

# Defending Computer Networks

## *Lecture 12: NIDS*

Stuart Staniford

Adjunct Professor of Computer Science

# Logistics

- Quiz 2 next Tuesday (10/13/15)
- Quiz 1 graded (in CMS, papers in front)
  - Range 5-11, mean 9.4 +/- 1.4
- Go through answers

# Assigned Reading

- Roesch, M. Snort – *Lightweight Intrusion Detection for Networks*
- [http://static.usenix.org/publications/library/proceedings/lisa99/full\\_papers/roesch/roesch.pdf](http://static.usenix.org/publications/library/proceedings/lisa99/full_papers/roesch/roesch.pdf)

# Nuclear power plants in ‘culture of denial’ over hacking risk

Sam Jones, Defence and Security Editor

[Share](#) ▾

[Author alerts](#) ▾

[Print](#)

[Clip](#)

[Comments](#)



Nuclear power plants around the world are harbouring a “culture of denial” about the risks of cyber hacking, with many failing to protect themselves against digital attacks, a review of the industry has warned.

A focus on safety and high physical security means that many nuclear facilities are blind to the risks of cyber attacks, according to the report by think-tank Chatham House, citing 50 incidents globally of which only a handful have been made public.

# Seoul suspects N. Korea in subway cyber attack

SEOUL • North Korea is suspected of having launched a cyber attack last year on the South Korean capital's subway system that carries millions of commuters every day, a Seoul lawmaker said yesterday, citing intelligence reports.

The attack, staged between March and August, affected several servers of Seoul Metro, which runs four major subway lines, ruling party legislator Ha Tae Kyung said.

# Main Goals for Today

- Network Intrusion Detection

# Network Intrusion Detection

- Basic idea:
  - Examine network traffic looking for evidence of attacks.
  - Idea is not to impose policy (firewall)
  - But specifically detect/id/block attacks.

# Example NIDS Rule

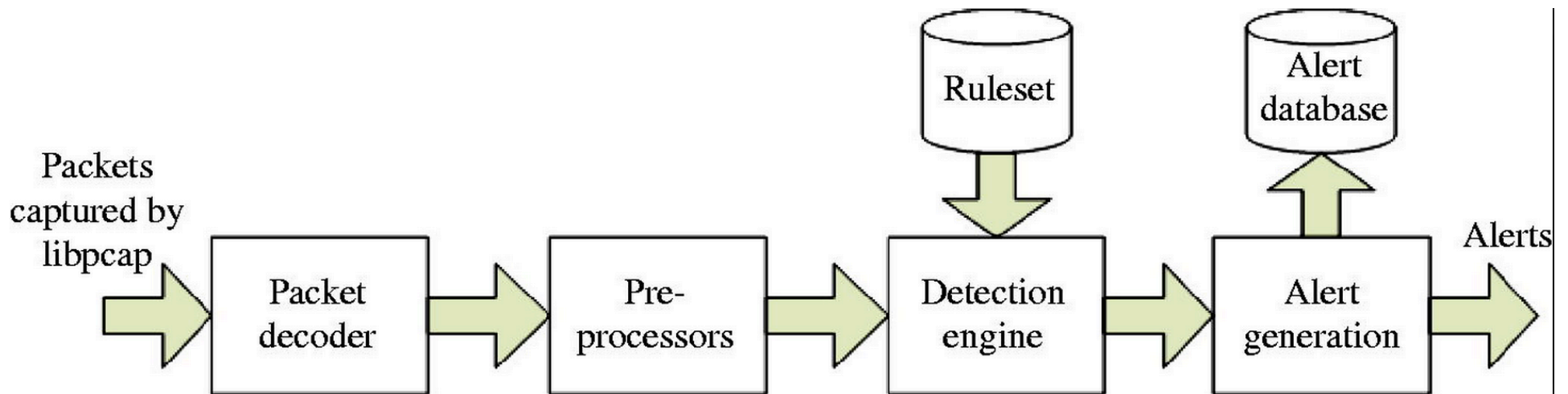
- alert ip \$EXTERNAL\_NET any -> \$HOME\_NET any (msg:"INDICATOR-SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; fast\_pattern:only; metadata:ruleset community; classtype:shellcode-detect; sid:648; rev:14;)



# High Points of NIDS History

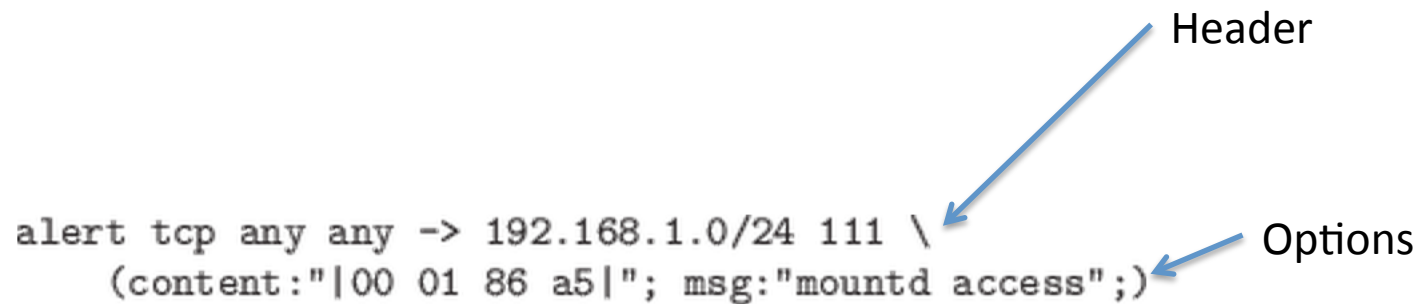
- Heberlein et al NSM – 1989
- Wheelgroup NetRanger - 1995
- Snort – 1998
- Intruvert – 2000
- FireEye – 2004 (but really 2007)
- Focus on Snort here, as conveniently accessible.

# Overall Snort Architecture



# Anatomy of a Snort Rule

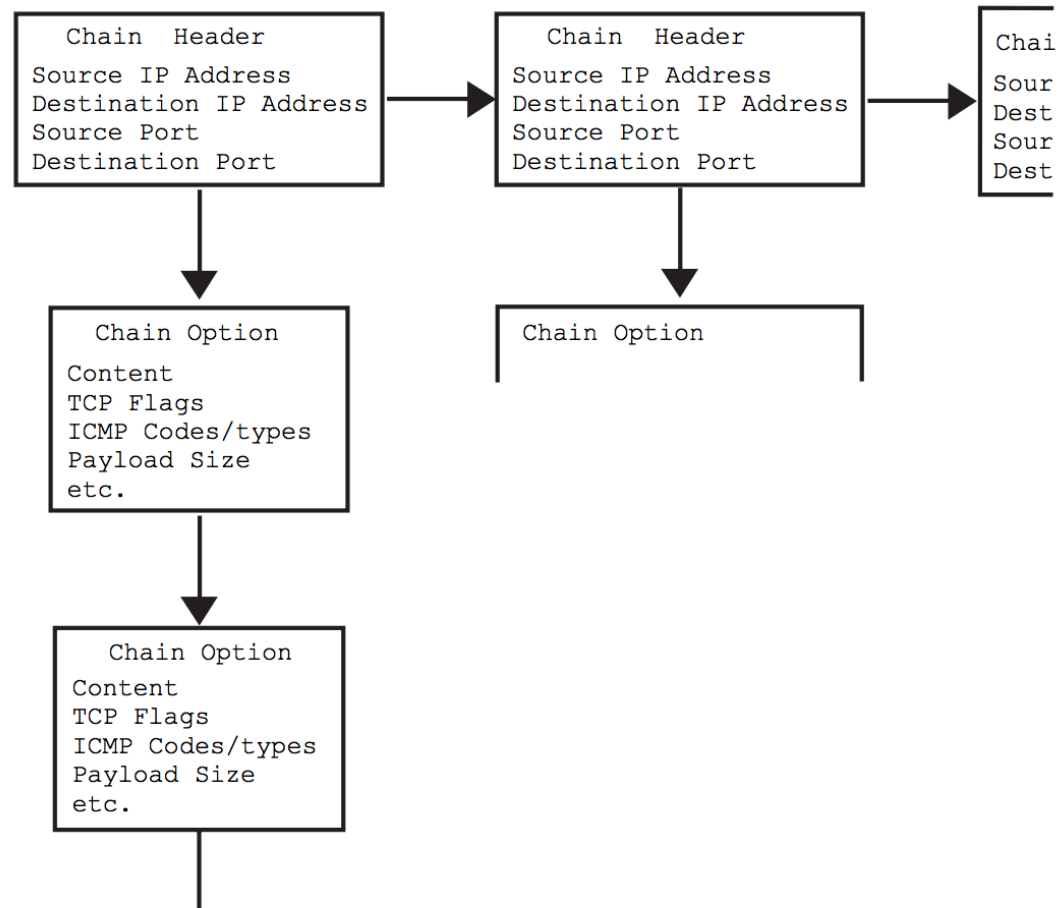
```
alert tcp any any -> 192.168.1.0/24 111 \
  (content:"|00 01 86 a5|"; msg:"mountd access");
```



The diagram shows a Snort rule with two blue arrows pointing to specific parts of the rule. One arrow points from the word 'Header' to the backslash character at the end of the first line of the rule. The other arrow points from the word 'Options' to the opening parenthesis of the options list in the second line of the rule.

**Figure:** Sample Snort Rule

# 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;)
```

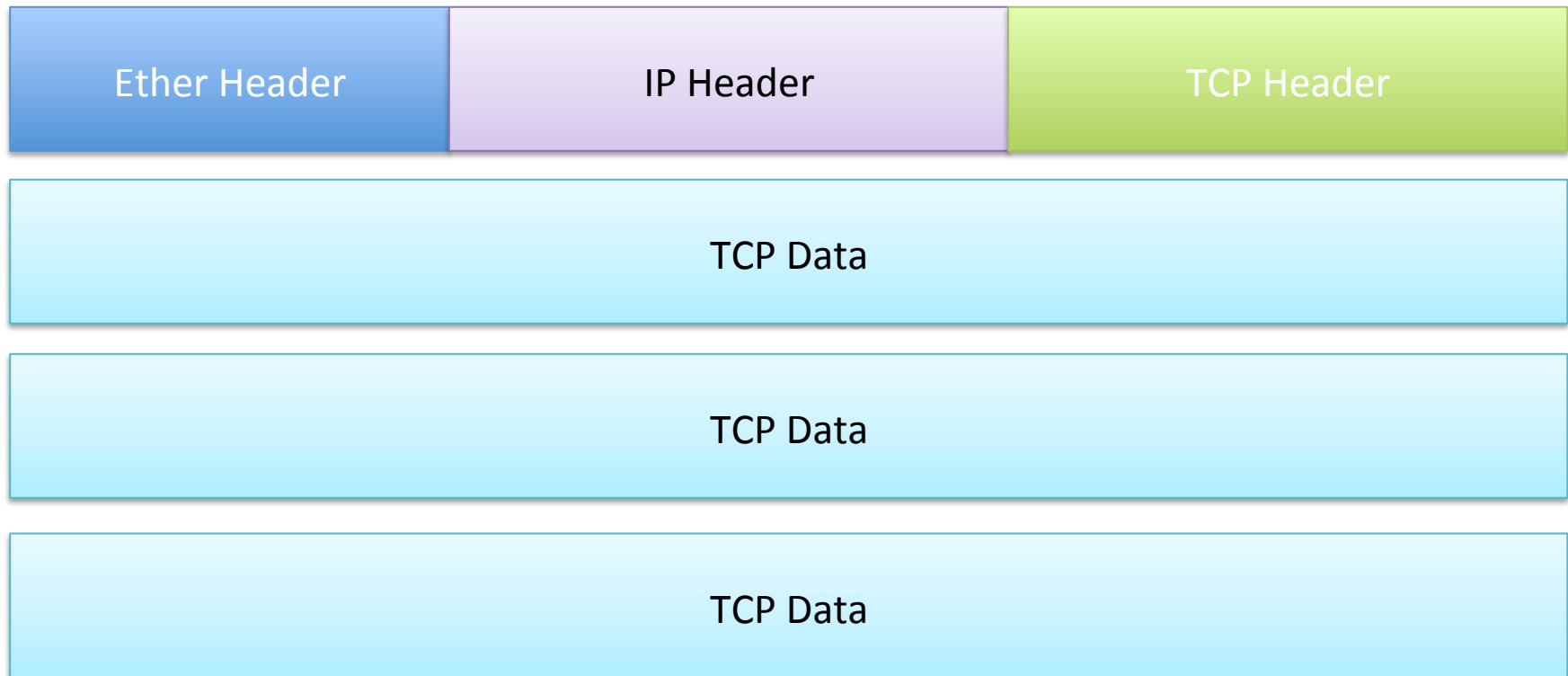


# Background on Example 3

- <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"



# 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 last 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

0	4	8	16	19	24	31
Version	IHL	Type of Service	Total Length			
Identification			Flags	Fragment Offset		
Time to Live	Protocol		Header Checksum			
Source IP Address						
Destination IP Address						
Options					Padding	

# 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

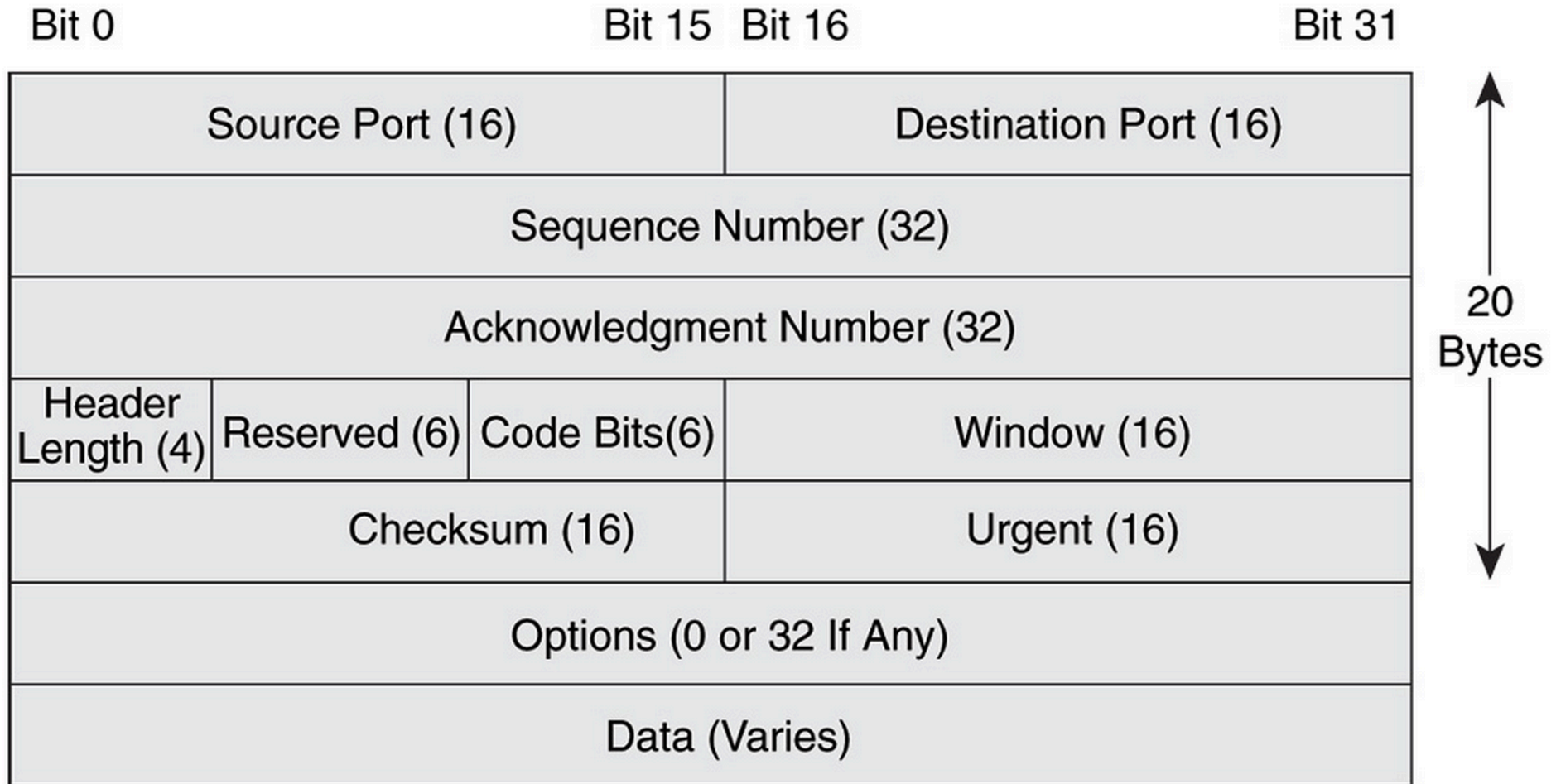
Offset of this segment in the original ip data, in eight byte blocks



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

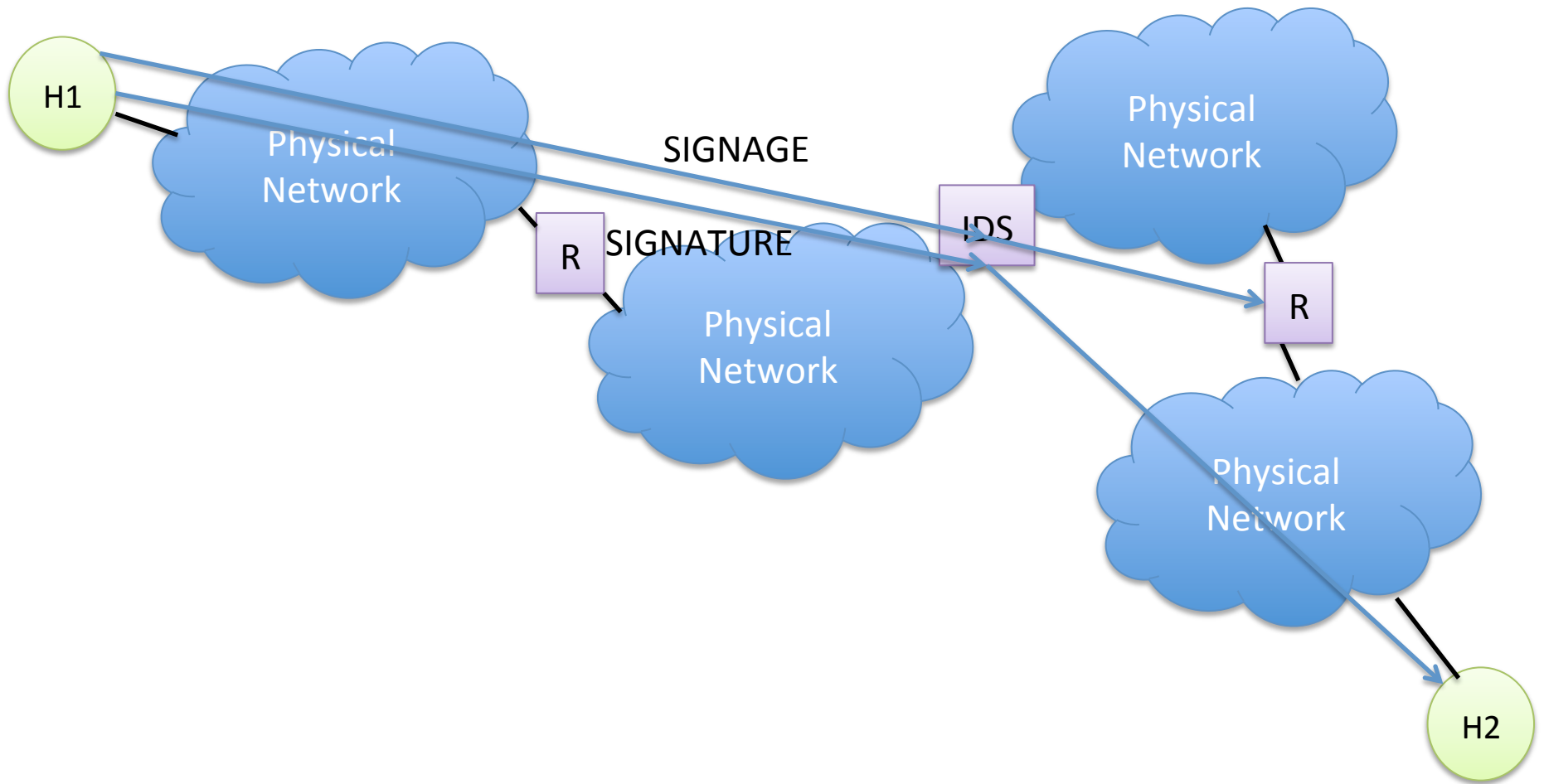
# Evading NIDS: Mac address

- 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

0	4	8	16	19	24	31
Version	IHL	Type of Service	Total Length			
Identification			Flags	Fragment Offset		
Time to Live	Protocol		Header Checksum			
Source IP Address						
Destination IP Address						
Options					Padding	

# Evading NIDS: TTL Field



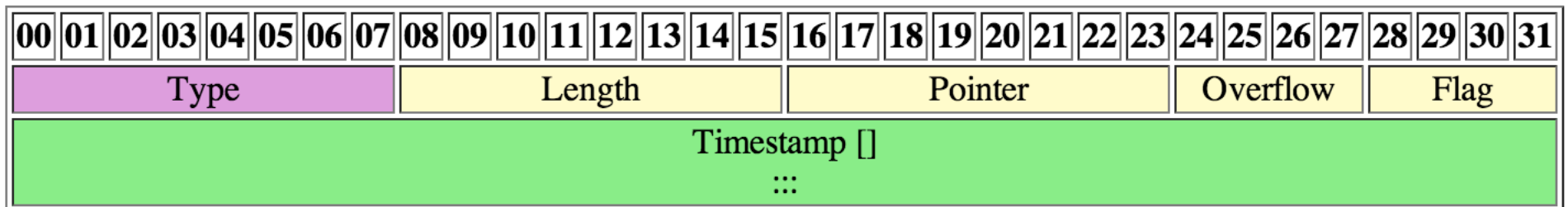
# Fragmentation Variant Strategy

- 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