CCEAP Documentation

Steffen Wendzel[1],[2]

[1]Worms University of Applied Sciences, Worms, Germany / [2]Fraunhofer FKIE, Bonn, Germany / Contact: wendzel (at) hs-worms (dot) de

A note before you start reading: You are currently reading the CCEAP tool documentation. However, we also published academic work on CCEAP:

- S. Wendzel and W. Mazurczyk: An Educational Network Protocol for Covert Channel Analysis Using Patterns (poster), in Proc. ACM Conference on Computer and Communications Security (CCS), pp. 1739-1741, ACM, 2016. You can also download the conference poster here, it summarizes the content of the actual paper.

Please send requests and feedback to the author: Steffen Wendzel, www.wendzel.de (steffen (at) wendzel (dot) de). Research on covert channel teaching is currently performed by Steffen Wendzel and Wojciech Mazurczyk.

Introduction

Basics of Information Hiding

The *Covert Channel Educational Analysis Protocol* (CCEAP) is a network protocol designed for teaching covert channels to professionals and students. It can also be used a simple covert channel traffic generator. For an introduction on network covert channels (network steganography), please have a look at this publication.

Videos: - Quick 4 min Introduction into Information Hiding in Computer Networks - 30 min Introduction into Network Steganography and Hiding Patterns – the foundation of CCEAP

CCEAP in one Paragraph

CCEAP implements a network protocol. The protocol is explicitly vulnerable against several so-called hiding patterns. Hiding patterns represent the core ideas of how secret data can be hidden in network transmissions. The core idea is that CCEAP allows students to model the protocol structure in a way that it represents covert channels. In other words, CCEAP can be used to teach students covert channel techniques in an easy manner.

Why use CCEAP?

The key advantage and unique feature of CCEAP is that it is based on hiding patterns. As shown in (Wendzel et al., 2015), more than hundred known network steganography methods can be represented by only 11 hiding patterns. Hence, it is enough to teach students these few hiding patterns instead of >130 hiding techniques.

Another advantage of CCEAP is that basically all methods for hiding data in network transmissions can be explained using only one protocol (=CCEAP) instead of several network protocols (IP, TCP, UDP, HTTP, ...). For this reason, switching network protocols while explaining hiding methods to students is not necessary anymore. The protocol's structure is simple and self-explanatory and its implementation is kept at a minimum level of code lines to make it especially accessible to students.

Installation

The tool requires `gcc` and Linux (or similar, e.g. Cygwin under Windows). Run `make` to build the two components of the tool: *`client`* and *`server`*.

A First Test-run

Open two terminals. Run `make runser` in one terminal to start the server (alternatively, run `./server -P 9999 -v`) and run `make runcli` in the other terminal (alternatively, `./client -D 127.0.0.1 -P 9999 -v`). You should see some data transmission going on that will transfer a couple of packets and then terminate. The parameter `-v` is just used for verbose output while `-D` and `-P` specify the server IP and the TCP port to use.

Teaching Process in a Nutshell

The educational process is split into two parts. First, a lecturer explains the fundamentals of CCEAP and network steganography to the students. Second, students perform their exercises, which are afterwards evaluated by the teacher.

1. Preparing Students to Use CCEAP

The lecturer is required to introduce

- the CCEAP protocol (see below), and
- hiding patterns as they are described on our project website that also points to several of our recent publications in which we introduce the topic.

Please Note: Currently, CCEAP supports most of the storage channels patterns as well as some timing channel patterns. However, a few hiding patterns as well covert channels for OSI layers 1-2 are not supported. For a detailed explanation of these terms, please see our book Information Hiding in Communication Networks, Chapters 2 and 3.

2. Exercises

In the exercises, students try to create covert channels **or** they try to determine the type (pattern) of a given covert channel. Therefore, the lecturer can perform one of two types of exercises:

- The lecturer can ask the *students to establish a hidden communication that represents a given pattern.* The pattern can be one of the known patterns of the available pattern collection, e.g. Value Modulation or Reserved/Unused. Therefore, the students need to find out how to create a covert channel for the particular pattern using the CCEAP protocol by analyzing the protocol structure and the `client` tool's command line parameters.
- Alternatively, the lecturer can ask the students to *determine the hiding pattern for a given traffic recording or for given tool parameters* that were used to run the client.

Sample exercises can be found in the CCEAP repository under the following URL https://github.com/cdpxe/CCEAP/tree/master/sample_exercises. Also, cf. section *Example Walk-through for an Exercise* in *this file* for an explanation of how exercises are performed.

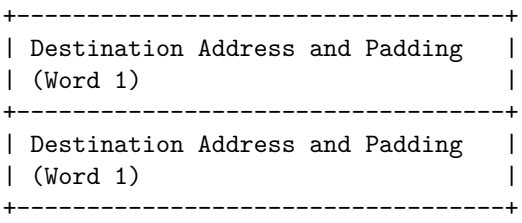
CCEAP Protocol

The protocol contains two components: the CCEAP *main header* and *option headers* (between 0-128). The option headers will be attached to the main header.

Main Header

The main header contains three 32 bit words. The first word contains a *Sequence Number* (8 bit), the indicator of how many *option headers* are attached (*Number of Options*, 8 bits), the length of the destination address (measured in bytes, called *Destination Length*, 8 bits) and an unused *Dummy* field (8 bits). The following two words contain the *Destination Address* (can be an arbitrary ASCII value) that is padded with `X` bytes if too short for two words.

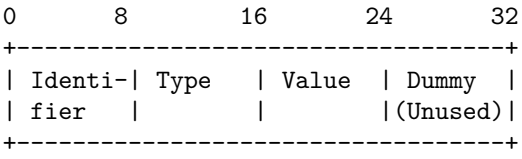
```
0       8        16       24       32
+---------------------------------+
| Seq.No.| Number | Dest.  | Dummy  |
|        | Options| Length |(Unused)|
```



The sequence number is incremental and starts with 1 by default (this can be changed using command line parameters). The *Number of Options* field is zero by default.

Options Header

The options header contains a freely definable *Identifier*, *Type* and *Value* field (8 bits each) and another *Dummy* field (also 8 bit).



Tool Architecture and User-Guide

The implementation of the CCEAP protocol is split into two components, a client and a server. The client transfers CCEAP protocol packets to the server. Users can freely define how the particular fields of the CCEAP header are filled and how the *options headers* are embedded. The server, on the other hand, simply displays the received content which allows to view whether packet data were transmitted in the desired way.

All actions can be performed over the network or on the local host. The easiest way is to open two X terminals (e.g. GNOME Terminal) and start the server in one terminal and the client in the other.

Example Walk-through for an Exercise

Sample exercises (incl. solutions) are available here. However, for now let us assume that the exercise given to the students was to create a covert channel that uses the Reserved/Unused pattern. The students look up the pattern description, which tells:

The covert channel encodes hidden data into a reserved or unused header/PDU element.

By analyzing the protocol structure of CCEAP, the students find an unused header field, namely the *Dummy* field, which is 8 bits in size. The field could be used to place hidden data in the way as described by the pattern. By running `./client -h`, the students check whether they can manipulate the Dummy field in some way:

```
$ ./client -h
CCEAP protocol implementation. Copyright (C) 2016 Steffen Wendzel
...
The following parameters are supported by CCEAP client v.0.4.1:
...
-D x   Destination IP x to connect to
-P x   TCP port x to connect to
...
-u x   Use digit x instead of 0 as `dummy' value in the main header
```

The parameter `-u` can obviously be used for creating a Reserved/Unused pattern-based covert channel. To this end, the students run the server on the localhost port 2222:

```
$ ./server -P 2222
...
CCEAP - Covert Channel Educational Analysis Protocol (Server)
   => version: 0.4.1, written by: Steffen Wendzel, www.wendzel.de
```

Please note that the use of an additional `-v` parameter provides verbose output when used with `server` or `client`. The parameter `-h` works also for both, `server` and `client`.

Next, the students run the client and connect to localhost (127.0.0.1), port 2222. They set the value of the Dummy field to 42:

```
$ ./client -D 127.0.0.1 -P 2222 -u 42
...
connecting ... connected.
sending: ..........
sending done.
```

By default, the client sends 10 packets with incremental sequence number. Therefore, the server will receive and display now 10 packets with the same content. Of course, such parameters can be changed as desired. The typical packet output is shown as follows:

```
received data (12 bytes):
 > time diff to prev pkt: 0.000
 > sequence number:       1
 > destination length:    0
 > dummy value:           42
 > destination + padding: XXXXXXXX
 > number of options:     0
```

More detailed outputs will be provided when the server receives more complex packets from the client, e.g. packets containing options headers.

More examples are available here.

Supported Hiding Patterns

Currently, the below listed hiding patterns are supported by CCEAP.

Storage Channel Patterns:

- Size Modulation
- Sequence Modulation
- Add Redundancy
- Random Value
- Value Modulation
- Reserved/Unused

Timing Channel Patterns:

- Artificial Message/Packet Loss (a.k.a. PDU Corruption)
- Artificial Retransmission
- Manipulated Message Ordering
- Inter-arrival Time
- Rate/Throughput, indirectly supported via the *Interpacket Time* Pattern

An explanation of how each pattern is addressed by CCEAP can moreover be found in our CCS'16 poster (Wendzel and Mazurczyk, 2016), cf. *Further Reading* at the end of the document.

Using CCEAP as a Traffic Generator

The tool can be used easily as a traffic generator, also when the semantics of the actual protocol are not required. For instance, the packet size can be manipulated to simulate packet-size covert channels (Size Modulation pattern). In fact, the author used CCEAP to generate traffic for some of his own papers.

Inofficial add-ons to automatically encode files into inter-arrival times / sequence numbers

With the inofficially provided tool `iat_encode` (encodes an input file into timing values), one can easily create timing channels (running `./client -t ´./iat_encode [input-file] [low-time] [high-time]´`, where the time values must be provided in units of ms. This would then represent the Inter-arrival Time pattern.

Similarly, the Manipulated Message Ordering pattern can be represented using `./seq_encode [input-file] 256 2` with parameter `-s` (instead of `./iat_encode` with parameter `-t`). The value `256` tells the tool the maximum allowed sequence number (in CCEAP, the sequence number field has 8 bits, so we need to use all sequence numbers *mod 256*) and the `2` represents the number of sequence numbers to be swapped (only the value 2 is supported). In other words, there is no need to alter any of these two numbers.

However, *both tools are not officially part of CCEAP and they are not necessary to realize these patterns with CCEAP, i.e. they are just additional tools to make it a bit easier to use CCEAP.*

Further Reading

The following subsections provide a list of papers which are directly linked or related to CCEAP.

Publications about CCEAP:

- S. Wendzel and W. Mazurczyk: An Educational Network Protocol for Covert Channel Analysis Using Patterns (poster), in Proc. ACM Conference on Computer and Communications Security (CCS), pp. 1739-1741, ACM, 2016. You can also download the conference poster here, it summarizes the content of the actual paper.

Publications on Information Hiding Patterns:

- S. Wendzel, S. Zander, B. Fechner, C. Herdin: Pattern-based Survey and Categorization of Network Covert Channel Techniques, ACM Computing Surveys, ACM, 2015. An early version of this document is available for free: download.
- W. Mazurczyk, S. Wendzel, K. Cabaj: Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach, in Proc. Second International Workshop on Criminal Use of Information Hiding (CUING 2018) at ARES, pp. 10:1-10:10, ACM, 2018.
- Chapter 3 of: W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, K. Szczypiorski: Information Hiding in Communication Networks, Wiley IEEE-Press, 2016.
- S. Wendzel, C. Palmer: Creativity in Mind: Evaluating and Maintaining Advances in Network Steganographic Research, Journal of Universal Computer Science (J.UCS), 2015.
- S. Wendzel, W. Mazurczyk, S. Zander: Unified Description for Network Information Hiding Methods, Journal of Universal Computer Science (J.UCS), 2016.

Publications on Fundamentals of Network Covert Channels/Network Steganography:

- W. Mazurczyk, S. Wendzel: Information Hiding – Challenges for Forensic Experts, Communications of the ACM, Vol. 61(1), pp. 86-94, January 2018. Summarizing video
- W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, K. Szczypiorski: Information Hiding in Communication Networks, Wiley IEEE-Press, 2016.
- S. Zander, G. Armitage, P. Branch: A survey of covert channels and countermeasures in computer network protocols, Computing Surveys and Tutorials, IEEE, 2007.
- S. Wendzel, W. Mazurczyk, L. Caviglione, M. Meier: Hidden and Uncontrolled–On the Emergence of Network Steganographic Threats, Proc. ISSE, Springer, 2014.
- J. Lubacz, W. Mazurczyk, K. Szczypiorski: Principles and overview of network steganography, IEEE Comm. Magazine, IEEE, 2014.
- B. Carrara, C. Adams: A Survey and Taxonomy Aimed at the Detection and Measurement of Covert Channels, Proc. 4th ACM Workshop on Information Hiding and Multimedia Security, ACM, 2016.
- C. H. Rowland: Covert channels in the TCP/IP protocol suite, First Monday, Vol. 2(5), 1997.
- S. Wendzel, J. Keller: Low-Attention Forwarding for Mobile Network Covert Channels, Proc. Communications and Multimedia Security, LNCS 7025, pp. 122-133, Springer, 2011.

Websites:

- Network Information Hiding Patterns Project
- Europol EC3 initiative Criminal Use of Information Hiding (CUING)
- TU Vienna Network Covert Channel Lab, cf. T. Zseby, F. Iglesias Vázquez, V. Bernhardt, D. Frkat, R. Annessi: A Network Steganography Lab on Detecting TCP/IP Covert Channels, IEEE Transactions on Education, IEEE; 2016.