Covert Channel Detection Methods

Adrián Ondov, Pavol Helebrandt
Institute of Computer Engineering and Applied Informatics
Slovak University of Technology
Bratislava, Slovakia
adrian.ondov@stuba.sk, pavol.helebrandt@stuba.sk

2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3503-2033-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICETA57911.2022.9974878

***Abstract*—The modern networking world is being exposed to many risks more frequently every day. Most of systems strongly rely on remaining anonymous throughout the whole endpoint exploitation process. Covert channels represent risk since they exploit legitimate communications and network protocols to evade typical filtering. This firewall avoidance sees covert channels frequently used for malicious communication of intruders with systems they compromised, and thus a real threat to network security. While there are commercial tools to safeguard computer networks, novel applications such as automotive connectivity and V2X present new challenges. This paper focuses on the analysis of the recent ways of using covert channels and detecting them, but also on the state-of-the-art possibilities of protection against them. We investigate observing the timing covert channels behavior simulated via injected ICMP traffic into standard network communications. Most importantly, we concentrate on enhancing firewall with detection and prevention of such attack built-in features. The main contribution of the paper is design for detection timing covert channel threats utilizing detection methods based on statistical analysis. These detection methods are combined and implemented in one program as a simple host-based intrusion detection system (HIDS). As a result, the proposed design can analyze and detect timing covert channels, with the addition of taking preventive measures to block any future attempts to breach the security of an end device.**

***Index Terms*—covert channel, statistical analysis, V2X communication**

I. Introduction

Covert channels can pose a potential threat to networking and data transferring. Their main purpose is to infiltrate some network without being noticed, or at least without being too suspicious. Nowadays, this is not considered to be an easy task, as the concept of network security is being modernized nearly every day. Hardware and software created for covert channel detection should therefore be taken into account when attempting to implement a successful and efficient covert channel.

One of the new areas where covert channels can have potentially great impact is connectivity of previously offline systems used in cars. The contemporary increase in number and importance of automotive systems connected to the cellular networks exposes these systems to new threats. These networks are IP based and thus are vulnerable to some of attack vectors known from traditional computer networks. [16] However, automotive applications require reliable and stable connection [15], so detection and prevention of attacks and covert channels is required. Since these systems are IP based, simulations using traditional computer networks facilitate accelerated prototype development and testing to better steer further research in this area.

Identify applicable funding agency here. If none, delete this.

After infiltrating the specific network, the exfiltration of data takes place. There are numerous ways that data can be secretly exfiltrated from a network, mainly by exploiting some of the well-known internet protocols (e.g. HTTP, DNS, and more), which are discussed in the following sections. Alternatively, a covert channel can help the threat actor to create a back door, which allows him to repeatedly enter the victim's computer remotely and much more easily.

On the other hand, nearly every action regarding the execution of a covert channel attack can be traced back, which mitigates the possibility of new attacks in the future. This is achieved with the help of current software solutions, but also with the knowledge gathered from previous attacks. This paper is aimed at detection of covert channels with the statistical analysis to efficiently detect covert channels on the endpoint level.

The rest of the paper is organized as follows. Section II describes covert channels state of the art together with tools and methods for detection of covert channels. Section III investigates the basic features of covert channels, their classification, prerequisites for covert channel to exist and issues in their detection. Design of our custom HIDS program for detection of timing covert channel using statistical analysis is introduced in Section IV, including prototype implementation details. Section V presents methodology used for verification of the proposed methods together with evaluation of test results. Conclusions and possible avenues for future improvements are discussed in VI.

II. Related Works

Since covert channels are considered to be a threat (mainly because of their anonymity), there were numerous studies focused on covert channels. Majority of them works with trends in steganography [2], [6], while trying to cover different fields of current networking, for example WebRTC protocol for audio/video communications [9], or IPv6. [6], [18]. Although secured variants of protocols are gaining popularity, common fields such as *DNS* [13], *TLS* [20] or *HTTP* protocols [5] are also being still used. These works include the execution, but also the detection of covert channels since both of these fields

are required to be properly analyzed to achieve breakthroughs in handling covert channels.

Zhan in [20] also covers detecting covert channels, especially the ones that are using common web services (e.g. Hypertext transfer protocol). Moreover, the paper analyzes *timing covert channels*, which utilize various timing aspects of packet transfer and delivery. [10] Research on covert channels in web communications in HTTP protocol include Caviglione [5] analyzing HTTP tunneling exploits, Cabuk, Brodley and Shields [4] mention *Reverse WWW shell* used to leak information through HTTP GET requests. Possible reason is that HTTP uses TCP/IP stack, and IP has its own vulnerabilities, coverd by [4].

The majority of information used in this paper will be derived from prior resources with the emphasis on detecting HTTP covert channels. Additional sources include technically oriented websites, mainly to express the specific examples of HTTP covert channel exploits. [14]

Tools and Detection Methods

The tools normally used is suitable for securing our networks from various threats. However, the question is - *can they protect our networks from covert channels?* The answer is not that simple, since it vastly depends on the way that security software works internally. *Snort* and *Suricata* are rule-based programs, which are not effective in detecting covert channels (especially the timing ones). The main reason is that Snort and Suricata do not support adding users' custom plugins unless they are added to the actual core code of these programs (e.g. by adding their software patches). Additionally, this kind of software does not support covert channel detection by default, and the rule-making process was created to secure networks differently, ultimately making it not suitable for our purposes. [11]

On the other hand, we have *Zeek* (formerly Bro). It is highly customizable software with the option to add custom plugins in the form of *scripts*. As stated in [11], "Bro passively observes what is happening in the network and reports whatever it sees. Although Bro also works as an IDS, it does so by adding the detection/analyze plugin based on the network traffic report." Zeek, therefore, acts according to its core functions, but also according to custom plugins added by the user. Implementing plugins is also much easier since this can be done separately and without interfering with the source code. Covert channel detection can therefore be achieved by various methods because it is only limited by the knowledge and skills of the user implementing them.

OSSEC is theoretically also capable of covert channel detection, but it uses different approach - *log mining*. It relies on analyzing events that already happened by analyzing logs. Log mining is strongly associated with *log parsing*, which is a process of extracting data from logs and using them in specific functions. Numerous methods are used in log parsing [12], making the whole concept of log mining possible. The same applies to *Samhain Labs*, as it also uses log analysis.

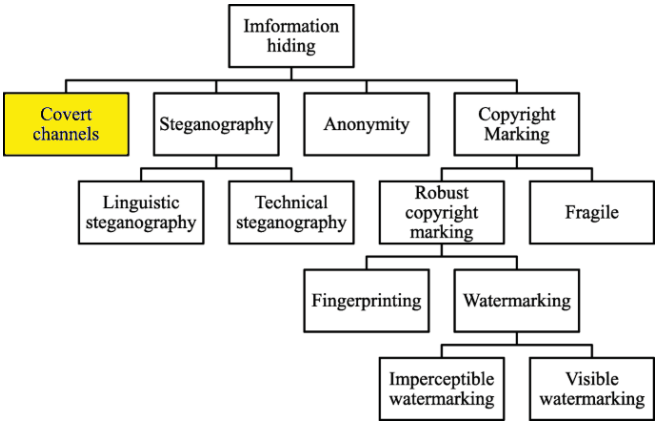


Fig. 1. The information hiding division tree. [17]

III. Covert Channel Characteristics

A. Covert channel definition

First of all, it is important to understand the meaning of the word *channel*. *Channel* can be represented as a connection between two or more devices that are used to transfer data. We can assume that *channel* is just a general medium for communication. However, we should be more interested in the word *covert*. The term *covert* is indicating something hidden or something that should not be discovered. This definition also applies to the whole term *covert channel* - it is a communication channel, which is used in the most undiscoverable manner, to prevent security appliances from detecting it, and possibly stopping its activity. People may think that this task could be easy with the use of modern technologies, but unfortunately, the truth is those network threats (and the assets which can deploy them) are being improved almost at the same pace as the network security is. [7], [8] Additionally, covert channels are a part of wider area of data manipulation called *information hiding* (or *data hiding*). Fig. 1 shows the position of different techniques in the information hiding division tree. Even if the following data hiding methods do not represent the subject of this paper, it is at least important to mention some of them, like *watermarking*, *fingerprinting* (both used for copyright marking), or *steganography*. Covert channels also have their place in this tree, which has been intentionally marked. As we can see in the Fig. 1, covert channels are one of the fundamental information hiding techniques.

B. Covert Channel Detection Issues

There are several problems, which are making covert channel detection difficult. The first one is the *universal accessibility*. With the help of modern technologies, we can establish communication channels anywhere in the world, making the distance between connected devices negligible. The problem is that this can be easily exploited - a variety of devices (which can be accessed remotely) are available to any threat actor.

After universal accessibility, another problem emerges - *communication protocol security*. Many of the protocols which

are considered to be obsolete and (to some degree) dangerous are still being used. A typical example is *Hyper-text transfer protocol*, or HTTP, which is considered to be vulnerable because of the lack of encryption. Data sent via HTTP protocol are unencrypted by default, indicating that methods like the HTTP man-in-the-middle attack are only a question of capturing such data and altering them. HTTP also introduced a lot more different vulnerabilities across its "golden age", namely tunneling attacks (Blind request tunneling, Non-blind request tunneling or Web cache poisoning), Cookie covert channel [14], etc.

Another good example could be the *Domain Name System* protocol, short for DNS. This protocol uses messages named *DNS Query and DNS Response* to translate a specific domain to the IP address associated with it. It could be problematic to remember the IP address of our favorite web page, but fortunately, with the DNS protocol being fully implemented, it is not necessary. All we need to remember is the domain name, which in most cases consists of a familiar name of some web page. Although it is a very useful protocol, it also comes with a portion of vulnerabilities. Probably one of the most common ones is the *DNS tunneling* exploit. [19] It is also considered to be a form of the covert channel since it uses query messages to exfiltrate file data from a victim's device. The whole process is fairly easy to do, we can divide it into four steps:

1) *File conversion* - selected file is converted to a hexadecimal string.
2) *String separation* - newly created string is divided into several smaller strings.
3) *Querying* - smaller strings are then added to the domain name used in the DNS Query message. Several queries are then sent to DNS server (which is also exploited in most cases).
4) *Data recovery* - hexadecimal strings from domain names are retrieved and used to build the desired file, making it available to threat actor.

C. Types of Covert Channels

Covert channels take advantage of many protocols and their downsides to efficiently transfer data. Since several vulnerabilities can be exploited, it is important to have a structure categorizing covert channels into smaller, more specific types. Firstly, covert channels are commonly divided into main categories - *storage* and *timing* covert channels.

Storage covert channels take advantage of different storage spaces in the communication process, whether it is a misused field in the protocol header, or some more dynamic storage spaces, which are used in the data exchange between several protocols. A good example is probably the use of *steganography* to hide data into a legitimate object, e.g. picture, video, or audio file. This file is then used in legitimate network communication without raising any suspicion. Additionally, we can consider the *header alteration* as another form of storage channel (e.g. changing the packet header, modifying the structure of a packet, etc.). [21]

Timing covert channels rely on seemingly unimportant time information to secretly forward data bit by bit. A good example of a timing channel is using the inter-packet arrival time, which is the amount of time elapsed between receiving two consecutive packets. If the time is higher (e.g. more than 100 milliseconds), the receiver will interpret it as a binary 1. On the other hand, when the time is lower (e.g. less than 20 milliseconds), the receiver will interpret it as a binary 0. Even though the throughput of this method is low, the anonymity is very high, making the covert channel efficient enough. [3]

Secondly, judging by the information stated in section III-B, we can assume that covert channels are protocol-dependent, meaning that they behave according to the rules of protocol, which is exploited by the covert channel, therefore we can divide covert channels differently - categorization by associated protocol. Since many protocols can be exploited, we have decided to include only some of the most known protocols or services.

Hyper-text transfer protocol - HTTP (as stated in section III-B) has many different vulnerabilities, plain-text payload being the most dangerous one. Even though we already use *HTTP Secure* (or HTTPS), some websites still rely on basic HTTP. Tunneling and man-in-the-middle attacks can be considered to be the most frequent ones. [14]

Domain name system - DNS (as stated in section III-B) comes with many exploitations as well. Similar to HTTP, tunneling is also a problem in the DNS protocol. [19]

Voice over IP - one of the bigger problems of audio payloads is the *audio steganography* [22]. A small number of bits are hidden in various places within the payload, making them nearly invisible and easy to transfer from one device to another. Steganography can provide a big bandwidth and efficiency to any threat actor since the hidden message is hardly recognizable inside a legitimate payload.

IPv6 - there already are some parts of IPv6 that can be exploited, for example the *Traffic class field* in the IPv6 header, which is susceptible to storage covert channels. However, it can be provided only with a limited amount of values, so the bandwidth is smaller, yet it still can transfer data anonymously. [5]

D. Covert Channel Prerequisites

At this point, we can understand that there are many possible places where any kind of data can be hidden and secretly moved to another device. It is clear that modern types of these threats are developed rapidly, so we should adjust our countermeasures accordingly. Before talking about more recent efficient ways of defending our assets, we would like to describe the actual *characteristics and prerequisites* that covert channels are built upon. The three main ones are *anonymity*, *bobustness* and *bandwidth*.

Anonymity is probably the most important one in any kind of attack, especially in covert channels. Should the covert channel be effective, its anonymity must be ensured. Another term, which is closely associated with anonymity, is *unobservability*

- a covert channel should resemble valid communications as much as possible to prevent any security measures from detecting it. This feature is considered the most difficult one to implement. [5]

Robustness represents the number of alterations that covert channel data can handle before becoming unusable. To successfully exfiltrate data from a device, it must handle the data efficiently and correctly, with minimum of alterations. [5]

Bandwidth indicates the amount of data that can be sent through a channel in a given time, its capacity. [5] Generally, bandwidth is measured in *bits per second* (or any derived values), but as covert channels rely on hiding data in legitimate traffic, are different metrics e.g. *bits per packet*.

IV. Proposed Design of Statistical Analysis Detection Tool

The goal of proposed solution is to design a host-based intrusion detection system utilizing statistical analysis to detect timing covert channels. It is to run as a separate executable file directly on an end device. This custom HIDS shall focus on analyzing the inter packet arrival time.

A. Prototype Implementation

The prototype implementation of the proposed uses three methods suitable for timing covert channel detection [1]. They are *Profiling*, *Regularity test* and *Entropy test*.

Profiling - represented by *Kolomogorov-Smirnov test*, this method is based on calculating the distance between two separate distributions of legitimate and covert data flow. The higher the distance, the higher is the possibility of an exploitation[1]. For example, the K-S test yields a variety of values, which can be used to decide if any deviation occurred. Values include the *critical test value* (labeled as D), or the *rate of significance* (labeled as D_α and the default value for this rate is 0.565).

To properly execute this test, we need to explicitly set the *null hypothesis*[2] that indicates the terms under which the distribution of data flow is *accepted*, or *rejected* otherwise. Moreover, we need to state our *alternative hypothesis* as well, which will be accepted if the null hypothesis is *rejected*. The K-S test statistic is defined as

$$D = max_{1 \leq x \leq N}[F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i)] \qquad (1)$$

where N is equal to a number of provided samples, i represents an index of the specific sample, and finally $F(Y_i)$ is a value given by the specific sample. Based on this equation, we calculate a maximum value from given formulas, which ultimately provides us with a value of D, or *critical test value*. Should the value of D be *higher* than the value of D_α, the null hypothesis is *not accepted*, and therefore, we accept our alternative hypothesis.

[1]This claim highly depends on the initial hypothesis stated before the actual test.

[2]Samples from a legitimate data flow are included when setting a null hypothesis.

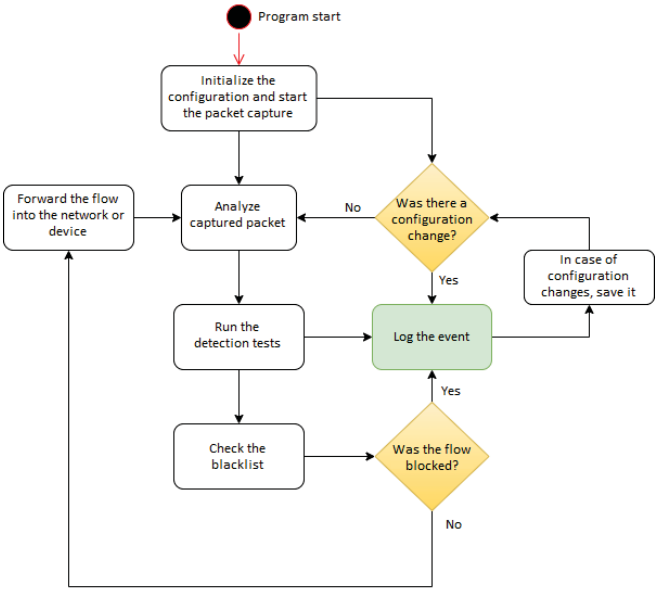


Fig. 2. Event logging configuration and handling logic.

Regularity test - this method works with the term *deviation*, meaning that it monitors the normal behavior in HTTP communication to establish a *network baseline*. If any data flow deviates from the existing baseline, it will be analyzed more intensely, since there will be a suspicion a covert channel is trying to be established. It is still in talks if this method will be implemented since it requires a big data set to establish a correct baseline. This process can take several weeks, so it is a very time-consuming task. On the other hand, it is possible to use values from other tests (profiling and entropy test) to establish a minimal baseline, which should be sufficient enough for this project.

Entropy test - entropy helps us to monitor the randomness of our network. With this method, there is a presumption that our typical HTTP flow has a certain value of entropy, which should be relatively high because the randomness of different timing values in basic HTTP communication is also high. However, if there will be any sign of a pattern forming, the program will be configured to analyze this behavior with higher priority. Moreover, with the ability to define the basic entropy manually, we will be able to observe the entropy differences more closely and with more control.

The prototype provides option of blocking packets from blacklisted flows and logs all events, as can be seen in Fig. 2.

To calculate a value of entropy for a specific data set, we can use the *Shannon Entropy* using formula

$$H(X) = -\sum_{i=1}^{n} P(x_i) * logP(x_i) \qquad (2)$$

where $P(x_i)$ stands for the *probability of the event x happening*, and $logP(x_i)$ represents the *uncertainty of the event x happening*. The sum of these values gives us the final

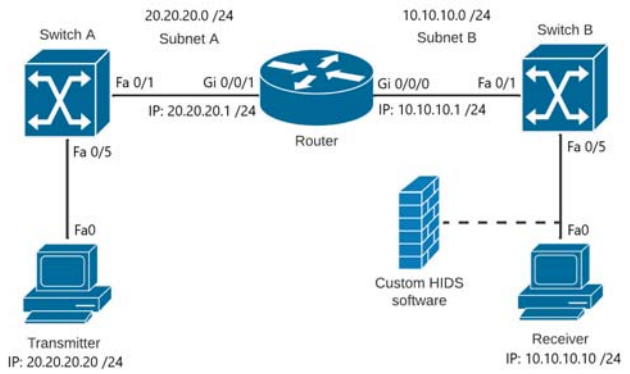


Fig. 3. Primary topology used for evaluation.

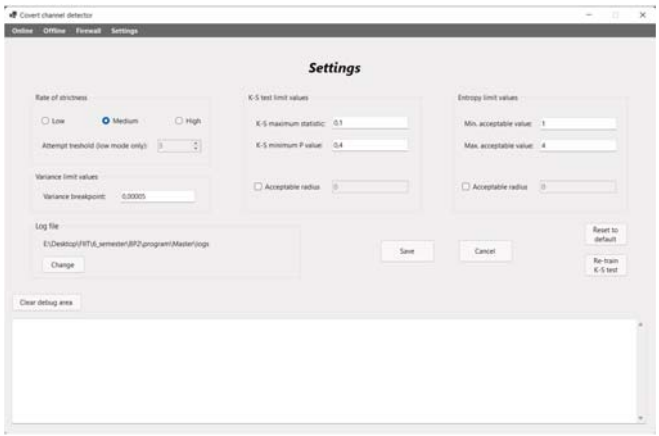


Fig. 4. Program settings configuration screen.

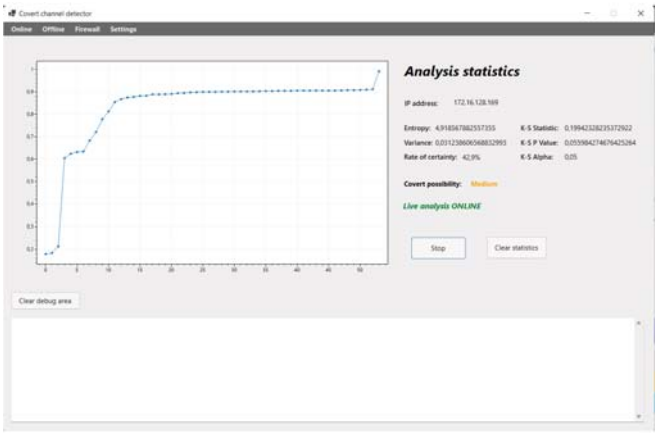


Fig. 5. Online analysis screen

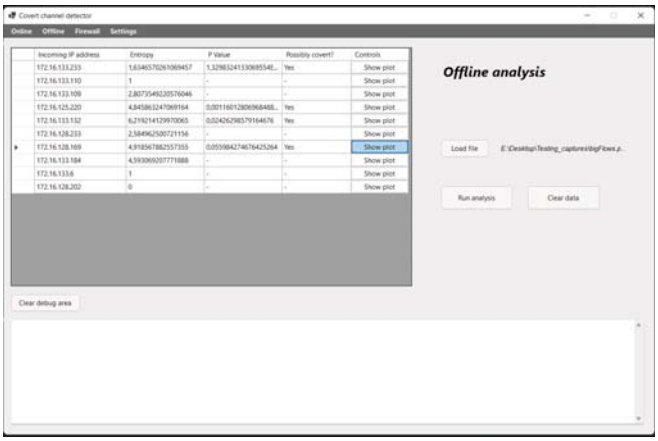


Fig. 6. Offline analysis screen

entropy. The higher the entropy, the higher the uncertainty of data flow timing aspects, which means covert channels should not be present.

Program offers numerous statistics that could be observed. The majority of them should be visible in the program, as it should offer a simple *GUI* with different tabs.

V. Verification

The verification of implemented solution consisted of the analyzing various PCAP files, which contained captured normal and hidden channel communication. To demonstrate that the program can deal with unexpected situations, these hidden channels were modified - either by changing inter packet intervals coding of 0 and 1, or an intentional interruption of covert channel. The testing was done using three topologies with varying number of routers and subnets between communication endpoints. These topologies are based on the primary testing topology, shown in Fig. 3.

Evaluation of the prototype implementation was done with using real time analysis of ICMP messages. Even though this part of the program is only at an early stage, it was able to successfully detect covert channels.

In addition to testing the basic program functionality, additional features were also tested. Among them were direct communication with the Windows firewall, automatic and manual creation of rules in the firewall to block potentially harmful traffic, manual program settings such as specific threshold values that are taken into account during analysis.

During the testing, it was possible to verify all the proposed functions - informing the user about a potential threat, displaying the blacklist and firewall rules. The manual setting of threshold values for statistical analysis was shown to be important for increased precision of detection.

Fig 4 displays settings screen that offers various options for configuration of the program. On this screen, the user can modify several aspects of the program - it is possible to change the path to the folder where the log file will be saved, it is also possible to modify the threshold values for the analysis, or the user can re-train the K-S test repeatedly. Thus, the user has the opportunity to customize the analysis according to his preferences. Alternatively, he can correct the analysis to a certain extent if he suspects that the analysis is not returning correct results.

Fig. 5 depicts an example of online analysis and statistics results of live traffic. On the other hand, offline analysis results from PCAP file can be seen in Fig. 6.

As for the non-functional requirements, the program meets the requirements such as preserving the integrity of the data, as it does not manipulate them in any way. The analysis is reliable and effective, and the program is flexible due to its configurability. In terms of scalability, the program modularity enables future improvements, either the GUI or additional detection methods.

VI. Conclusions

Covert channels are a dangerous tool for data transmission with a low chance of detection. Their detection and analysis is a difficult task that often requires extensive research into the networks and protocols over which they can implemented. Our work is dedicated to the analysis and detection of the presence of hidden channels with an emphasis on the detection of timing covert channels using inter-packet arrival intervals.

The results of the testing were positive, as we obtained a correct detection result in 17 out of 18 cases, which represents a success rate of 94%. The main contribution of this paper was the combination of multiple selected detection methods, thus we pointed out their effectiveness. In addition, we have created a scoring system as a result of the traffic analysis. This provides more information to the administrators to tune blacklisting of flows than just a is or is not a covert channel. We believe that by gradually improving our software, we will be able to make the detection mechanism even more reliable, thereby creating a safer environment for commodity end devices.

Acknowledgements

This publication has been written thanks to the support of the Operational Programme Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems (ITMS code: 313021W404), co-funded by the European Regional Development Fund (ERDF). This work was supported by Cultural and Educational Grant Agency (KEGA) of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the project No. 026TUKE-4/2021.

References

[1] Al-Eidi, S., Darwish, O., Chen, Y.: Covert timing channel analysis either as cyber attacks or confidential applications. Sensors (Basel) **20**(8). https://doi.org/10.3390/s20082417, published 2020 Apr 24.

[2] Ali, B.Q.A., Shahadi, H.I., Kod, M.S., Farhan, H.R.: Covert voip communication based on audio steganography. International Journal of Computing and Digital Systems **11**(1), 821–830 (2022)

[3] 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. p. 178–187. CCS '04, Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1030083.1030108

[4] Cabuk, S., Brodley, C.E., Shields, C.: Ip covert channel detection. ACM Trans. Inf. Syst. Secur. **12**(4) (apr 2009). https://doi.org/10.1145/1513601.1513604

[5] Caviglione, L.: Trends and challenges in network covert channels countermeasures. Applied Sciences **11**(4), 1641 (2 2021). https://doi.org/10.3390/app11041641

[6] Caviglione, L., Schaffhauser, A., Zuppelli, M., Mazurczyk, W.: Ipv6cc: Ipv6 covert channels for testing networks against stegomalware and data exfiltration. SoftwareX **17**, 100975 (2022)

[7] Choi, M.k., Robles, R.J., Hong, C.h., Kim, T.h.: Wireless network security: Vulnerabilities, threats and countermeasures. International Journal of Multimedia and Ubiquitous Engineering **3**(3), 77–86 (2008)

[8] Dastres, R., Soori, M.: A Review in Recent Development of Network Threats and Security Measures. International Journal of Information Sciences and Computer Engineering (Feb 2021)

[9] Figueira, G., Barradas, D., Santos, N.: Stegozoa: Enhancing webrtc covert channels with video steganography for internet censorship circumvention (2022)

[10] Goher, S.Z., Javed, B., Saqib, N.A.: Covert channel detection: A survey based analysis. In: High Capacity Optical Networks and Emerging/Enabling Technologies. pp. 057–065 (2012). https://doi.org/10.1109/HONET.2012.6421435

[11] Gunadi, H., Zander, S.: Comparison of ids suitability for covert channels detection. Tech. rep., Technical Report. Murdoch University. https://www. semanticscholar. org … (2017)

[12] He, P., Zhu, J., He, S., Li, J., Lyu, M.R.: An evaluation study on log parsing and its use in log mining. In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 654–661 (2016). https://doi.org/10.1109/DSN.2016.66

[13] Heinz, C., Zuppelli, M., Caviglione, L.: Covert channels in transport layer security: Performance and security assessment

[14] Huba, W., Yuan, B., Johnson, D., Lutz, P.: A http cookie covert channel. pp. 133–136 (11 2011). https://doi.org/10.1145/2070425.2070447

[15] Janeba, M., Lehoczký, P., Galinski, M., Milesich, T., Danko, J., Kotuliak, I.: Evaluation of lte and 5g qualitative parameters for v2x use cases. In: 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 165–169 (2022). https://doi.org/10.1109/ZINC55034.2022.9840619

[16] Khan, R., Kumar, P., Jayakody, D.N.K., Liyanage, M.: A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions. IEEE Communications Surveys & Tutorials **22**(1), 196–248 (2020). https://doi.org/10.1109/COMST.2019.2933899

[17] Liu, H., Lee, C.: High-capacity reversible image steganography based on pixel value ordering. EURASIP Journal on Image and Video Processing (54) (2019). https://doi.org/https://doi.org/10.1186/s13640-019-0458-z

[18] Lucena, N.B., Lewandowski, G., Chapin, S.J.: Covert channels in ipv6. In: Danezis, G., Martin, D. (eds.) Privacy Enhancing Technologies. pp. 147–166. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[19] Steadman, J., Scott-Hayward, S.: Dnsxd: Detecting data exfiltration over dns. In: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). pp. 1–6 (2018). https://doi.org/10.1109/NFV-SDN.2018.8725640

[20] Zhan, M., Li, Y., Yu, G., Li, B., Wang, W.: Detecting dns over https based data exfiltration. Computer Networks p. 108919 (2022)

[21] Zhang, Q., Zhang, X., Xue, Y., Hu, J.: A stealthy covert storage channel for asymmetric surveillance volte endpoints. Future Generation Computer Systems **102**, 472–480 (2020). https://doi.org/https://doi.org/10.1016/j.future.2019.08.016

[22] Özer, H., Sankur, B., Memon, N., İsmail Avcıbaş: Detection of audio covert channels using statistical footprints of hidden messages. Digital Signal Processing **16**(4), 389–401 (2006). https://doi.org/https://doi.org/10.1016/j.dsp.2005.12.001

ICETA 2022

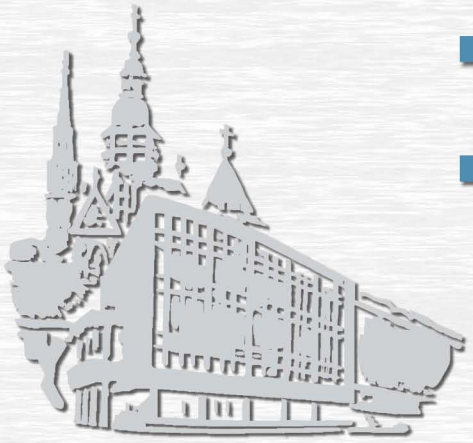
20th Anniversary of IEEE International Conference on Emerging eLearning Technologies and Applications

PROCEEDINGS

October 20 – 21, 2022

Grand Hotel Starý Smokovec, High Tatras, Slovakia

2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3503-2033-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICETA57911.2022.9974883



ICETA 2022

International Conference

20th Anniversary of IEEE International Conference on Emerging eLearning Technologies and Applications

IEEE Czechoslovakia Section

PROCEEDINGS

Information and Communication Technologies in Learning

October 20-21, 2022
Grand Hotel Starý Smokovec, High Tatras
Slovakia

2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3503-2033-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICETA57911.2022.9974615

COMMITTEES

Honorary Committee Chair

Kmeť Stanislav, Technical University of Košice

Honorary Committee

Molnár Ľudovít, Slovak University of Technology, Bratislava
Newman B. Harvey, California Institute of Technology, Pasadena
Rudas Imre, Óbuda University, Budapest

Program Committee Chair

Jakab František, Technical University of Košice

Program Committee

Babič František, Technical University of Košice
Bauer Pavol, Delft University of Technology
Bederka Andrej, Slovak National Coalition for Digital Skills and Jobs, Bratislava
Behun Marcel, Technical University of Kosice
Berke József, John von Neumann Computer Society, Budapest
Bieliková Mária, Slovak University of Technology, Bratislava
Bours Patrick, Norvwegian University of Science and Technology
Brestenská Beáta, Comenius University Bratislava
Cehlar Michal, Technical University of Košice
Chovanec Martin, Technical University of Kosice
Čičák Pavol, Slovak University of Technology, Bratislava
Čižmár Anton, Technical University of Kosice
Dado Milan, University of Žilina
Doboš Ľubomír, Technical University of Košice
Dolnák Ivan, University of Žilina
Doroshenko Anatoliy, Institute of Software Systems of NASU, Kiev
Drozdová Matilda, University of Žilina
Dzupka Peter, Technical University of Košice
Feciľak Peter, Technical University of Kosice
Galgoci Gabriel, AT&T Bratislava
Gavurova Beata, Technical University of Košice
Genči Ján, Technical University of Košice
Haľama Marek, Technical University of Košice
Hämäläinen Timo, University of Jyväskylä
Hautamaki Jari, University of Appl. Science
Hluchý Ladislav, Slovak Academy of Sciences, Institute of Informatics
Horváth Pavol, SANET - Slovak Academic Network, Bratislava
Huba Mikuláš, Slovak University of Technology, Bratislava
Hudak Radoslav, Technical University of Košice
Hvorecký Jozef, Vysoká škola technická a ekonomická in České Budejovice

2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3503-2033-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICETA57911.2022.9974915

Jelemenská Katarína, Slovak University of Technology, Bratislava
Juhár Jozef, Technical University of Košice
Juhas Gabriel, Slovak University of Technology in Bratislava
Kainz Ondrej, Technical University of Košice
Kess Pekka, University of Oulu
Kireš Marian, University of Pavol Jozef Šafárik
Klačková Ivana, University of Žilina
Klimo Martin, University of Žilina
Korshunov Alexander, Kalashnikov Izhevsk State Technical University of the name M.T. Kalashnikov
Kotuliak Ivan, Slovak University of Technology
Kováčiková Tatiana, COST Office, Brusel
Kovács Levente, Óbuda University, Budapest
Kyselovič Ján, Slovak Centre of Scientific and Technical Information, Bratislava
Lavrin Anton, Technical University of Kosice
Lelovsky Mario, Slovak IT Association Bratislava
Levický Dušan, Technical University of Košice
Lumnitzer Ervin, Technical University of Kosice
Mesaroš Peter, Technical University of Košice
Meško Dušan, Jessenius Faculty of Medicine, Martin
Michalko Miroslav, Technical University of Kosice
Modrak Vladimir, Technical University of Kosice
Mulesa Oksana, Uzhorod National University
Mytnyk Mykola, Ternopil National Ivan Pul`uj Technical University
Nakano Hiroshi, Kumamoto University
Pavlik Tomaš, Technical University of Košice
Pietrikova Alena, Technical University of Kosice
Pisutova Katarina, Comenius University
Piteľ Jan, Technical University of Košice
Poruban Jaroslav, Technical University of Košice
Radács László, University of Miskolc
Restivo Maria Teresa, University of Porto
Ristvej Jozef, University of Zilina
Salem Abdel-Badeeh M., Ain Shams University of Cairo
Semanišin Gabriel, University of Pavol Jozef Šafárik
Simonics István, Obuda University, Budapest
Šimšík Dušan, Technical University of Košice
Sinčák Peter, Technical University of Košice
Sobota Branislav, Technical University of Košice
Sovak Pavol, University of Pavol Jozef Šafárik
Stopjakova Viera, Slovak University of Technology in Bratislava
Strémy Maximilián, Slovak University of Technology in Bratislava - Advanced Technologies Research Institute – University Science Park CAMBO, Trnava
Stuchlikova Lubica, Slovak University of Technology
Šveda Dušan, UPJŠ Košice
Szabo Stanislav, Technical University of Košice
Szentirmai László, University of Miskolc
Turna Jan, Comenius University, Bratislava

Usawaga Tsuyoshi, Kumamoto University
Vagan Terziyan, University of Jyvaskyla
Vasková Iveta, Technical University of Košice
Vokorokos Liberios, Technical University of Košice
White Bebo, SLAC National Accelerator Laboratory, Stanford University
Zivcak Jozef, Technical University of Košice
Zolotová Iveta, Technical University of Košice

Organizing Committee Chair

Jakab František, Technical University of Košice

Organizing Committee

Fejedelem Štefan, elfa, s.r.o., Košice, Conference Manager
Rakoci František, elfa, s.r.o., Košice, Webmaster
Zámečníková Iveta, elfa, s.r.o., Košice, Finance Chair

TABLE OF CONTENTS

2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3503-2033-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICETA57911.2022.9974730