An Educational Network Protocol for Covert Channel Analysis Using Patterns

Steffen Wendzel[1] and Wojciech Mazurczyk[2]

[1] Hochschule Worms, Germany and Fraunhofer FKIE, Germany
[2] Warsaw University of Technology, Poland and University of Hagen, Germany

Several papers describe data hiding methods to create network covert channels. However, no work explains the actual process of analyzing a network protocol for potential covert channels in the context of recently introduced hiding patterns. This paper fills this gap by providing a pattern-based method for teaching network covert channel analysis with an educational network protocol.

The Core Idea in a Nutshell

We define a simple network protocol called the *Covert Channel Educational Analysis Protocol* (CCEAP, see Figure 1). CCEAP can be used in didactic environments. With our protocol, we lower the barrier for understanding network covert channels by eliminating the requirement for students to understand several network protocols in advance. In addition, we reduce the number of hiding methods that students need to understand to capture the full spectrum of hiding methods. CCEAP aims to serve as both, an assistance for teaching and for learning.

Background: Network Covert Channels

Covert channels are communication channels that can be used to perform stealthy network communications.

Typical scenarios for their application com-prise the exfiltration of confidential data or the hidden transfer of C&C data for botnets.

Key Related Work

A first approach for a systematic covert channel education was recently presented by Zseby *et al.* in *A Network Steganography Lab on Detecting TCP/IP Covert Channels* (IEEE Trans. Education, 59(3), 2016).

The concept of hiding patterns was introduced by Wendzel *et al.* in *Pattern-based Survey and Cate-gorization of Network Covert Channels* (ACM Computing Surveys, 47(3), 2015).

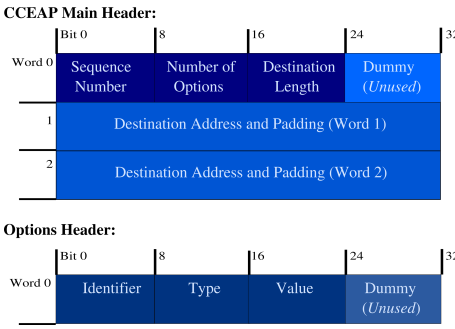

Fig. 1: The CCEAP main and options header.

What are Hiding Patterns?

Hiding patterns describe the abstract *core idea* of an information hiding *method* (also known as hiding *technique*). While more than hundred hiding methods are known, earlier work has shown that these hiding methods can be broken down into only 14 hiding patterns. If students learn these hiding patterns, they automatically understand the core concepts of all known hiding methods.

Being still in an early stage, CCEAP can already be used to create most of the known hiding patterns for network covert channels (Table 1).

Hiding pattern	Explanation	CCEAP realization
Patterns for storage channels		
Size Modulation (●)	Size of a PDU or particular header element is modulated	Modulation of address field length or of number of options
Sequence Modulation (●)	Alternation of header field order in a PDU	Alternation of order in which n options are encapsulated into the payload
Add Redundancy (●)	Creation of space within a PDU/PDU field by addition of redundant bits	Adding a new option of type X
Random Value (●)	Modification of a random value field	Filling the initial sequence number field with covert data
Value Modulation (●)	Selection of 1 out of n possible values for a header field	Change the case of ASCII letters used for the destination address (e.g. address 'AAaAa' could indicate the hidden message '11010')
Reserved/Unused (●)	Utilization of reserved or unused bits in header fields	Overwriting the reserved field 'dummy' in the main header.
Patterns for timing channels		
Artificial Message/Pkt Loss (●)	Artificial loss of network packets	Messages are sent with incremental sequence values. The user can force the program to exclude a selected sequence number, indicating a message loss.
Artificial Retransmission (●)	Introduction of artificial message duplications	Similarly as in case of Artificial Message Loss, the user can select the sequence number of a packet to be duplicated and that will thus be sent twice.
Manipulated Message Ordering (●)	Modulation of messages order	Define the order of sequence numbers in which messages are sent (instead of the default use of incremental sequence numbers)
Rate/Throughput (○)	Modulation of traffic rate / throughput	Modulation of several inter-arrival times as bursts (protocol-agnostic timing channel)
Interpacket Times (○)	Modulation of inter-arrival times between packets	Modulation of inter-arrival times between packets (protocol-agnostic timing channel)
Message Sequence Timing (○)	Modulation of messages to be sent during pre-defined time-slots	Message sending via pre-defined inter-arrival times (protocol-agnostic timing channel)
Collision / Frame Timing (●)	Artificial introduction of frame collisions (e.g. in CSMA/CD), followed by a timed re-transmission	Not implemented (requires a low-level protocol, OSI layer ≤ 2)
Temperature (●)	Modulation and measurement of CPU clock skew	Not implemented (requires a low-level CPU clock skew evaluation)

Tab. 1: How CCEAP can create network covert channels based on patterns. ● = protocol-aware hiding pattern (i.e. hiding method must be applied in the context of a particular network protocol), ○ = protocol-agnostic hiding pattern (i.e. independent of a protocol).

Teaching Workflow and Exercises

Figure 2 visualizes the workflow that we envisage for teaching CCEAP with its tool (open source). During the exercises, students need to understand a particular pattern and find a way to create a covert channel based on the pattern with the CCEAP tool. Exercises are given in form of a short sentence after introducing the tool and the patterns, e.g.:

`Find a way to establish a covert channel that uses the pattern X. Verify your approach with the CCEAP tool.'

Sample Exercise (`Value Modulation'): The students are asked to create a covert channel that represents a Value Modulation (Tab. 1). Firstly, the students verify the definition of a Value Modulation pattern, e.g. via available descriptions from the lecture or by using websites and papers.

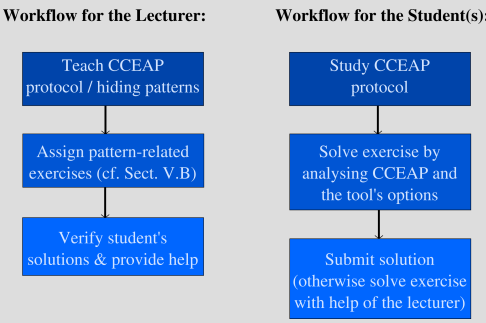

Fig. 2: Teaching workflow for lecturer and student(s).

Secondly, they determine how CCEAP can be used to represent the pattern. A typical scenario for the Value Modulation pattern that is mentioned in the publications is to signal hidden information by alternating between upper and lower case characters in plaintext protocols. CCEAP contains the `Destination Address' field and the students could decide to transfer data either to the address `ABC' or to the address `abc' to signal a covert zero or a one bit.

Fourth, to test their idea, the students check the parameters provided by the CCEAP tool and detect the command line parameter **-d** that specifies the destination address. The students start the server and let the client send two packets to the server, once with the former and once with the latter destination address. On the server, the students can see the output and finally submit the answer.

Selected Further Exercises: Pattern-based Exercises can be reversed and patterns can be combined to exercise their hybrid application.

The CCEAP code and its documentation are available for download: