Defending Computer Networks
Lecture 13: More NIDS

Stuart Staniford

Adjunct Professor of Computer Science

Logistics

- HW2
  - 48 hour extension to tomorrow midnight.
- HW3
  - Aiming to have it out next time
- Project descriptions
  - Also aim for next time
- Once projects are out, homeworks will get easier to compensate (no major coding).

Assigned Reading

- Ptacek and Newsham, *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*
  - https://sparrow.ece.cmu.edu/group/731-s08/readings/ptacek-newsham.pdf

J.P. Morgan's Cyber Attack: How The Bank Responded

J.P. Morgan Chase JPM -1.33% & Co. said Thursday that contact information for about 76 million households and about 7 million small businesses **was compromised in a cybersecurity attack** detected this summer. The attack went unnoticed for about two months until the bank caught wind. Here's how the attack and J.P. Morgan's response played out:

Mid-June: Hackers gain access to J.P. Morgan servers that housed user contact information of current and former customers who accessed chase.com or jpmorgan.com via web or mobile in past years, people familiar with the matter said. It's unclear how many years the information went back.

Mid-August: J.P. Morgan learns of the attack and stops it, identifying and closing all access paths over more than 90 servers, people familiar with the matter said. A couple hundred employees across J.P. Morgan's technology and cybersecurity teams begin working to examine data on any server that was compromised, led by a core team of around 20 bank executives, including COO Matt Zames. Mr. Zames begins to regularly brief bank management in weekly Monday morning operating committee meetings, people familiar with the meetings said.

August 27: Bloomberg and The Wall Street Journal report the Federal Bureau of Investigations is probing a possible computer hacking attack on J.P. Morgan and possibly other financial institutions. The FBI later says in a statement it is "working with the United States Secret Service to determine the scope of recently reported cyber attacks against several American financial institutions."

http://blogs.wsj.com/moneybeat/2014/10/03/j-p-morgans-cyber-attack-how-the-bank-responded/

J.P. Morgan Hackers Attempted to Infiltrate Other Financial Institutions

Federal Officials Asked Banks, Other Institutions to Check For Cyberattack Signs

Hackers who breached J.P. Morgan Chase JPM -1.61% & Co.'s computer network earlier this year also tried to infiltrate a number of other financial institutions, but the companies believe they were unsuccessful, according to people familiar with the investigation.

Federal officials asked a group of large banks and other financial institutions last month to check if they had seen indicators associated with the cyberattack that resulted in the theft of account information for millions of J.P. Morgan customers this summer, these people said.

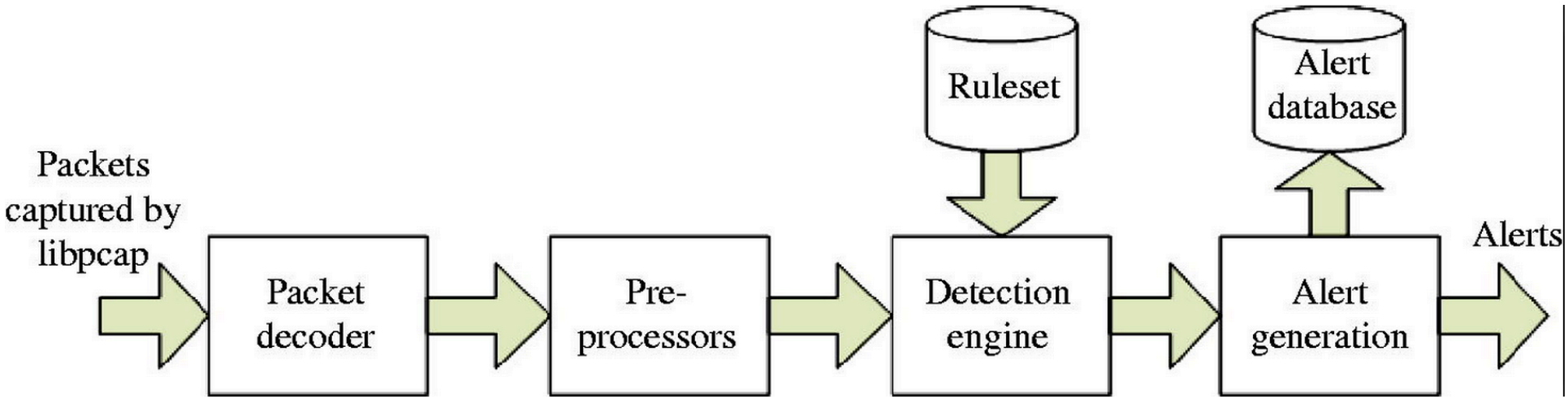
A number of financial institutions responded that they had seen traffic from the suspect computer addresses linked to the hackers, but that they didn't believe they had been breached, the people said.

Rather, the hackers, whose identity remains unknown, appeared to be "probing," or searching for weaknesses on the firms' digital perimeters.

Main Goals for Today

- More Network Intrusion Detection

Overall Snort Architecture



Snort Detection Engine Data Structure

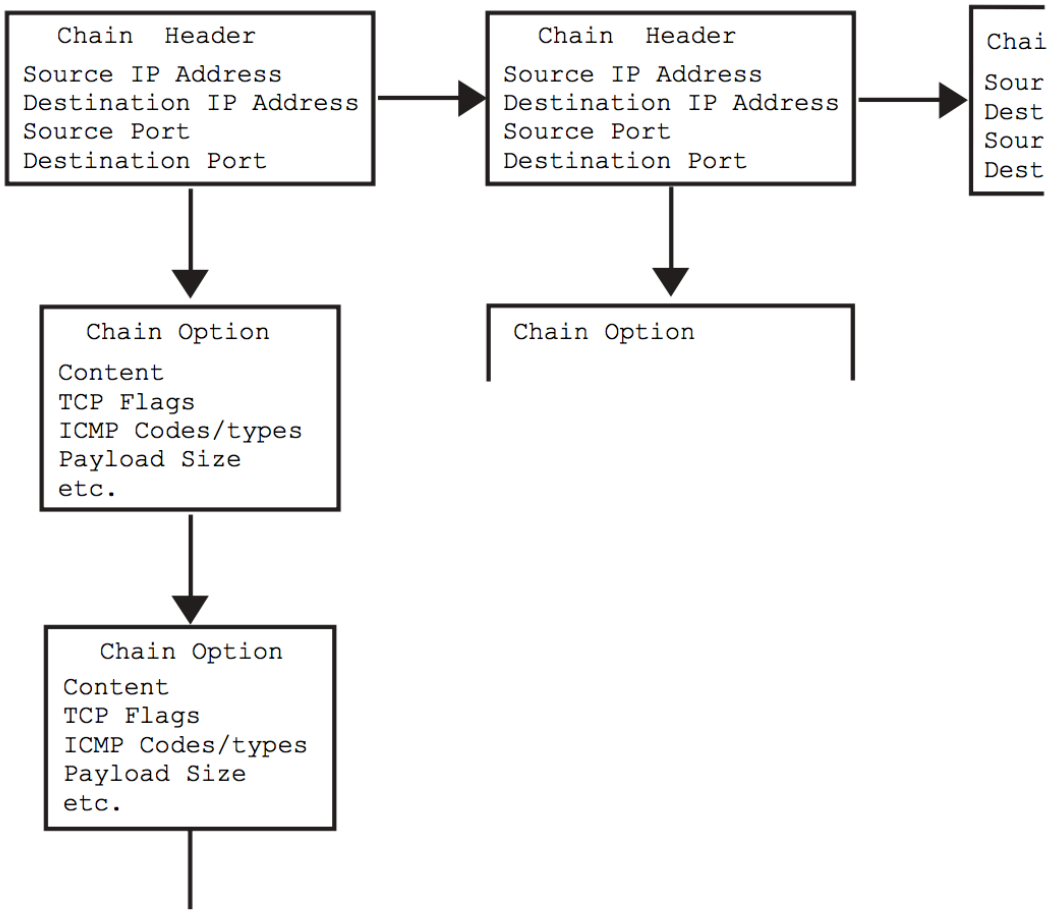


Figure 3: Rule Chain logical structure.

Snort Rule Example 1

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SERVER-WEBAPP HyperSeek hsx.cgi directory traversal attempt"; flow:to_server,established; content:"/hsx.cgi"; http_uri; content:"../../"; http_raw_uri; content:"%00"; distance:1; http_raw_uri; metadata:ruleset community, service http; reference:bugtraq,2314; reference:cve,2001-0253; reference:nessus,10602; classtype:web-application-attack; sid:803; rev:21;)

Snort Rule Example 2

alert tcp **$EXTERNAL_NET $HTTP_PORTS -> $HOME_NET** any (msg:"*EXPLOIT-KIT Multiple exploit kit Payload detection - readme.exe*"**; flow:to_client,established**; content:"**filename=";** http_header; content:**"readme.exe"**; within:12; fast_pattern; http_header; **content:"|0D 0A|"**; within:4; http_header; metadata:policy balanced-ips drop, policy security-ips drop, service http; reference:cve,2006-0003; reference:cve,2007-5659; reference:cve,2008-0655; reference:cve,2008-2992; reference:cve,2009-0927; reference:cve,2010-1885; reference:cve,2011-0559; reference:cve,2011-2110; reference:cve,2011-3544; reference:cve,2012-0188; reference:cve,2012-0507; reference:cve,2012-1723; reference:cve,2012-1889; reference:cve,2012-4681; reference:url,blog.webroot.com/2011/10/31/outdated-operating-system-this-blackhole-exploit-kit-has-you-in-its-sights/; classtype:trojan-activity; sid:25387; rev:3;)

Snort Content Modifiers

- Offset (start looking n bytes into packet/flow)
- Depth (stop looking n bytes into packet/flow)
- Distance (start looking n bytes from previous match)
- Within (stop looking n bytes from previous match
- Nocase (ignore case in matching)

Snort Rule Example 3

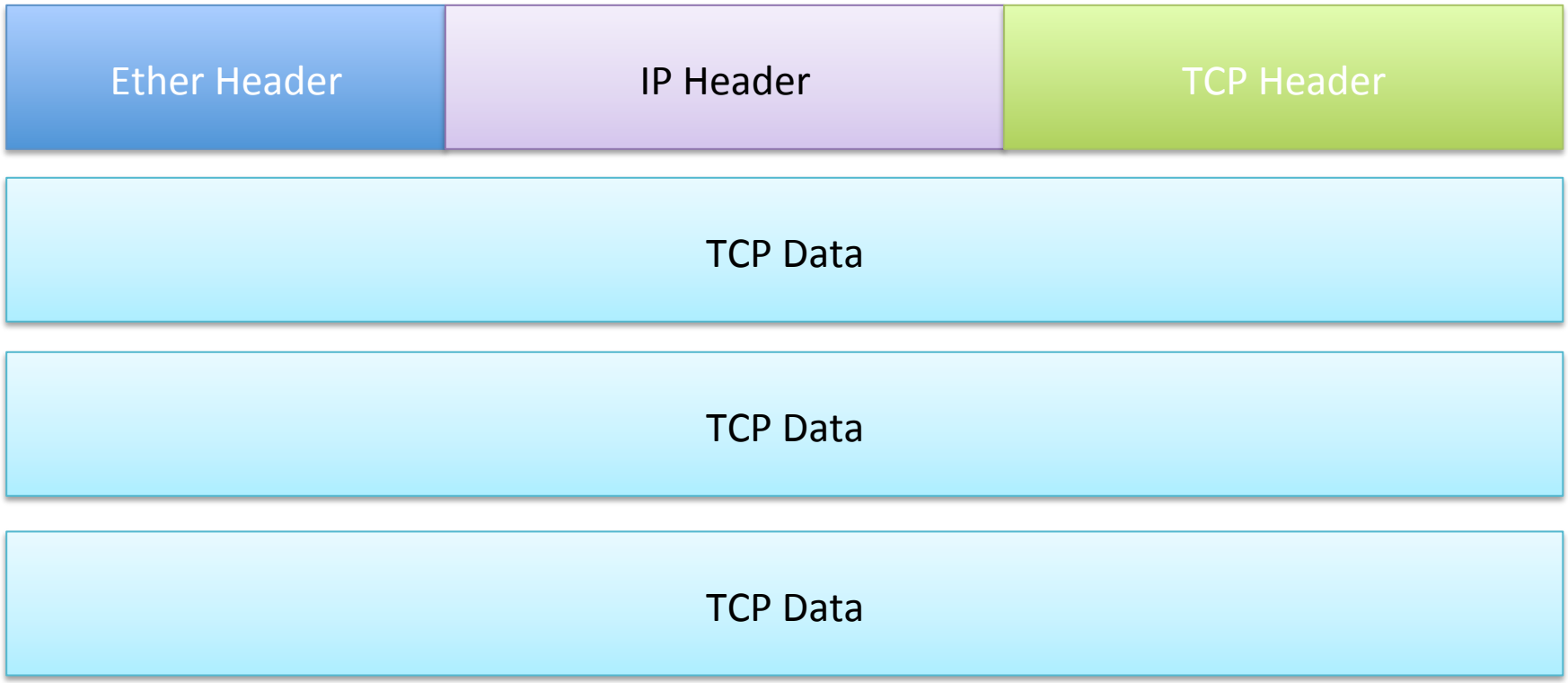
alert tcp **$EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any**
(msg:"*BROWSER-IE IE5 compatibility mode use after free attempt*";
flow:to_client,established; file_data; content:**"meta http-equiv=|22|X-UA-Compatible|22| content=|22|IE=5|22|"**; fast_pattern:only;
content:**".runtimeStyle.setExpression"**;
content:**"document.body.innerHTML"**; metadata:policy balanced-ips drop, policy security-ips drop, service http; reference:cve,2013-3121;
reference:url,technet.microsoft.com/en-us/security/bulletin/MS13-047;
classtype:attempted-user; sid:26851; rev:3;)

Background on Example 3

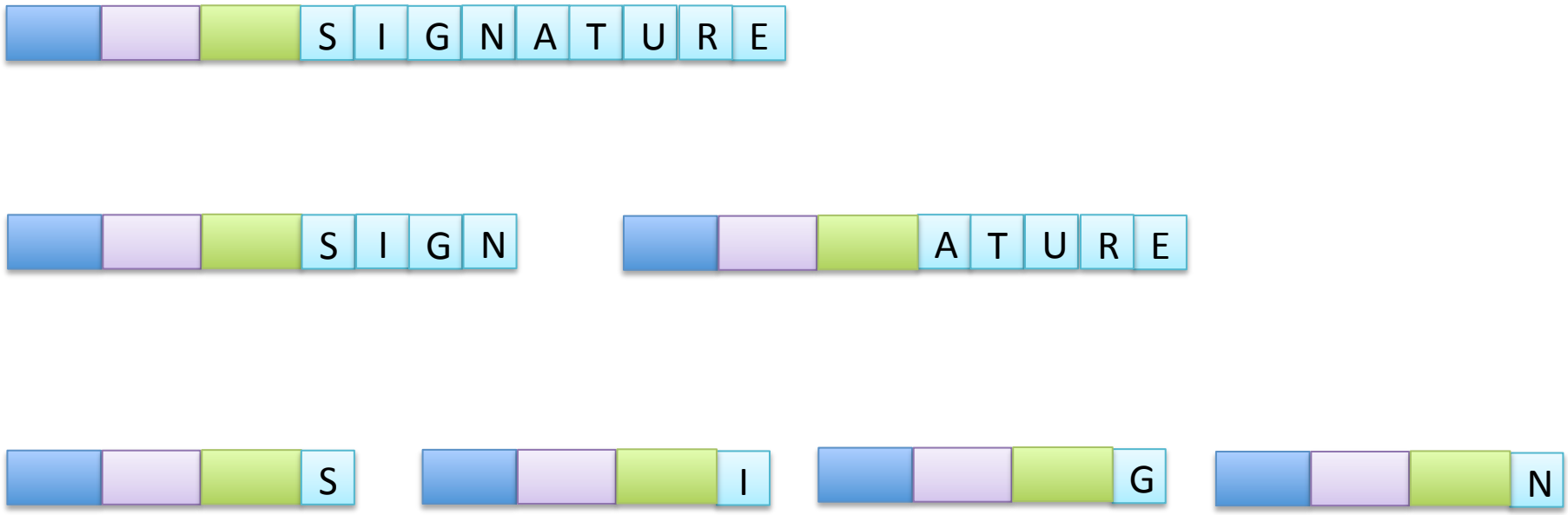
- http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3121
- http://www.securityfocus.com/bid/60390
- http://technet.microsoft.com/en-us/security/bulletin/ms13-047

A remote code execution vulnerability exists when Internet Explorer improperly processes script while debugging a webpage. The vulnerability may corrupt memory in a way that could allow an attacker to execute arbitrary code in the context of the current user within Internet Explorer. An attacker could host a specially crafted website that is designed to exploit this vulnerability through Internet Explorer and then convince a user to view the website.

Evading NIDS: TCP

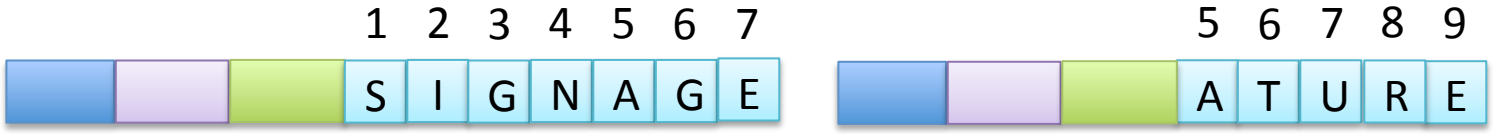


Variants



Clearly we have to reassemble TCP before looking for “SIGNATURE”

But what about this case?



In Snort

- Stream5 preprocessor
  - Buffers packets and reassembles stream
  - Passes onto detection engine
  - Target based
    - changes behavior according to target OS
    - Static configuration
  - Can detect reassembly anomalies
    - But off by default
    - Probably too many fps

`policy <policy_id>`	The Operating System policy for the target OS. The policy_id can be one of the following:

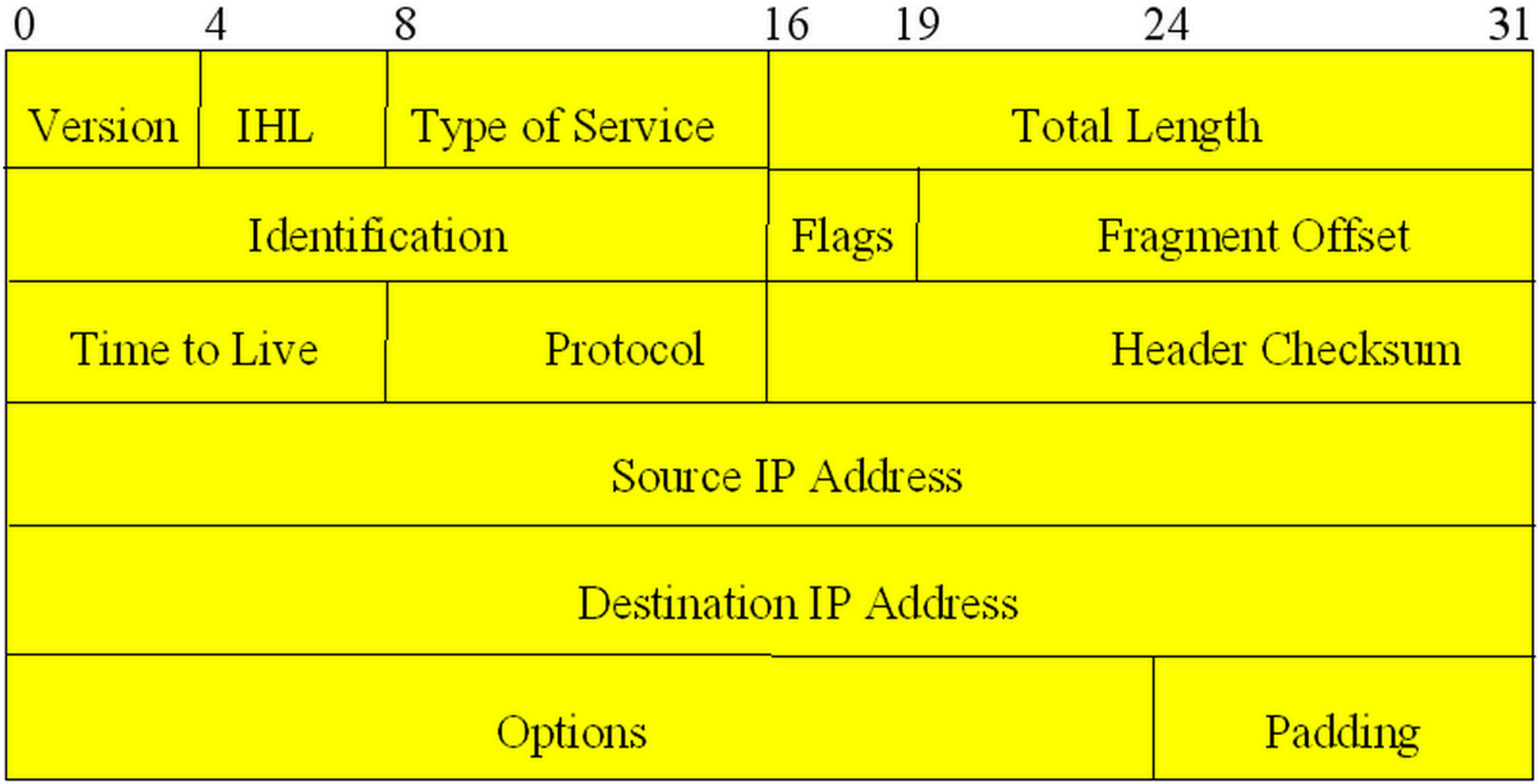
Policy Name	Operating Systems.
`first`	Favor first overlapped segment.
`last`	Favor first overlapped segment.
`bsd`	FresBSD 4.x and newer, NetBSD 2.x and newer, OpenBSD 3.x and newer
`linux`	Linux 2.4 and newer
`old-linux`	Linux 2.2 and earlier
`windows`	Windows 2000, Windows XP, Windows 95/98/ME
`win2003`	Windows 2003 Server
`vista`	Windows Vista
`solaris`	Solaris 9.x and newer
`hpux`	HPUX 11 and newer
`hpux10`	HPUX 10
`irix`	IRIX 6 and newer
`macos`	MacOS 10.3 and newer

Snort flow sub-keywords

Option	Description
`to_client`	Trigger on server responses from A to B
`to_server`	Trigger on client requests from A to B
`from_client`	Trigger on client requests from A to B
`from_server`	Trigger on server responses from A to B
`established`	Trigger only on established TCP connections
`not_established`	Trigger only when no TCP connection is established
`stateless`	Trigger regardless of the state of the stream processor (useful for packets that are designed to cause machines to crash)
`no_stream`	Do not trigger on rebuilt stream packets (useful for dsize and stream5)
`only_stream`	Only trigger on rebuilt stream packets
`no_frag`	Do not trigger on rebuilt frag packets
`only_frag`	Only trigger on rebuilt frag packets

http://manual.snort.org/node33.html#SECTION00469000000000000000

Evading NIDS: Fragmentation



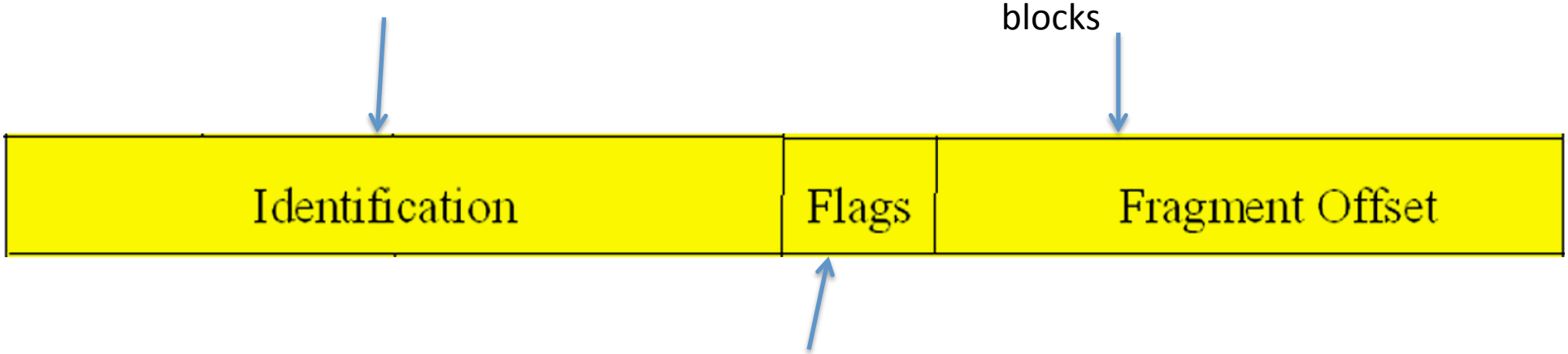
Reason for Fragmentation

- 2 byte length: 64kB IP packet
  - Actually more through special jumbo options
- Physical layers generally smaller
- Historically endpoints would not know MTU size in middle
  - "MTU discovery" nowadays.
- So if a packet too big for physical network arrives at router
  - Need to split it into pieces

How Fragmentation Works

All fragments of a given packet have same id

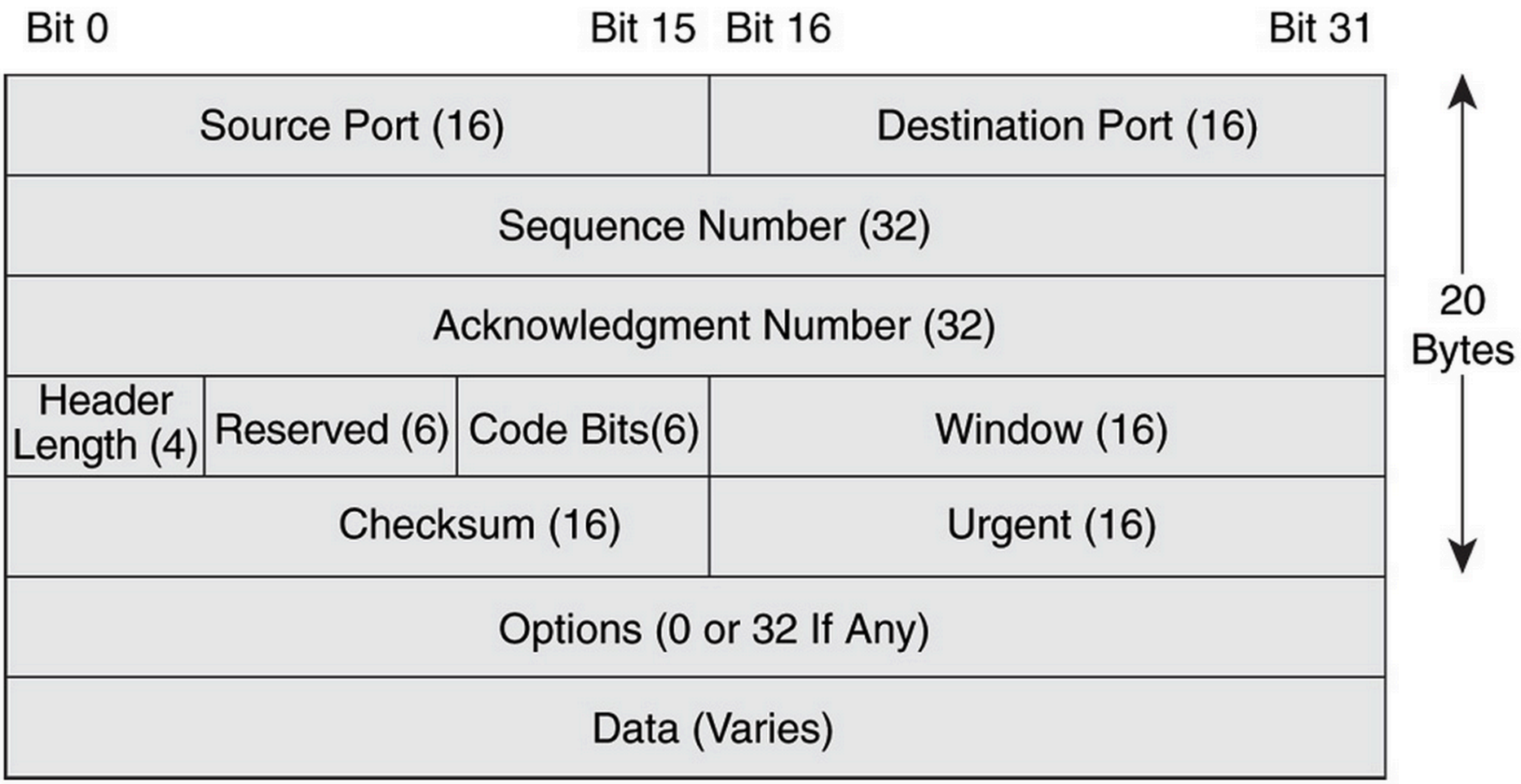
Offset of this segment in the original ip data, in eight byte blocks



MF flag bit says whether to expect any more packets

Note that fragmentation can be used to break up the TCP header, not just the TCP data

TCP Header in Fragmentation



Using Fragmentation for Evasion

- IDS must reassemble fragments before doing TCP processing
  - Can look for signs of abusive fragmentation
- Overlapping fragments are host dependent
  - Possibility of evasion
- Hosts will timeout partial fragment streams
  - IDS must match host timeout behavior

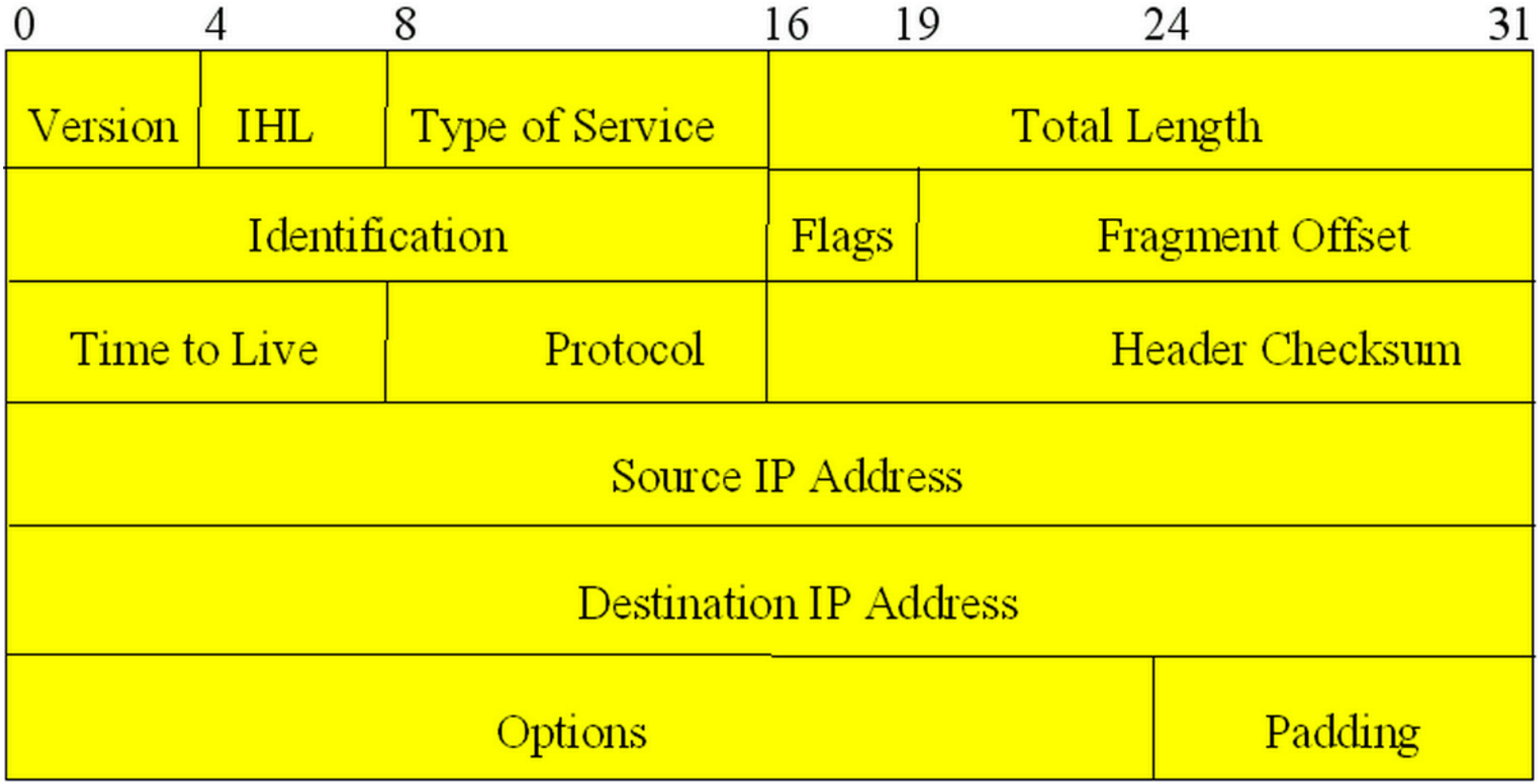
Snort solution

- Frag3 reassembly preprocessor
  - Buffer and reassemble fragments
  - Has to come before TCP reassembly
    - Since cannot even reliably infer tcp header until defragging is done
  - Target based
    - Again, based on a static policy
  - Can alert on anomalies

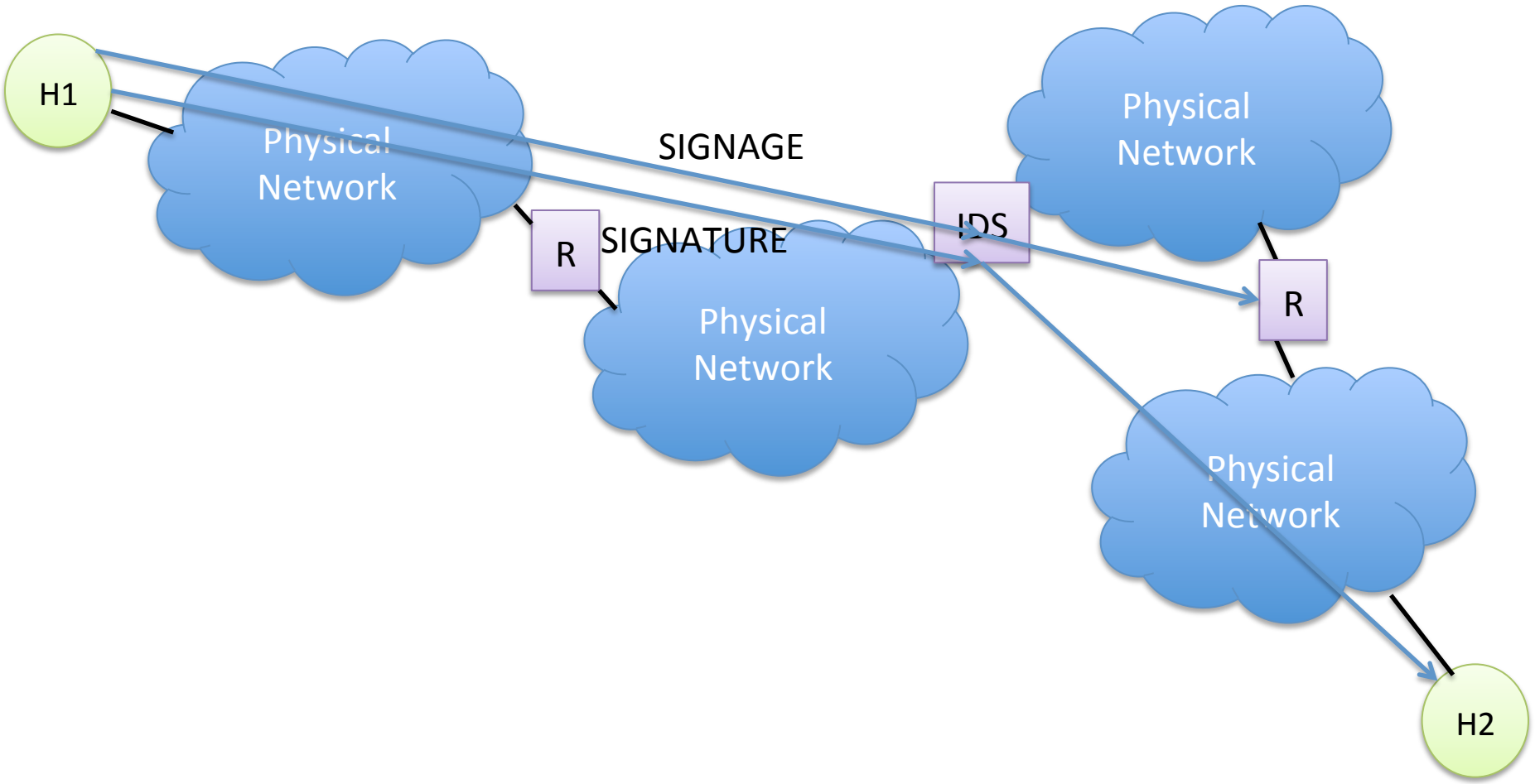
Evading NIDS: Mac address

- Only works if on same L2 network as NIDS
- Add extra packets directed to bad Mac address
  - But with correct destination IP
  - If IDS is not careful, it will process promiscuously
    - Where end-client won't
- Note there are possible legit reasons for Mac address to change during a connection
  - Eg route flapping
  - So just looking for a changing Mac will have some FPs.

Evading NIDS: TTL



Evading NIDS: TTL Field

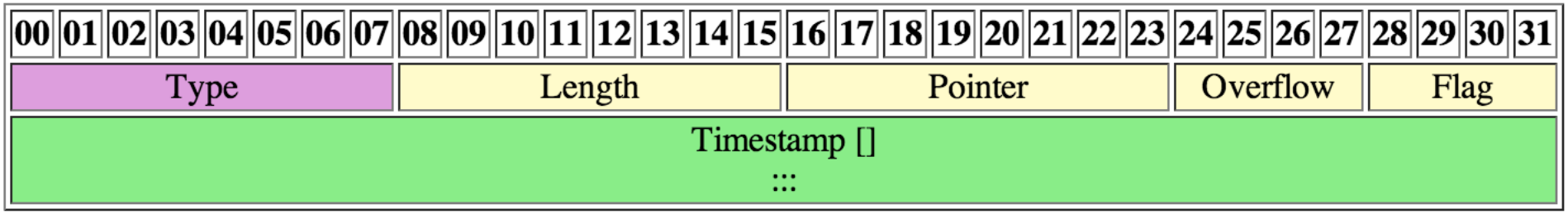


Fragmentation Variant Strategy

- Similar to TTL
- There is a DF bit in “Flags” field in IP header
- Means “Don’t Fragment”
- On certain packets, set this then set packet size greater than MTU at some part of route
- Routers will drop those packets, not deliver
- Can be used as an evasion strategy

IP Timestamp Option Evasion

- IP Options allow additional fields to be added to IP packet header
  - For special purposes
  - IHL field > 5 signals presence of options
- Timestamp recording (RFC 781)
- Packet will be dropped if timestamp option malformed



Effects of Evasions

- Force the IDS to know a great deal about the network
  - Distance to end points (TTL)
  - MTUs in physical networks (DF bit)
  - Nature of end-client (reassembly algorithms)
- OTOH
  - Many of these strategies are themselves somewhat suspicious
  - IDS can use them as evidence
    - maybe, care needed on FPs

Strategies for Defeating Evasions

- Target based
  - IDS needs to figure out nature of all machines on network
    - Active fingerprinting (integration with vuln scanner)
    - Passive fingerprinting
    - Manual, static
      - not scalable unless network pretty homogeneous
  - Do TCP, Frag, etc reassembly however appropriate
    - IDS implementors have a lot of work to do

Strategies for Evasions (2)

- Normalization
  - If IDS is inline (IPS = Intrusion Prevention System)
  - Then IPS can rewrite packet stream to make it unambiguous
  - Solves problem pretty well in principle
  - Places different set of demands on IPS
    - Better not break anything in rewriting those packets!
    - Latency
    - Reliability – MTF
      - Disks on box
    - Typically customers start in non-inline mode, and then move to inline as they gain confidence