

A Network Steganography Lab on Detecting TCP/IP Covert Channels

Tanja Zseby, *Member, IEEE*, Félix Iglesias Vázquez, *Member, IEEE*,
Valentin Bernhardt, Davor Frkat and Robert Annessi,

Abstract—This paper presents a network security laboratory to teach data analysis for detecting TCP/IP covert channels. The laboratory is mainly designed for students of electrical engineering but is open to students of other technical disciplines with similar background. Covert channels provide a method for leaking data from protected systems, which is a major concern for big enterprises and governments. The inclusion of covert channels in the curricula of network security students and network data analysts is therefore considered a valuable extension. In the lab exercises presented, students learn how covert channels in TCP/IP network traffic can be hidden and detected. Since the detection of covert channels requires an in-depth understanding of protocol standards and typical behavior of TCP/IP flows, the lab also provides a “playground” in which students can deepen their communication networks knowledge. Students learn how to use and interpret statistical analysis to discover abnormal patterns and footprints in network data. They are also trained to deal with noisy scenarios which increase ambiguity and uncertainty. The laboratory was first implemented during the winter semester 2014 with a class of 18 students at TU Wien, Austria. This experience showed that students consolidated the targeted skills as well as increased their interest in the topics explored. All exercises and datasets for the introduced “Network Security Advanced” lab are made publicly available.

Index Terms—Communication system security, data analysis, engineering education, security

I. INTRODUCTION

COVERT channels make use of communication networks in ways not intended in the original design of the communication protocols. Covert channels utilize control fields (e.g., TCP/IP headers) or manipulate time-related properties (*covert timing channels*, Fig. 1) to hide information. The transmitted message is disguised as control data or random communication peculiarities, and travels undetected in the network, other than by the sender and the receiver of the covert communication. Hence, the goal of a covert channel is not only to prevent a hidden message being read by third parties, but also to conceal evidence that such communication has taken place at all. Although covert channels can be used for ethically acceptable applications (e.g., to bypass censorship in non-democratic regimes), they are particularly suited for criminal activities, such as illegal transfer of sensitive information, data leakage and data theft, or hidden malware command and control structures. They are hard to detect with standard attack detection methods and should be included in today’s network security education.

T. Zseby, F. Iglesias, V. Bernhardt, D. Frkat, and R. Annessi are with the Institute of Telecommunications, TU Wien, Gusshausstrasse 25/E389, 1040 Vienna, Austria. T. Zseby and F. Iglesias contributed equally to this work.
Manuscript received xxx.



Fig. 1. Example of a covert timing channel. Clandestine information is conveyed by using the delay between consecutive packets.

The lab introduced here is intended as an advanced hands-on exercise for students already familiar with concepts and methodologies related to network security and data analysis techniques. It is designed as an advanced lab that students could take having completed the lab presented in [1]. The educational goals can be summarized as follows: a) to teach students how covert channels in communication networks are created and how they can be detected; b) to increase students’ data analysis skills using statistical methods; c) to deepen students’ knowledge about the TCP/IP protocol and properties of normal TCP/IP traffic; and d) to train students’ skills to solve complex problems that have no clear path to a solution, by employing conjectural reasoning, drawing upon diverse analysis, testing multiple hypotheses and working patiently until a solution is reached.

Network security classes are often supplemented by practical exercises that help students to comprehend and deepen theoretical knowledge while acquiring the skills required for scientific work in that area. Several authors have published guidelines for, and their practical experience with, installing network security labs, e.g., [1]–[8]. Comprehensive overviews on covert channels are offered in [9] and [10], with TCP/IP being specifically treated in [11]. Methods to conceal information in different types of signals are summarized in [12]. In [13] a practical exercise investigating bouncing covert channels is proposed for research and educational purposes.

This paper presents a lab with several exercises for teaching covert channel detection methods. The TU Wien, Austria, first implemented the laboratory in an advanced Network Security course (NetSec Advanced) during the winter semester of 2014, with a class of 18 students. The class consists of a theory part with lectures and a lab part. All exercises and datasets for the lab are publicly available in order to encourage adoption of the introduced class by other instructors of network security in electrical engineering and computer science [14].

II. EDUCATIONAL AIMS

The educational aims pursued by the Network Security Advanced Lab are that students should:

- 1) *Learn about covert channel hiding/detection methods.*
In the theory lectures students learn the basics of network steganography. They learn how information can be hidden in TCP/IP header fields or timing, and which fields are better suited for hiding information. They also learn how digital signatures can be used to establish subliminal channels. In the lab students apply their knowledge to detect covert channels, observing TCP/IP flows and deciding, based on their analysis and their theoretical knowledge which fields may contain hidden information. They also become familiar with software and tools for traffic generation and exploration.
- 2) *Apply statistical methods for knowledge discovery.*
In the theory lecture students learn the basics of network traffic analysis, univariate and multivariate data analysis, the use of time series analysis, histograms, and distributions. In the lab they apply this knowledge to generate and test hypotheses about data. They learn to discover anomalies by applying their existing knowledge of the phenomena under test to the outcomes of statistical analysis.
- 3) *Understand TCP/IP flow behavior.*
In the theory lectures students learn typical protocol behaviors (IP, TCP, UDP, ICMP), i.e., the common values and distributions that header fields show in different implementations and scenarios. They also learn the differences between hiding encrypted and plaintext messages. In the lab students apply this knowledge in analyzing deviations from typical protocol behavior in order to identify potential covert channels. From the hands-on exercises they gain a deep understanding of TCP/IP flows on the Internet.
- 4) *Learn exploratory and forensics analysis skills.*
In the theory lectures students learn the basics of data analysis, as well as the pitfalls and common mistakes. In the lab they gain practical experience in dealing with ambiguity and uncertainty. The lab is specially focused on reinforcing exploratory thinking in data analysis. The lab prepares students to face problems that involve ambiguity, degrees of freedom, partial solutions, and uncertainty, skills that are difficult to convey in theory classes.

Table I summarizes the educational aims, the theoretical knowledge taught in the lecture, the exercises proposed, and the methods students are intended to use to solve them, and links these to students' observed performances.

III. KEY DESIGN DECISIONS

A. Student Prerequisites

Students are expected to have a good background in TCP/IP networks and basic knowledge of network security and data analysis. The lab is planned as an advanced lab for students who have already passed the Network Security lab described

in [1], so students should be able to perform traffic analysis tasks using tools like Wireshark[15], MATLAB[16], and RapidMiner[17].

Students work in pairs. One reason for this, beyond encouraging teamwork as a value in itself, is that the exercises require many data exploration and interpretation tasks. These tasks are easier to accomplish if students can discuss their findings in a team, instead of working alone. In addition, students with different backgrounds can profit from combining their skills, and a teamwork setting is more realistic for a future work environment. Having larger groups of students would make it difficult to assess the individual performance.

B. Exercise Overview

The lab consists of three exercises; these are consecutive, related challenges presented to students in the form of a story. In the first exercise – *Live Capture* – students are required to capture traffic data between two hosts, find the covert channel and decipher it. In the second exercise – *Offline Analysis* – students explore big traffic capture files containing many communication flows between several hosts; they must discover the flow with the covert channel and decode the hidden message. Finally, for the third exercise – *Using Covert Channels* – students utilize a covert channel to bypass security barriers and obtain some sensitive data.

C. Laboratory Setup

The physical layout of the laboratory, Fig. 2, has two computers per student team. The first, called *Main*, is used as a workstation. The second, called *Sec*, is required for the third exercise, where it simulates a remote host, so its human interface devices are blocked and students can only see the activities displayed on the screen.

A *Tutor's PC* controls all the lab machines, has access to students' workspaces, and triggers online communications by connecting to the *Ex1 server*. The *Ex1 server* is ultimately the device that establishes all IP communications for online exercises. Three additional servers (*Googel*, *Gockel*, *Goggel*) simulate external, neutral and bouncing hosts to be used by students during lab sessions.¹

Details of the exercises are explained in Section IV. Specific setup requirements are:

- *Exercise 1: Live Capture.* In this online exercise, covert channels are generated with CCHEF [18] by the *Ex1 server*. CCHEF is a software framework for empirically evaluating covert channels in network protocols. CCHEF needs existing network traffic on which to build the covert communication; this traffic is created with Iperf [19].

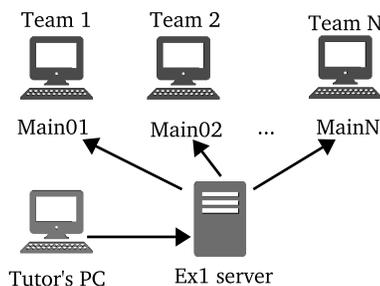
- *Exercise 2: Offline Analysis.* In the offline analysis part, covert channels are hidden inside datasets of real traffic captures (*pcap* files). Therefore only one computer is required.

¹The lab has been tested with machines with the following characteristics: *Main*, *Sec* and *Tutor's PC*: 8x Intel(R) Core(TM) i7-4770T CPU @ 2.50GHz, Memory: 16GB, OS: Ubuntu 12.04 LTS, kernel Ubuntu 3.11. *Ex1 server*: 16x Xeon 2.4GHz, Memory: 25GB, OS: Debian 6.0.10. *Googel*, *Gockel* and *Goggel*: 4x Xeon 2.4GHz, Memory: 16GB RAM, OS: Debian 6.0.10.

TABLE I
 NETWORK SECURITY ADVANCED LAB SUMMARY TABLE
 (Marks: Excellent, Good, Satisfactory, Adequate, Failed)

| Educational aims | Theory Part | Exercises | Solving method | Students' performance | Observed difficulties |
|---|---|-----------|--|------------------------------------|--|
| Learn about covert channel hiding/detection methods | Suitability of TCP/IP header fields for information hiding, subliminal channels in signatures | 1, 2, 3 | Diverse (<i>see specific exercise below</i>) | Good | Diverse (<i>see specific exercise below</i>) |
| Apply statistical methods for knowledge discovery | Network traffic analysis methods | 1 | Step by step analysis | Good (most) and Satisfactory (few) | Interpretation of statistics and graphs |
| Understand TCP/IP flow behavior | Typical protocol behavior, typical distribution of header field values | 3 | Design of exploit methods | Excellent | Occasional conceptual misunderstandings |
| Train exploratory and forensic analysis skills | Data analysis basics, common mistakes | 2 | Combination of analysis methods (free choice of tools) | Good | Initial solving plan. Cross referencing results of different analyses. |

Exercise 1: Live Capture

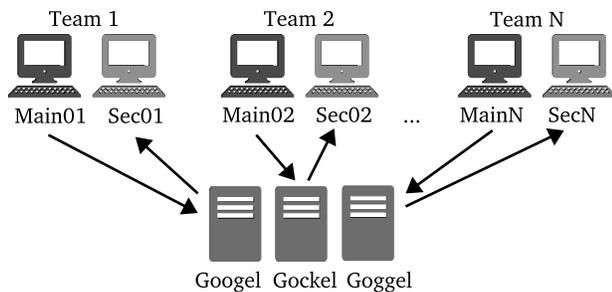


Exercise 2: Offline Analysis



be hacked. The mouse and the keyboard of the *Sec* machine are locked, leaving only the screen available for students to monitor their remote activities. Every *Sec* computer contains a copy of a *malware* implemented in Python⁵ that waits for incoming TCP packets on port 8888 and basically performs remote command execution. *Googel*, *Gockel* and *Goggel* simulate external hosts, which must be used by students as bouncing servers to circumvent firewall policies protecting *Sec* machines. Finally, *tgn*⁶ and *netcat*⁷ tools are intended to be used by students to operate the malware and obtain the data. A detailed description of how to implement the lab is available at [14].

Exercise 3: Using Covert Channels



D. Lab Sessions

The lab consists of six two-hour sessions. The lab design allowed an average of one session for exercise 1, three sessions for exercise 2, and one session for exercise 3. An additional final session was added to finish exercises, check results, and complete lab deliverables. Each team is also required to reserve a 30-minute time-slot for the oral reviews that take place some days after lab sessions are over.

The lab carried out during the winter semester 2014 verified the lab timing design; all teams had enough time to conclude the exercises.

Fig. 2. Hardware requirements for the lab exercises.

The offline exercise allows students to work with data from a more complex and realistic network scenario with many different traffic flows.

The original versions of the datasets used are publicly available for download from the NETRESEC webpage [20]. These *pcap* files are split into 100 MB files to give every team an equivalent amount of traffic to check. By using CCHEF, Wireshark, and other network utilities such as *tcpdump*², *tcpwrite*³ or *editcap*⁴, selected flows of every 100 MB *pcap* file were manipulated for the lab to inject a covert channel.

• **Exercise 3: Using Covert Channels.** For this exercise, the *Sec* computer plays the role of a remote machine that is to

IV. EXERCISE DESCRIPTION

To make the lab experience more enjoyable, and to place the exercise in an applied context, the exercises are devised as a story in which students play the role of the security staff at a government department (Ministry). The lab consists of three exercises to be solved sequentially, since ongoing solutions provide clues to sort out subsequent challenges. Each of these exercises addresses the goal of *learning methods to hide and detect covert channels in TCP/IP traffic* mentioned in Section II.

A. Exercise 1: Live Capture

• **Story.** The security staff of the Ministry has confiscated a suspicious laptop. Data analysts are required to check the incoming traffic and search for a covert communication.

²<http://www.tcpdump.org/>

³<http://tcp replay.synfin.net/wiki/tcpwrite>

⁴<https://www.wireshark.org/docs/man-pages/editcap.html>

⁵<https://www.python.org/>

⁶<http://netexpect.org/wiki/>

⁷<http://nc110.sourceforge.net/>

- **Setup.** The *Tutor's PC* triggers the IP connections between *Ex1 server* and the *Main* machine of each team. A covert channel is hidden in the communication, which lasts for ten seconds and is repeated every two minutes.

- **Required tasks.** Students must capture the incoming traffic, filter it to isolate meaningful data, find the covert channel, and decode the hidden message.

- **Methodology and goals.** This first exercise is devised as a *warm up* where students are guided step-by-step throughout a set of well-defined tasks. Students observe the captured data, apply filters to remove irrelevant information and make inferences according to the analysis of statistics (univariate and multivariate analysis, histograms, scatter plots). This exercise mainly pursues the *apply statistical methods* goal described in Section II.

- **Tools used.** Wireshark (capturing and filtering) and RapidMiner (filtering, statistics, distributions). In addition, students are free to use other tools such as MATLAB, Octave, LibreOffice Calc, or their own scripts (e.g., Python and Perl) for message decoding.

- **Example.** Fig. 3 shows examples of typical analysis situations that students face in Exercise 1. The histogram (Fig. 3, upper left) of the Destination Port of a flow shows three different values, which may be caused by hidden information. However, the time evolution shape of the values (Fig. 3, upper right) refutes the existence of a covert channel as values follow repetitive and non-mixed patterns that *a priori* cannot mask any meaningful information. An analogous histogram is shown for the Time to Live values of a different flow (Fig. 3, lower left). In this second case, the visual inspection of the time evolution triggers the suspicion of a covert channel (Fig. 3, lower right). Students are expected to reach such conclusions by themselves.

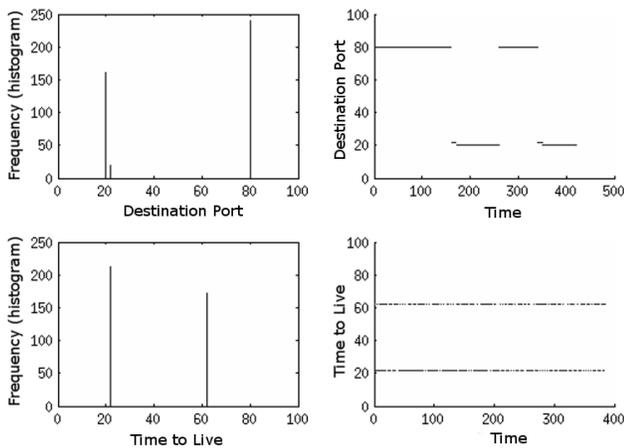


Fig. 3. Example of analysis in Exercise 1: the histograms (left) and the time series (right) of the values of the Destination Port (upper) and Time to Live (lower) in two different flows are displayed.

B. Exercise 2: Offline Analysis

- **Story.** *The Ministry suspects that some data leakage has occurred at some point during recent days. Data analysts are*

Listing 1. Example of an output of the script that checks multimodality. The two asterisks (*) are added for the purposes of this paper, and do not appear in students' outputs.

```
$ python feat_power.py --input file.csv --feature
  TTL --sort-by Source
```

| src.IP | pkts | Unique | Main |
|-----------------|-------|--------|-----------|
| 192.168.204.172 | 5662 | 1 | 1.0000 |
| 192.168.205.012 | 3 | 1 | 1.0000 |
| 192.168.200.001 | 19 | 1 | 1.0000 |
| 192.168.195.049 | 16909 | 2 | 1.4773 * |
| 192.168.198.057 | 1564 | 24 | 15.3842 * |
| 192.168.198.012 | 14 | 2 | 1.3086 |
| 192.168.213.165 | 54 | 1 | 1.0000 |

provided with collected traffic files that cover communications among multiple sources and destinations. They are required to find any covert channel and identify the implicated hosts.

- **Setup.** One-hour traffic captures of about 100 MB with anonymized payloads are given to each team of students. Additional scripts for data aggregation and multimodality estimation are provided to ease data exploration.

- **Required tasks.** Students must find the covert channel and decode the message.

- **Methodology and goals.** This second exercise is the most challenging and time demanding. Students must apply what they learned during the first exercise as well as their own intuition. Multimodality estimation and the combined application of filters and other analysis tools are introduced theoretically. Students are expected to draw upon various different analysis outcomes in order to discover the covert channel. This exercise mainly pursues the *train explorative and forensics analysis skills* goal described in Section II.

- **Tools used.** Python scripts for data aggregation and multimodality estimation. Students are free to use also other available tool for analysis and filtering (Wireshark, RapidMiner, MATLAB, Octave, LibreOffice Calc, own scripts).

- **Example.** Listing 1 shows an example of a typical analysis situation that students face in Exercise 2. In the example, students used a script to list existing source IP addresses in a traffic capture file. As the input argument they selected TTL (Time to Live) to check whether this field could be hiding a covert channel. The script shows how many packets each source sends, calculates the different unique TTL values that every source deploys (column *Unique*) and outputs the estimation of main TTL symbols (column *Main*). This is an exploratory step where students should realize that only a few sources (here marked with ‘*’) are susceptible to contain a covert channel in the TTL field. Sources that always send packets with the same TTL value (*Unique* = 1) are not able to hide any information in the TTL field; sources that send few packets are also unlikely to carry hidden information in the TTL. For instance, the sixth row in Listing 1 can in this case be discarded as the potential sensitive data would consist of just 14 bits. This step provides filtering criteria and its results must be matched with additional analysis to progressively isolate the flow with the covert channel.

Listing 2. Sequence of “tgn” commands to execute a “ls” in the *Sec* machine through a bouncing server and the malware.

```
$ #l
$ sudo tgn "ip (src=192.168.67.38,
dst=192.168.67.200,ttl=80)/tcp(src=8888,
dst=80,window=16384,syn,seq=107,ack-seq=42) "

$ #s
$ sudo tgn "ip (src=192.168.67.38,
dst=192.168.67.200,ttl=80)/tcp(src=8888,
dst=80,window=16384,syn,seq=114,ack-seq=42) "

$#[CARRIAGE RETURN]
$ sudo tgn "ip (src=192.168.67.38,
dst=192.168.67.200,ttl=80)/tcp(src=8888,
dst=80,window=16384,syn,seq=12,ack-seq=42) "
```

C. Exercise 3: Using Covert Channels

This exercise is based on the article published in [13].

- **Story** *The recipient machine of the data leakage has been identified. The Ministry wants to apply counterespionage measures to obtain information about the attacker. Experts are required to operate a malware that creates a backdoor into the remote host by using a covert channel.*

- **Setup.** A *Sec* machine (symbolizing the remote host) is prepared for every student team by disabling its human interface devices (mouse and keyboard), so that students can only access it from their *Main* computer by using the installed backdoor. The backdoor is deployed by a malware that executes commands concealed in the headers of TCP packets. The *Sec* computer is provided with security barriers, so students cannot connect directly to the malware but are forced to connect through bouncing servers.

- **Required tasks.** Students must connect to the malware and obtain information about the system and network configuration of the remote host.

- **Methodology and goals.** Here students take a hacker's role, hence they are encouraged to deduce what is happening in the network – from the network security perspective – based on the results of their exploratory actions. This exercise is partially a tutorial to introduce some network tools and show the complexity of some covert channel techniques. It mainly pursues the *understanding TCP/IP flow behavior* goal described in Section II.

- **Tools used.** `tgn` (traffic generation) and `netcat` (backdoor implementation), also general-purpose network utilities (e.g. `ping`, `ifconfig`).

- **Example.** Listing 2 displays a sample set of commands that students could deploy to execute the “ls” command on the backdoor created by the malware. The source IP 192.168.67.38 has been forged, it is actually the intended address to which the bouncing server 192.168.67.200 should respond to the TCP SYN attempt. The idea is that the bouncing server will answer to 192.168.67.38:8888 with a SYN-ACK packet containing an *ack-seq* equal to the *seq* in the original packet plus one. The malware is prepared to interpret and process *ack-seq* values that arrive at the TCP Port 8888 as ASCII characters.

V. EVALUATION OF THE ACQUIRED KNOWLEDGE

During the registration period for the first offering of the new class, in winter semester 2014, 25 students registered in order to access the slides and material⁸. Of those, 15 – 20 students regularly attended the lectures and 18 students took the theoretical exam before the lab started. A total of 18 students attended the lab.

A. Evaluation Format

Each student team (pair) is required to prepare a report, in which they write their results and answer a set of questions that appear throughout the lab handout. Marking criteria and the handout are available at [14] and in the Appendix to this paper. The evaluation process is completed with a lab review, as proposed in [8]. Here reviews took approximately 30 minutes per team and consisted of asking a minimum of three questions. Each student was required to answer at least one question without assistance from the team mate.

Students can obtain 30 points in total for the report and can gain up to a further ten points for their individual performance during the review. At least 21 points are required to pass the lab. The final grade is awarded on a combination of points from the theory exam and the lab exercise.

B. Testing Students' Skills and Knowledge

Students' work during lab sessions was evaluated based on the written report and the oral review. For the report, instructors checked each of the aspects listed below. Values in brackets show the maximum amount of points awarded for each part, yielding a total maximum of 30 points.

- *Exercise 1* (7 points). The ability to correctly filter data and interpret univariate and multivariate analyses was the main skill under test.
- *Exercise 2* (9 points). Agility in data exploration and non-guided problem solving was the main skill under test.
- *Exercise 3* (7 points). Interpretation of the TCP/IP flow context and the smart deployment of traffic tools were the main skills under test.
- *Presentation* (3 points). For document layout, result presentation, plots, and clarity of explanations.
- *Performance* (4 points). For the timing, quality and soundness of the applied methodologies.

In the oral reviews an instructor went through each report with the students under review, analyzing and pointing out the main misunderstandings and aspects that required further discussion. Students were also asked with questions that covered the following three issues (10 points in total):

- Consolidation of concepts, e.g., *What is a covert timing channel? How is the throughput of a timing channel compared to other covert channel types?*
- Understanding of methods and tools, e.g., *Write the tgn commands necessary to send the word 'OK' to host B by a covert channel hidden in the Source Address field.*

⁸Registration is required to gain access to material, but carries no other obligations.

- Interpretation of analysis outcomes. *Imagine the following scenario: in a flow between a source host and a destination host with 1000 packets, the "Length" field shows 100 different values and 27 main symbols (multimodality). Could the flow contain a covert channel? In such a case, could it be concealing ASCII characters?*

Questions were individually prepared for each student team based on a previous assessment of their reports. The objective was not only to evaluate students, but also to help them consolidate the necessary skills and knowledge.

C. Student Marks

Table II presents a statistical view of final student marks. These are normalized and presented on scale of 0 to 1, where 0 is a very poor performance, and 1 is an excellent performance. A population of 18 students does not provide enough statistical evidence to formulate general conclusions; nevertheless, based on the evaluations and the tracking of lab sessions, the lab was successful in teaching the educational goals outlined in Section II.

TABLE II
DISTRIBUTION OF STUDENTS MARKS:

1-Excellent performance, 0-Poor performance (Key: s.dev.: standard deviation, C.int.: confidence interval 95%, sts.: students)

| | | mean | s.dev. | C.int. | max. | min. | sts. |
|--------|--------------|------|--------|--------|------|------|------|
| Report | Exercise 1 | 0.76 | 0.15 | 0.07 | 1.00 | 0.57 | 18 |
| | Exercise 2 | 0.88 | 0.15 | 0.07 | 1.00 | 0.56 | 18 |
| | Exercise 3 | 0.92 | 0.10 | 0.05 | 1.00 | 0.71 | 18 |
| | Presentation | 0.78 | 0.14 | 0.06 | 1.00 | 0.67 | 18 |
| | Performance | 0.78 | 0.19 | 0.09 | 1.00 | 0.38 | 18 |
| Review | Questions | 0.71 | 0.17 | 0.08 | 1.00 | 0.40 | 18 |

Overall, Table II shows that students passed the lab with good marks and no major difficulties. The high average value for Exercise 3 is due to the fact that the exercise focused on showing students new schemes for covert channels and traffic generation tools, and was therefore deliberately designed to be a less complex exercise than the others.

Exercise 2 was conceived to be the most difficult part of the lab. Table III shows that it took students the longest time to perform, yet marks in Table II show that students performed satisfactorily. This is partly due to the fact that, although the injected covert channels and datasets were different for each team, students were able to share their *know-how* and suggest possible strategies for each other to follow. Such information-sharing among students was expected and welcomed.

TABLE III
TIME REQUIRED FOR COMPLETING LAB EXERCISES

For every exercise, min. and max. values correspond to the times of the fastest and slowest teams respectively

| | mean | s.dev. | min. | max. |
|------------|--------|--------|--------|--------|
| Exercise 1 | 2h 19m | 57m | 1h 22m | 4h 00m |
| Exercise 2 | 4h 20m | 1h 14m | 2h 08m | 6h 10m |
| Exercise 3 | 2h 14m | 30m | 1h 30m | 3h 00m |

D. Student Feedback

A total of 14 students completed the anonymous, standard TU Wien evaluation form (available at [14]) that posed 18 questions about the degree to which the course achieved its goals, with responses being on a Likert-type scale ranging from 1 (strongly agree) to 5 (strongly disagree).

For the questions related to the course preparation, students reported that they attended 66% to 100% (on average 90.9%) of the classes and spent two to five hours (on average 2.75 hours) per week for the course. The differences in the effort spent might be due to student preference, but was also caused by the heterogeneity of the class. The class was attended mainly by electrical engineering students, but some students were from computer science. Furthermore, some students had done their Bachelor's degree at universities other than TU Wien. Some of them (e.g., exchange students visiting TU Wien for a single semester) could not attend the basics NetSec lab, and thus lacked some experience in the field. This is also in line with instructors' observations during the lab of differences in student performance, with some groups having to put in extra effort, mainly due to their unfamiliarity with the tools and methods. None of the students had worked with covert channels before, so the lab content was new to all of them.

Questions about the course implementation (e.g., organization, content, answers to questions) received a positive rating, ranging from 1.0 to 1.36. Also the four questions about students' perception of knowledge gained, increase in their skills, and the usefulness of the class, obtained excellent evaluations. Detailed results are shown in Table IV.

TABLE IV
STUDENTS SELF-ASSESSMENT OF ACQUIRED SKILLS

1-strongly agree (positive), 5-strongly disagree (negative)

| Question | max-min | mean |
|---|---------|------|
| The course raised my interest in exploring the topic further. | 1 - 2 | 1.29 |
| Information was provided during the course on how I will be able to use the contents in the future. | 1 - 3 | 1.50 |
| The course increased my knowledge. | 1 - 3 | 1.14 |
| I am capable of using the knowledge I gained from the course. | 1 - 3 | 1.21 |

Students could also comment on what they enjoyed and what could be improved. These comments acknowledged a good structure and composition of the exercises. Furthermore, students enjoyed the diversity and the freedom of choice of tools for problem solving ("*diversity of the three exercises*", "*freedom regarding tools*") and the fun with the content ("*labs were fun and engaging*", "*the moments: when you successfully finish an exercise*"). For improvements, students suggested "*tool-tutorials before the class*", and additional information on RapidMiner and shell programming. Other suggestions asked for extensions to other domains (e.g., wireless, smart grid) and even for additional exercises ("*more exercises!; to be honest, I could have done another three exercises, it was fun!*").

VI. DIFFICULTIES DURING THE LAB EXECUTION

The lab experience carried out during the winter semester 2014 at TU Wien, as presented in this paper, entailed some inherent difficulties. The main ones are presented below.

A. Preparation

- *Laboratory Setup.* The preparation of the network for the lab as shown in Fig. 2 took a considerable amount of time, effort, and resources, for example in terms of duplication of required machines per team (*Main* and *Sec*), suitable configuration of *Sec* hosts, adjusting server policies, and so on. Future versions of the lab are being prepared based on Virtual Machines, thus achieving an easier and more robust setup.
- *Preparing datasets for the Offline Analysis.* Exercise 2 demands a different dataset for every team, each containing communications between many different sources and destinations. A covert channel must be injected in one of the flows and, in order to prevent cheating, the characteristics of flows and covert channels must be different. Moreover, solving the exercise must present an equivalent challenge for each team. Given that, the careful preparation procedure is manual and time-demanding. Currently alternatives to lighten the preparation effort of Exercise 2 are explored.

B. Lab Design

- *Students' background knowledge.* Being an *advanced* lab, the idea of giving students more freedom with solving techniques (compared with an *introductory* lab [1]) was adopted deliberately. The different backgrounds mainly affected the time students required to solve the exercises, but did not affect the quality of their performance. Table III shows the difference between the slowest and fastest groups, which differed depending on the exercise. In any case, the slowest groups needed up to triple the time needed by the fastest ones. As for the complete lab, the first group to finish required five hours, whereas the slowest needed exactly ten hours.
- *Unequal performance within a team.* One of the risks of allowing teamwork in labs is that, in some cases, team members do not make an equivalent effort. Two of the nine teams did not share tasks equitably, with one student carrying out most of the work while the other was more of a passive observer. This was clearly detected during oral reviews. Such cases correspond to marks below 0.5 in the *Performance* evaluation and the oral review (Table II, "min." column).

C. Students' Deadlocks

- *Interpretation of statistics and graphs. Application of methods.* The main problems observed in Exercise 1 were of interpreting statistics and graphs, and in following step-by-step analysis methodologies. Students' errors were due to: a) an inability to deduce filtering rules from statistic results; b) oversights, missing steps; c) problems interpreting reality from time series, histograms and 2D/3D plots.

- *Cross-referencing results of various methods.* Solving Exercise 2 is easy and fast by cross-referencing results from various analysis perspectives and tools. However, most students initially tackled the exercise by using brute force and trial and error. Except for one case, students needed at least a complete session (two hours) of exploring methodology before finding the right analysis paths. In any case, all teams came to understand that they needed to compare diverse perspectives to arrive at a solution.

VII. LESSONS LEARNED

This section describes lessons learned, and planned future improvements.

1) "Keep it Fun!"

To devise the lab exercise within a story about espionage, information leakage and hacking was an undeniable success. Students enjoyed the lab and this increased their interest in the topics under study. Although dramatically exaggerated on purpose, the story was intended also to establish a stronger link between technical studies and their application in the real, current social context.

2) Encourage Conjectural Reasoning

From the outset, the lab was devised to empower data exploration, reasoning based on conjecture, and the exploitation of non-directly-related resources, which inherently are basic skills for researchers and data analysts. Real problems do not always show a clear and straightforward solution or respond nicely to a suitable solving methodology. Students were expected to encounter some dead ends to train themselves to deal with some frustration during their research. Dealing with some tough challenges in their lab experiments made their final findings even more rewarding.

3) Introduce further demanding tasks

The related literature is full of various methodologies to hide data in TCP/IP flows. In order to reduce the complexity of the lab, the number of covert channel techniques under study was initially limited to a small set. Students' good performance in the lab suggests that future students can probably deal with a wider scope, and tackle covert channels generated from completely different perspectives, although it will demand an even deeper understanding of TCP/IP structures and mechanisms.

4) Free Choice of Tools

In the basic Network Security lab [1] students became familiar with a set of various data analysis tools. In the Network Security Advanced Lab students were free to use own tools and scripts. This freedom trains them to compare and select suitable tools for the different tasks they have to work on. The considerations of suitability of various tools for a given task is a valuable skill for

future research.

These four conclusions are based on the high performance rates, the quality of the reports, and the answers during oral reviews, as well as subjective observation by instructors and tutors who kept track of students' development during lab sessions.

Table V shows a comparison between the advanced lab described here and the NetSec introductory lab presented in [1], by considering students' marks. Students of the advanced lab were mostly students who had already passed the introductory lab. The difficulty of the labs was designed based on the students' background knowledge, and are considered equivalent. The evaluation criteria are the same. Given these conditions, a substantial increment is observed in the average mark of the advanced lab. Also, the standard deviation is considerably lower, expressing a higher homogeneity in the students' performances.

TABLE V
 COMPARISON BETWEEN LABS BASED ON STUDENTS' MARKS

1-Excellent performance, 0-Major difficulties (Key: s.dev.: standard deviation, C.int.: confidence interval 95%, sts.: students)

| | mean | s.dev. | C.int. | sts. |
|-----------------|------|--------|--------|------|
| NetSec [1] | 0.75 | 0.20 | 0.06 | 41 |
| NetSec Advanced | 0.80 | 0.08 | 0.04 | 18 |

The deeper consolidation of skills and knowledge in the second lab is an obvious goal for instructors, who as noted above identify the importance of "keeping it fun" as the key element in the improvement. Even students with the lowest marks showed interest and skills in the reviews, thus deserving to pass the lab. This point is also observable in the students' feedback sheets, where two remarkable outcomes are: a) the extremely high average rate obtained by the assessment *The course increased my knowledge* (1.14, Table IV), and b) the many comments expressing amusement.

This assessment, as well as the also high-rated assessment *I am capable of using the knowledge I gained from the course* (1.21, Table IV) support the conclusions reached in *Encourage Conjectural Reasoning* and *Free Choice of Tools*. Instructors observed that the free selection of tools made students feel comfortable, but also realize the limitations of their initial approaches and resources. Because of this, and the deep exploration conducted in the exercises, instructors observed that students gradually gained flexibility and a deeper understanding of the scenarios under analysis.

VIII. CONCLUSION

This paper has presented a class on network steganography with focus on teaching data analysis techniques for detecting TCP/IP covert channels. Students learn how information can be hidden and detected in network protocols, and gain an in-depth understanding of standards and typical behavior of TCP/IP flows. Students learn how to select suitable tools for their analysis and how to autonomously investigate exercise questions in a hands-on, fun environment. They are also

trained to deal with results from noisy scenarios with ambiguity and uncertainty. The first implementation of the lab at TU Wien showed that students not only acquired the aimed skills but also enjoyed the class and were eager to learn more. All exercises and datasets for the lab are made publicly available to allow other instructors to implement similar classes.

ACKNOWLEDGMENTS

The authors thank Joaquín Moreno Garijo for the contribution with his article about bouncing covert channels in [13].

APPENDIX – MARKING CRITERIA

A. Report, Exercises

Mistakes subtract points. There are four kinds of mistakes:

- *Negligible* (−0p). Worth nothing or correcting, but does not make outcomes wrong or affect the correct interpretation of results, e.g., significant typos, ambiguous phrasing, result wrongly transcribed.
- *Minor* (−0.5p). Wrong outcome/reasoning that does not derive from a major conceptual mistake or that ignores a not-so-important aspect of the exercise, e.g., a graph with wrong labels/units, result values that do not match the reasoning.
- *Significant* (−1.5p). A result/reasoning that does not meet the basic expectations of the intended exercise/question. A result that reveals a lack of fundamental understanding of the exercise purposes. A clear/evident contradiction/nonsense.
- *Major* (−2.5p). An exercise or a part of an exercise is completely wrong or missing.

B. Report, Presentation

The presentation of results and the report layout is assessed as follows: 3 (excellent), 2 (normal), 1 (on the borderline), 0 (bad).

C. Evaluation During Lab Sessions (Timing and Quality)

Based on instructors' observations, the performance during lab sessions is evaluated as follows: 4 (excellent timing, very good/imaginative solving methods), 3 (good timing, satisfactory performance), 2 (normal timing), 1 (slow but everything finished, too much help required), 0 (not finished).

D. Oral reviews

Questions should reveal if students understood what they did, what tools are for, how to use them, how to apply the knowledge to other fields or situations, etc. Reference marks:

- 0 – The student does not attend the review, does not answer anything or shows absolute ignorance about the exercise aspects/purposes and the deployed tools.
- 2 – The student answers wrongly all or most of the questions. They lack insight into the exercise insights and have no knowledge about the tools. In some cases they tried to get a free ride with another better performing student.

- 6 – The student has consolidated/understood just the basic knowledge of the exercise (the minimum required).
- 8 – The student shows a good understanding of tools and exercise aims, but ignores the most advanced aspects.
- 10 – The student shows interest and a deep understanding of the exercise, even when dealing the trickiest aspects.

REFERENCES

- [1] T. Zseby, F. Iglesias Vazquez, A. King, and K. Claffy, "Teaching network security with IP darkspace data," *IEEE Trans. Edu.*, vol. PP, no. 99, pp. 1–7, 2015.
- [2] M. Micco and H. Rossman, "Building a cyberwar lab: Lessons learned: Teaching cybersecurity principles to undergraduates," in *Proc. 33rd SIGCSE Tech. Symp. Computer Science Education*. New York, NY, USA: ACM, 2002, pp. 23–27.
- [3] P. J. Wagner and J. M. Wudi, "Designing and implementing a cyberwar laboratory exercise for a computer security course," in *Proc. 35th SIGCSE Tech. Symp. Computer Science Education*. New York, NY, USA: ACM, 2004, pp. 402–406.
- [4] R. Abler, D. Contis, J. Grizzard, and H. L. Owen, "Georgia tech information security center hands-on network security laboratory," *IEEE Trans. Edu.*, vol. 49, no. 1, pp. 82–87, Feb 2006.
- [5] M. Wannous and H. Nakano, "NVLab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies," *IEEE Trans. Learn. Tech.*, vol. 3, no. 2, pp. 129–138, April 2010.
- [6] I. Marsa-Maestre, E. de la Hoz, J. Gimenez-Guzman, and M. Lopez-Carmona, "Using a scenario-generation framework for education on system and internet security," in *IEEE Global Engineering Education Conf. EDUCON*, Marrakesh, Morocco, April 2012, pp. 1–7.
- [7] I. Marsa-Maestre, E. de la Hoz, J. M. Gimenez-Guzman, and M. A. Lopez-Carmona, "Design and evaluation of a learning environment to effectively provide network security skills," *Computers & Education*, vol. 69, pp. 225 – 236, 2013.
- [8] L. Ben Othmane, V. Bhuse, and L. Lilien, "Incorporating lab experience into computer security courses," in *World Congr. Computer and Information Technology (WCCIT)*, June 2013, pp. 1–4.
- [9] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *Commun. Surveys Tuts.*, vol. 9, no. 3, pp. 44–57, 2007.
- [10] S. Wendzel, S. Zander, B. Fechner, and C. Herdin, "Pattern-based survey and categorization of network covert channel techniques," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 50:1–50:26, Apr. 2015.
- [11] S. J. Murdoch and S. Lewis, "Embedding covert channels into TCP/IP," in *Proc. 7th Int. Conf. Information Hiding*, ser. IH'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 247–261.
- [12] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec 2005.
- [13] J. Moreno Garijo, "Covert channels," 2010, Security A(r)tWork, [posted 26-October-2010]. [Online]. Available: <http://www.securityartwork.es/2010/10/26/covert-channels/>
- [14] Communication Networks Group of TU Wien, "TU Wien NetSec Lab," 2014. [Online]. Available: <http://www.tc.tuwien.ac.at/netsec-lab>
- [15] "Wireshark." [Online]. Available: <http://www.wireshark.org/>
- [16] "MATLAB." [Online]. Available: www.mathworks.de/products/matlab/
- [17] RapidMiner, "Community edition, RapidMiner version 5.3." [Online]. Available: <http://rapidminer.com/>
- [18] S. Zander and G. Armitage, "CCHEF - Covert Channels Evaluation Framework Design and Implementation," CAIA, Centre for Advanced Internet Architectures, Swinburne University of Technology, Tech. Rep. 080530A, 2008.
- [19] "Iperf 2 (last update)," 2014. [Online]. Available: <https://iperf.fr/>
- [20] "Traffic Captures Corresponding to the U.S. National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC)," 2012. [Online]. Available: <http://www.netresec.com/?page=MACCDC>

Communication Systems (FOKUS) in Berlin and worked as a visiting scientist at the University of California, San Diego.

Félix Iglesias Vázquez was born in Madrid, Spain, in 1980. He received the Ph.D. degree in technical sciences in 2012 from TU Wien, where he currently holds a University Assistant position. He has worked on fundamental research and project development for several Spanish and Austrian firms, and lectures in the fields of electronics, physics and automation. His research interests include machine learning, data analysis and network security.

Robert Annessi received the B.Sc and M.Sc degrees in computer engineering from TU Wien in 2011 and 2014 respectively. Before his studies, he gained practical experience working for an Austrian ISP. Currently, he is doing his Ph.D in the area of secure group communication for critical infrastructures; his further research interest include anonymous communication networks and smart grid security.

Davor Frkat is a teaching assistant and Master student at the Institute of Telecommunications, TU Wien. His research interests include network security, darkspace analysis and covert channels.

Valentin Bernhardt is a teaching assistant and Bachelor's student at the Institute of Telecommunications, TU Wien. His research interests focus on covert channels and machine to machine communication in smart grids.

Tanja Zseby is a professor of communication networks in the Faculty of Electrical Engineering and Information Technology at TU Wien. She received her Dipl.-Ing. degree in electrical engineering and her Ph.D. (Dr.-Ing.) from Technical University Berlin, Germany. Before joining TU Wien she led the Competence Center for Network Research at the Fraunhofer Institute for Open