NETWORK STEGANOGRAPHY – CAN BEHAVIOURAL TYPES MAKE THE DIFFERENCE?

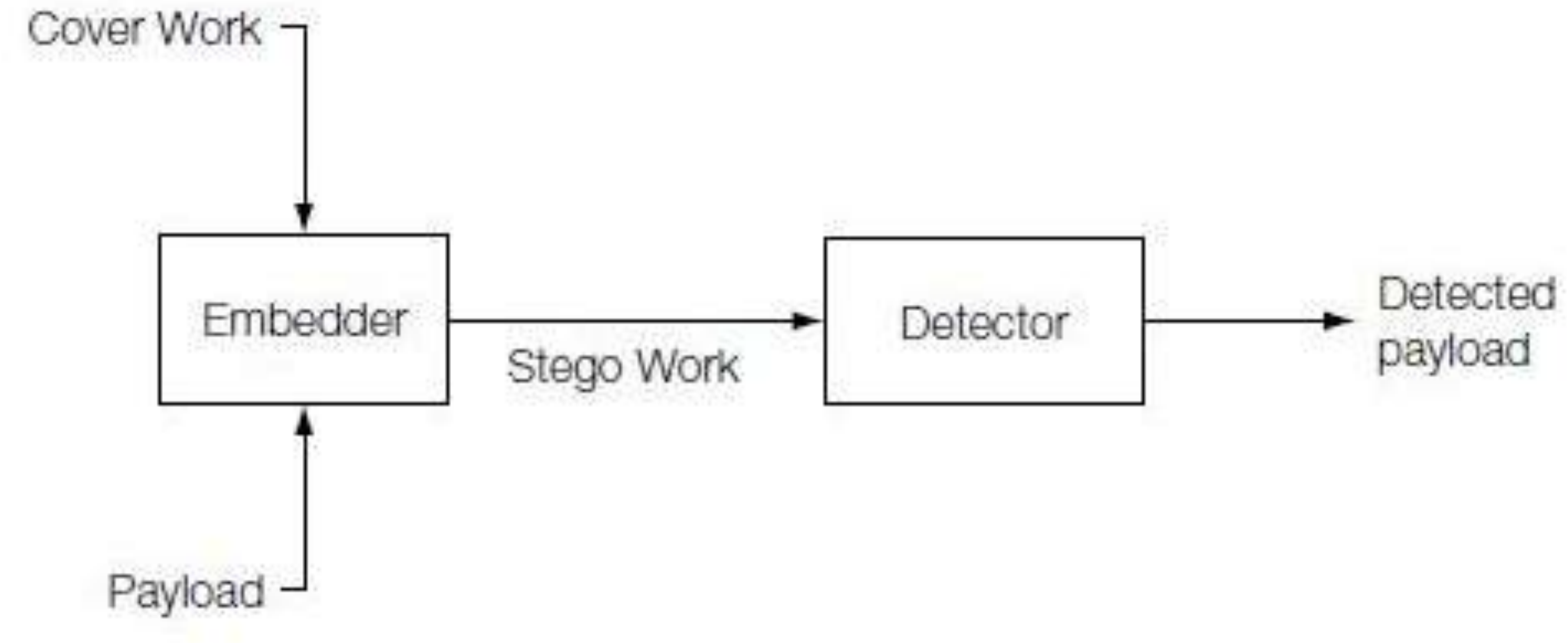
Aleksandra Mileva
Faculty of Computer Science
University “Goce Delčev”
REPUBLIC OF MACEDONIA

COST Action 1201
WG/MC Meeting 17-18 March 2016, Malta

What is steganography?

- Steganography is the art of concealed communication (the practice of **undetectably** altering given carrier to embed a secret message).
  - The very existence of a message is secret.
  - Invisible ink, wax tablet, messenger's body, ...
  - Digital media - picture, text, video, audio, ...
- The original unaltered carrier is sometimes referred to as the **cover Work**, and the obtained altered carrier is known as **stego Work** or **steganogram**.
- A generic steganography system consists of **embedder** and **detector**.
- Possible uses can fall into the category of legal or illicit activity

A generic steganographic system



What is steganography?

The best carrier for secret messages must possess two features:

1. It should be popular, that is, the usage of such a carrier should not itself be considered an anomaly.
2. The steganogram insertion-related modifications of the carrier should not be “visible” to the third party not aware of the steganographic procedure.

What is network steganography?

- **Network steganography** is the art of hiding secret data in legitimate transmissions in communication networks without destroying the used hidden data carrier.
  - data carriers – different network protocols
  - while trying to conceal the presence of hidden data from network devices.
- The best choices of network protocols as a carrier of secret data are the most popular and most used protocols
  - HTTP, IP, TCP, DNS, RTP, DHCP, ARP, ...
- Network-level embedding allows for leakage of information (even very slow) during long periods of time and, if all the exchanged traffic is not captured, then there is nothing left for forensics experts to analyze.

Network steganography methods

Several characteristics of communications are utilized for steganographic methods (Zielinska et al, 2014):

- The communication channel is not perfect—errors are a natural phenomenon
- Most protocols bear some quantity of redundant information
  - Sometimes, different kind of duality is present in some protocols.
- Not every protocol is completely defined, and most of the specifications permit some amount of freedom in implementation, which can be abused

Network steganography methods (Mazurczyk et al, 2008)

- Methods that modify protocol data unit (PDU), including fields with protocol control information from protocol header or/and the protocol payload.
  - examples: reserved, pad or undefined fields and fields with random values, like IP *Identication* or TCP *Initial Sequence Number (ISN)* fields, or in Skype communication by replacing the encrypted silence with secret data bits
- Methods that modify the structure of PDU streams (time-relations between PDUs), by PDU reordering, intentional losses, use of interpacket delays, modification of timestamps, etc.
- Hybrid methods, which involve a combination of previous two types of methods.

Covert channels

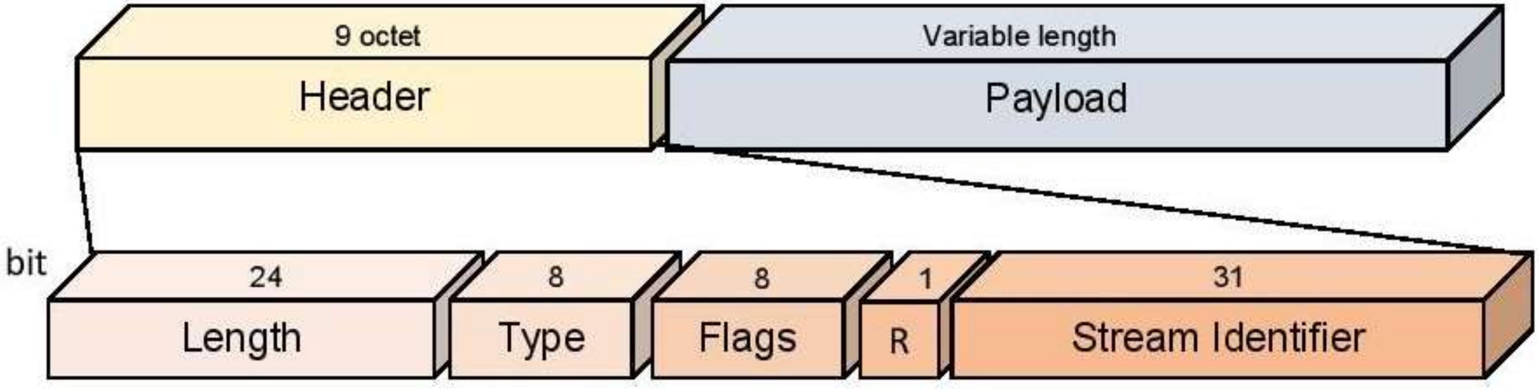
- A **covert channel** is any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy
  - Storage covert channel - usually one process writes (directly or indirectly) to a shared resource, while another process reads from it
  - Timing covert channel is essentially any technique that conveys information by the timing of events, in which case the receiving process needs a clock.
    - Counting channel
    - Active vs passive
  - Noisy vs noiseless
  - Indirect vs direct
  - Payload tunnel

HTTP/2 as an example

- New protocol (RFC 7540, May 2015), build with security in mind, but still, covert channels can be build.
- HTTP/2 is a binary request/response protocol with many improvements and benefits:
  - Multiplexing and concurrency: Several HTTP requests can be sent on the same TCP connection as separate streams, and their responses can be received out of order in the same streams.
  - Server push: If the server has a knowledge that some resources are needed and will be requested later for a given web site, the server can send these resources without a request, and the client will cache the resources till later;
  - Header compression;
  - Stream dependencies and priorities: the client can indicate to the server, which of the streams are more important than the others, and need to be delivered first.

HTTP/2 as an example

- HTTP/2 connection is a TCP connection between client and server which consists of three elements:
  - Stream: A bidirectional flow that carries messages between the endpoints;
  - Message: Logical HTTP message consisting of one or more frames;
  - Frame: The smallest unit of communication that carries the specific type of data. There are 10 types of frames: DATA, HEADERS, PRIORITY, RST STREAM, SETTINGS, PUSH PROMISE, PING, GOAWAY, WINDOWS UPDATE and CONTINUATION.



Some similar HTTP/1.1 covert channels suppressed in HTTP/2

1. HTTP/1.1 treats any amount of consequent linear white space characters (optional line feed [CLRF], spaces [SP] and tabs [HT]) present in the header, in the same way as a single space character (Kwecka Z. 2006), so, for example:
   - [HT] can be a binary one and
   - [SP] can be a binary zero
2. Header names are case-insensitive in the HTTP/1.x, so, one can use different capitalization for the header field values for covert channel (Dyatlov A, Castro S. 2003). Header fields names must be converted to lowercase prior to their encoding in HTTP/2.

Some new HTTP/2 covert channels

(paper in review)

- One can use a protocol feature that has dual nature, i.e., the same feature can be obtained in more than one way; or deployment of feature is not mandatory.

Covert channel 1

- DATA, HEADERS and PUSH PROMISE frames, use padding optionally as a security feature to obscure the size of messages. When padding is used, the third flag, PADDED (0x8), is set to 1, and at the beginning of Frame Payload there is a 8-bit field Pad Length containing the length of the frame padding in units of octets. When no padding is used, there are two representations with the same effect:
  - PADDED flag set to 0, and
  - PADDED flag set to 1, together with Pad Length field set to 0.

Some new HTTP/2 covert channels

(paper in review)

Covert channel 2

- An HTTP request consists of:
  - one HEADERS frame, followed by zero or more CONTINUATION frames, containing the header block
  - zero or more DATA frames containing the payload body
  - optionally, one HEADERS frame, followed by zero or more CONTINUATION frames containing the trailer-part
- DATA and CONTINUATION frames are variable-length sequences of octets and they can be sent in different number!
- So, we can create one covert channel using:
  - odd number of DATA (or CONTINUATION) frames to be binary 1, and
  - even number of DATA (or CONTINUATION) frames to be binary 0.

Some new HTTP/2 covert channels

(paper in review)

Covert channel 3

- For better compression efficiency, the HTTP/2, differently from the rules in HTTP/1.x, can allow separation of cookie-pairs from one Cookie header field into several Cookie header fields, each with one or more cookie-pairs.

- So, because of this duality, we can create a one directional covert channel from a client to the server by:
  - only one present Cookie header field to be binary 1, and
  - more than one present Cookie header fields to be binary 0.

Some new HTTP/2 covert channels

(paper in review)

Covert channel 4

- There are three different representations of the literal header field: with incremental indexing (starts with binary sequence 01), without indexing (starts with binary sequence 0000), and never indexed (starts with binary sequence 0001).

- We can create another covert channel using the following:
  - literal header field with incremental indexing or without indexing representation to be binary 0, and
  - literal header field never indexed representation to be binary 1.

Can BETTY research make the difference?

- Specially, one large group of network covert channels uses duality that exists in many protocols or non-mandatory options in them. Sometimes, these features are needed for security or performance reason, but sometimes, there are present without any additional reason.

- Building some tool, that will check and find these features in a given protocol, and report them to the designers?!
  - will eliminate redundancy and deal with unspecification!

THANKS FOR YOURS ATTENTION