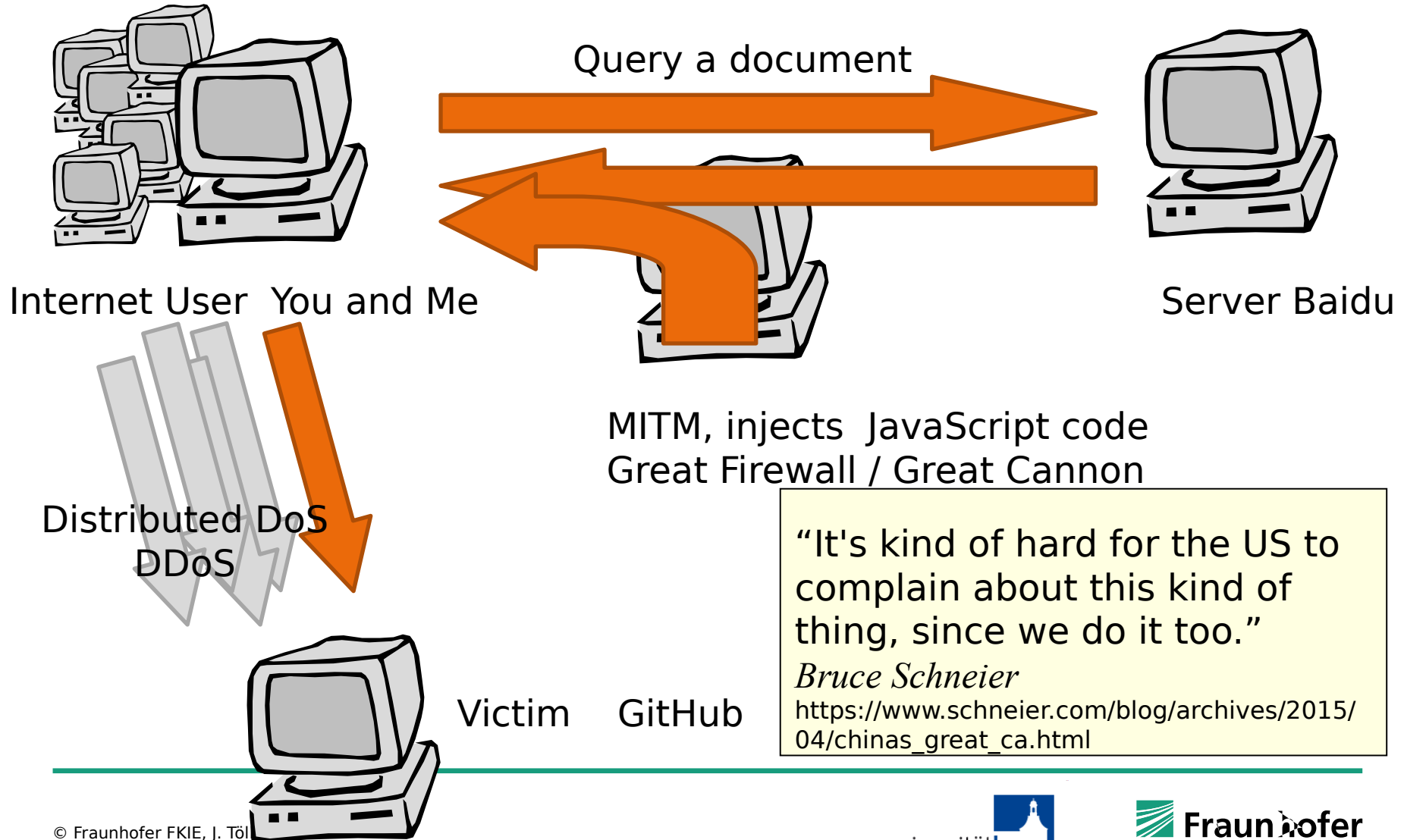


Agenda

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“



Agenda

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

RST Attack: Background

RST 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
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

- $x \in [\min, \max]$ 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?

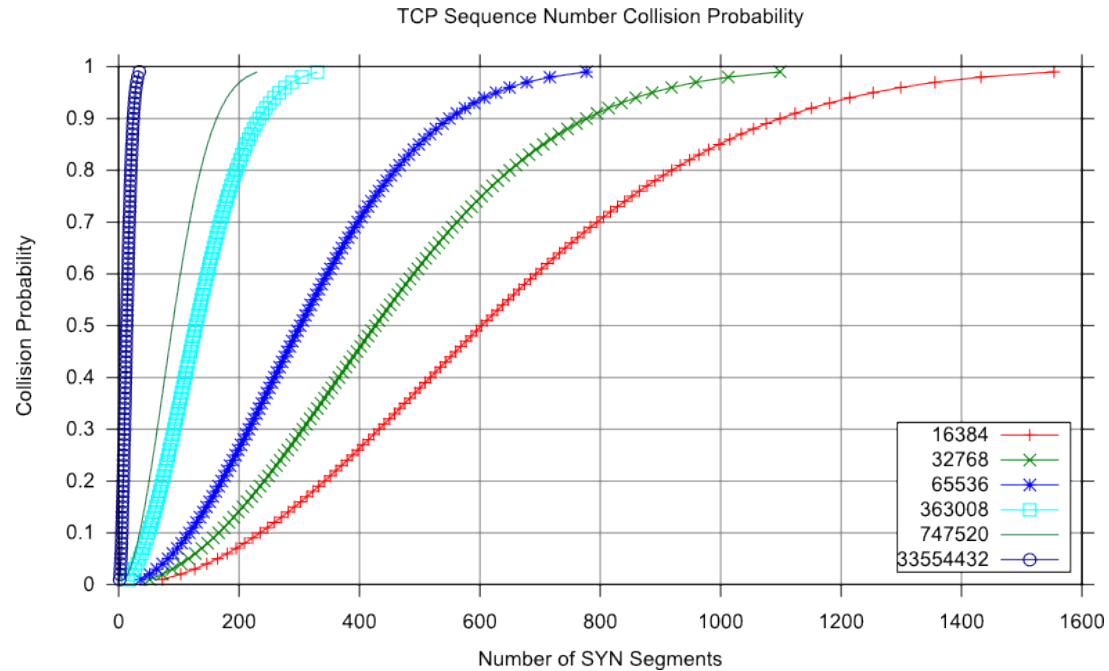


The number of sessions needed can be approximated by

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

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

RST Attack: TCP Window Size



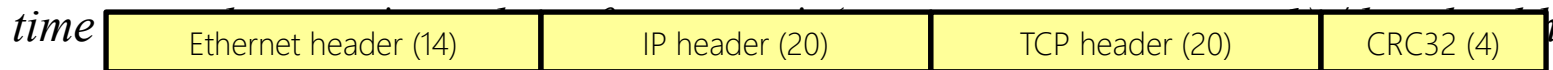
Platform	Window Size	Population	Number of Sessions (Coll. > 50%)
CISCO	16384	262144	603
Windows	64240	66858	305
Linux	5840 * 27	5746	90
Sun OS	32768 * 210	128	14

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?



Frames are padded to the minimum size of 512 bits.

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

That is fast!

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

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.

From:
Angela Merkel
Berlin



To:
Prof. Dr. P. Martini
Bonn

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)

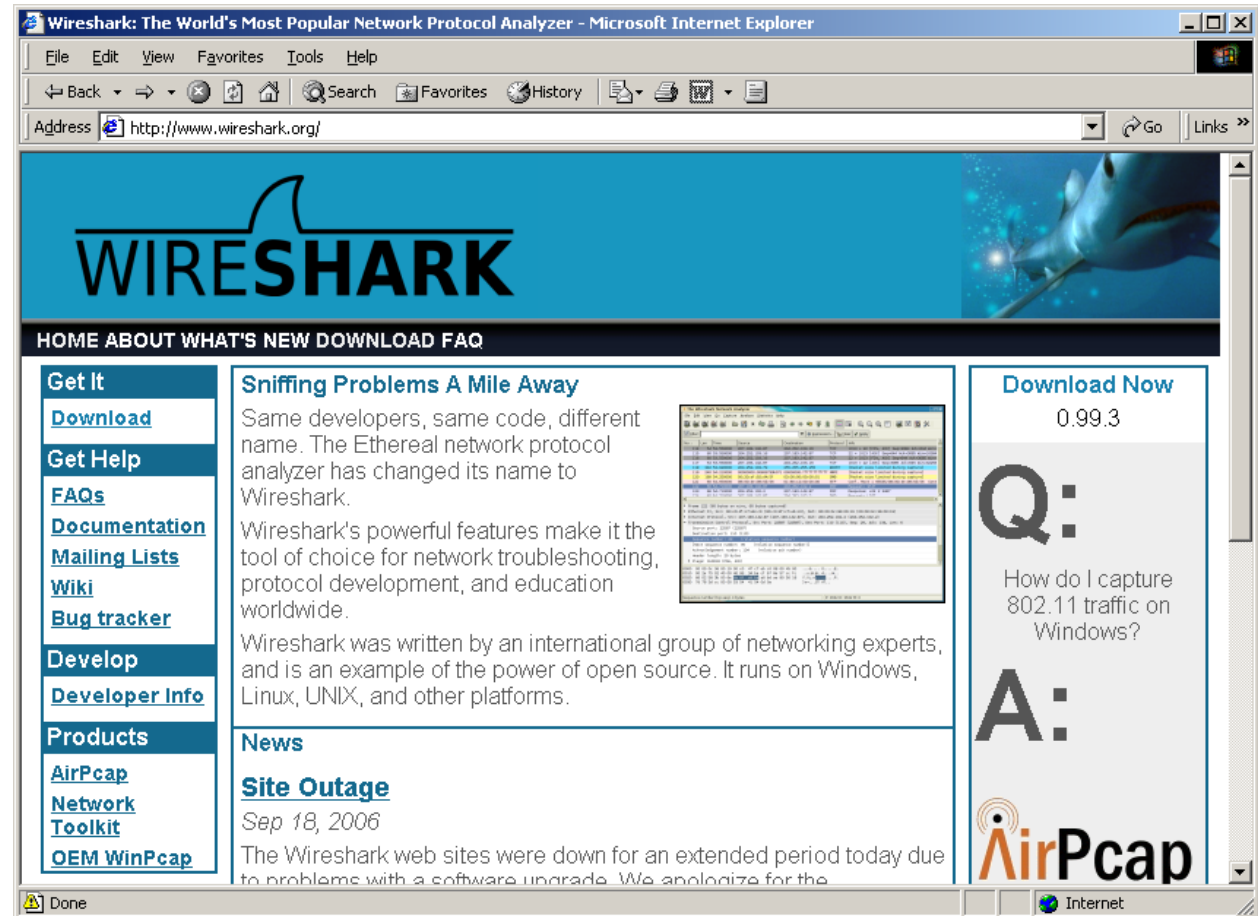
...

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>?

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

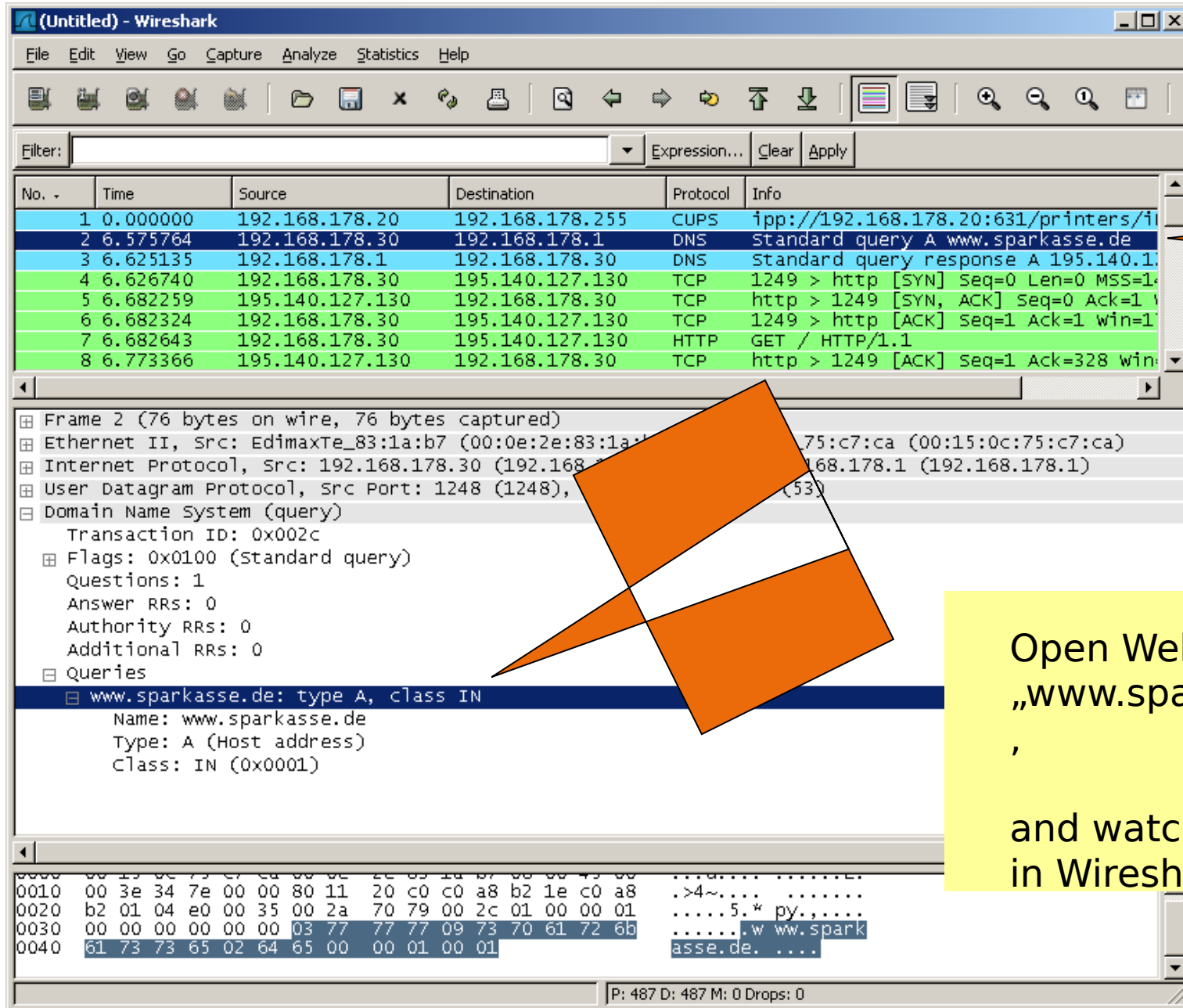
This means sending an IP packet to this server.

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...



The image shows a Wireshark packet capture window titled "(Untitled) - Wireshark". The packet list on the left shows eight packets. Packet 2 is a DNS standard query for www.sparkasse.de. Packet 3 is a DNS standard query response from 195.140.1.1. Packets 4 through 8 show an HTTP connection established and a GET request for /printers/.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.178.20	192.168.178.255	CUPS	ipp://192.168.178.20:631/printers/
2	6.575764	192.168.178.30	192.168.178.1	DNS	Standard query A www.sparkasse.de
3	6.625135	192.168.178.1	192.168.178.30	DNS	Standard query response A 195.140.1.1
4	6.626740	192.168.178.30	195.140.127.130	TCP	1249 > http [SYN] Seq=0 Len=0 MSS=1
5	6.682259	195.140.127.130	192.168.178.30	TCP	http > 1249 [SYN, ACK] Seq=0 Ack=1
6	6.682324	192.168.178.30	195.140.127.130	TCP	1249 > http [ACK] Seq=1 Ack=1 win=1
7	6.682643	192.168.178.30	195.140.127.130	HTTP	GET / HTTP/1.1
8	6.773366	195.140.127.130	192.168.178.30	TCP	http > 1249 [ACK] Seq=1 Ack=328 win=

The packet details pane shows the structure of the selected packet (Frame 2):

- Ethernet II, Src: EdimaxTe_83:1a:b7 (00:0e:2e:83:1a:b7), Dst: 08:00:0c:27:c7:ca (00:15:0c:75:c7:ca)
- Internet Protocol, Src: 192.168.178.30 (192.168.178.30), Dst: 192.168.178.1 (192.168.178.1)
- User Datagram Protocol, Src Port: 1248 (1248), Dst Port: 53 (53)
- Domain Name System (query)
 - Transaction ID: 0x002c
 - Flags: 0x0100 (Standard query)
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
- Queries
 - www.sparkasse.de: type A, class IN
 - Name: www.sparkasse.de
 - Type: A (Host address)
 - Class: IN (0x0001)

The packet bytes pane shows the raw data of the selected packet, with the IP address 192.168.178.30 and the destination IP 192.168.178.1 highlighted.

Open Web-Browser
„www.sparkasse.de“
,

and watch packets
in Wireshark...

DNS Spoofing: ...and an Answer

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

Wireshark (Untitled) - Wireshark

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.178.20	192.168.178.255	CUPS	ipp://192.168.178.20:631/printers/i
2	6.575764	192.168.178.30	192.168.178.1	DNS	standard query A www.sparkasse.de
3	6.625135	192.168.178.1	192.168.178.30	DNS	standard query response A 195.140.1
4	6.626740	192.168.178.30	195.140.127.130	TCP	1249 > http [SYN] Seq=0 Len=0 MSS=1
5	6.682259	195.140.127.130	192.168.178.30	TCP	http > 1249 [SYN, ACK] Seq=0 Ack=1
6	6.682324	192.168.178.30	195.140.127.130	TCP	1249 > http [ACK] Seq=1 Ack=1 win=1
7	6.682643	192.168.178.30	195.140.127.130	HTTP	GET / HTTP/1.1
8	6.773366	195.140.127.130	192.168.178.30	TCP	http > 1249 [ACK] Seq=1 Ack=328 win=

Frame 3 (92 bytes on wire, 92 bytes captured)

Ethernet II, Src: Avm_75:c7:ca (00:15:0c:75:c7:ca), Dst: EdimaxTe_8 (00:0e:2e:83:1a:b7)

Internet Protocol, Src: 192.168.178.1 (192.168.178.1), Dst: 192.168.178.30

User Datagram Protocol, Src Port: domain (53), Dst Port: 1248 (1248)

Domain Name System (response)

- Transaction ID: 0x002c
- Flags: 0x8180 (Standard query response, No error)
- Questions: 1
- Answer RRs: 1
- Authority RRs: 0
- Additional RRs: 0

Queries

- www.sparkasse.de: type A, class IN
 - Name: www.sparkasse.de
 - Type: A (Host address)
 - Class: IN (0x0001)

Answers

- www.sparkasse.de: type A, class IN, addr 195.140.127.130

The answer is
195.140.127.130

0020 b2 1e 00 35 04 e0 00 3a eb fd 00 2c 81 80 00 01 ...5...:

0030 00 01 00 00 00 00 03 77 77 77 09 73 70 61 72 6bw ww.spark

0040 61 73 73 65 02 64 65 00 00 01 00 01 c0 0c 00 01 asse.de.

0050 00 01 00 00 00 b8 00 04 c3 8c 7f 82

P: 487 D: 487 M: 0 Drops: 0

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