TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

institute of telecommunications

Exercise 1: Live Capturing

Network Security - Advanced Topics (VU 389.160), Winter Semester 2018/2019

Communication Networks Group at the Institute of Telecommunications

You are working in Vienna at Austrians largest IT security company, and – right now – auditing network security in the data center of the Ministry of Cyber Affairs. Suddenly, two members of the security staff show up at your office and ask you to come with them immediately.

While walking along the corridors of the Ministry, the security personal informs you that a suspicious individual has been detected and his notebook (Figure 1) confiscated. The notebook is connected to the Ministry's internal network. The Ministry suspects that a severe information leak is taking place and their network operators assume the source of the leak to be right in the middle of their network test laboratory. The confiscated notebook might be the receiver of the stolen data.

Figure 1. Suspicious notebook

Knowing the characteristics of the Ministry's (extremely strict) security measures, you have the feeling that a covert channel is being used to transmit the data (otherwise it would probably have been detected by payload checkers of their intrusion detection system). Once you arrive at the room where the confiscated notebook is located, the ministry staff urges you to search for the data leakage, discover the source, and recover the stolen data if possible.

The security personal just handed the credentials over to you and, thus, you can login to the notebook to start looking for any covert communication ...

1 Capturing traffic

(Step 1) Open Wireshark and get ready to capture the traffic. In order to do that click the "List the available capture interfaces..." button. In the newly opened panel select the local network interface ("em1"). "Start" allows you to obtain data arriving to your machine. A few minutes of captured traffic should be sufficient to detect the covert channel (if one does actually exist).

1.1 Rearranging traffic features

In the Wireshark environment you should see a large amount of captured traffic traces. As you know, covert channels can be masked in numerous fields and attributes. For this exercise you can reduce the possibilities to some specific traffic features that belong to the IP, TCP, and UDP headers and are commonly checked for covert channels.

(Step 2) The first thing that you should do is to rearrange the traffic in a way that you get the following distinctive information for every captured packet:

No., Time, SrcIP, DstIP, Protocol, IP Length, IP Identification, TTL, SrcPort, DstPort, TCP flags, IP version, IP flags, DSCP and Fragment Offset.

We recommend adjusting Whireshark parameters to display the proposed features as the following "Field Types" (respectively): Number, Time, Source address, Destination address, Custom (ip.proto), Packet length (bytes), Custom (ip.id), Custom (ip.ttl), Src port (unresolved), Dest port (unresolved), Custom (tcp.flags), Custom (ip.version), Custom (ip.flags), DSCP Value, Custom (ip.frag_offset).

Important: Make sure that you remove the "Information" column before proceeding! Otherwise, some analysis tools may get confused by the field's content[1].

Rep:1.a

What is the IP address of the suspicious notebook (i.e., your machine)?

[1] Feel free to leave the field in but do remember when you run into troubles with some analysis tool!

2 Analysis

2.1 Filtering irrelevant data

An essential part of discovering covert channels consists of filtering non-pertinent information. In this way, you can reduce the scope and focus on suspicious data. Do that progressively by iterative steps of filtering and analysis.

(Step 3) After answering *Rep:1.a* and *Rep:1.b*, you should be able to apply filters and reduce the total amount of data to only relevant traffic (i.e. data from the machine leaking information to the suspicious notebook). You can do that either in Wireshark or later in Rapidminer (read the next subsection before carrying out *Step 3*).

Figure 2 shows all possible combinations of different covert sender and covert receiver locations. During the course of this lab, you can safely assume that covert sender and covert receiver are located at the overt sender and overt receiver, respectively (i.e., case 1 in Fig. 2).

Important: This implies that neither source nor destination addressees are spoofed!

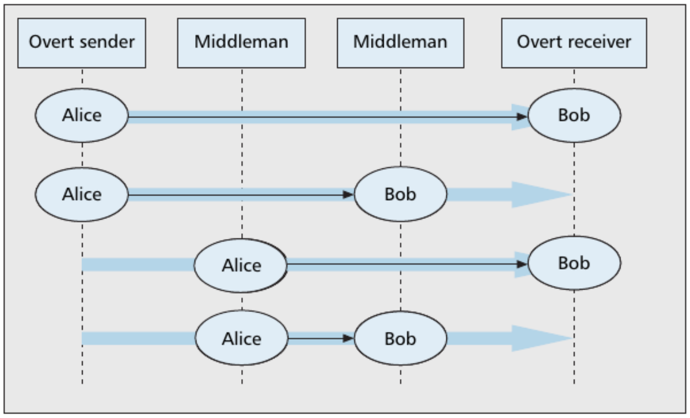

Figure 2. Possible combinations of different covert sender/receiver locations [?]

Important: You may assume that the network's gateway is not involved in covert communication.

> **Rep:1.b**
> What is the IP address of the machine presumably leaking information?

2.2 Exporting and importing CSV files

No matter how you filter irrelevant data, you will need to use specialized tools to perform some analysis (for which Wireshark may not be the best tool). Data can be exported from the Wireshark environment into Comma Separated Values (CSV) files, a general, widely-used text format intended for easy data storage and exchange. **(Step 4)** To do this, in Wireshark, go to "File>Export>as CSV...", mark "Displayed" if you applied any filter, and save the data as a file.

Note that Wireshark outputs some values in hexadecimal form, which can result in problems for the subsequent analysis tools. To avoid running into such problems a preprocessing step is required. **(Step 5)** In the "*workfiles*" folder you will find a small script named "only_decimal.sh". Use it to transform hexadecimal values in your CSV file into decimal and generate a new CSV file.

```
% ./workfiles/only_decimal.sh in.csv > out.csv
```

(Step 6) After converting the captured traffic to the desired CSV format, open Rapidminer and import the new CSV file. Keep the header generated by Wireshark to identify the columns as the corresponding features. Feature types should be as follows: No. (numerical), Time (numerical), SrcIP (nominal), DstIP (nominal), Protocol (nominal), Length (numerical), IP Identification (numerical), TTL (numerical), SrcPort (nominal), DstPort (nominal), TCP flags (nominal), IP version (nominal), IP flags (nominal), DSCP (nominal), Fragment Offset (numerical). Remember that defining features as *numerical* or *nominal* affects how analysis tools deal with the given variables. Spend some seconds checking that you understand the proposed classification in *numerical* or *nominal* types. If your are not completely certain about the difference, take a short look at the corresponding Wikipedia page. [2]

> **Rep:1.c**
> Give a detailed (but brief) explanation of the steps you carried out to filter irrelevant data (either Wireshark or Rapidminer). Do also specify the keywords and operators required.

2.3 Univariate analysis

You are now ready to perform the first quick analysis with Rapidminer. **(Step 7)** Go to the Results panel (F9) and have a look on the statistics (Meta Data View) and the Histograms (Plot View>Histograms). Check all the different features.

> **Rep:1.d**
> Which features are not viable to mask a covert channel and could be removed from the analysis? List the rejected features and provide short but meaningful reasons for rejection.

2.4 Time series and bivariate analysis

Data can be explored in depth by checking the time evolution of the feature usage and also crossing features each other. **(Step 8)** In the Results panel of Rapidminer (F9), have a look on the Scatter Plot (Plot View>Scatter). Select "Time" or "No." for the "x-Axis" and for the "y-Axis" check the remaining features (the ones that you do not have discarded yet). In addition, you can also use the "Color Column" to cross a third feature.

[2] https://en.wikipedia.org/wiki/Level_of_measurement

Rep:1.e

From the remaining features, which ones are not viable to mask a covert channel and could be removed from the analysis? List the newly rejected features and provide short but meaningful reasons for rejection.

Rep:1.f

Do you think that you have found the covert channel? Give a detailed description of where the covert channel is occurring (`feature_value:covert_symbol` relationship) and provide a capture of the plot where the abnormal behavior of the suspicious feature is isolated and clearly visible.

3 Decoding the hidden channel

Detecting a (presumably) covert communication is one thing, being able to decode the encoded communication is another. Without any clue, to elucidate the codification can be a extremely demanding task. Fortunately, the security staff has confiscated a piece of paper in one of the notebook owner's pockets (Figure 3).

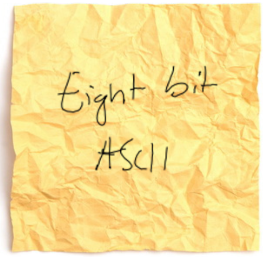

Figure 3. Confiscated note

Now you are quite sure that the attackers have probably never heard about Kerckhoffs' principle. Will you be able to discover the hidden message?

Important: While decoding, make sure to know the exact size of the data (not all fields are either single bits or entire bytes)!

(Step 9) Come back to Wireshark, apply filters and manage the workspaces in a way that you isolate the feature containing the covert channel. Again, export it as CSV file. Alternatively, you can also use RapidMiner for this task.

Rep:1.g

Write in the report the way you filtered the data (e.g., the formula) and the steps carried out to prepare the required file.

(Step 10) Now it is time to face the decoding task. With the gathered information and the note found in the notebook owner's pocket, it should be possible to figure out what information was being stolen. In order to do that you can use any tool you find suitable, such as MATLAB, Octave, LibreOffice Calc, any programming or scripting language, (or maybe even manually), ..

Rep:1.h

Write in the report the decoded message. Explain clearly how you carried out the decoding task (step by step in a numbered list).

Rep:1.i

Report briefly any additional comment or observation related to the exercise solving to be considered during the review of your exercise.

(Step 11) Important: Call a tutor when you finish this point before continuing with the next exercise.

References

[1] S. Zander, G. J. Armitage, and P. Branch. A survey of covert channels and countermeasures in computer network protocols. 9(1-4):44–57.