CS5321 Network Security
TCP Sequence Number Inference

Week 5 Paper Summary

Transmission Control Protocol (TCP)

Layer			Protocol data unit (PDU)	Function[26]
Host layers	7	Application	Data	High-level protocols such as for resource sharing or remote file access, e.g. HTTP.
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

Material source: wikipedia.org

TCP Sequence numbers

Why:

- TCP **preserves order** of the packets

What:

- Each packet can be **acknowledged by** its **sequence number**
- Sequence number **increments by** a number of **bytes** sent in a packet

TCP Sequence numbers

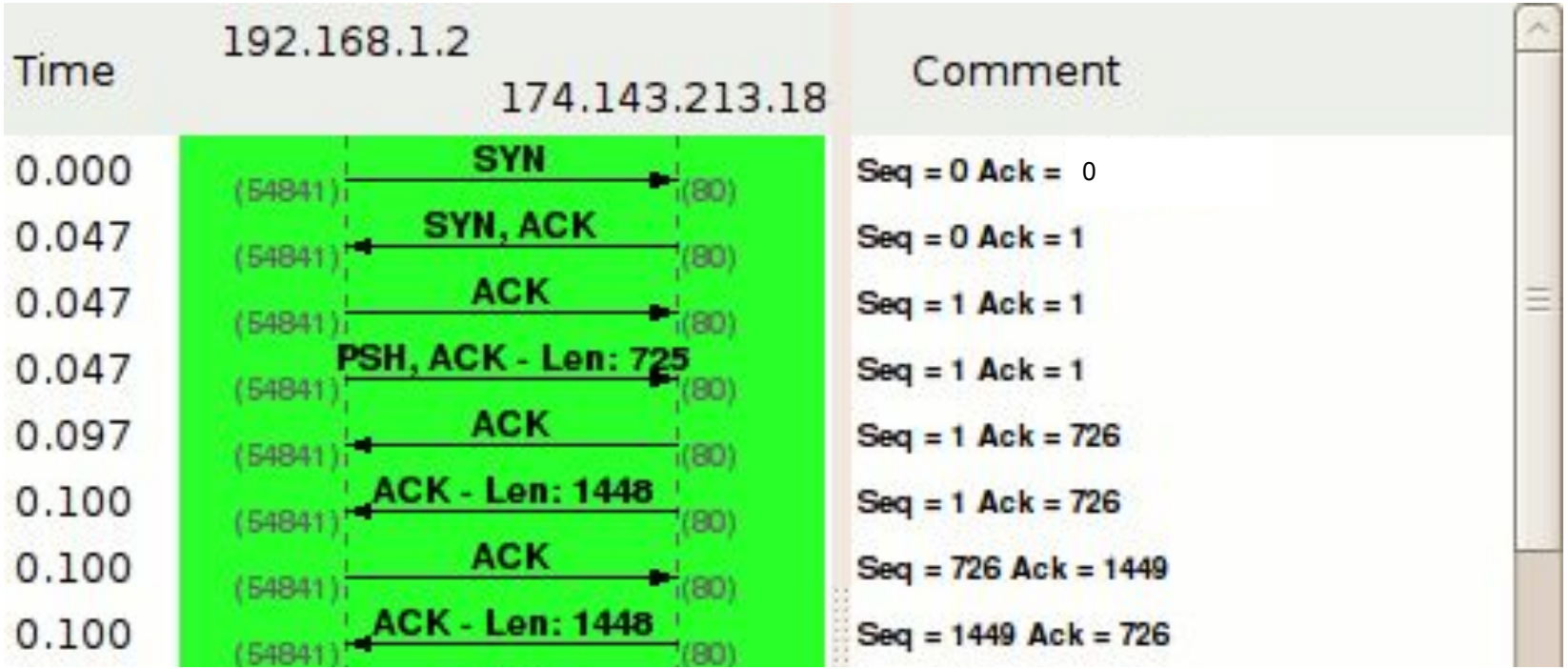


Image source: packetlife.net

What is the issue with acknowledging every packet?

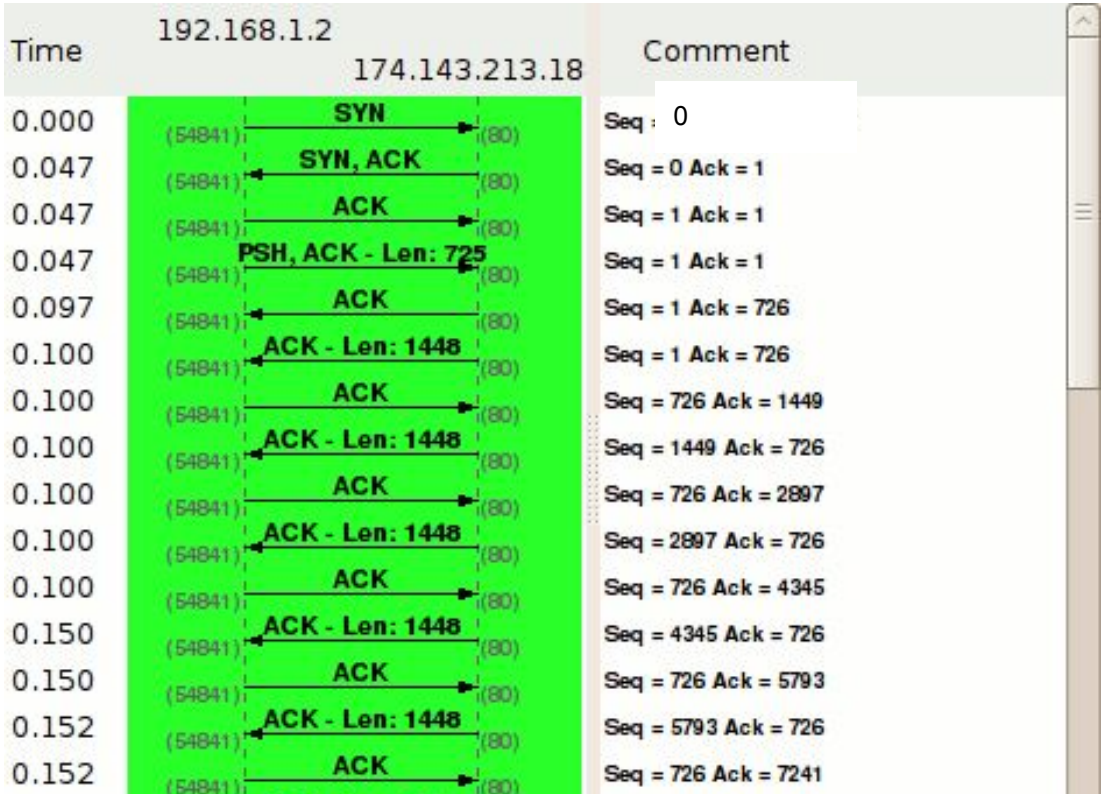


Image source: packetlife.net

TCP Sliding Window

Why:

- TCP **preserves order** of the packets
- TCP tries to **maximize network throughput**

What:

- TCP Sliding Window protocol allows to have up to WINDOW_SIZE **bytes in-flight** (i.e. sent but not acknowledged by a receiver)

Nota bene:

- There are **two windows** -- one for each direction of the connection

Communication with TCP Sliding Window in place

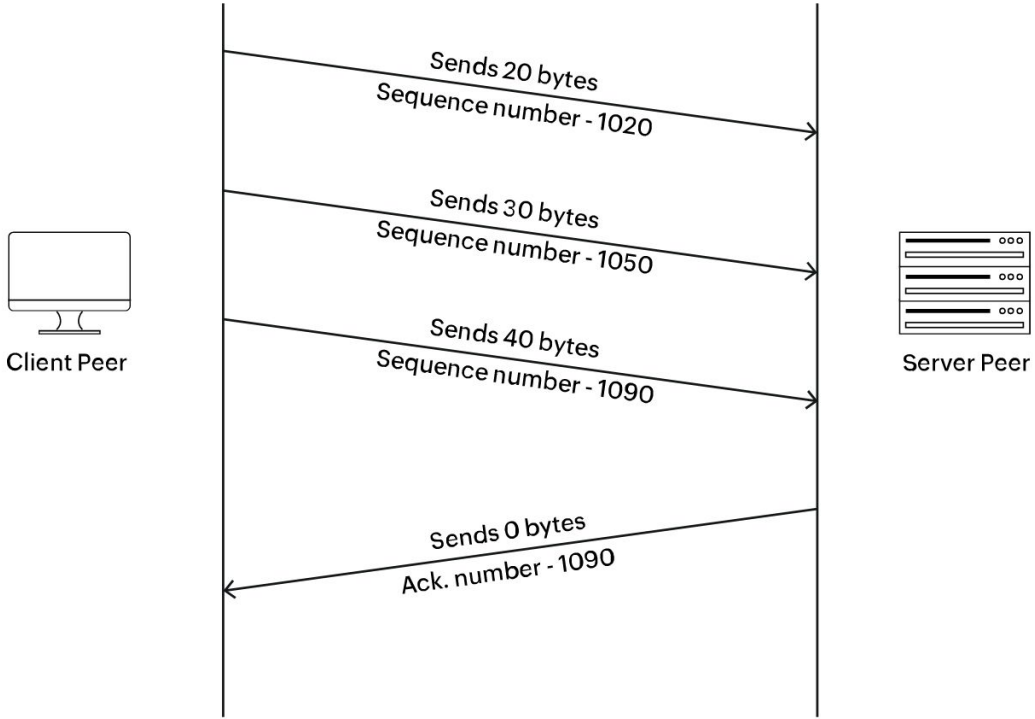


Image source: site24x7.com

Paper: *"Collaborative TCP Sequence Number Inference Attack"*

Year: 2012

Authors: Zhiyun Qian, Z. Morley Mao, Yinglian Xie

Why is inferring TCP Sequence Numbers interesting?

1. Injection of data of attacker's choice into TCP connection
2. TCP Reset attack

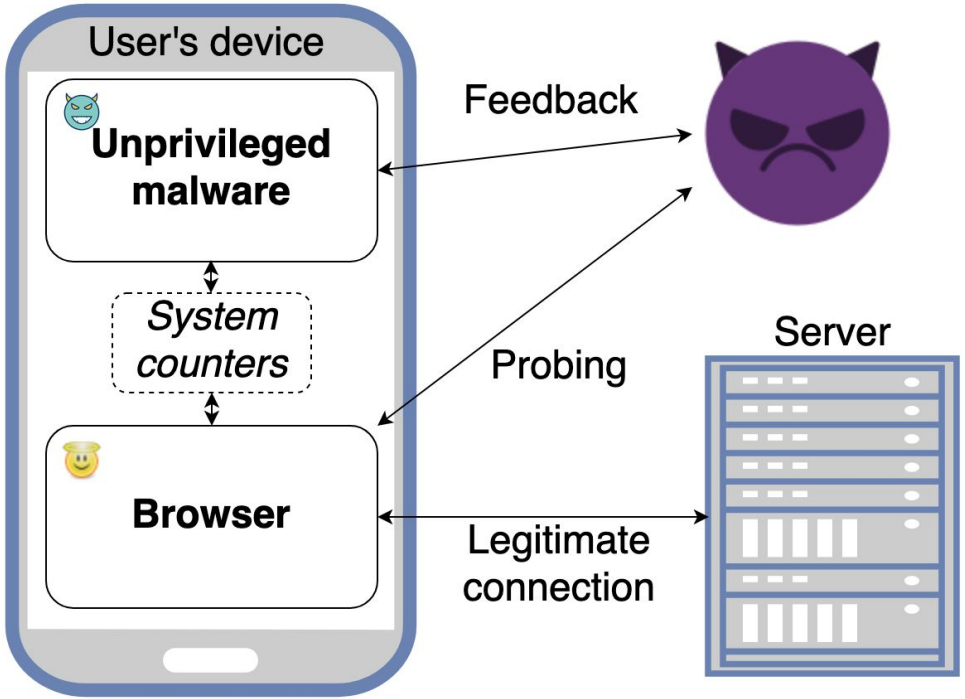
How?

Side channels!

Threat model

Requirements for the attacker:

1. Malware on the client with Internet access
2. Malware that can run in the background and read packet counters
3. Malware that can read the list of active TCP connections and their four-tuples
4. A predictable external port number if NAT is deployed



What can unprivileged malware obtain directly?

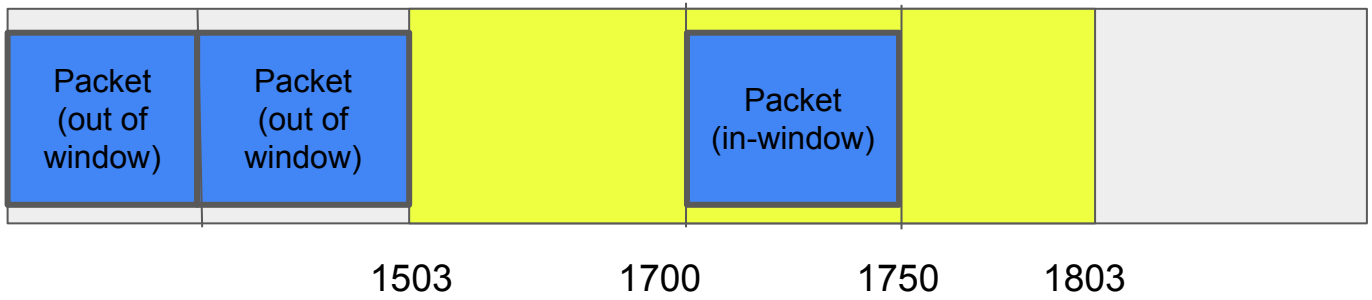
1. **Four-tuple of the connection**: source/destination IP addresses and source/destination port numbers
2. Values of **system-wide packet counters**

Relevant logic of Linux's TCP/IP stack

If:

➔ the packet contains no error (checksum matched content etc.)

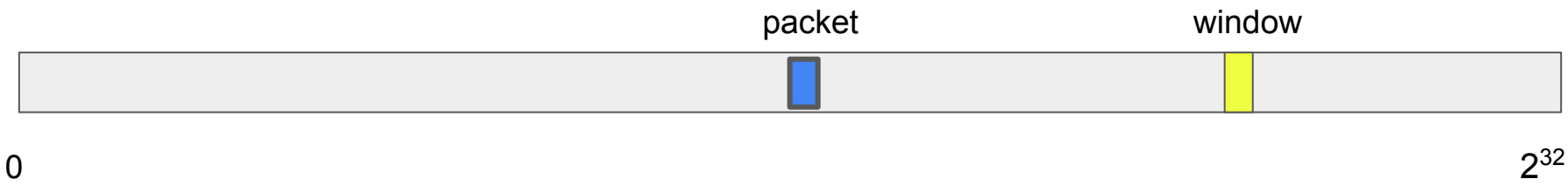
➔ the sequence number is **out of window**



Relevant logic of Linux's TCP/IP stack

If:

- ➔ the packet contains no error (checksum matched content etc.)
- ➔ the sequence number is **out of window**
- ➔ packet contains **non-zero payload**
- ➔ **sequence number is less than the beginning of the window**



Relevant logic of Linux's TCP/IP stack

If:

➔ the packet contains no error (checksum matched content etc.)
➔ the sequence number is **out of window**
➔ packet contains **non-zero payload**
➔ **sequence number is less than the beginning of the window**

Then:

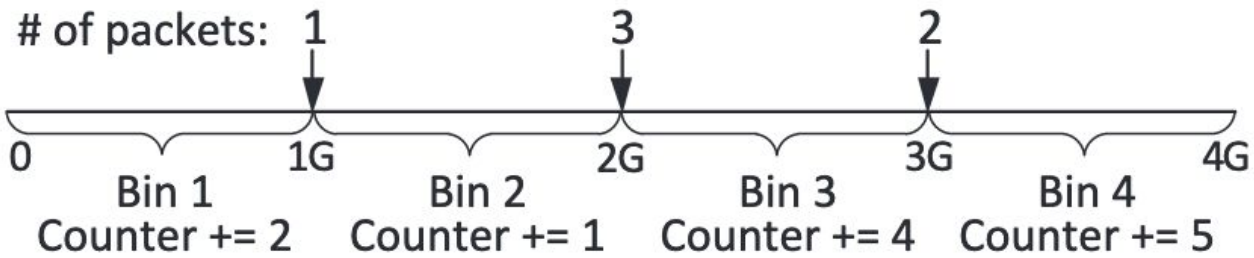
➔ System-wide counter **DelayedACKLost** is incremented

Nota bene: this counter is not noisy, i.e. only rarely increments naturally.

How to put incrementation of DelayedACKLost to use?

- Run **binary search** to find the expected sequence number, i.e. the beginning of the window => **32 packets**
- Run **N-way search** => even less packets (tradeoff)

N-way search: how? **Golomb ruler**



Overall results

The range of the attacks possible due to sequence number inference:

- client-side TCP Injection,
- passive TCP hijacking
- active TCP hijacking,
- server-side TCP injection

At least some variations of the attack were possible:

- On MacOS/BSD
- On Linux

Is this paper still relevant?

1. The attack is not relevant for connections with enabled SSL/TLS.
2. The fix hindering relevant side-channels from unprivileged programs was suggested by authors to be merged to Linux.

> Default protocol https is used by 81.5% of all the websites.

Material source: w3techs.com