TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

institute of telecommunications

Exercise 2: Offline Analysis

Network Security - Advanced Topics (VU 389.160), Winter Semester 2015/2016

Communication Networks Group at the Institute of Telecommunications

1 Exercise 2: Offline Analysis

Your effort in decoding the covert channels has confirmed that indeed some data leakage operations are taking place within the Ministry at this moment. However, the arrested person seems not to be talking too much. Moreover, the machine from where the information was sent belongs to a bachelor trainee who assures that he does not know anything; actually, he claims that his computer must have been hacked.

In your opinion, the decoded messages found in the notebook clearly show that there is a problem within the minister's office network. Again, you suspect that covert channels are used.

After reporting your conjecture to your superiors, they ask you to go on with your investigation.

1.1 Exploring the captured traffic

While the security staff physically checks the rooms of the minister, you quickly go back to your office and ask the network operators for recent traffic captures of the minister's office network. Fortunately, they do have them. Together with your team, you proceed to search a covert channel that could have past undetected so far by the ministry's security barriers. Your main goal is to unmask IP addresses of the sender and receiver of a possible covert channel. Therefore, you must discover which traffic feature is used to conceal data, and, if possible, what kind of information is clandestinely sent through such security hole.

Now it is time to apply the methods and tools you have learned in Exercise 1. You can find the captured traffic as a pcap file to analyze in your `workfiles` folder (`/home/team??/workfiles/team??ex21.pcap`).

Rep:2.a – Steps

Find and decode the covert channel. In the report, depict briefly every step so that your exploration can be successfully understood and reproduced. For every step (in list format), provide sound arguments or reasoning that support your decisions. In addition, specify all the information that characterizes the covert channel.

Note: Also, briefly comment on wrong attempts or approaches which did not lead to any improvement but allowed you to progress in your reasoning. Be short, accurate, and organized.

Hint: In order to facilitate solving this exercise, the following hint is provided: the covert channel is concealed in a *flow* with ≥ 400 packets.

Important: Be sure that you have read all the exercise sheet up to section 1.2 before you start solving this task.

Rep:2.b – Message

Write in the report the message contained in the discovered covert channel.

You remember some tools that a colleague recently told you about during a coffee break. These tools might be useful to seize covert channels and discriminate irrelevant data and connections. Thus, you start reading about them:

1.1.1 IP address flow information

Theory To check aggregated data of traffic flows can be useful to identify or filter suspicious IP addresses. In normative documents [1], a Flow (traffic) is defined as *a set of packets or frames passing an Observation Point in the network during a certain time interval. All packets belonging to a particular Flow have a set of common properties*. For this exercise, let us consider that one flow contains all packets going from one Source Address to one Destination Address. For instance, a TCP connection between client A and server B would be expressed with two flows: A to B and B to A.

Usage In the `workfiles` folder you can find a script `showinfo.py`. This script takes the following arguments: a) a CSV file to analyze and b) a specific "source" or "destination" IP address. The script outputs some information about the activity of the provided "source" or "destination" and its main flow (in terms of packet volume).

Example Consider an hypothetical CSV traffic capture file called `example.csv`. The command

```
% ./workfiles/showinfo.py --input example.csv
--destination-ip 121.39.55.24
```

outputs the following information:

```
Number of packets: 6
Number of connecting partners: 4
Most connections: 200.133.27.44
Number of packets for 200.133.27.44: 3
```

Results show that the host 121.39.55.24 has received only 6 packets from 4 different hosts, and that the source address 200.133.27.44 sent the most packets (3) to the selected host. With so few packets from a single source, it is unlikely that the host 121.39.55.24 has received a covert communication in the analyzed capture. The situation changes for this second example[1], however:

```
% ./workfiles/showinfo.py --input example.csv
--source-ip 110.23.9.32
```

```
 Number of packets: 645
 Number of connecting partners: 2
 Most connections: 128.34.112.110
 Number of packets for 128.34.112.110: 620
```

Here, the selected host has sent a considerable number of packets (645) to two different hosts, mostly to 128.34.112.110 (620 packets). Of course, this is not by itself suspicious to conceal a covert communication, since this is can be a perfectly normal scenario. Nevertheless, the introduced tool can be helpful to quickly display some relevant flow information about a specific "source" or "destination" under observation.

1.1.2 Multimodality estimation

Theory A suitable way of detecting possible covert channels is estimating the number of principal symbols $M(X)$ (multimodality estimation [2]). A rough estimation can be obtained as follows:

$$M(X) = \frac{\sum n_i^2}{\max\left(n_i^2\right)} \qquad (1)$$

where $M(X)$ stands for the number of principal symbols of the observed histogram X. X represents the relative frequencies of occurrences of different states of a phenomenon (i.e. a specific traffic-flow feature). n_i refers to the number of occurrences of a particular state i. Terms are squared to emphasize dominant symbols and fade less frequent symbols.

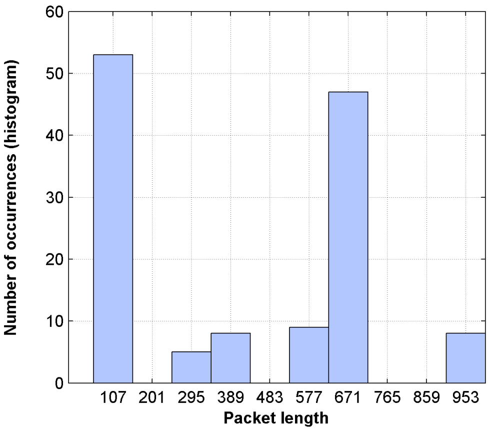


Figure 1: Histogram of packet lengths for an example case

Example This measure can be understood more clearly with an example. Assume we want to know whether a specific IP address *A* is using the packet length *L* to clandestinely convey information to destination address *B*. We filter the traffic and analyze the histogram of the packet length values used in the A→B flow (Figure 1). Based on the displayed histogram, a 2-symbol covert channel with two modes of length 107 and 671 could be in use. Therefore, we can consider this connection suspicious. The same information is provided by equation 1 without having to visually check the histogram. For the displayed case, out of 130 packets, length occurrences are: 107 (53 times), 295 (5), 389 (8), 577 (9), 671 (47) and 953 (8). The number of principal symbols can be estimated by

$$M(L)|_{A,B} = \frac{53^2 + 5^2 + 8^2 + 9^2 + 47^2 + 8^2}{53^2} \simeq 1.8 \quad (2)$$

The result ($\simeq 1.8$) gives an approximation on the number of principal symbols, i.e. 2.

Usage In the `workfiles` folder you find a script `srcfeat_modes.py`, which takes the following arguments: a) a CSV file to analyze, b) a feature to check, and c) a selection between "sources" or "destinations". It outputs a list, showing the following information:

X.X.X.X$_1$, No. of packets$_1$, No. of states$_{1,f}$, $M_1(f)$

X.X.X.X$_2$, No. of packets$_2$, No. of states$_{2,f}$, $M_2(f)$

...

[1] Note that, in the first example, we have selected to check the host in its role as a destination (`--destination-ip`), and in the second example as a source (`--source-ip`).

where *X.X.X.X*$_i$ is the source or destination address i and *No. of packets*$_i$ refers to the number of packets sent or received by i. *No. of states*$_{i,f}$ stands for the total number of observed states of feature f used by i, and $M_i(f)$ is the estimation of the number of principal symbols for feature f used by i. For the example above, the output would look like:

A, 150, 6, 1.8

This means that A sends 150 packets, from which the majority is shared between 2 out of 6 observed states.

This technique can be used to filter suspicious sources and destinations by observing the relative frequency occurrences of different features; mainly for those features where normal or random distributions are expected (i.e. where multimodal distributions are rare). Also $M(X)$ values close to 1 usually indicate the absence of a covert channel (since only a single dominant symbol hardly allows efficient communication). **Important!** There are multiple situations in which this detection scheme can fail. For example, when the analyzed time scope is too large compared to the time during which the covert channel is used (then symbols used for hidden communication become less dominant), also when code symbols correspond to feature ranges instead of specific values, or when there is a significant amount of noise caused by non-related traffic.

1.1.3 Mean of autocorrelation coefficients

Theory In some cases, traffic hiding covert channels can be detected by looking at the time evolution of field values. For instance, some fields, such as the IP *Identification*, are expected to follow self-correlated patterns. Breaking such patterns can suggest the existence of a covert channel.

We define ρ_A as the average of a set of selected autocorrelation coefficients (absolute values) with different lags (e.g. $A = \{1, ..., 5\}$):

$$\rho_A = \frac{1}{N} \sum_{\tau}^{A} |R_\tau| \tag{3}$$

where N is the number of elements in A. R_τ is the autocorrelation coefficient of the time series $Y_1, ..., Y_z$ for the lag τ, defined in equation 4:

$$R_\tau = \frac{E[(Y_t - \mu)(Y_{t+\tau} - \mu)]}{\sigma^2} \tag{4}$$

with μ and σ^2 being the mean and variance of the time series, and E the expected value.

Usage In the `workfiles` folder you can find a script `autocorr.py`, which takes different arguments that identify a feature and a flow inside a CSV traffic file (run `autocorr.py` without any arguments for additional information). The script outputs a score $\in [0, ..., 1]$, which is the calculation of ρ_A. Values close to "0" means no correlation, whereas close to "1" indicates a high self-correlation.[2]

Example For the first example, imagine that host A sends 104 consecutive packets to destination B using two different TCP Destination Ports, 80 and 443. Packet by packet, the Destination Port sequence could be as follows:

```
80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,
80,80,80,80,80,80,80,80,80,80,80,80,80,80,433,433,433,433,
433,433,433,433,433,433,433,433,433,433,433,433,433,433,433,
433,433,433,433,433,433,433,433,433,433,433,433,433,433,433,
80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,
80,80,80,80,80,80,80,80,433,433,433,433,433,433,433,433
```

`autocorr.py` obtains $\rho_A = 0.78$ for such sequence. It is not common in normal traffic to observe a highly overlapping destination ports (over long time periods) and, therefore, autocorrelation values are expected to be high.

In a second example, however, "Hello world!" has been binary encoded, using Destination Port 80 to mask '0' and Destination Port 443 to mask '1'. The 104-packet sequence in this second example is mixed and chaotic, showing a low $\rho_A = 0.09$:

```
80,433,433,80,433,80,80,80,80,433,433,80,80,433,80,433,80,433,
433,80,433,433,80,80,80,433,433,80,433,433,80,80,80,433,433,
80,433,433,433,433,80,80,433,80,80,80,80,80,80,433,433,433,
80,433,433,433,80,433,433,80,433,433,433,433,80,433,433,433,
80,80,433,80,80,433,433,80,433,433,80,80,80,433,433,80,80,
433,80,80,80,80,433,80,80,80,80,433,80,80,80,80,433,80,433,80
```

Important! Whether to expect high or low self-correlated sequences of values in a flow strongly depends on the traffic feature under analysis as well as the type of searched covert channel technique. Deep knowledge and understanding of traffic features and covert channels are mandatory for effective detection.

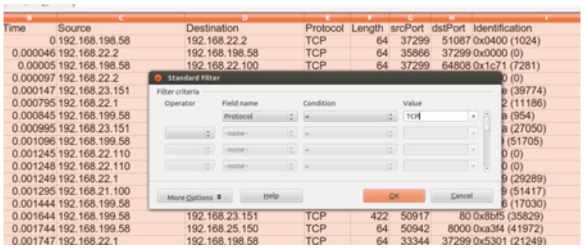
1.1.4 Combining tools and filters

This second set of exercises requires the combined deployment of introduced tools and methods. Correct preprocessing and filtering is determinant to remove irrelevant data and to progress in the task of seizing the covert channel. The exploration entails repeatedly filtering, generating CSV files, analyzing CSV files, suggesting and testing hypothesis, and improving filtering criteria.

For example, LibreOffice spreadsheets can be a suitable tool for filtering data. Results from Wireshark, Rapidminer[3] and the `srcfeat_modes.py` script can be saved as CSV files and imported into a spreadsheet. Then, filters can be applied using: "Data>Filter>Standard Filter..." by selecting diverse criteria (Figures 2 and 3).

[2] Note that the flow must contain a sufficient number of packets in order to provide a meaningful, reliable estimation.

[3] Have a look on the "Filter Examples" and "Write CSV" operators in Rapidminer.



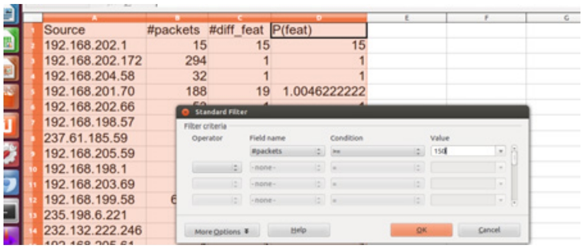
Time	Source	Destination	Protocol	Length	srcPort	dstPort	Identification
0	192.168.198.58	192.168.22.2	TCP	64	37299	51087	0x0400 (1024)
0.000046	192.168.22.2	192.168.198.58	TCP	64	35866	37299	0x0000 (0)
0.00005	192.168.198.58	192.168.22.100	TCP	64	37299	64808	0x1c71 (7281)
0.000097	192.168.22.2						0 (0)
0.000147	192.168.23.151						e (39774)
0.000795	192.168.22.1						2 (11186)
0.000845	192.168.199.58						a (954)
0.000995	192.168.23.151						a (27050)
0.001096	192.168.199.58						(51705)
0.001245	192.168.22.110						0 (0)
0.001248	192.168.22.110						0 (0)
0.001249	192.168.22.1						9 (29289)
0.001295	192.168.21.100						9 (51417)
0.001444	192.168.199.58						6 (17030)
0.001644	192.168.199.58	192.168.23.151	TCP	422	50917	80	0x8bf5 (35829)
0.001744	192.168.199.58	192.168.25.150	TCP	64	50942	8000	0xa3f4 (41972)
0.001747	192.168.22.1	192.168.198.58	TCP	64	33344	37299	0x5301 (21249)

Standard Filter

Filter criteria

Operator	Field name	Condition	Value
	Protocol	=	TCP
	-none-	=	
	-none-	=	
	-none-	=	

Figure 2: Filtering entries of a Wireshark CSV output file



Source	#packets	#diff_feat	P(feat)
192.168.202.1	15	15	15
192.168.202.172	294	1	1
192.168.204.58	32	1	1
192.168.201.70	188	19	1.0046222222
192.168.202.66			
192.168.198.57			
237.61.185.59			
192.168.205.59			
192.168.198.1			
192.168.203.69			
192.168.199.58	6		
235.198.6.221			
232.132.222.246			

Standard Filter

Filter criteria

Operator	Field name	Condition	Value
	#packets	>=	150
	-none-	=	
	-none-	=	
	-none-	=	

Figure 3: Filtering entries of an output file obtained from the multimodality script

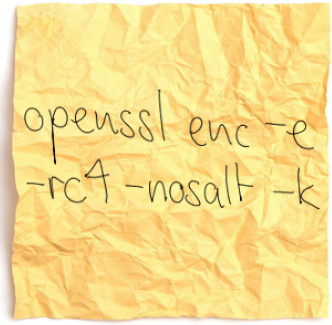
1.2 Encrypted covert data

Your suspicions about the usage of covert channels within the minister's office network to clandestinely convey information proved to be true eventually. Your findings indicate that another covert communication is likely taking place right now and, according to the previously discovered message, it might be encrypted this time. Quickly, you ask the network operators for the newest traffic captures of the minister's office network.

While gathering your team, a member of the ministry's security staff approaches. He informs you that another suspect was taken into custody and hands you a note (Figure 4) that was found in the suspect's pocket. Clearly, the security staff does not have the slightest clue about the meaning of the note. You, on the other hand, immediately recognize its value; this might actually facilitate decrypting the covert message.

Finally, everybody in your team starts checking the most recent traffic files. All information you can disclose about the intruders and the transferred data are of utmost importance!

You can find the captured traffic as a pcap file to analyze in your `workfiles` folder (`/home/team??/workfiles/team??ex22.pcap`).



openssl enc -e
-rc4 -nosalt -k

Figure 4: Note from the second suspect.

Rep:2.c – Encryption/Decryption

What does the note reveal about the used encryption? How can you use that information the decrypt the message?

Rep:2.d – Steps

Find, decode, and decrypt the covert channel. In the report, depict briefly every step so that your exploration can be understood and reproduced. For every step (in list format), provide sound arguments or reasoning that support your decisions. In addition, specify all the information that characterizes the covert channel.

Note: Also, briefly comment on wrong attempts or approaches that did not lead to any improvement but allowed you to progress in your reasoning. Be short, accurate, and organized.

Hint: Use the hint within the covert message discovered in the exercise 2.1 for solving this exercise. Note that the hint provided in the exercise 2.1 description is no longer valid. Therefore, the length of the flow containing this new covert channel can be arbitrary.

Rep:2.d – Message

Write in the report the message contained in the discovered covert channel.

References

[1] RFC 7011 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. Technical report, Internet Engineering Task Force (IETF), September 2013.

[2] B. W. Silverman. Using kernel density estimates to investigate multimodality. *J. Royal Stat. Soc. B*, 43(4):97–99, 1981.