Journal of Computer Virology and Hacking Techniques (2022) 18:117–126
https://doi.org/10.1007/s11416-021-00396-w

ORIGINAL PAPER

Covert timing channels: analyzing WEB traffic

Mehrdad Nasseralfoghara[1] · **Hamid Reza Hamidi**[1]

Received: 27 October 2020 / Accepted: 7 August 2021 / Published online: 21 August 2021

Abstract
In case there is a communication contrary to the system security policies, a covert channel has been created. The attacker can easily disclosure information from the victim's system with just one public access permission. Covert timing channels, unlike covert storage channels, do not have memory storage and they draw less attention. Different methods have been proposed for their identification, which generally benefit from the shape of traffic and the channel's regularity. The application nature of HTTP protocol allows the creation of a covert timing channel based on different features of this protocol (or different levels) that has not been addressed in previous researches. This research tries to study the effect of using different features (or levels) of HTTP protocol on identifying the covert channel. The amount of channel's entropy could be manipulated by changing the channel's level or adding intentional noise on the channel to protect from the analyzer's detection. The difference in the placement of the covert channel and the detector causes the amount of channel entropy to be far from the detection threshold. Therefore, we concluded that the analyzer must investigate traffic at all possible levels. Adding noise on the covert channel decrease its capacity, but as entropy increases, it would be harder to detect it.

Keywords Information security · Convert channel · Timing channel · WEB · Entropy

1 Introduction

A hidden communication to transmit hidden messages contrary to system rules is called a covert channel [1–9]. The covert channel can be used to disclose sensitive user information such as a password [10]. These channels can also enable remote control of embedded systems such as car [1] or artificial heart [2] and endangering people's lives.

Covert channels are categorized in several ways, as can be seen in Table 1. Different classification criteria are derived from covert channel components including hidden communication, message exchange, and security breaches.

The covert channel can be categorized into the covert timing channel and the covert storage channel. In the covert timing channel, similar to Fig. 1, information is exchanged using shared timely resources [11].

Unlike covert storage channel, covert timing channel has no memory at all. As a result, if the message is not received at the specific time, the information is irreversible [7]. For example, in the web context by specifying a time interval and monitoring whether to send a request at this interval, similar to Fig. 2, a covert timing channel can be created [12]. Due to public access to the web and the lack of users' sensitivity, this platform has a high capacity for such attacks.

The Hypertext Transfer Protocol (HTTP) is the foundation for information communication on the web. This protocol works based on "request-response" model. Typically, the response is important to the user. However, a covert timing channel on the web platform can use message sequencing, message delays [13] or message headers [12] to send information. The parameters that the normal user does not need to pay attention to.

The next section discusses the methods for countering the covert timing channels. After reviewing related research, the proposed approach is described and the results are evaluated.

✉ Hamid Reza Hamidi
hamidreza.hamidi@eng.ikiu.ac.ir

Mehrdad Nasseralfoghara
mnaser1992@gmail.com

[1] Faculty of Engineering, Imam Khomeini International University, Qazvin, Iran

Table 1 Classifications of covert channels

Taxonomy	Criteria	Research
Timing, storage, hybrid	Modulation medium	[3, 4]
Simplex, full duplex, half duplex	Modulation mode	[3, 5]
Noisy, noiseless	Channel noise model	[5]
Detectable, undetectable, secure undetectable	Modulation type	[5]
Hardware, network, operating system	Modulation environment	[6]
Host-based, network, air-gap	Reference monitor	[5]
Value-based, transition-based	Hidden message	[7]
Steganography, open	Channel cover model	[5, 8]
Active, passive	Role of sender	[9]
Mediated, shared	Channel attacker model	[3, 5]
Invasive, semi-invasive, non-invasive	Covert exploit	[5]

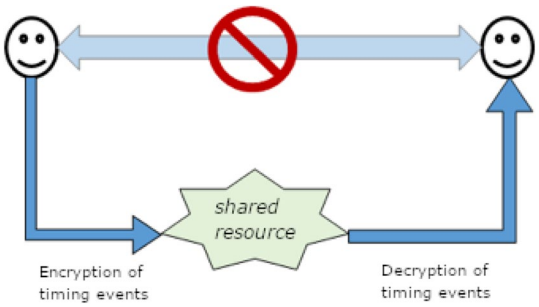

Fig. 1 Covert timing channel mechanism

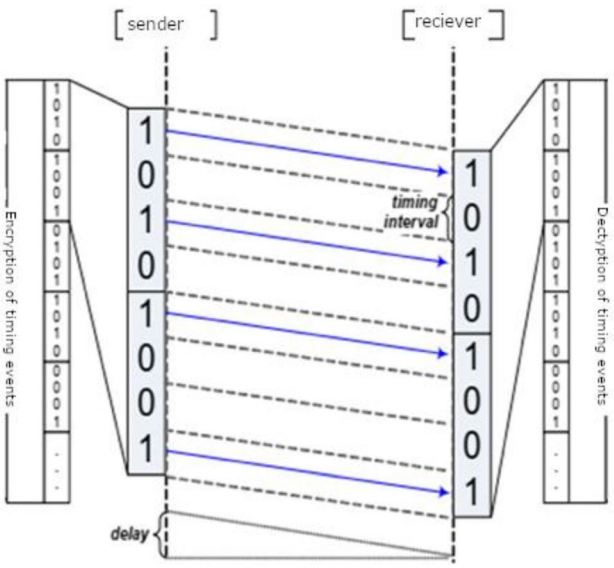

Fig. 2 How covert timing channel works [11]

2 Countering method

Detecting the covert channel requires heavy computational burden and is usually accomplished after the channel starts operating. However, since the covert channel is created based on shared resource access, it may be removable by restricting access to resources. Therefore, the first attempt to counter the covert timing channel is to eliminate it. Because the channel generally uses authorized traffic, eliminating it may result in disruption of the involved systems, which is not desirable. The next attempt is to reduce the channel capacity. This can also reduce the efficiency of the systems.

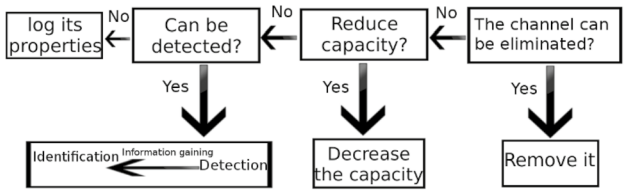

Fig. 3 Countering covert timing channels

The next step is detecting the covert channel. Since channel detection may not affect the performance of the involved systems, it is of particular importance. If not detected, the required alert is recorded where users are notified of the risk [10]. The countering procedure can be expressed as Fig. 3.

Covert timing channel detection tests are divided into two general categories; shape and regularity tests. The shape of traffic is described by first-order probabilities such as median and distribution, while channel rules are described by second-order or higher probabilities such as the correlation between information [14].

It is possible to detect changes in the shape of traffic with a statistical test. These tests examine changes in the statistical distribution. One of these tests is called Kolmogorov–Smirnov, which measures the maximum distance between two statistical distributions [15, 16].

The regularity test is based on the variance of delays between packets. This method is based on the fact that in normal traffic, the variance of delays changes over time,

while in a covert timing channel this value remains almost constant [11]. In another method, it is assumed that the delays between packets are approximately equivalent to a normal distribution. Note that if we create a histogram of the delays, we expect the median values of the histogram to be the most frequent. The existence of a two-dimensional or a multidimensional distribution indicates the existence of a covert timing channel [12].

In another study, the compression test was used [17]. Based on this criterion, network delays are first collected, then arranged from top to bottom and the relative difference between the delays is calculated. Finally, the percentage of numbers that are less than a threshold is used as the test criterion and compared to regular traffic. This method works well for noisy covert channels. By increasing the noise in the covert channel, the accuracy of this method is reduced [18].

Another method is entropy-based detection. In this method, the irregularity of normal traffic and suspicious channels are compared. The entropy calculation is based on the distribution of delay samples between messages. Any deviation from this training distribution entails a lower entropy and may indicate that information is being diverted by an unauthorized channel [14, 19, 20].

The main characteristics of a covert timing channel are its robustness, capacity and invisibility [21]. The channel strength indicates the maximum noise tolerance on the channel while the message decoding is retained by the receiver. Channel capacity is the amount of useful information transmitted by the channel per unit of time. And invisibility is the likelihood of external auditors detecting the channel. There is a lot of work being done on covert channels when they use the delay between packets. Generally, one or more attributes are important for the channel's design. Some studies are looking to build a covert channel that is just be robust. Some others are just looking for high capacity. Of course, this feature alone could not guarantee to stay with high capacity because if the covert channel is detected and confronted, its capacity may also be limited. Generally, the high capacity of the channel attracts attention and is in contrast to invisibility. There must be a trade-off.

3 Related researches

Investigating research in the area of covert timing channels based on messaging shows which features (robustness, capacity, and invisibility) have been important to researchers.

A research into the creation of the covert channel is done when the transmitter and receiver first agree on a set of time intervals. Each time interval represents a number that depends on how the message is converted. For example, if the incoming message is converted to binary and then sent, two numbers are agreed for the time intervals for sending a bit. Then, according to these intervals, the sender sends the message. Finally, the noise is added to channel to make difficult the detection [11].

In another study, ordinary traffic is first analyzed. Then, using the famous statistical distributions, the closest distribution to this traffic is selected. Next, the incoming message encoded. Each symbol is mapped to a random number and then a set of random numbers is generated by a statistical distribution. This statistical distribution will generate new traffic carrying the hidden message. The recipient must have the agreed mapping table. The aim was to bring concealed traffic closer to regular traffic and to avoid channel detection [22].

When there is a router or other network agent between sender and receiver, the delay of the messages will no longer be in accordance with the sender's desire and is based on queue theory. The Markov process is a mathematical model for scheduling tasks in queue theory. One of the effects is the reduction of channel capacity. Some researchers have attempted to increase the channel capacity under these conditions using mathematical formulas and statistical distributions [13, 23].

Some work has tried to robust the channel and stay invisible from detector systems. Generally, error correction codes are added to the original message when sending a message by the sender [24–26]. Instead of adding error correction code and retrieving the original message, they have expanded the message itself. So that for low and acceptable errors, the original message is visible to the receiver. Statistical distributions have also been used to improve the detection capability. The nearest distribution to the normal traffic is selected and a hidden message is sent at this distribution.

In covert timing channels, the capacity of the covert channel can be increased by reducing the binary string. By converting the incoming message to Huffman code, researchers have been able to increase the covert channel capacity [27, 28].

The use of the statistical distribution of ordinary traffic in concealed traffic, also known as model-based concealed channeling, is done [29]. In this research, the hidden message is mapped to a number. Further delays are added to create the desired distribution. In setting the delays, a recursive algorithm is used that makes hidden message delays dependent on each other. Therefore, the receiver needs to execute a recursive algorithm. This also strengthens the covert timing channel.

Normal traffic has no specific distribution and is random. Bringing the statistical features of hidden traffic distribution to ordinary traffic is a good idea to do [30]. In this study, both the shape and the regularity of usual traffic have been mimicked, which make difficult to be detected by different types of security systems.

Table 2 Related studies of covert timing channels

What to improve	Main idea	Research
Invisibility	Adding noise to the channel	[11]
Invisibility	Model-based distribution	[16, 22]
Robustness	Neutralize the effect of queue delay using mathematical formulas	[13, 23]
Invisibility	Obfuscation of receiver code	[32]
Robustness	Extending message length	[24, 25]
Invisibility	Select closest distribution to usual traffic	[24, 25]
Invisibility	Balancing parameters passed on Markov distribution	[17]
Invisibility	Simulate channel's distribution to the ordinary traffic distribution	[33, 34]
Invisibility	Coding of network traffic	[35]
Invisibility	Match the shape of channel to ordinary traffic	[36]
Capacity	Huffman code	[27]
Robustness	Fountain code	[28]
Invisibility	Modeling statistical distribution based on ordinary traffic	[29]
Robustness	Using the recursive algorithm to relating the delays	[29]
Invisibility	Bringing the statistical features of covert channel to normal traffic distribution	[30]
Capacity, robustness, invisibility	Using physical layer of network	[37]

We summarize the covert channel time domain studies in Table 2. The covert timing channel in these studies was implemented based on the delay between packets. The HTTP attributes allow to create a timing channel on each of these attributes (or levels), and none of the related studies have examined the impact of timing channel creation on different levels of the web to channel invisibility. For this reason, our study attempts to evaluate some factors that are effective in detecting covert timing channel in the web.

4 Entropy-based detection

The research environment is assumed to be a network with a number of transmitters-receivers and a traffic analyzer. Message exchange is done on HTTP and timing channel is based on packet delays. Figure 4 shows the components of a sample request of HTTP protocol. A sender must send the request according to the location of receiver (IP address and port). The message text contains the requested page (resource) address, the desired function/method as well as the parameters of requested method. Therefore, the content of hidden message can be created as follows, from top to bottom levels (attributes):

1. IP address and port number.
2. Requested page address.
3. Function/method.
4. Requested method's parameters and their values.

To create a covert channel, a transmitter must follow the requirements of HTTP request with respect to the receiving location (IP address and port), the content of hidden message, the delay for bit "0", the delay for bit "1" (similar as Fig. 2), and the creation level of covert channel.

It is assumed that the covert channel receiver also knows the channel details. The recipient can retrieve the hidden message by collecting incoming messages based on HTTP and the delay between requests. Because traffic is in the public domain, so many unrelated messages may also have been sent to the covert channel receiver that we call noise.

The proposed method analyzes the web traffic at the same time as the transceivers. Figure 5 shows the components of the simulation environment.

The proposed algorithm of this paper is to use the entropy of message exchange traffic. In information theory, the entropy criterion is used to measure the randomness of data. Suppose x is a random variable that can accept alphanumeric values of the set S. The probabilistic distribution is [31]:

$$P_X(x) = \Pr\{X = x\}(x \in S)$$

Fig. 4 A HTTP request format

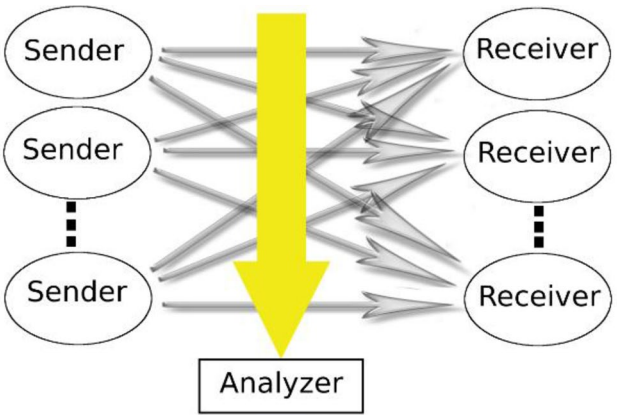

Fig. 5 Experiment components

Therefore, the entropy of the discrete random variable x, when I is information content and b is the entropy value in the information unit, is defined as follows [31]:

$$\mathrm{H}(X) = \sum_{i=1}^{n} \mathrm{P}(x_i)\mathrm{I}(x_i) = -\sum_{i=1}^{n} \mathrm{P}(x_i)\log_b \mathrm{P}(x_i).$$

For example, suppose we want to send a binary string in a communication channel. The string contains 25% of the digit "1" and 75% of the digit "0" and the unit of information is a bit, so $b=2$. In this case:

With this definition, we can obtain the entropy of any set of data. Data that is more regular has the lower entropy value. The highest entropy value occurs when all probabilities are equal (the data has a maximum coincidence).

$$\mathrm{H}(X) = -\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) \approx 0.81$$

The proposed algorithm of this study calculates the randomness of the messages between separate pairs (sender-receiver) by calculating the entropy of these messages.

$$\mathrm{H}_e = -\left(P_e \log_2\left(\frac{1}{Pe}\right) + (1-P_e)\log_2\left(\frac{1}{(1-Pe)}\right)\right)$$

In this study, the covert channel only uses two symbols ("0" and "1") to send information. Therefore, the channel entropy value is 1 for binary string. If another (wanted or unwanted) symbol is added to these two main symbols, the irregularity of these symbols increases and therefore entropy increases. If the probability of error (eg, due to unintended delays) is considered Pe, then the channel entropy increases to the following value [31]:

An entropy-based analyzer needs to set a "threshold" for entropy to detect the covert timing channel. Whenever it finds a relatively regular behavior whose entropy is less than this "threshold", it declares it a covert channel. In this study, we assume a maximum of 20% error ($Pe=0.2$), so the threshold is equal to 0.5.

5 Evaluation

Numerous experiments have been conducted to evaluate the effect of different parameters on the detection of covert timing channels. A specified transmitter and receiver within the context of an active public network perform creating a covert channel. The receiver is deployed on a virtual machine outside the local network (Internet) and the transmitter is deployed on a computer on the local network. Messages have been exchanged across the web using HTTP and hiding is based on the delay between messages. It is therefore essential to record the timing of packets on the receiving and analyzer side. Network messages are recorded on the network router side (for analyzer use). We first made sure the receiver was able to receive the message correctly, and then enabled the analyzer to detect the channel.

In all experiments conducted in this study, the sender sent messages with a length of 200 symbols. The exchange of messages has been done on a public network and therefore there exist noise (unwanted error) caused by normal network traffic. The receiver knows how to decode messages. Our evaluations are based on the following three parameters:

- "Threshold": the channel detection limit by the analyzer, assuming 20% error is a constant value, 1.721.
- "Max_CTC": The receiver calculates the entropy of the in-channel messages after correctly decoding the messages. Given the presence of noise, this entropy is definitely greater than the ideal entropy (value 1).
- "Min_Noise": The messages in public traffic that are not hidden within the channel were considered noise and their entropy was calculated. In our tests, there was only normal network traffic noise. If the transmitter also generates fictitious noise to increase channel invisibility, the entropy will certainly be greater than this.

If "Threshold" exceeds "Max_CTC" and less than "Min_Noise", then the analyzer can correctly identify the covert channel. Otherwise, it either does not recognize the channel (False Negative) or mistakenly considers noise as channel (False Positive).

The covert channel was created at two levels (page and method) and the analyzer performed entropy evaluation at three levels (port, page and method). Next, we evaluate the results of our experiments depending on what the sampling level of the analyzer was relative to the covert channel.

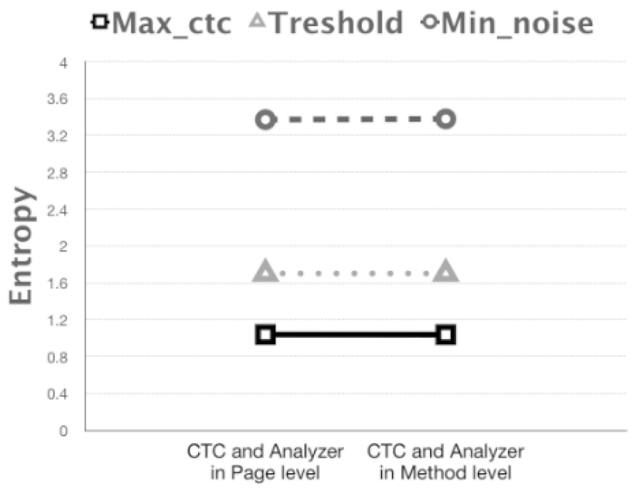

Fig. 6 Channel and analyzer on the same level

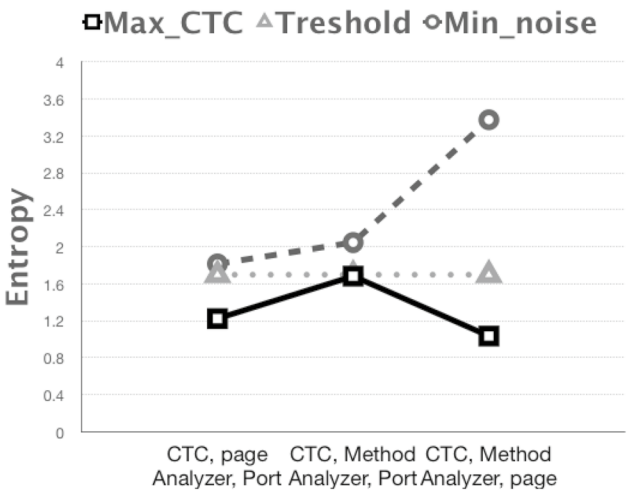

Fig. 7 Channel at lower level

5.1 Channel and analyzer on the same level

When the covert timing channel (CTC) is created at the page level, if the analyzer is also at the page level then the channel and the analyzer are in the same level. For the channel at the method level, the same level analyzer is at the method level.

As can be seen in Fig. 6, the analyzer was able to identify correctly the covert channel with respect to threshold value ("Threshold" is greater than "Max_CTC" and less than "Min_Noise").

5.2 Channel at lower level

The channel is created at two levels (page and method) so there are only three states where the channel is at the lower level: CTC at page and analyzer at port, CTC at method and analyzer at page or port level. When analyzer is on upper level of CTC, it can monitor channel messages. In addition to covert channel messages, other messages can fall into analyzer's perspective. For example, when analyzer is on port and the covert channel is created on page, in addition to the requests on particular page, it also picks up the rest of the requests on other pages of that address. Therefore, if there are too many requests to other pages, the analyzer will be mistaken.

As shown in Fig. 7, the covert channels were also correctly identified by the analyzer in all three states, but interestingly, the analyzer detected the other covert channels from the unused ports in the experiment. The "Min_Noise" and "Max_CTC" reported by the analyzer is a bit unexpected. If we are optimistic, additional channels and noise detected as system processes such as trying to connect to virtual server, updating website content or any other system process in background.

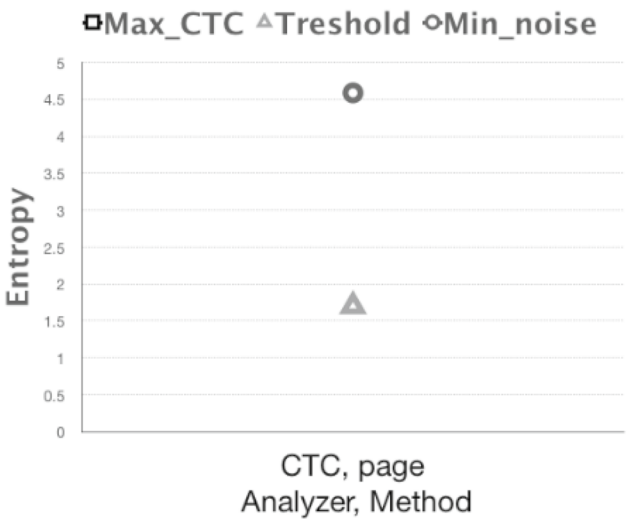

Fig. 8 Analyzer at lower level

However, in the most pessimistic way, a covert channel has attacked the system. According to the analyzer, the values of the channel delays and unexpected noises were in the range of zero to one second. Because of the initial assumption of covert channel strength, the analyzer calculates the delays in seconds. The result is that the background processes change the chart.

5.3 Analyzer at lower level

According to channel and analyzer levels, only when the channel is on page and analyzer on method, analyzer is in lower level.

As shown in Fig. 8, in this case, "Max_CTC" exceeds "Min_Noise" and both have significant distances beyond "Threshold" level, so the covert channel is not detected.

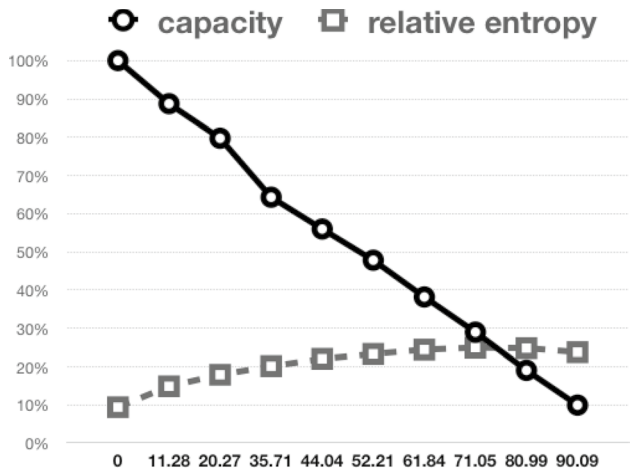

Fig. 9 Linear decreasing of channel's capacity

When requests are sent to a page, the headers of the HTTP are seen irregularly, so analyzer sees a random sequence whose entropy is higher than the threshold.

5.4 Impact of capacity on invisibility

Although the purpose of this study was not to create a covert channel with a low probability of detection (high invisibility), we attempted to measure the impact of analyzer's position related to the channel on diagnosis.

In an experiment, we examined the entropy behavior of the covert channel by increasing the noise to the channel (intentional error) and thus reducing its capacity. In this experiment, the transmitter, receiver, and analyzer were mounted on a computer to avoid general network traffic (unintentional noise). By increasing the noise in the covert channel, the length of its sequences increases. Then the entropy rate also increases. We use relative entropy to eliminate the entropy dependence on sequence length [31]:

$$\mathrm{H(X)} = -\sum \frac{\mathrm{PX(x)\ log\ PX(x)}}{\log\left(|S|\right)}$$

The channel capacity is equal to 1 when there is no noise (intentional or unintentional error). As the noise increases, the useful capacity of the channel decreases as follows [31]:

$$C = 1 - \left(P_e \log_2\left(\frac{1}{Pe}\right) + (1 - P_e)\log_2\left(\frac{1}{1 - Pe}\right)\right)$$

If the noise level reaches 50% ($Pe = 0.5$), the channel capacity becomes zero. That is, the recipient cannot detect the hidden message. However, if the noise level exceeds 20%, the analyzer loses the detection capability, given that the "minimum" noise entropy is lower than the hypothetical value of this study as the "threshold" of channel detection (1.721).

As shown in Fig. 9, with the increase in intentional noise, the entropy of channel and its invisibility increases, but on the other hand, the channel capacity decreases as well. Increasing the noise to a degree that reduces the capacity at 20%, the entropy level also exceeds the "threshold" level (assumed 20%), after which the analyzer cannot detect and the channel invisibility is maintained.

5.5 Impact of intentional noise

We measured the effect of intentional noise on channel detection in all possible cases of relative location of the covert channel and the analyzer in other experiments. We have created 10 covert channels within the context of an active public network. The receivers were deployed on separated virtual machines outside the local network (Internet) and the transmitters are deployed on a computer on the local network. Network messages were recorded on the network router side (for analyzer use). We first made sure the receivers were able to receive the message correctly, and then we activated the messages recording process. This experiment was repeated 20 times with different intentional noise on the channels. For each case of analyzer location, we measured "false positive" and “false negative” rates.

We present the results as a detection error tradeoff (DET) graph, Fig. 10. Adding noise increases the covert channel invisibility to the analyzer of the same-level with a threshold value of 1.7. The analyzer coverage could be increased but the probability of detecting a "false positive" error may be increased too. Reducing "threshold" decreases the coverage of analyzer. However, it is more likely to detect "false negative" errors. When the analyzer is at a higher level than the channel, invisibility is better than the case when the analyzer is in lower level. On a public network, we expect most messages to have public requests (at a higher level), so the message rate at the higher levels is much higher than the message at the lower levels. Therefore, if the analyzer is at lower levels, it sees more entropy and, as a result, the channel is invisible.

5.6 Impact of web protocols

We next investigated the impact of different web protocols in all possible cases of relative location of the covert channel and the analyzer in other experiments. We redesigned and implemented covert channels and analyzer using SMTP[1] and FTP[2] requests. Then using SMTP and FTP protocols, we repeated the previous experiments (Sect. 5.5.) with the same procedure and measured "false positive" and “false

[1] Simple mail transfer protocol.

[2] File transfer protocol.

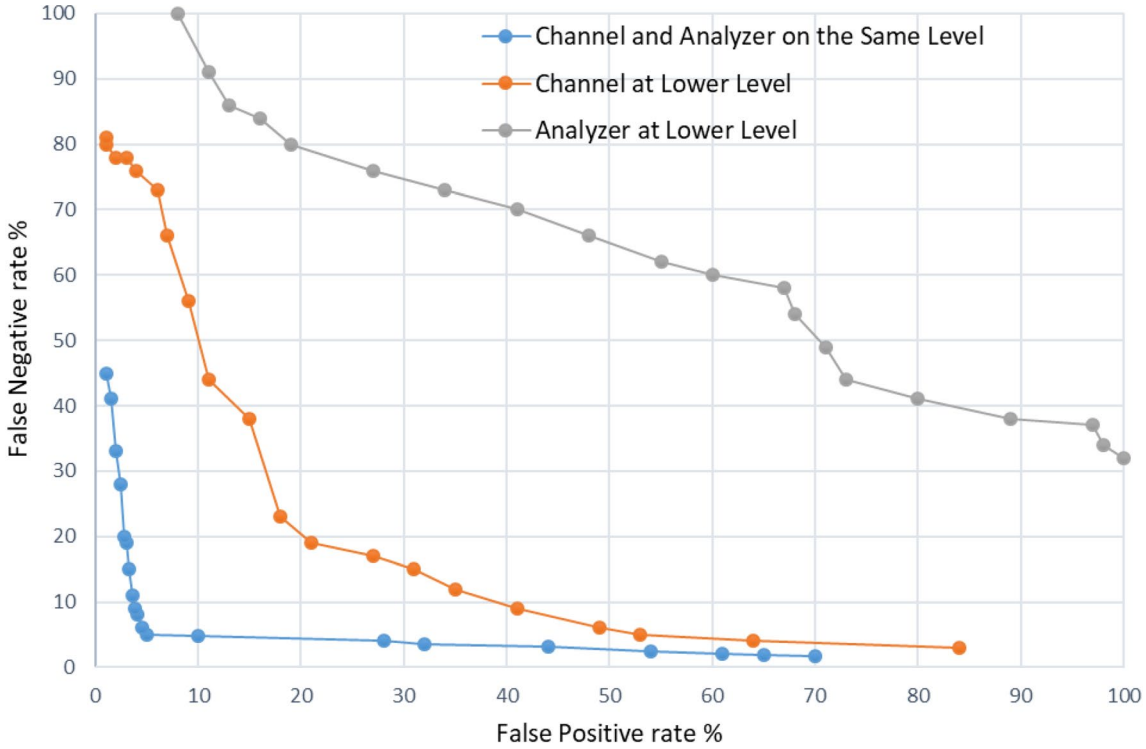

Fig. 10 Detection error tradeoff (DET) graph related to the levels of analyzer

Table 3 Statistical analysis results for HTTP, SMTP and FTP on false detection rates

	HTTP		SMTP		FTP		HTTP-SMTP		HTTP-FTP	
	FP	FN	FP	FN	FP	FN	*p* value	t value	*p* value	t value
Both on the same level	5.1	5	5.05	4.97	4.98	5.05	0.15	1.43	0.32	0.98
Channel at lower level	21.85	19.12	21.76	19.06	21.57	19.32	0.66	0.413	0.56	1. 32
Analyzer at lower level	67.21	57.97	67.43	57.45	67.19	57.01	0.47	2.44	0.72	2.11

negative" rates. Paired-samples T-test examination of difference was utilized to comprehend the measurable centrality of the exploratory protocols on the detection rates. We did not find significant statistical differences between HTTP, SMTP and FTP protocols in false detection rates (in all cases p value > 0.05), as shown in Table 1. Therefore, the used web protocol has no impact on detection rates.

5.7 Discussion

In all researches that have been proposed to identify the covert timing channel (Table 3), an analyzer has to examine the traffic of message transmission. These studies have measured the effect of different coding methods, noise levels or statistical models on the accuracy of the analyzer. The applications based on web protocols have numerous functions and parameters that could greatly increase the scope of channel building. Even when there is no complexity in coding methods, noise levels or statistical models, changes in the functions or parameters of web requests can confuse the analyzer. This paper evaluated the effect of the relative position of the analyzer and the channel in terms of these functions and parameters. We found that the used web protocol did not have a significant effect on the probability of analyzer deception. However, the relative position of the analyzer and the channel (in terms of the used function or parameter).

6 Conclusions

We used an entropy-based method for analyzing covert timing channel. This idea has already been used in other studies, but we examined the sensitivity of various parameters affecting invisibility. The analyzer's level as well as the effect of increased intentional noise (or reduced channel capacity) on channel invisibility have been demonstrated in the experiments.

We evaluated a new criterion called the relative position of covert channel and analyzer. Proper position of the analyzer is important for detecting the covert channel. When the analyzer is at higher levels, due to more requests than covert channel requests, the false positive rate is increased. In addition, when the analyzer is at lower level, the false negative is increased because the amount of channel's entropy is measured more than its real value. The entropy threshold is also important to distinguish covert channels. This threshold value can be calculated in the learning phase.

The created covert channel in our study was one of the simplest channels. Complicating its various parameters can make it harder to be detected. For example, the covert channel can contain up to two valid symbols that send a pair of "00, 01, 10, 11" in any inter-packets' delay. In addition, more complex coding such as Huffman code can be used to encode messages to use less bits and thus less time to send message. In this case, the channel capacity will increase. In the experiments, we sent only the hidden message itself. In cases that are more complex, error detection or error correction codes can be added to the message.

In this study, the main characteristics of the channel (IP, port, page and method) were constant throughout the lifetime of a channel. For example, the sender always uses the same IP and port. To make detection more difficult, the sender can change IP, port or both during a specified time interval. The receiver must be aware of these changes. In addition, the receiver may not be stationary; in fact, the process that is considered to be the receiver requires several IPs.

Funding This research received no external funding.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Sommer, F., Jürgen, D., Reiner, K.: Survey and classification of automotive security attacks. Information **10**(4), 148 (2019). https://doi.org/10.3390/info10040148
2. Mikhail, F., Flor, A., Steinmetzer, D., Paul Gardner, S., Hollick, M.: Survey and systematization of secure device pairing. Commun. Surv. Tutor. IEEE **20**(1), 517–550 (2018). https://doi.org/10.1109/COMST.2017.2748278
3. US Department of Defense.: Trusted Computer System Evaluation Criteria. ISBN 978-0-333-53947-7, Palgrave Macmillan, London (1985). https://doi.org/10.1007/978-1-349-12020-8_1.
4. Gligor,V.D.: A guide to understanding covert channel analysis of trusted systems. National Computer Security Center (US). Meade, Maryland, NCSC-TG-030 (1994)
5. Carrara, B., Adams, C.: A survey and taxonomy aimed at the detection and measurement of covert channels. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 115–126 (2016). https://doi.org/10.1145/2909827.2930800
6. Okhravi, H., Bak, S., King, S.T.: Design, implementation and evaluation of covert channel attacks. In: IEEE International Conference on Technologies for Homeland Security (HST), pp. 481–487 (2010). https://doi.org/10.1109/THS.2010.5654967
7. Wang, Z., Lee, R.B.: New constructive approach to covert channel modeling and channel capacity estimation. In: International Conference on Information Security, pp. 498–505 (2005). https://doi.org/10.1007/11556992_37
8. Changxiang, S., et al.: Survey of information security. Sci. China Ser. F Inf. Sci. **50**(3), 273–298 (2007). https://doi.org/10.1007/s11432-007-0037-2
9. Xiaosong, Z., et al.: A covert channel over volte via adjusting silence periods. IEEE Access **6**, 9292–9302 (2018). https://doi.org/10.1109/ACCESS.2018.2802783
10. Mazurczyk, W. et al.: Information Hiding in Communication Networks: Fundamentals, Mechanisms, Applications, and Countermeasures, Wiley (2016). https://doi.org/10.1002/9781119081715
11. Cabuk, S., Brodley, C.E., Shields, C.: IP covert timing channels: design and detection. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 178–187 (2004). https://doi.org/10.1145/1030083.1030108
12. Berk, V., Giani, A., Cybenko, G., Hanover, N.: Detection of covert channel encoding in network packet delays. Rapport technique TR536, de lUniversité de Dartmouth, p. 19 (2005)
13. Coleman, T.P., Kiyavash, N.: Sparse graph codes and practical decoding algorithms for communicating over packet timings in networks. In: 42nd Annual Conference on Information Sciences and Systems, CISS 2008, pp. 447–452 (2008). https://doi.org/10.1109/CISS.2008.4558568
14. Yao, S., Yang, W., Liusheng, H.: Concealed in web surfing: behavior-based covert channels in HTTP. J. Netw. Comput. Appl. **101**, 83–95 (2018). https://doi.org/10.1016/j.jnca.2017.10.019
15. Chen, A. et al.: Detecting covert timing channels with time-deterministic replay. In: 11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14) (2014)
16. Beyrami, B., Dehghani, M., Saleh Esfahani, M.: Covert timing channel detection based on statistical methods. J. Electron. Cyber Defence **2**(5), 13–24 (2014). (**(in Persian)**)
17. Kiyavash, N., Coleman, T.: Covert timing channels codes for communication over interactive traffic. In: ICASSP IEEE International Conference on Acoustics, Speech and Signal processing, pp. 1485–1488 (2009). https://doi.org/10.1109/ICASSP.2009.4959876
18. Cabuk, S., Brodley, C.E., Shields, C.: IP covert channel detection. ACM Trans. Inf. Syst. Secur. (TISSEC) **12**(4), 22 (2009). https://doi.org/10.1145/1513601.1513604
19. Nasseralfoghara, M., Hamidi, H.: Web covert timing channel detection based on entropy. Electron. Cyber Defense **8**(3), 13–23 (2021). (**in Persian**)
20. Nasseralfoghara, M., Hamidi, H.: Entropy-based analyzing anomaly WEB traffic. High Speed Netw. **26**(4), 255–266 (2020)
21. Brown, E., Yuan, B., Johnson, D., Lutz, P.: Covert channels in the HTTP network protocol: channel characterization and detecting man-in-the-middle attacks. In: International Conference on Cyber Warfare and Security, p. 56 (2010)
22. Gianvecchio, S., Wang, H., Wijesekera, D., Jajodia, S.: Model-based covert timing channels: automated modeling and evasion. In: International Workshop on Recent Advances in Intrusion Detection, pp. 211–230 (2008). https://doi.org/10.1007/978-3-540-87403-4_12
23. Coleman, T.P., Kiyavash, N.: Practical codes for queueing channels: an algebraic, state-space, message-passing approach. In: Information Theory Workshop, ITW'08, IEEE, pp. 318–322 (2008). https://doi.org/10.1109/ITW.2008.4578677

24. Liu, Y., Ghosal, D., Armknecht, F., Sadeghi, A.-R., Schulz, S., Katzenbeisser, S.: Hide and seek in time—robust covert timing channels. In: European Symposium on Research in Computer Security, pp. 120–135 (2009). https://doi.org/10.1007/978-3-642-04444-1_8
25. Liu, Y., Ghosal, D., Armknecht, F., Sadeghi, A.-R., Schulz, S., Katzenbeisser, S.: Robust and undetectable steganographic timing channels for iid traffic. In: International Workshop on Information Hiding, pp. 193--207 (2010). https://doi.org/10.1007/978-3-642-16435-4_15
26. Saadati, M., Dehghani, M., Saleh Esfahani, M.: Simulation and evaluation of jitter and packet loss noises influence on covert timing channel performance. J. Electron. Cyber Defence **2**(3), 35–49 (2014). ((**in Persian**))
27. Liu, J., et al.: A detection-resistant covert timing channel based on geometric huffman coding. In: International Conference on Wireless Algorithms, Systems, and Applications, Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94268-1_26
28. Archibald, R., Ghosal, D.: A covert timing channel based on fountain codes. In: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 970–977 (2012). https://doi.org/10.1109/TrustCom.2012.21
29. Ahn, T.S. et al.: Turbo equalization for covert communication in underwater channel. In: Eighth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE (2016). https://doi.org/10.1109/ICUFN.2016.7537071
30. Wang, J. et al.: Implementing a covert timing channel based on mimic function. In: International Conference on Information Security Practice and Experience, Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06320-1_19
31. Han, T.S., Kobayashi, K.: Mathematics of information and coding. Am. Math. Soc. (2007). https://doi.org/10.1090/mmono/203,PMid:17014848
32. Stillman, R.M.: Detecting IP covert timing channels by correlating packet timing with memory content. In: Southeastcon, IEEE, pp. 204–209 (2008). https://doi.org/10.1109/SECON.2008.4494286
33. Liu,G., Zhai, J., Dai, Y., Wang, Z.: Covert timing channel with distribution matching. In: International Conference on Multimedia Information Networking and Security, MINES'09, vol.1, pp. 565–568 (2009). https://doi.org/10.1109/MINES.2009.28
34. Liu, G., Zhai, J., Dai, Y.: Network covert timing channel with distribution matching. Telecommun. Syst. **49**(2), 199–205 (2012). https://doi.org/10.1007/s11235-010-9368-1
35. Zander, S., Armitage, G., Branch, P.: Stealthier inter-packet timing covert channels. Networking **2011**, 458–470 (2011). https://doi.org/10.1007/978-3-642-20757-0_36
36. Walls, R.J., Kothari, K., Wright, M.: Liquid: a detection-resistant covert timing channel based on IPD shaping. Comput. Netw. **55**(6), 1217–1228 (2011). https://doi.org/10.1016/j.comnet.2010.11.007
37. Lee, K.S., Wang, H., Weatherspoon, H.:{PHY} covert channels: can you see the idles? In: 11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14), pp. 173–185 (2014)

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com