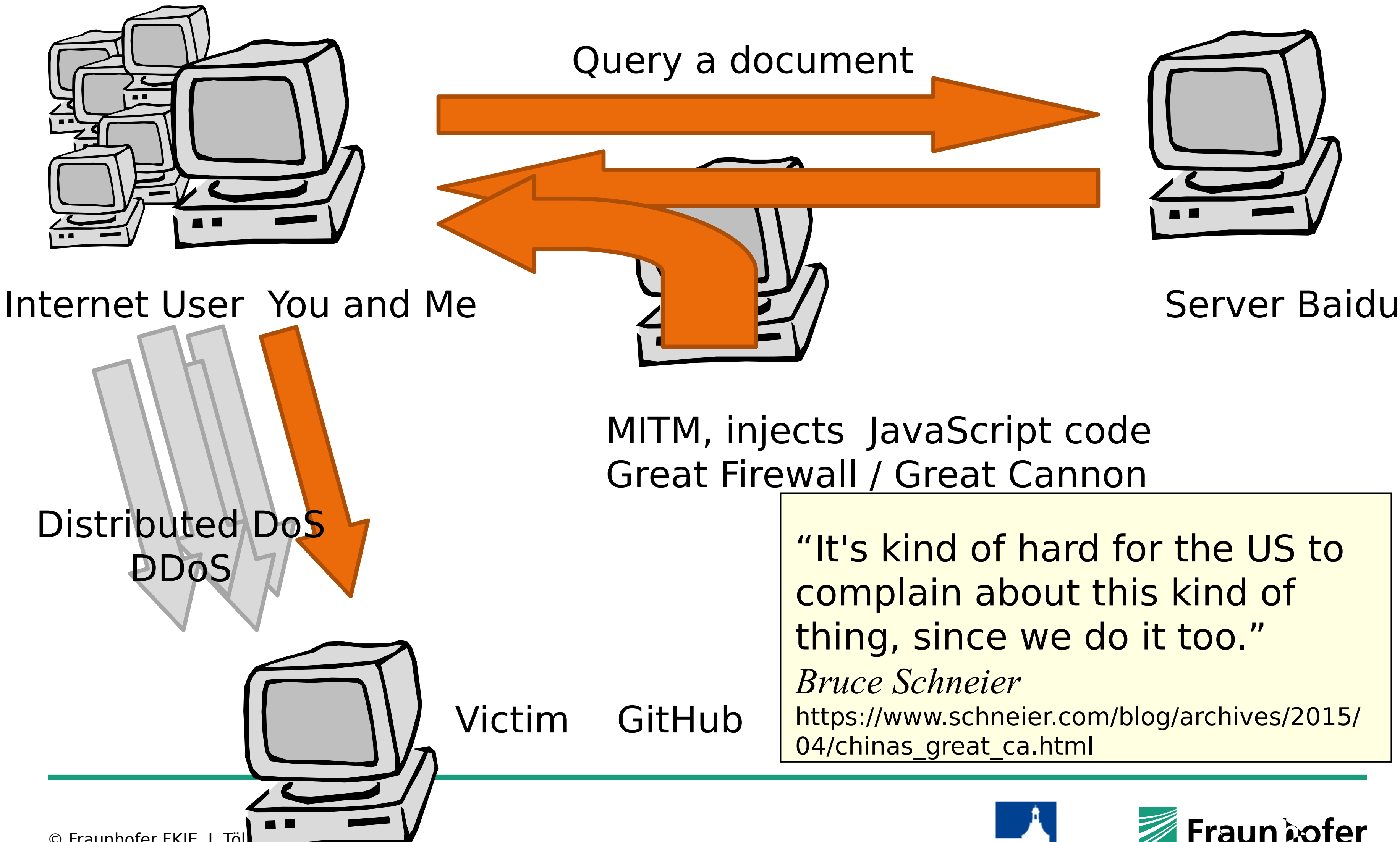
Agenda

1. TCP Refresher
2. Session Hijacking
3. (TCP) DoS Attacks
   DoS in a modern version
   „The great cannon of china“
4. The RST Attack
5. DNS Spoofing

„The great cannon of China“



“It's kind of hard for the US to complain about this kind of thing, since we do it too.”
Bruce Schneier
https://www.schneier.com/blog/archives/2015/04/chinas_great_ca.html

Agenda

1. TCP Refresher
2. Session Hijacking
3. TCP DoS Attacks
4. The RST Attack
5. DNS Spoofing

RST Attack: Background

T attack: Injection of a spoofed RST segment that terminates a TCP session.

Attacker **knows**

- Destination IP address
- Destination TCP port

Attacker has to **spoof**

- Source IP address
- Source TCP port

Attacker has to **guess**

- Sequence number?
- Acknowledgement number?

RST attacks pose a big threat especially against long-term connections (e.g., for BGP, VPNs, SSL/TLS, IRC, Databases, …) where recovering from lost connections takes lots of time and resources.

RST Attack: Source Port and Source IP Address and ISN

The source **IP address** has to be known to perform a RST attack.

The source **TCP port** is chosen by the operating system. It increases the difficulty by 216.

- Only true for pseudo-random port selection (e.g., in OpenBSD)
- Most **d**
  interv

```
$ cat /proc/sys/net/ipv4/ip_local_port_range
32768    61000
```

- Source port selection might be **predictable**

The **sequence number** is…

- accepted if their sequence number lies in the current window
- x ∈ [min, max] versus x = y

RST Attack: Attacking Multiple Sessions – Birthday Attack

Birthday Paradox: In a group of 23 randomly chosen people the probability that two of them have been born on the same day is more than 50 percent.*

* assuming that birthdays are evenly distributed.

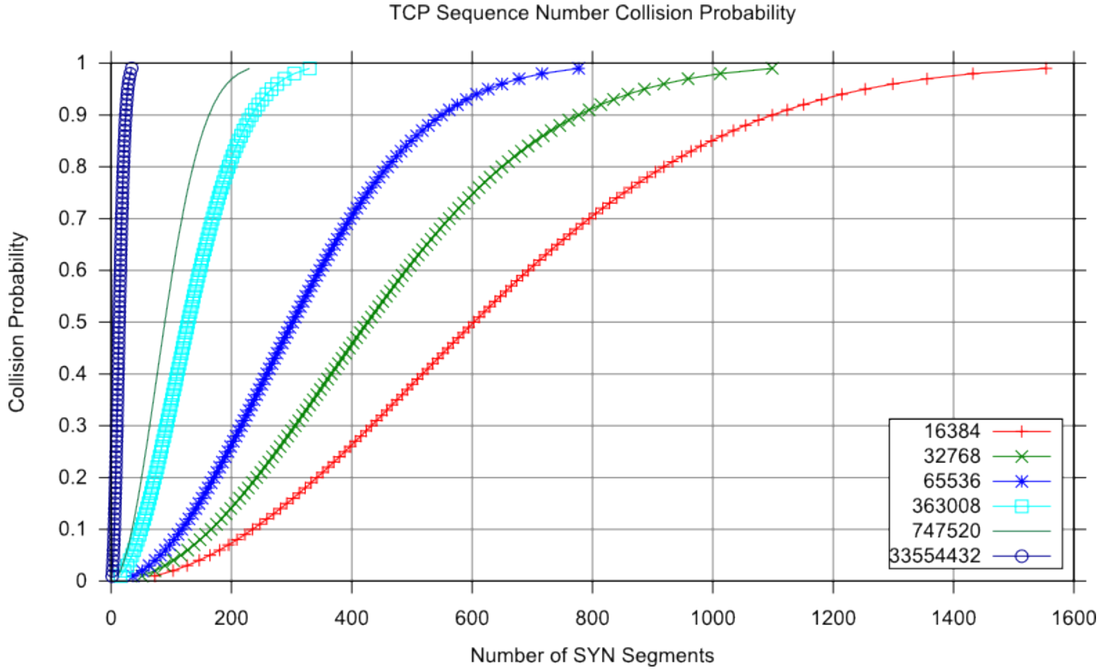
How many sessions must be present to generate a sequence number collision with a probability of at least 50 percent?

The number of sessions needed can be approximated by

$$n(p;H) \approx \sqrt{2H \ln \frac{1}{1-p}},$$

with p being the required minimum probability and H the size of the population (all possible sequence numbers).

RST Attack: TCP Window Size



Platform	Window Size	Population	Number of Sessions (Coll. > 50%)
CISCO	16384	262144	603
Windows	64240	66858	305
Linux	5840 * 27	5746	90
Sun OS	32768 * 210	128	14

RST Attack: Example – HTTP Server on Linux

An attacker wants to reset a random connection to a Linux HTTP server. How long does it take to perform an attack with a 50% success probability on an average 100Mbit/s network?

90 parallel HTTP sessions are not much on an average site. By sending 90 RST segments with randomly chosen sequence numbers, the attacker can make use of the Birthday Paradox.

So how long does it take to send the packets?

time

Ethernet header (14)	IP header (20)	TCP header (20)	CRC32 (4)

Frames are padded to the minimum size of 512 bits.

eth frame ***# of parallel sessions*** ***range of ports*** ***100Mbit/s***

That is fast!

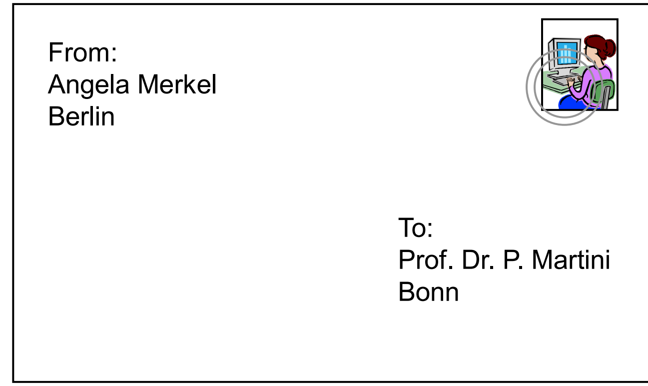
$$512 * 90 * (61000 - 32768 + 1) / 100\,000\,000 \text{ bit/s} = \mathbf{13s}$$

Agenda

1. TCP Refresher
2. Session Hijacking
3. TCP DoS Attacks
4. The RST Attack
5. DNS Spoofing

DNS Spoofing: Spoofing in General

Spoofing is the usage of a **fake identity**.



In human communication usage of fake identities is common as well, but *common sense* often helps to discover these attacks.

In network protocols information identifying the sender is **usually unprotected**. The recipient often simply **trusts** the sender entry.

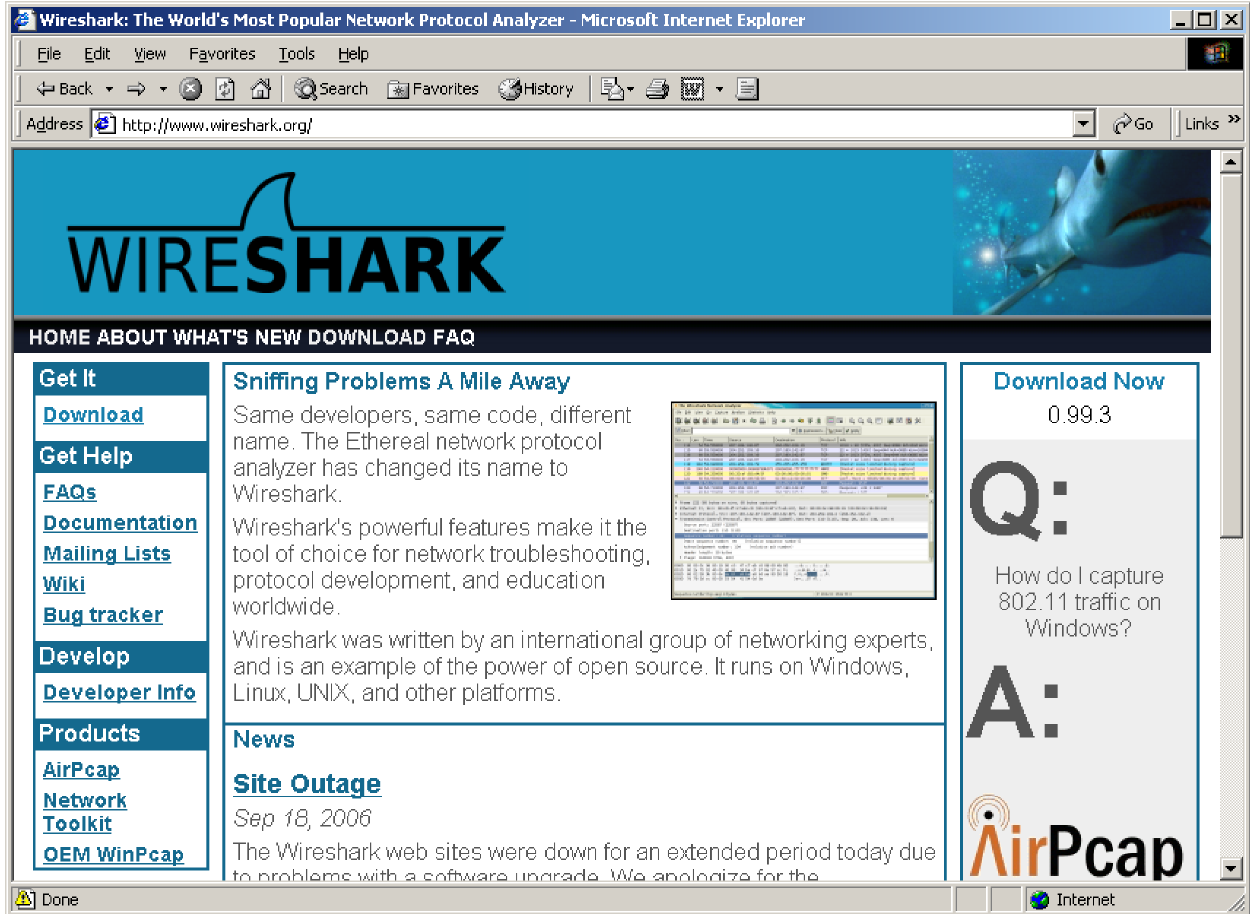
This works well for

- ARP spoofing
- DNS spoofing (e.g. cache poisoning)
- IP spoofing
- ICMP spoofing (e.g. redirect)
- Mail spoofing (sending mails with forged sender)
- Web spoofing (URL rewriting)
- ...

DNS Spoofing: Watch your Network

To understand **network protocol vulnerabilities**, it is helpful to see...

- how networks work,
- how network packets look like,
- and what kind of traffic you will find in your network.



A **network traffic analyzer** is an interesting tool to learn a lot about your network.

http://www.wireshark.org (formerly known as **ethereal**)

DNS Spoofing: Background

What happens, if you **open a web page**, e.g. http://www.sparkasse.de?

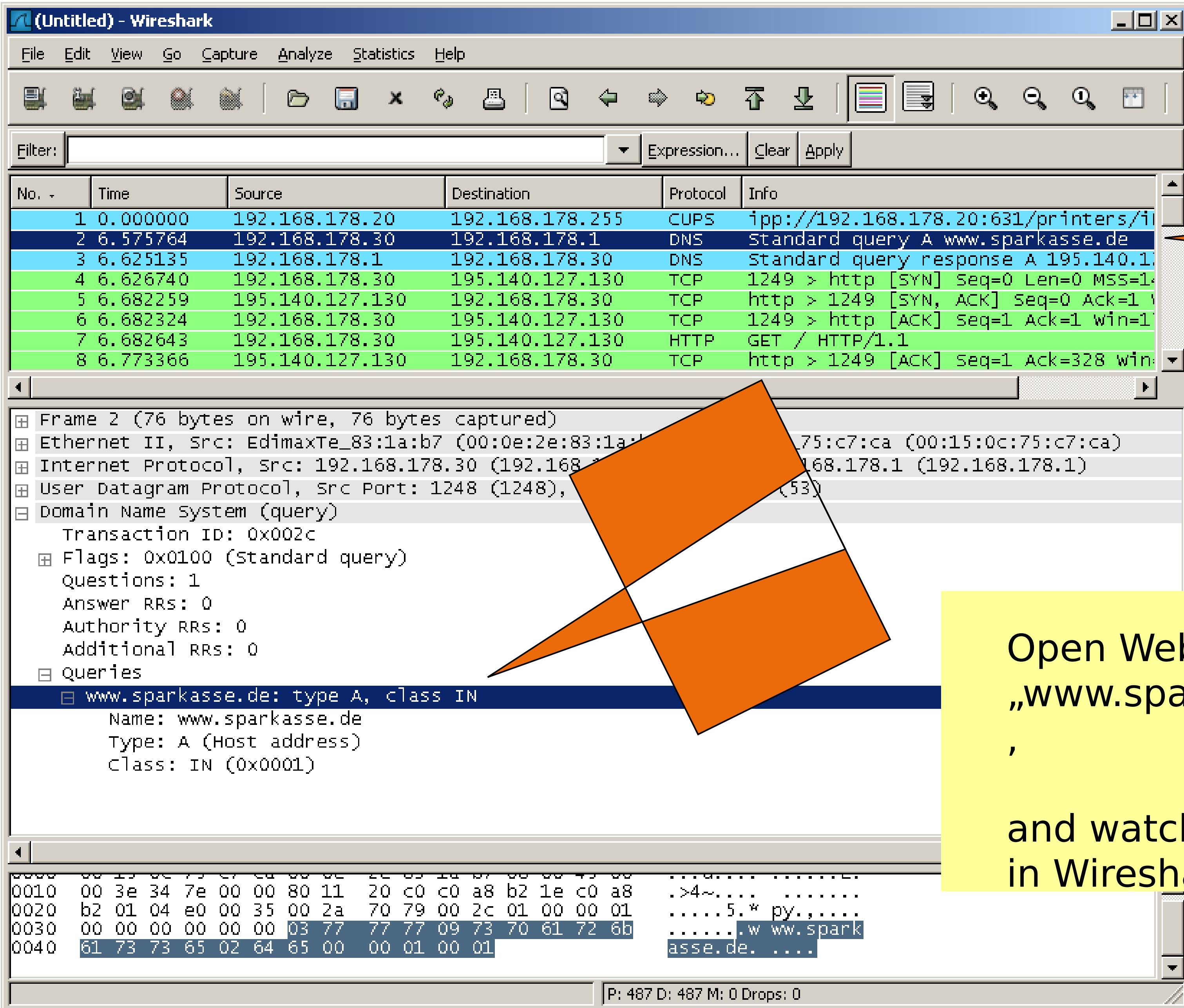
he Web browser has to contact the webserver of the domain **sparkasse.de**

This means sending an IP packet to this server.

And this means asking for the IP address belonging to **www.sparkasse.de**

This is done by the **Domain Name System**.

DNS Spoofing: A Question…

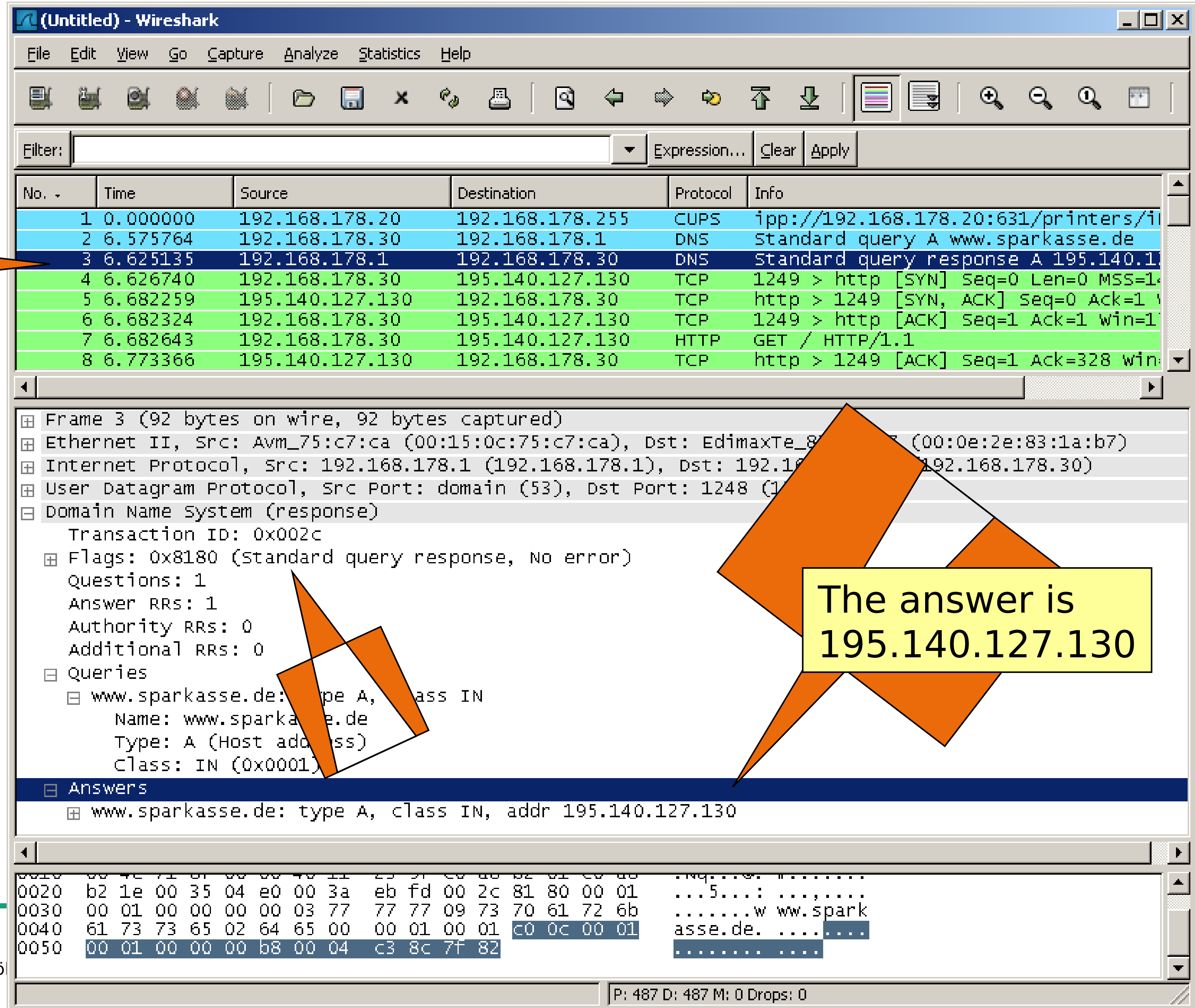


Open Web-Browser „www.sparkasse.de“ ,

and watch packets in Wireshark…

DNS Spoofing: ...and an Answer

An answer,
0,049 sec.
later...



The answer is
195.140.127.130

DNS Spoofing: The Attack

Now imagine that an attacker...

- has **access to the network** with your computer,
- **sees your DNS query**,
- and **sends a prepared answer** with another IP address.

If he is faster than the real answer,

- the asking program receives the **attacker's answer** first
- accepts it, if the **Transaction ID** fits,
- and „believes" the answer. **Even worse, it stores it in its cache**.

Your browser displays the URL **http://www.sparkasse.de**, but displays the web page of the attacker.

This works with every http web page and with every DNS query. More about
secured web pages later. The domain name sparkasse.de is only an example!

DNSsec

RFC 3833 (called *DNS Threat Analysis*) analyzes potential attacks targeted to the Domain Name System., e.g. the example given above.

Countermeasure:

Use cryptographic protection

-> DNSsec (DNS security extensions) was published in 2005

(see RFC 4033, 4034, 4035)

Idea:

Sign (cryptographic signature, see chapter 4) answers of DNS servers.

Summary

Summary

- TCP Refresher
- Session Hijacking
- TCP DoS Attacks
- The RST Attack
- DNS Spoofing