A Review on Covert Techniques

Luis Barroso
Department of Computer Science
Faculty of Sciences
University of Porto

Marcelo Santos
Department of Computer Science
Faculty of Sciences
University of Porto

***Abstract*—Due to the widespread adoption of the internet and its services, protocols have been established and new ones arise every year. Either for securing connections, ciphering information or service authentication, protocols place an important part in network communications. The TCP/IP has become one of the ubiquitous protocol suites for secure communication, and therefore, a desirable target for covert information encapsulation. In this article, we will discuss the art of unauthorized data transfer - covert techniques - for data encapsulation in protocol data packets, emphasizing headers fields manipulation.**

I. Introduction

Throughout the years, network transmissions have become more than a local communication between two or more computers. Communications are now, more then ever, spread through large geographical ranges over the Internet by either services or single users. The large amount of data transferred every day imposed the need for protocols to ensure integrity and privacy of such traffic. Covert techniques appeared as an exploit to those protocols.

There was always a need for secret communications, either by corporations trying to protect their latest products or governmental agencies exchanging classified information.

Covert communications appear in scenarios of hidden communication. Suppose a situation of corporate espionage or a journalist wishing to send secret information through censorship channels. Some agencies restrict information exchange using external means, such as pen drives or applications such as skype and it is in this scenario that covert techniques appear the most.

Covert channels perception was originally acquainted in [3], although newer definitions have now arisen. A covert channel can be defined as a mean to transfer information hidden in a communication channel without any awareness of such transmission. Contrary to criptography - where we wish to hide the content of the communication - covert techniques aim to hide the very existence of such communication. Sometimes, a covert channel can be defined as a type of network steganography. However, they are not the same, and so it is important to distinguish them. While steganography is defined as the method to conceal the fact that a communication is being made, for instance hiding content in an audio, video or text file, a covert channel requires exploiting a protocol in order to send hidden messages, for example, using TCP/IP optional header bits to send a covered message. [8]

To further understand the content of this paper, we must also describe overt channels. An overt channel is defined as an open communication path, within a computer system or network, designed for transferring authorized data.

In the following sections we will present and analyze a set of covert techniques and some known applications, sections II and III, respectively . In section IV, we will mention some countermeasures in order to protect a system from potentially harmful covert channels usage.

II. Covert Techniques

In this section we will highlight some techniques of how undisclosed communications can be embedded in overt channels. Only a few examples will be exposed, and its selection was based on their usage and exploit frequency.

A. Unused Header Bits

By exploiting protocols, such as TCP/IP, it is possible to encode a covert channel using reserved or unused bits of their headers, as proven in [6]. If there is no confirmation on the receiver or the protocol specifications do not impose explicit values, hidden data can be transmitted, (e.g. in "type of service" field of the IP header). In figure 1 we can see TCP/IP header and their exploitable fields, marked as underlined.

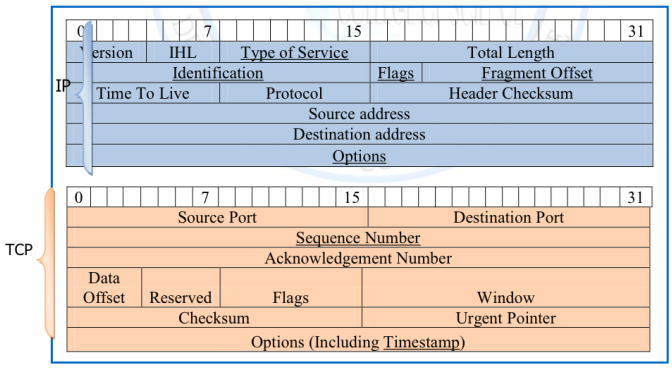

Fig. 1. TCP/IP Header Structure (from [5])

Another possible exploit regards padding bits. Again, if the no specific value is imposed for padding, information can be covert within those bits.

B. Optional Header Fields

In spite of the usage of predefined header extensions regarding discretionary information transport on requisition, several protocols consent unheralded data to be carried in header

extensions in order to augment the proficiency of protocols. One simple example is to covert masked data as an IP address in route record option.

C. Semantic Overloading of Header Fields

A different approach regarding covert techniques is the semantic overloading. It consists of exploiting syntactic variations of the overt channel to encode covert data whilst the channel is maintained semantically identical. For example, hidden content can be encoded using TCP sequence numbers in TCP header. In order to do it, the client chooses the ISN (Initial Sequence Number), and it should be carefully chosen to prevent new incarnations sequence numbers to overlap with the ISNs previous ones. One example of a covert channel created in these circumstances is the use of each ISNs most significant byte, while enforcing the remainders to be set as zero , as proven in [8]. Higher layer protocols, mainly text-based ones, like Hypertext Transfer Protocol (HTTP), offer further opportunities. By simply varying the use of upper and lower case, or the amount of spaces interleaving words, covert channel can be created.

D. Packet and Message Sequence Timing

Another technique relies on sequence timing. To establish a covert channel, in every time interval the sender adjusts its packet rate, while on the receiver's side, in order to decode the concealed data, he needs to measure the rate of the packets in each time interval. However, packet timing channels required synchronization mechanisms at both, sender and receiver sides, in order to alter the packet rate and obtain proper readings at the destination.

E. Payload Tunneling

This technique consists of using the payload tunneling one protocol into another. The major goal of this approach is to bypass firewalls responsible for restraining outgoing transmissions to a brief set of authorized application protocols, such as HTTP.

Such methods can even be applied to Domain Name Server (DNS), tunneling information through the protocol. In this case, the client would request a name resolutions for the host in the form of `host.covertserver.com`, where `covertserver.com` would be a modified DNS server participating in the covert channel, and `host` would be encoded covert data. All the covert information would be sent from the DNS server to the client in the DNS responses as text records.

III. Applications

In this section we cover some practical implementations that successfully transmitted data in a covert channel exploitation.

A. Covert Communication with skype

Skype is currently one of the most used P2P communication systems with a number of users of around 35 million [9]. Using the IP protocol, skype communication is made in an high cryptographic manner. Such communication aims to guarantee privacy for skype users, but it also covers said communication from firewalls as they usually do not verify ciphered contents and therefore creates an ideal ambient for covert communication.

Skype uses the UDP protocol for communications and, as as many other protocols, it is susceptible to covert techniques, such as network covert storage channels through packet field manipulation.

In [4], the authors successfully used skype's 70 bit packets, that do not carry speech, to conceal communications successfully. Such exploit is due to skype's method of transmitting data. Even though no dialog is being performed, like text or audio communication, skype continuously send data packets during the session time, as explained in [9].

B. Covert Communication in Social Networks

With facebook's acceptance and usage spreading worldwide, it became a craved target for covert channels.

Such exploitation was performed in [7]. In this article, the authors have successfully implemented means to use social networks as a pipe for covert communications, specifically targeting facebook.

The authors created an application, named FaceCat which operates based on users facebook accounts. Firstly, the software reaches for long-term cookies stored in cache. After successfully retrieved said cookies, the software starts to operate as an authenticated facebook user.

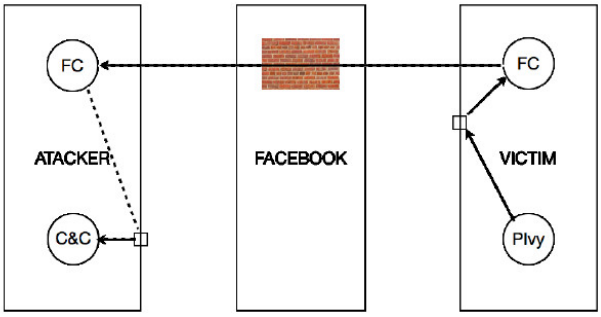

Fig. 2. FaceCat operation method (from [7])

By manipulating cookies, a TCP session is established between the master node wall, the "attacker", and unrelated account walls, the "victims". The master node writes its own cookie on his wall, and awaits for connections. FaceCat is now able to read the cookie and, using a method, which authors called "pass-the-cookie", gain the ability to write on master's wall.

Communications works using `Base64` encoding as well as sequence numbers. The authors choose facebook's mobile interface as it was easier to parse and obtain the encoded messages and use PoisonIvy as a Remote Administration Tool (RAT), with the purpose of interpreting concealed data.

C. Covert Communication in TCP/IP

We must start this discussion by firstly introducing some concepts regarding TCP/IP. TCP/IP is a computer networking

model and a set of communication protocols widely used on the Internet. It is composed by the TCP protocol for reliable communication and IP protocol for routing functions. The protocol's header is the combination of both, TCP and IP.

In [6], Craig H. Rowland successfully implemented undisclosed communications using TCP/IP packet headers, adopting three different approaches.

a) Manipulation of the IP Identification Field: TCP/IP uses the IP identification field to reassemble packet ordering at the destination node. If - by some reason - a packet was to be lost along the way, the destination router would be aware of the lack of such packet and could not reconstruct data accurately until a retransmission of the packet would be received. By using a simple method of placing the ASCII representation of the characters he wished to encode in the identification field, Rowland managed to pass the word "HELLO" hidden, being subsequently reconstructed at the destination node. This method consists of having the client host to build a packet with the correct destination host, encoded IP ID field and data regarding the source host. The remote host, while listening on a passive socket, receives the packet and decodes the information.

Although effective, this implementation is easily detectable by firewalls and there is a high probability of losing data due to the need of packet overwriting by routers (TTL for example).

b) Initial Sequence Number Field: The second approach taken by the author consists in modifying the Initial Sequence Number Field. This field is used in the three-way handshake implemented by TCP in order to establish a reliable protocol negotiation with a remote server. It comes as an ideal field to conceal communications as it has a reserved 32 bit size. As in the previous example, the author encapsulates ASCII coding that refer to a given character in this field. They define the communication as a synchronized communication and encapsulate the ASCII code, taking in account the generation of more realistic sequence numbers through divisions.

c) The TCP Acknowledge Sequence Number Field "Bounce": Finally, the author refers to a third and last method entitled as The TCP Acknowledge Sequence Number Field "Bounce". In this method, the author uses basic IP spoofing (packet manipulation in order to forge the sender IP address) and bouncing technique (using IP spoofing, a packet is sent to a given server that then replies with an ACK/SYN with ISN + 1). Basically, a packet is created with forged source IP address, port, destination IP address (the target system), destination port and a TCP SYN number forget with the data they wish to transmit. Then, it is sent to a bounce server that receives the packet, increments ISN by one number and replies to the forget IP address in the packet. The receiver system expects communication from the bounce server and when received, it interprets the ISN number minus one and therefore the ASCII value of the character.

IV. Countermeasures

Covert channels can be created in many different ways, as seen previously in section II, therefore the first action against these channels is to identify them. Such identification can be obtained using ad hoc strategies or based on formal methods, either in single host systems or network protocols, as referred and explored in [8].

Covert channels existence is mostly due to imperfections, namely design oversights and intrinsic system flaws. Although a covert channel generated from oversights can be revised once noticed, those stemmed from system weaknesses imply remodeling the system in order to eliminate them. Hence, covert channels should be preferably detected and disposed throughout the design phase.

If a covert channel cannot be disposed, in case it was not correctly identified during the design phase and posterior measures would imply system's inefficiency, the capacity of the channel should be reduced. Such reduction depends on the exposure of the system, since restraining the capacity means delaying the system mechanisms or enforcing noise, leading to aggravated performance.

If a covert channel cannot be retained it should be audited, imposing some level of prevention to channel's usage. Whenever covert channel's capacity is too low to be considered relevant, or the channel cannot be audited, it should be documented, allowing people to know its existence and possible hazards.

A. Eliminating Covert Channels

In order to eliminate covert channels the general approaches are:

- Host Security
- Network Security
- Traffic Normalization

Despite the existence of different procedures, regarding covert channels, they cannot/shall not be replaced with each other, since their applicability and outcome differ.

While host security may prevent the channel's exploitation in some situations, protecting hosts from directed attacks, it cannot remove covert network channels. For example, and citing from [8], *If hosts are secured from being hacked, hackers cannot exploit covert channels. However, this does not protect against data ex-filtration by insiders, nor does it solve the covert channel problem in other scenarios such as censorship circumvention*.

However, with network security is is possible to resist to tunneling channels by blocking susceptible protocols. One example, shown in [8], is the firewall blockage to Internet Control Message Protocol (ICMP). Nonetheless, there are protocols in the Internet that cannot be blocked, given their importance. In these cases the action would be replacing them by improved versions with limited occurrences of covert channels.

In regard to traffic normalization, by normalizing protocol headers, header extensions and padding, can mitigate simple storage covert channels, as demonstrated in [1].

B. Limiting Covert Channel Capacity

As aforementioned, when a covert channel cannot be eliminated, the consequent procedure is limiting its capacity.

Such limitation can achieved in several different manners, like limiting the allowed host-to-host connections, setting a fixed size for packets, introducing noise to the channel by inserting dummy packets in the network or buffering and delaying packets, among others proposed in [8], [2]. Despite being effective to specific exploitation, some methods are only applicable in closed networks, due to the inherent efficiency reduction.

C. Auditing Covert Channel

In order to audit effectively a channel, one needs to know its standard behavior to correctly identify when it is being exploited or used in an irregular manner. However, some variations can, sometimes be hard to identify, when the covert channel is identical to the regular use of the protocol.

Some covert channels are based on nonstandard behavior of protocols, such as using reserved bits and padding as mean to communicate, thus making it easy to identify such channels, since some protocols mandate specific values to be filled in these places.

Depending on the type on exploit performed, there are several proposals to audit covert channels, by analyzing traffic rate over time changes or inspecting packet inter-arrival time distribution. Further methods and details can be seen in [8].

V. Conclusion

As covert techniques are fairly new, there is still some work to be done regarding it's detection and prevention. As long as new protocols continue to burst, new covert techniques are sure to tag along. Since countermeasures are still rough, we strongly believe that covert techniques have all the means to continue growing and so, data confidentiallity may be at risk. Furthermore some investment should be done in security software such as firewalls or anti-virus in order to be able to analyse network traffic and protocol integrity and effectively prevent unauthorized data transmitions.

References

[1] Mark Handley, Vern Paxson, and Christian Kreibich. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In *USENIX Security Symposium*, pages 115–131, 2001.

[2] Myong H Kang, Ira S Moskowitz, and Stanley Chincheck. The pump: A decade of covert fun. In *Computer Security Applications Conference, 21st Annual*, pages 7–pp. IEEE, 2005.

[3] Butler W. Lampson. A note on the confinement problem. *Commun. ACM*, 16(10):613–615, 1973.

[4] Wojciech Mazurczyk, Maciej Karas, and Krzysztof Szczypiorski. Skyde: A skype-based steganographic method. *arXiv preprint arXiv:1301.3632*, 2013.

[5] Asst Prof Dr Ziyad Tariq Mustafa and Authman Waleed Khalid. Packet steganography using ip id.

[6] Craig H. Rowland. Covert channels in the tcp/ip protocol suite. *First Monday*, 2(5), 1997.

[7] Jose Selvi. Covert channels over social networks. In *SANS Institute Reading Room site.* SANS Institute, 2012.

[8] Sebastian Zander, Grenville Armitage, and Philip Branch. A survey of covert channels and countermeasures in computer network protocols. *Communications Surveys & Tutorials, IEEE*, 9(3):44–57, 2007.

[9] Jiangtao Zhai, Mingqian Wang, Guangjie Liu, and Yuewei Dai. Skylen: a skype-based length covert channel.