports of each practical exercise done must be aggregated in a final report with all the practical exercises done by the group (classes and extra classes). All the reports must be delivered to the professor through Moodle, only.

Note: Consult Moodle to confirm the dates for the delivery of the reports.

Prerequisites

"A laboratory is as vital to a computer-security specialist as it is to a chemist or biologist. It is the studio in which you can control many variables that come to bear upon the outcome of your experiments. And network security, especially, is a field in which **the researcher must understand how a diverse range of technologies behave at many levels**. For a moment, just consider the importance of the production network to most organizations. They must rely on an always-on functioning, which means that many tests and evaluations must be developed in a lab on a network that has been specifically designed for such experiments.
…
Building a lab requires you to become familiar with the basics of wiring, signal distribution, switching, and routing. You also need to understand how you might tap into a data stream to analyze or, potentially, attack the network. **The mix of common network protocols must be understood; only by knowing what is normal on the network can you recognize and isolate strange behavior.**" – **Michael Gregg**, "Network Security Test Lab – A Step-by-Step Guide", Wiley.

In addition to the study of the theory that is necessary to have success in each of the topics related to the course unit of SRC, the implementation of the practical exercises implies the need to install in your computer some specific software tools (which ones depends on the exercises you choose to do). For many of the practical exercises, the lab that is needed can be based on the SEED labs (more information below on Software installation):

- New SEED info: https://seedsecuritylabs.org/labsetup.html
- Old SEED info: https://seedsecuritylabs.org/lab_env.html

- SEED slides about the labs: https://www.handsonsecurity.net/resources.html

Note: Run the software tools in virtual machines, containers, or cloud whenever possible.

Example of some tools you may need to solve the exercises (you don't need to install now all of them, install only the ones you need to do each of the practical exercises you chose):

- CyberChef [https://gchq.github.io/CyberChef/]
- VirtualBox [https://www.virtualbox.org/] (VMware Player for some experiments, if you prefer, but most of the exercises were tested with VirtualBox and/or Docker)
- Containers [https://www.docker.com/] (If you decided to use containers)
- Kali Linux [https://www.kali.org/] (lots of tools, some included some that you can download)
- Wireshark [https://www.wireshark.org/]
- Nmap [https://nmap.org/]
- OpenSSL [https://www.openssl.org/]
- GNS3 with images of Cisco *routers* c7200 and c3750 [https://www.gns3.com/]
- …

You can explore other software tools. You only need to justify in your report why you chose them.

You can explore the tools available in the net, some free, some with a trial period, some paid, but it isn't pretended only a type of approach of the kind "script kiddie", using only tools done by others without, at least, know how they work, and the protocols involved.

POLITÉCNICO DE LISBOA

Lab Software installation

Many of the proposed exercises are based in the "SEEDLabs - Hands-on Labs for Security Education" site (https://seedsecuritylabs.org/). But there are exercises that are done with other tools as the Packet Tracer or the GNS3 network simulator. In the VirtualBox it is possible to install more machines than the ones referred to in the SEED labs, its needs depend on the exercises you choose to do. Some of those extra machines are Kali Linux and Metasploitable.

Some software, installation and configuration instructions can be obtained from:

- SEEDlabs2.0: https://seedsecuritylabs.org/labsetup.html
  There are several possibilities for the installation. You can use the cloud if you have an account but if you haven't then it is much cheaper to use a virtual machine on your computer, namely VirtualBox (Ubuntu 20.04 VM): https://seedsecuritylabs.org/Labs_20.04/
  Manual for the installation using VirtualBox: https://github.com/seed-labs/seed-labs/blob/master/manuals/vm/seedvm-manual.md
- If needed, the former version is SEEDUbuntu16.04: https://seedsecuritylabs.org/Labs_16.04/
- You need to have 3 machines configured in the VirtualBox for some of the labs. We can call them Alice, Bob and Cain (you can use other names, for example: User, Server, Hacker). Activate only the machines you need to the lab you are doing or else your computer can become very slow, or worse. This depends on the hardware characteristics your computer has.
- **For the creation of Alice, Bob and Cain you must install one VM following the instruction above and then clone the other two machines.** Some more instructions for this in the VM Manuals for Ubuntu 20, 16 and 12) (https://seedsecuritylabs.org/labsetup.html).
- It is useful to install a machine with Kali in the VirtualBox so you can also do some tests with Kali in the VirtualBox Network (VM NATnetwork) where all the VMs are connected. For Kali:
  - https://www.kali.org/
  - https://www.kali.org/docs/introduction/
  - Install VirtualBox with at least 3 virtual machines (cloned) (see SEED Labs) and one with Kali Linux. Use NATnetwork to connect them in VirtualBox.
    Note: Problems changing the Kali keyboard layout?
    https://mayadevbe.me/posts/linux_keyboard_layout/
  - For the more curious there are some free courses about Kali: https://kali.training/

More information about the practical works based in the SEED labs (https://www.handsonsecurity.net/resources.html):

- https://seedsecuritylabs.org/Labs_20.04/Networking/
- https://seedsecuritylabs.org/Labs_20.04/Crypto/
- https://seedsecuritylabs.org/Labs_20.04/Web/

[***This possibility was removed recently.*** *The only thing remaining is: https://www.handsonsecurity.net/*] There is a book that is part of the bibliography of the curricular unit of SRC ("**Internet Security: A Hands-on Approach, Second Edition**", **Wenliang Du**), but the students can also obtain an old SEED lab manual ("SEED: A Suite of Instructional Laboratories for Computer - **SE**curity **ED**ucation", Wenliang (Kevin) Du)) from: http://www.cis.syr.edu/~wedu/seed/Labs/SEED_Book_1_2011.pdf.

Other site that has some interesting information is: https://github.com/Samsar4/Ethical-Hacking-Labs. It is a good place to start being a "*script kid*"!

Many other practical exercises are possible, for example about Firewall with Iptables, etc, but their complexity can be higher. The former learning experience of some students, that are from several different courses, with the Linux

programming (Python and C), virtual machines, containers, … as well with Windows and Linux OS somehow limits the deepness of the type of practical exercises proposed here.

Note: Can be proposed to the professor other practical exercises (from SEED labs or others), but it is always decision of the professor to accept or not accept them.

For more ideas for practical exercises, you can consult several sites as:

- https://seedsecuritylabs.org/
- https://www.kali.org/
- https://cyberlab.pacific.edu/courses/comp178/labs
- https://www.cryptool.org/en/
- https://github.com/Samsar4/Ethical-Hacking-Labs
- …

but you need to remember that this course unit (SRC) is about **network security** and **not** applications or operating systems security, so the practical exercises done must be related, above all, with **network security**.

Bibliography

- "Internet Security, A Hands-on Approach, Second Edition", Wenliang Du (this is the name of the one book only version but there is a two separated books version, the computer one and the Internet one).
- https://github.com/Samsar4/Ethical-Hacking-Labs [**Note:** Lots of information to who likes cybersecurity]
- "The Network Security Test Lab: A Step-by-Step Guide, 1st Edition", Michael Gregg, Wiley
- "Learn Kali Linux 2019", Glen D. Singh, Packt
- "SEED: A Suite of Instructional Laboratories for Computer SEcurity Education", Wenliang (Kevin) Du [http://www.cis.syr.edu/~wedu/seed/Labs/SEED_Book_1_2011.pdf] [**Note:** PDF version, old and limited, of the new books by Wenliang Du]
- See also the Internet links in the text.

POLITÉCNICO DE LISBOA

Practical exercises

1. Knowledge that should have been acquired in previous cybersecurity disciplines.

1.1 Regular expressions

What this regular expression does? [1]

(?:(?:\d|[01]?\d\d|2[0-4]\d|25[0-5])\.){3}(?:25[0-5]| 2[0-4]\d|[01]?\d\d|\d)(?:\/\d{1,2})?

Any suggestions to improve the previous regular expression?

[https://regexr.com/] [https://regex101.com/] [https://en.wikipedia.org/wiki/Regular_expression]

Note: You may get some help from ChatGPT, or others, but you shouldn't only copy and paste the answer. You must understand it very well before. Remember that what ChatGPT doesn't know it invents.

1.2 Decode

The following coded message was detected in transit: [1]

A. Using the CyberChef application, decode the message indicating what it is.

   01010100 01110101 01110010 01101101 01100001 00100000 01100100 01100101 00100000 01010011 01010010 01000011 00100000 01100100 01100101 00100000 00110010 00110011 00101111 00110010 00110100

B. What are the differences between ASCII, UTF8, UTF16 and Unicode?

1.3 Decode

Using the CyberChef application: [1]

A. The following coded message was detected in transit:

   i) TmFkYSBkZSBjb3BpYXIgZGEgTmV0IG91IGRlIHVzYXIgbyBjaGF0R1BUIQ==

      Using the CyberChef application, decode the message indicating what it is.

   ii) What type of encoding is used in this message? What is this encoding used for?

   iii) Do the encoding of the names of the members of your group with the same code.

B. The following coded message was detected in transit:

   wTBQfebwJvCy9Txw4o3NG3FZgPD3i73zpRTynyCBLC96NuQxhbF2K

   i) Using the CyberChef application, decode the message indicating what it is.

   ii) What type of encoding is used in this message?

C. What are the main differences between the encodings used in the previous two points?

D. What are these encodings used for?

E. Can it be said that the text in the previous points are encrypted?

1.4 Coding

The following coded message was detected in transit: [1]

A. Using the CyberChef application, decode the message indicating what it is (Note: It is a readable string).

37 55 71 6e 78 43 4a 4d 54 7a 4a 66 68 47 6d 4d 62 37 44 32 50 75 55 50 61 33 71 77 33 31 34 6d 7a 70 38 6a 71 41 56 38 6d 4a 44 76 5a 76 64 6a 33 62 47 66 53 79 34 68 4c 59 68 6e 69 79 43 75 5a 57 6d 6e 72 39 78 63 44 6d 55 53 53 75 52 52 47 67 76 6d 57 47 46 67 79

B. What type of encoding is used in this message?

1.5 Decode [1]

The following coded message was detected in transit:

U1JDKE8gc2VncmVkbyBkaXN0byDDqSBuw6NvIHNlIGRlc2lzdGlyIGZhY2lsbWVudGUuIFNlciBwZXJzaXN0ZW50ZS4pCg==

Additional information made it possible to understand, by default, that the decoded message has the format SRC(...). Note: The result hasn't sintax errors in portuguese.

A. Using the CyberChef application, decipher it, knowing that it starts with SRC(.....).

B. What kind of transformation has the message undergone?

1.6 Caeser Cipher [1]

The Caesar Cipher, also known as the shift cipher, is one of the oldest and simplest ways of encrypting a message. Here the shift can be different from 3. In addition to the class slides, you can find more information at: https://privacycanada.net/classical-encryption/caesar-cipher/. Two messages were intercepted.

NMX(Dnoj nzmqz kvmv oznoz yz phv xdamv hpdoj xjiczxdyv.)

POZ(Jxfp al jbpjl. X afcfzriaxab fox pryfkal klp bubozfzflp moxqfzlp.)

Use the tool https://www.dcode.fr/.

A. Number of characters in the rotational (number of times rotated).

B. Type of alphabet used (coding).

C. Discover the original message.

D. Change the alphabet to ASCII Table. What is the result of encrypting both? Do the reverse process. Is it possible to reverse the cipher? Justify.

1.7 Using the CyberChef application [1]

In the Input part, write: "SRC(Vamos ver os hashes.)".

A. Calculate the MD5 hash.

B. Change the message to SRC(Vamos ver os hashs.). Calculate the MD5 hash. Compare with the previous one and justify.

C. Go to https://www.dcode.fr/md5-hash. Redo the hash calculation (MD5 encode) and check if the results are different. Do both sites use different algorithms? Justify

D. Try now, still on dcode.fr, reversing the hashes using MD5 decrypt. Check if you can reverse the hash and recover the original message. Justify.

E. Try decryption at https://www.md5online.org/. What is the result? Considering the result of the previous paragraph, can it be said that hashes guarantee confidentiality, integrity and authenticity of the message?

F. Try decrypting the hash of the previous paragraph on the following website, which uses dictionary attacks: https://crackstation.net/. Can you decrypt? Justify.

G. Try it with the following hash: 662af1cd1976f09a9f8cecc868ccc0a2. Did you manage to decrypt? What is the original message? Why were you able to decrypt this via Crackstation and the previous one not? Try for example SRC2223, calculate the hash and check if it is possible to reverse it.

H. Find another possible hash and password, without downloading the wordlist, that can eventually be reversible. Justify.

I. What is the reversibility of the previous paragraphs?

J. Is it true that there is a MD5 collision between these two strings:

```
d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325 71415a085125e8f7cdc99fd91dbdf280373c5b
d8823e3156348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70
```

and

```
d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b6708280d1ec69821bcb6a8839396f965ab6ff72a70
```

K. What is the hash (message digest) of each of the strings above?

Reference: https://www.mscs.dal.ca/~selinger/md5collision/

1.8 Using the CyberChef application. Encryption/decryption [2]

In the Input part, write: “Learn Computer Network Security doing practical exercises.” (without the quotation marks).

A. Clear the Recipe.

B. Encrypt with DES, CBC Mode, using:

   i) Key = ffffffffffffffff (8 bytes)

   ii) IV = 00ff00ff00ff00ff (8 byte)

C. Take note of the result.

D. Do Input and Output have the same size in number of bytes? Justify.

E. Encrypting using the same DES but using other modes is the result the same?

F. By encrypting in one mode (Mode) and decrypting with another, can the message be decrypted? Justify.

G. Clear the Recipe. Place the result from the previous paragraph as Input. Check if you can decrypt with DES using the same data as in the previous section, simulating that you had access to the Key and the IV. Justify.

H. Change just 1 bit of the Key. Can you recover the original message?

I. Change 1byte or 4 bits of the IV. Can you recover the original message?

J. Is the protocol resistant to key tampering or IV changes?

K. Change the entire IV. Can you recover the original message? Finish.

1.9 Using the CyberChef application. Encryption/decryption [2]

In the Input part, write: "The Computer Network Security subject is just one of several subjects taught in the area of cybersecurity at DEETC." (without the quotation marks).

A. Clear the Recipe.

B. B. Encrypt with 3DES (Triple DES Encrypt), CBC Mode, using:

   i) 1. Key = ffffffffffffffff ffffffffffffffff ffffffffffffffff (24 bytes, no spaces)

   ii) 2. IV = 00ff00ff00ff00ff (8 byte)

C. Take note of the result.

D. Compare with the previous exercise. Is the encryption result the same? Justify.

E. Clear the Recipe. Place the result from the previous paragraph as Input. Check if you can decrypt with 3DES (Triple DES Decrypt) using the same data as in the previous section, simulating that you had access to the Key and IV. Justify

F. Change just 1 bit of the Key. Can you recover the original message?

G. Change 1byte or 4 bits of the IV. Can you recover the original message?

H. Then compare DES and 3DES CBC in terms of robustness. Equals? Is 3DES better? What are the main differences between the two? What justifies the use of Triple DES as an alternative to DES?

1.10 Linux passords [2]

A Linux server was compromised, and the following hashes were detected in /etc/passwd (Note: In these experiments, do not use YOUR real passwords) [you can use the previous tools, for example CyberChef and/or Crackstation]:

root: f4244e539b46496d146fa1159a2a188aeed7295f

admin: addbd3aa5619f2932733104eb8ceef08f6fd2693

Joe: 906b8e437c1fd25c70efd94e78f5e23b

Security: aa1722b7818cf3a9ced224805f3ee5fd

A. What is the structure of passwords in Linux? What hashes are currently most used to protect passwords in Linux? [https://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils]

B. Why do /etc/passwd and /etc/shadow files exist for Linux to handle passwords?

C. Indicate each user's original password and hashing method.

D. Do the hashes above already include salt? Would these be the hashes that Linux would put in /etc/shadow?

E. What difference does the number of rounds make in calculating SHA1?

F. Make a 1-character change to each password and check whether it remains reversible. Present password and hash.

G. Do you identify any bad security practices?

1.11 Steganography [2]

It is intended to transmit secret information to a person. However, this information has to be delivered by someone who cannot know that this secret message exists, although they have access to the file where it is. So, you've been reading about Steganography and using this method to convey the message hidden within an image. Like this:

A. Create a small text file on your Kali machine with the message to be transmitted inside.

B. Download the following image (Note: You can use any other image):

   https://blog.hubspot.com/hs-fs/hubfs/Sales_Blog/1-min.jpg?width=1196&name=1-min.jpg

   Note: If the file type isn´t the expected one (file 1-min.jpg) find a way to convert it.

C. Using the steghide application that you may need to install (sudo apt install steghide) insert the file created in section a) hidden inside the image in section b.

   steghide embed -ef <file_to_be_embedded> -cf <image_file_where_embedded>

D. Once inserted, check that you can open the file normally and that you find the image unchanged. The file is now ready to send.

E. Imagine that you received the file. The first thing you will do is analyze the image file. Check that it opens the image correctly.

F. Use the EXIFTOOL tool (sudo apt install libimage-exiftool-perl -y) to analyze the image and see if you can find signs that there is a component beyond the image.

G. Use the binwalk tool (sudo apt install binwalk -y) and check if there is a text file in addition to the jpg file.

H. If you find it, use the command to remove the text file from within the image file

   steghide extract -sf <ficheiro_de_imagem_onde_embeber>.

I. Repeat the entire process for a .zip file (compress the text file before placing it on the image).

J. With your groupmate, each choose your own .JPG image and insert messages and files crossed between them, via steganography.

K. Demonstrate all steps in the report.

Reference: https://infosecwriteups.com/beginners-ctf-guide-finding-hidden-data-in-images-e3be9e34ae0d