Network layer security

- Abusing address and length fields
- IP fragmentation problems
- IP Options
- TTL and ICMP

<u>Reading material:</u>
IP Security: Security assessment of the IP protocol by CPNI (see Canvas)
or RFC 6274: https://tools.ietf.org/html/rfc6274 (same content)

Attacking a device or a system… Where? How?

Invalid input validation is a major problem!

Cisco Adaptive Security Appliance (ASA) Software

Proven Firewall and Network Security Platform

The Cisco ASA Family of security devices protects corporate networks and data centers of all sizes. It provides users with highly secure access to data and network resources - anytime, anywhere, using any device. Cisco ASA devices represent more than 15 years of proven firewall and network security engineering and leadership, with more than 1 million security appliances deployed throughout the world.

Even firewalls have problems:

CVE-2019-1873 A vulnerability in the cryptographic driver for Cisco Adaptive Security Appliance Software (ASA) and Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to cause the device to reboot unexpectedly. The vulnerability is due to incomplete input validation of a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) ingress packet header. An attacker could exploit this vulnerability by sending a crafted TLS/SSL packet to an interface on the targeted device. An exploit could allow the attacker to cause the device to reload, which will result in a denial of service (DoS) condition. Note: Only traffic directed to the affected system can be used to exploit this vulnerability. This vulnerability affects systems configured in routed and transparent firewall mode and in single or multiple context mode. This vulnerability can be triggered by IPv4 and IPv6 traffic. A valid SSL or TLS session is required to exploit this vulnerability.

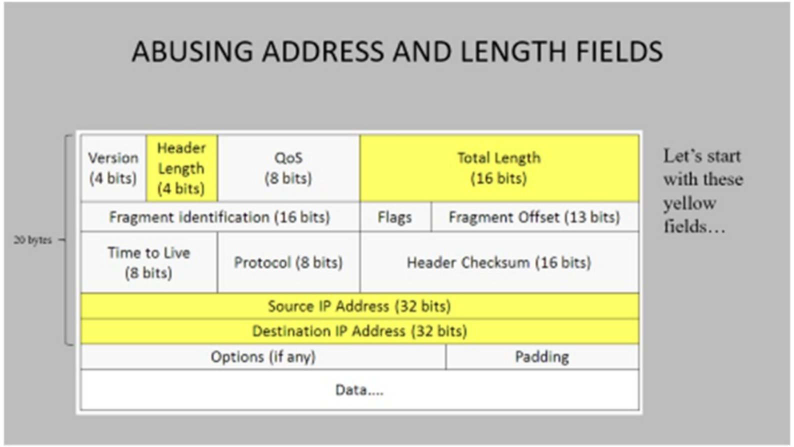
IP version 4 packet format

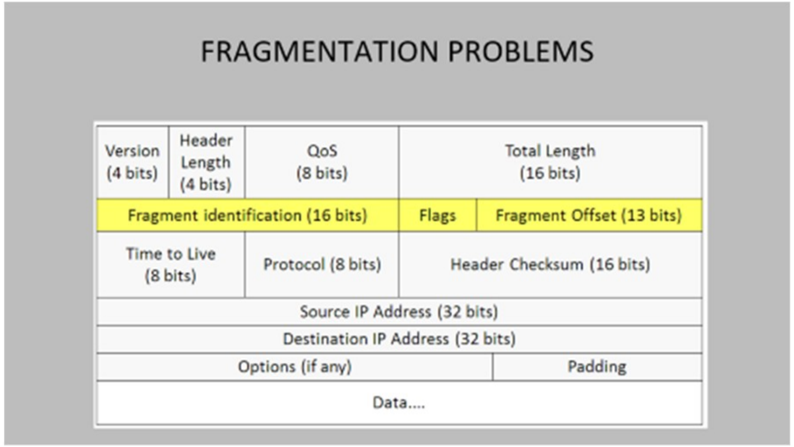
Bit 0 Bit 31

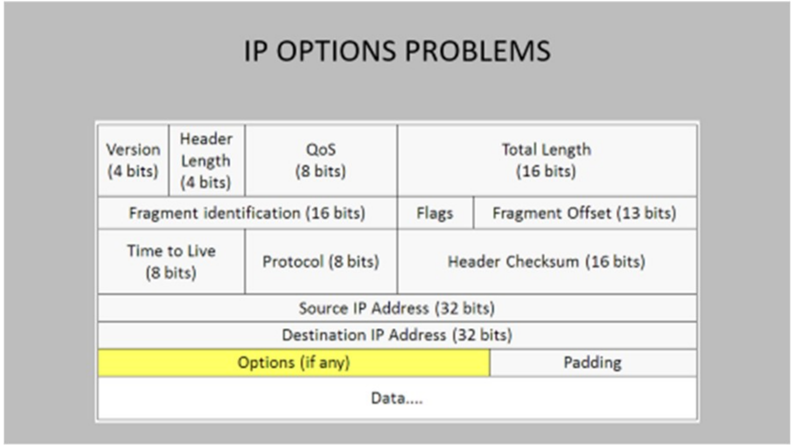
<table>
<tr><td>Version
(4 bits)</td><td>Header
Length
(4 bits)</td><td>QoS
(8 bits)</td><td colspan="2">Total Length
(16 bits)</td></tr>
<tr><td colspan="3">Fragment identification (16 bits)</td><td>Flags
[DF, MF]</td><td>Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live
(8 bits)</td><td>Protocol (8 bits)
1=ICMP, 6=TCP,
17=UDP, 50=ESP, …</td><td colspan="2">Header Checksum (16 bits)</td></tr>
<tr><td colspan="5">Source IP Address (32 bits)</td></tr>
<tr><td colspan="5">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="4">Options (if any)</td><td>Padding</td></tr>
<tr><td colspan="5">Data....</td></tr>
</table>

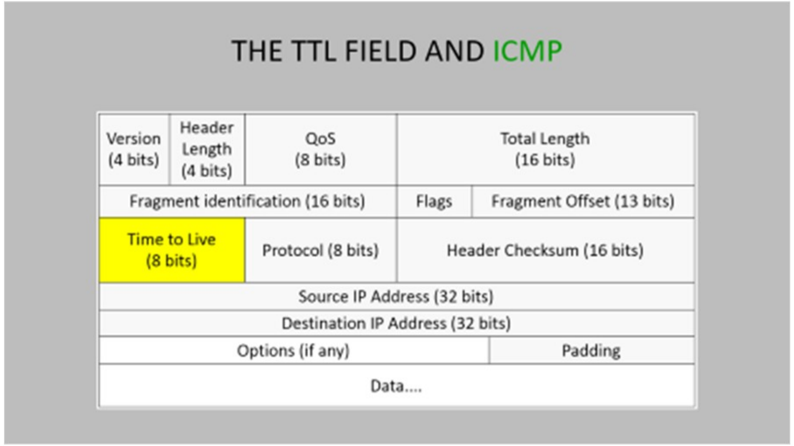
20 bytes

All fields can and will be abused









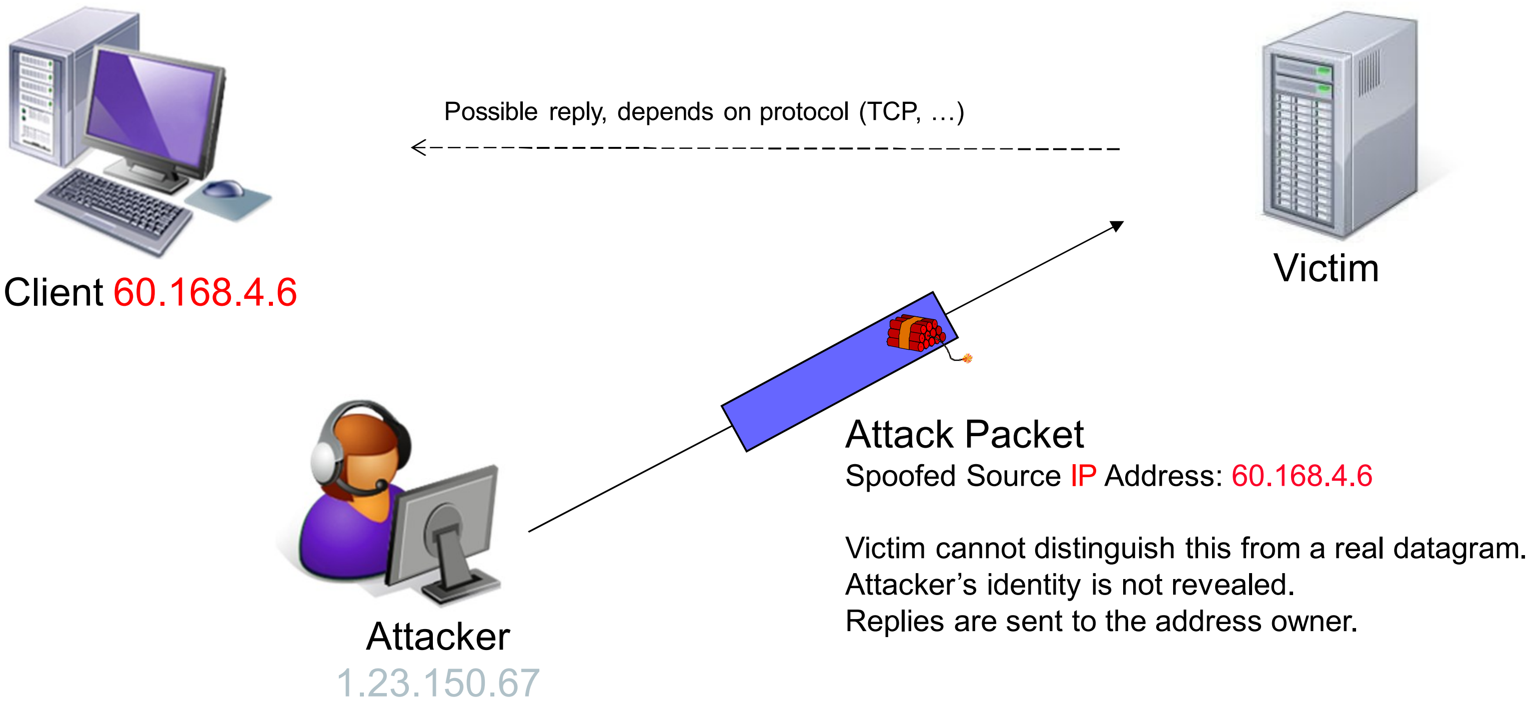
ABUSING ADDRESS AND LENGTH FIELDS

20 bytes

<table>
<tr><td>Version
(4 bits)</td><td>Header
Length
(4 bits)</td><td>QoS
(8 bits)</td><td colspan="3">Total Length
(16 bits)</td></tr>
<tr><td colspan="3">Fragment identification (16 bits)</td><td>Flags</td><td colspan="2">Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live
(8 bits)</td><td>Protocol (8 bits)</td><td colspan="3">Header Checksum (16 bits)</td></tr>
<tr><td colspan="6">Source IP Address (32 bits)</td></tr>
<tr><td colspan="6">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="5">Options (if any)</td><td>Padding</td></tr>
<tr><td colspan="6">Data....</td></tr>
</table>

Let’s start with these yellow fields…

IP Address Spoofing



A Google search for IP Address Spoofing

45

Is it really that easy for an attacker to forge an IP address in the wild?

Sure, if I don't care about actually receiving any responses, I can very easily send out packets using any source address I like. Since many ISPs don't really have good egress rules, anything I forge generally will be delivered.

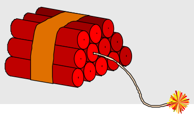
If the attacker actually needs two way communication it becomes very difficult.

Little Proof of Concept for Zordeche's Answer (with ubuntu):

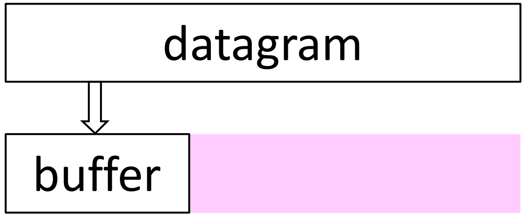
```
$ sudo apt-get install hping3
$ sudo hping3 -1 --spoof 11.10.10.20 www.google.com
HPING www.google.com (eth0 64.233.169.105): icmp mode set, 28 headers + 0 data bytes
```

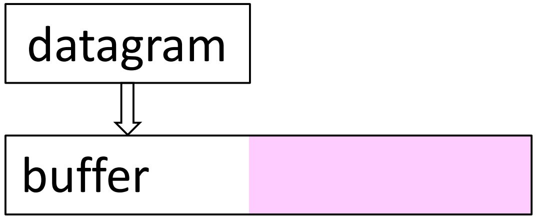
Attack is easy to detect: just check that outgoing IP addresses belong to us

Abusing the address and length fields

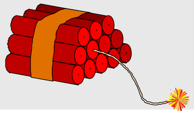


- IP address spoofing: send a message with a faked source address
  - Gives sender anonymity – attacker cannot be identified
  - Can blame someone else for sending packets
  - Can exploit trust between hosts
  - Never rely on IP addresses for security
- What if header length field is < 20 bytes (less than header size)?
  - What if zero? Can the receiver handle this? Maybe triggers a division by zero?
- What if datagram length differs from actual length?
  - If actual length > header specified length, a buffer overrun may occur
  - If actual length < header specified length, the packet may be placed in a too large buffer and the old contents may also be forwarded
  - Many protocols and devices have had such problems.
    Always check the input!

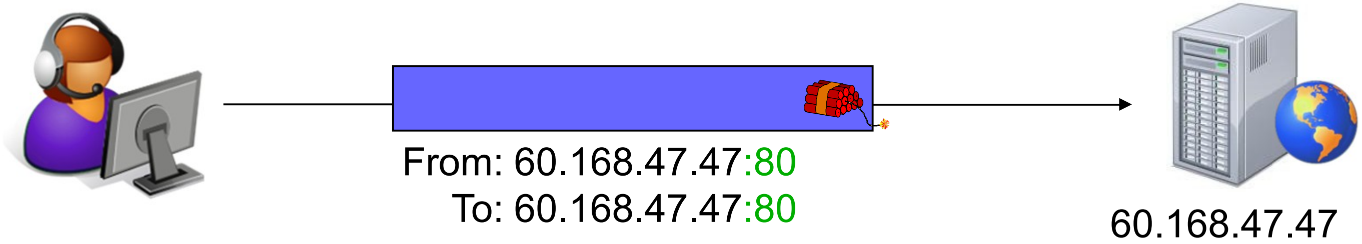




Address spoofing: the LAND attack



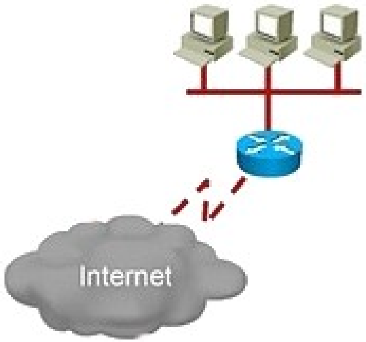
- Send victim a packet with victim's own IP address as both source and destination and possibly also with the same port number for source and destination



- Has triggered bugs in many implementations
- In the past, many computers, switches, routers, and even printers, crashed
- Although this has been a well-known problem for a very long time, in Windows Vista (beta), the bug was back again…

Ingress and egress filtering

- All networks should do ingress and egress filtering
  - Check all incoming and outgoing traffic to make sure your addresses are ok
- Border routers and firewalls should drop:
  - Broadcast addresses as destination
  - Multicast addresses as source - invalid (224/4)
  - Unassigned addresses (240/4): 240.0.0.0 – 255.255.255.254 reserved for future use
  - IP packets from an interface where the route to that network is through another interface
    - Not always practical – but definitively possible in most corporate networks
- Invalid addresses
  notation here is {network_prefix, host}, 0 means all zeroes, -1 all ones)
  - {0, 0} and {0, host} invalid   (but may be used as source in local DHCP requests during boot)
  - {-1, -1} broadcast cannot be source
  - {our_network, -1} local broadcast address, invalid as source
  - {127, *} invalid, “localhost” should not be seen on networks

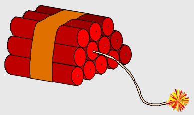


More details in RFC 6274 and in the firewall lecture

FRAGMENTATION PROBLEMS

<table>
<tr><td>Version
(4 bits)</td><td>Header
Length
(4 bits)</td><td>QoS
(8 bits)</td><td colspan="3">Total Length
(16 bits)</td></tr>
<tr><td colspan="3">Fragment identification (16 bits)</td><td>Flags</td><td colspan="2">Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live
(8 bits)</td><td>Protocol (8 bits)</td><td colspan="3">Header Checksum (16 bits)</td></tr>
<tr><td colspan="6">Source IP Address (32 bits)</td></tr>
<tr><td colspan="6">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="5">Options (if any)</td><td>Padding</td></tr>
<tr><td colspan="6">Data....</td></tr>
</table>

IP fragmentation: Teardrop attack



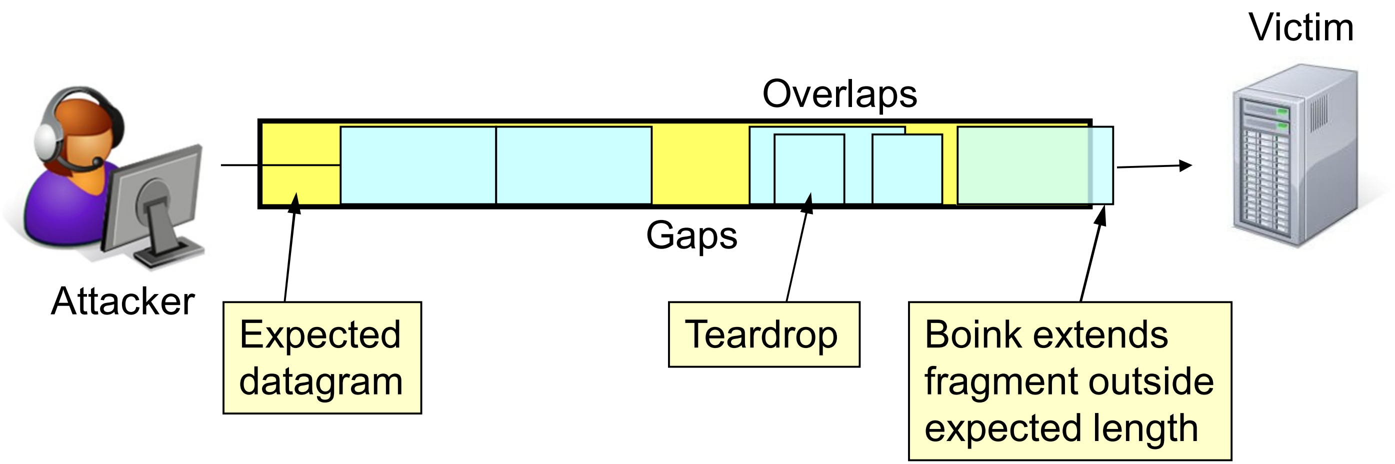
- Fragmented datagrams may trigger reassembly bugs
  - Packets may not make sense when reassembled
  - Several variants exist: Teardrop, New teardrop, Boink, ...
  - Bugs in Windows, Linux, Cisco routers, etc.
  - New versions were released each time vendors had patched their software…
- Teardrop: Send two fragmented IP datagrams:
  - The second (red) has a payload that fits entirely within the first (blue):

  - Linux was optimized to not copy overlapping parts of datagrams:

```
Bytes_to_copy = end_of_second_fragment – end_of_first_fragment
```

  - Bytes to copy becomes negative and is later treated as an unsigned integer -> Number of bytes to copy becomes very large – overwrites the whole system

More about Teardrop, Boink, …



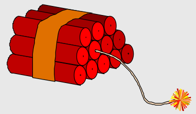
Holes in a datagram means that it is placed in a buffer while waiting for the missing parts:

Long wait → Buffer space exhausted
DoS problem?

Datagram may not make sense when reassembled. Firewalls confused?

Which is the correct data, the first received or the last?

IP fragmentation: Ping-of-Death attack



- Normal fragmentation method:
  - All fragments of a datagram have the same fragment identification (ID)
  - Fragment offset tells where to place it in the final datagram
  - The more fragments (MF) bit is 0 in the last fragment
  - The receiver doesn't know in advance number of fragments

MF	Fragment offset	Fragment ID	
1	0	15260	First packet
1	!=0	15260	
0	!=0	15260	Last paket
0	0	15260	Invalid

- IP Length Field
  - Tells length of entire datagram (16 bits = maximum 65,535 bytes)
  - But… fragmentation may create datagrams longer than 65,535 bytes: [Offset=65,000, size=10,000]
  - Many systems could (can?) not handle these invalid oversized datagrams!

- Ping-of-Death attack:
  send IP oversized datagrams, for example an ICMP echo ("ping")

Detection in firewalls

Example: Cisco firewall (ASA) detects these attacks:

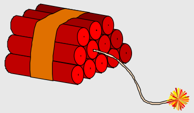
1102	400008	IP Impossible Packet	Attack	Triggers when an IP packet arrives with source equal to destination address. This signature will catch the so-called Land Attack.
1103	400009	IP Overlapping Fragments (Teardrop)	Attack	Triggers when two fragments contained within the same IP datagram have offsets that indicate that they share positioning within the datagram.
2154	400025	Ping of Death Attack	Attack	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP), the Last Fragment bit is set, and (IP offset * 8) + (IP data length) > 65535

Fragment ID reveals information

More details in the paper (or RFC 6274)

- Fragment IDs should be unique for each datagram sent from a host
  - Fragment ID should be unique per {source, destination, protocol}
- Some systems increase fragment ID for each packet they transmit
  - Leaks information about total packet transmission rate
  - Makes "Idle" scanning possible – next slide…
- Older Linux and Solaris: ID unique per IP address pair
  - Result: my connection does not update other connection's fragment IDs – good
  - Still possible to count number of systems behind a NAT gateway or a load balancer since each system has its own counter (may not matter?)
- Linux: sets ID = 0 if DF flag is set (i.e. almost always)
  - This behavior has been standardized [RFC 6864]
  - Some non-RFC compliant network devices may still fragment – then collisions will occur
  - If collision: TCP or UDP checksum will catch this
  - But some UDP implementations have the checksum disabled by default…

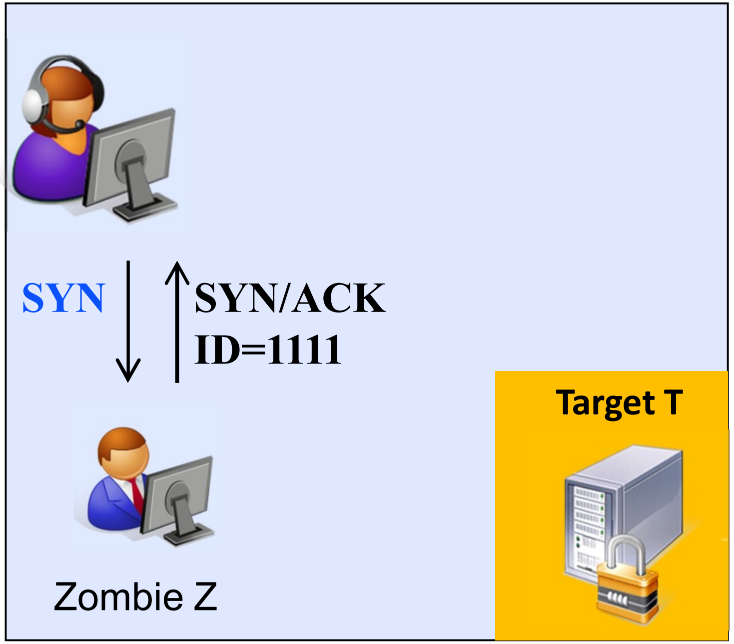
Idle (or ”Dumb”) Scanning [1]



Works if IP (fragment) ID:s are not unique per IP address

Problem: Target T does not talk to Eve. Maybe another system Z is trusted by T?

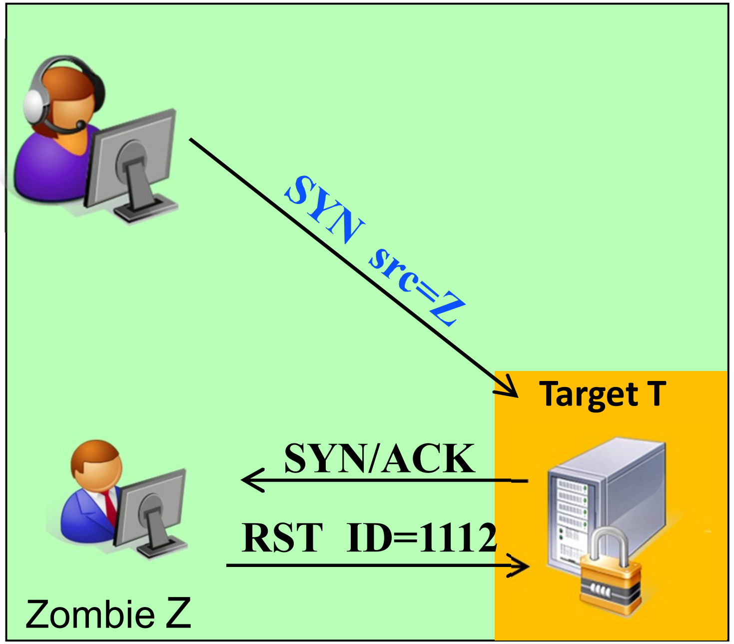
1. Send SYN to the zombie Z to find out what its fragment ID counter is.
2. Send a forged TCP SYN packet <u>to T</u> with the <u>Z as the source</u>.
3. If T trusts Z and answers, Z gets an unwanted SYN/ACK packet.
4. Z answers with a RST packet and its fragment ID counter is increased.
5. Probe Z again to determine if T has answered Z or not.



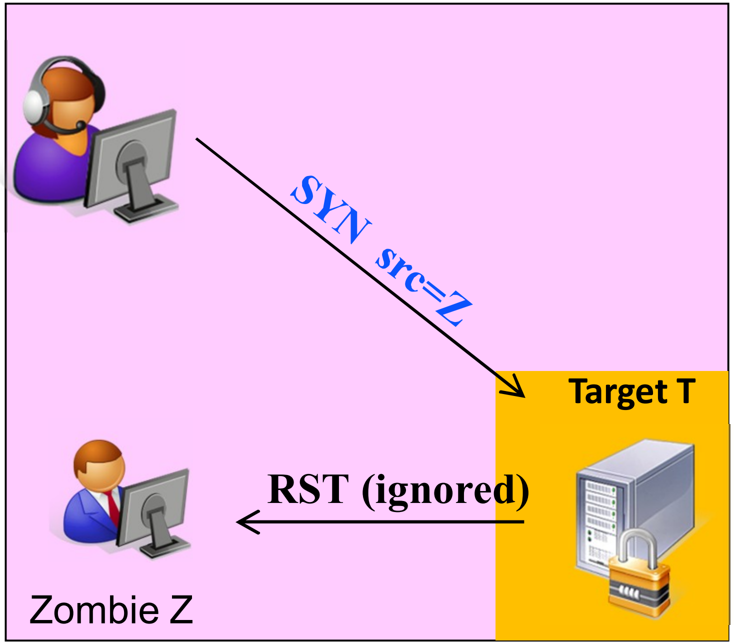
More info: https://insecure.org/nmap/idlescan.html

Idle (or “Dumb”) scanning [2]

Port **Open** (and Z trusted):

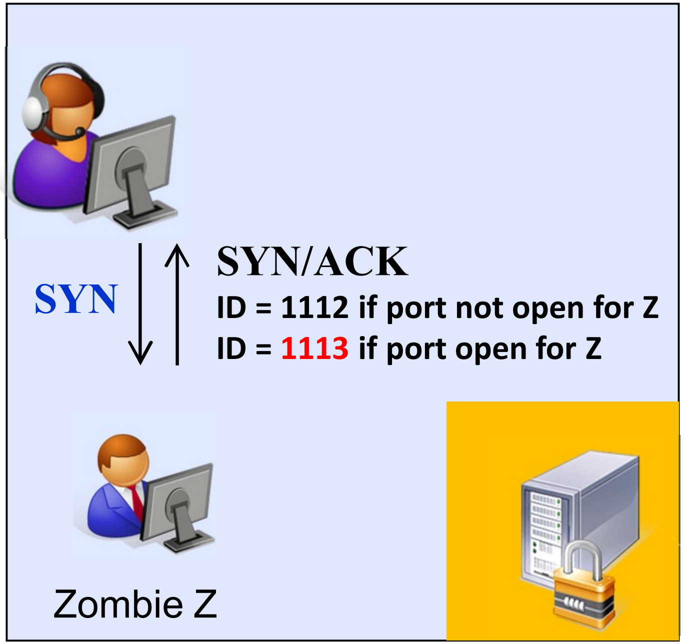


Port **Closed**



Later, Eve will not know whether it is the port that is closed or if it is Z that is not trusted.

Idle (or “Dumb”) scanning [3]



- Note that the zombie is not cracked – it just increases fragment ID numbers in a non-optimal way
- Shows whether computers filter out requests based on IP addresses
  - Can find out which systems trust each other
  - If a system (the “zombie”) is trusted, it is a suitable target
  - It could be a web server or home computer
  - Not uncommon that sensitive hosts filter addresses like this
- Tools like NMAP optimizes this scan
  - Sends 100 requests to target(s) before contacting the zombie
  - If a reply is as received, the port range is narrowed down
- Fragment numbers can also reveal if a host has many aliases or belongs to a load-balanced cluster
- **What to learn:** Filter incoming traffic – don’t allow external traffic with internal source addresses

Idle (or “Dumb”) scanning

```
# nmap -P0 -p- -sI kiosk.adobe.com www.riaa.com

Starting nmap V. 3.10ALPHA3 ( www.insecure.org/nmap/ )
Idlescan using zombie kiosk.adobe.com (192.150.13.111:80)
Interesting ports on 208.225.90.120:
(The 65522 ports scanned but not shown below are in state: closed)

Port        State       Service
21/tcp      open        ftp
25/tcp      open        smtp
80/tcp      open        http
111/tcp     open        sunrpc
135/tcp     open        loc-srv
443/tcp     open        https
1027/tcp    open        IIS
1030/tcp    open        iad1
2306/tcp    open        unknown
5631/tcp    open        pcanywheredata
7937/tcp    open        unknown
7938/tcp    open        unknown
36890/tcp   open        unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 2594.472 seconds
```

Don’t do this outside the lab. Packets with false source addresses may (should!) trigger alarms at Chalmers or at your ISP!

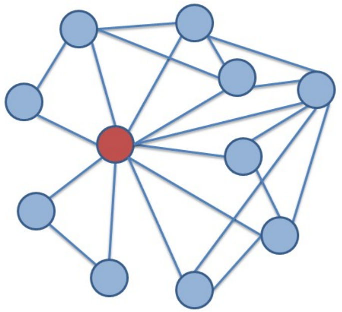
IP OPTIONS PROBLEMS

<table>
<tr><td>Version
(4 bits)</td><td>Header
Length
(4 bits)</td><td>QoS
(8 bits)</td><td colspan="3">Total Length
(16 bits)</td></tr>
<tr><td colspan="3">Fragment identification (16 bits)</td><td>Flags</td><td colspan="2">Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live
(8 bits)</td><td>Protocol (8 bits)</td><td colspan="3">Header Checksum (16 bits)</td></tr>
<tr><td colspan="6">Source IP Address (32 bits)</td></tr>
<tr><td colspan="6">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="5">Options (if any)</td><td>Padding</td></tr>
<tr><td colspan="6">Data....</td></tr>
</table>

IP options, source route

More details in the paper (or RFC 6274)

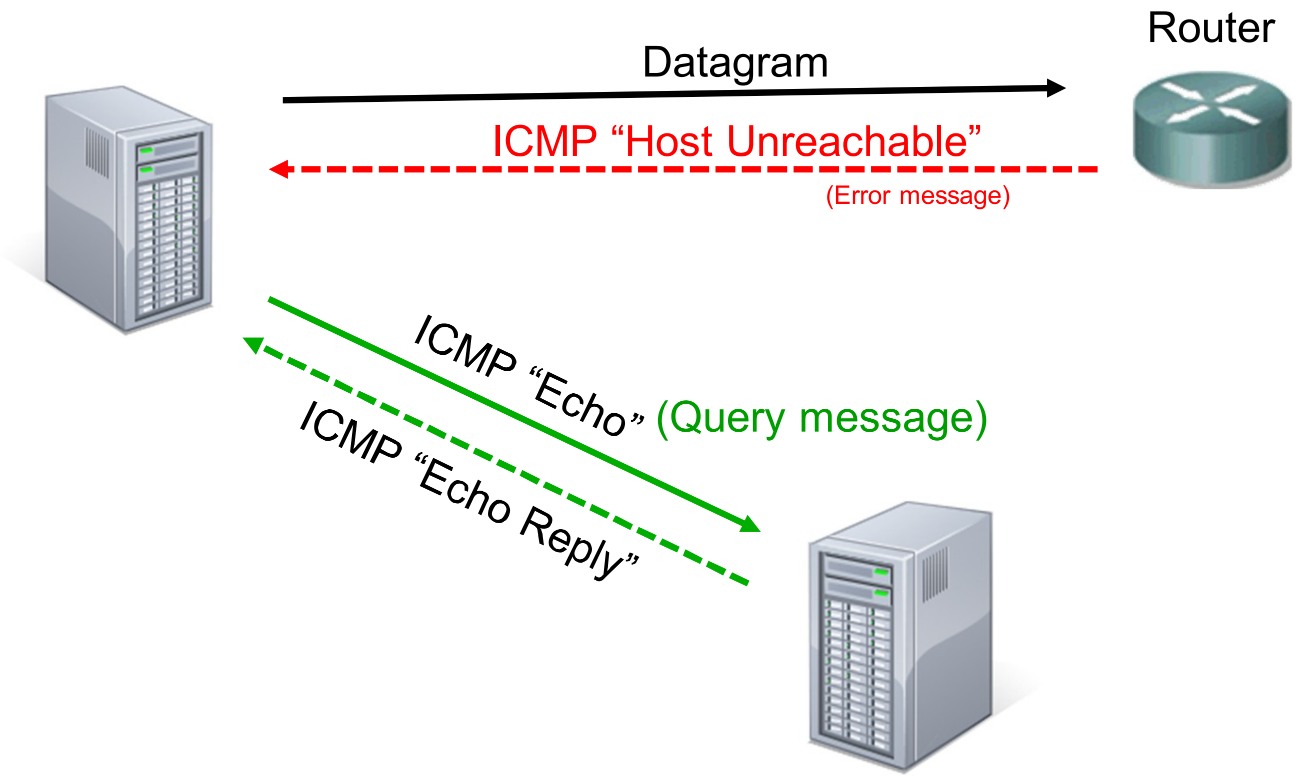
- The LSR and SSR (loose/strict source route) options have well-known security implications
  - Source route option: “send packets through the following networks”
  - Windows Vista and later (2007) refuse to accept datagrams with LSR/SSR options
  - Not used seriously today – most devices DROP it by default!
- The option can be used by an attacker to:
  - Reach otherwise unreachable systems
  - Send packets to RFC 1918 internal addresses (e.g. destination 10.0.0.1)
  - Avoid passing through links with firewalls/IDS systems
  - Establish connections in a stealthy way
  - Learn about the topology of a network
  - Perform bandwidth-exhaustion attacks (packets bouncing between systems)
- Also consider to drop all other datagrams with options in firewalls
  - Both for incoming and outgoing datagrams, many options are obsolete today



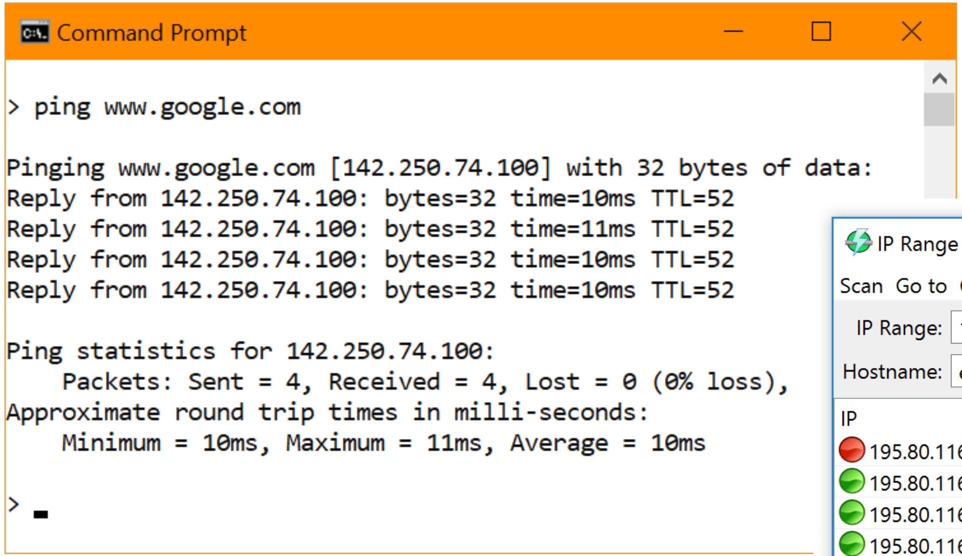
THE TTL FIELD AND ICMP

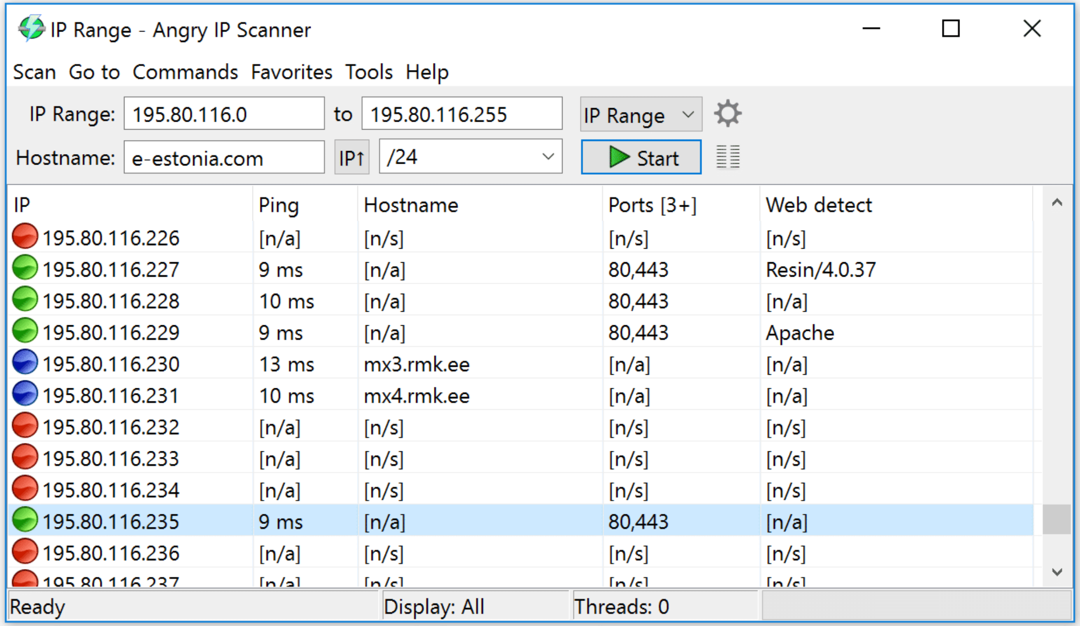
<table>
<tr><td>Version
(4 bits)</td><td>Header
Length
(4 bits)</td><td>QoS
(8 bits)</td><td colspan="3">Total Length
(16 bits)</td></tr>
<tr><td colspan="3">Fragment identification (16 bits)</td><td>Flags</td><td colspan="2">Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live
(8 bits)</td><td>Protocol (8 bits)</td><td colspan="3">Header Checksum (16 bits)</td></tr>
<tr><td colspan="6">Source IP Address (32 bits)</td></tr>
<tr><td colspan="6">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="5">Options (if any)</td><td>Padding</td></tr>
<tr><td colspan="6">Data....</td></tr>
</table>

Examples of ICMP messages



Ping – to find hosts





Sends ICMP echo messages
Receives ICMP echo reply messages

Network Mapping

- Subnet addresses can be figured out using ping to a broadcast address
  - If more than one reply is given, it is a broadcast address
  - Example: if ping to 62.2.15.8 replies with multiple ICMP echo replies
  - Automatically done by nmap
- Info about an address can also be received from the whois database:
  - www.ripe.net European registry
  - www.arin.net American registry
  - www.apnic.net Asian registry
- Example from ripe.net about 193.44.158.105 (www.telia.com):
  - `inetnum:        193.44.158.0 - 193.44.158.255`
  - `netname:        TELIANET`
  - `descr:          TeliaSonera Network Services`
  - `address:        Marbackagatan 11`
  - `address:        SE-123 86 Farsta`
  - `address:        Sweden`
  - `address:        Abuse and intrusion reports should`
  - `address:        be sent to: abuse@telia.com`
  - …

Traceroute to www.mit.edu

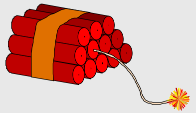
TTL = 1, 2, 3, …

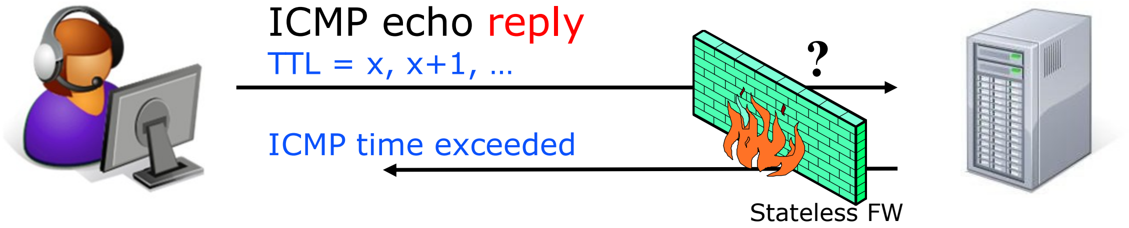
"ICMP time exceeded" sent by this node

```
% tracert www.nist.gov
Tracing route to www.glb.nist.gov [132.163.4.162] over a maximum of 30 hops:
  1     1 ms     1 ms    <1 ms  cth22a-gw.chalmers.se [129.16.22.1]
  2     1 ms     1 ms     1 ms  core1-ed-m-gw.chalmers.se [129.16.2.233]
  3     1 ms     1 ms     1 ms  optosunet-lr1-core1-gw.chalmers.se [129.16.2.193]
  4     8 ms     8 ms     8 ms  m1fre.sunet.se [193.11.0.1]
  5     9 ms     8 ms    13 ms  t1fre-ae5-v1.sunet.se [130.242.83.46]
  6     8 ms     8 ms     8 ms  se-fre.nordu.net [109.105.102.9]
  7    17 ms    23 ms    17 ms  dk-ore.nordu.net [109.105.97.130]
  8    27 ms    27 ms    27 ms  nl-sar.nordu.net [109.105.97.137]
  9   112 ms   112 ms   112 ms  us-man.nordu.net [109.105.97.139]
 10   106 ms   106 ms   106 ms  xe-2-3-0.118.rtr.newy32aoa.net.internet2.edu [109.105.98.10]
 11   139 ms   139 ms   139 ms  et-10-0-0.116.rtr.chic.net.internet2.edu [198.71.46.166]
 12   162 ms   161 ms   161 ms  ae0.3454.core-l3.frgp.net [192.43.217.223]
 13   190 ms   162 ms   162 ms  noaa-i2.frgp.net [128.117.243.11]
 14   162 ms   162 ms   164 ms  dsrc-rtr-xe-5-2-1-0.boulder.noaa.gov [140.172.2.26]
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
^C
```

ICMP echo messages are dropped by a router/firewall

TTL and firewalls – firewalking





- Internal systems can be probed and firewalls tested:
  - ICMP echo reply messages are forwarded by stateless firewalls since they may be replies to internal requests (ICMP echo messages)
  - Attacker sets TTL to expire somewhere after firewall to check for replies: ICMP time exceeded = firewall is stateless and one router found
- This technique is often called “firewalking”
  - Firewalls should not forward internal time exceeded replies to the outside
  - Firewalls should be stateful – don’t go for a cheaper solution

ICMP and firewalls

See RFC 5927 for more advise !

Always ALLOW:

- "Destination unreachable: fragmentation needed, DF set" [type 3, code 4]
  - Important information for us: Datagram is too large to be delivered
  - Operating systems performs "Path MTU Discovery" and expects a reply when a datagram is too big

Always DROP from internal network – allow from DMZ:

- "Echo request" [type 8], "Echo reply" [type 0] and "Time exceeded" [type 11]
- "Destination unreachable": net/host/protocol/port unreachable [type 3, code 0, 1, 2, 3]

Always DROP:

- "Source Quench" [type 4]
  - Means slow down. Now depreciated [RFC 6633]
- "Redirect" [type 5]
  - Tells host or router to send packets in another direction (through another network)
  - Attackers can place themselves in the path of a conversation and also create "black holes"
- Drop most other ICMP Messages exist (255 types) – see RFC 5927 (ICMP attacks against TCP)

Summary I

- Don’t rely on IP for security
  - Don’t authenticate a system or user based on the IP address only
- Always perform ingress and egress filtering
  - Remove reserved addresses, internal source address from outside, …
- Most fields in the IP header can be abused
  - Addresses can be spoofed
  - Fragmentation is complicated
  - IPv6 therefore does not allow intermediate nodes to do fragmentation!
  - Fragment ID may leak information
  - TTL can be abused (but also used as a security feature, see IDS lecture)
  - IP options are always dangerous (remove !)
- Make sure you are protected against old well-known attacks:
  - LAND, Teardrop, Ping-of-death, IDLE/Dumb, …
- ICMP: Filter incoming and outgoing messages with some care

Summary II

- The described ways to analyze and abuse IP is not limited to IP
  - It is valid for most protocols
  - All fields will be abused: false lengths, faked addresses, same sender as receiver address, …
  - We have not even discussed all fields (QoS)

- During protocol design, it is hard to foresee all possible side-effects
  - Fragmentation was really too complicated (size overflow, overlaps, holes, reassembly problems)
  - Fragment ID was a good idea, but turned out to be useful when probing systems for trust
  - Options were nice (?) but led to more problems than motivated by functionality

- With proper knowledge, it is possible to filter out IP-related problems in firewalls