

Defending Computer Networks

Lecture 11: DDOS/NIDS

Stuart Staniford

Adjunct Professor of Computer Science

Logistics

- HW2 is due tomorrow
- HW1 almost graded...

Assigned Reading

- Roesch, M. Snort – *Lightweight Intrusion Detection for Networks*
 - http://static.usenix.org/publications/library/proceedings/lisa99/full_papers/roesch/roesch.pdf
- 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>

Did hacker taunt Rutgers over latest cyber attack?

NEW BRUNSWICK — An alleged hacker appeared to taunt Rutgers University officials Monday as a cyber attack paralyzed the school's computer network.

The university was hit with a "denial of service" attack Monday morning that crashed Rutgers' websites and cut off internet and wifi access to tens of thousands of students, faculty and employees.

An alleged hacker who uses the screen name Exfocus **took credit for several similar attacks on Rutgers' computer networks** during the 2014-2015 school year.

'Digital India' making India a 'strategic' cyber attack target: Report

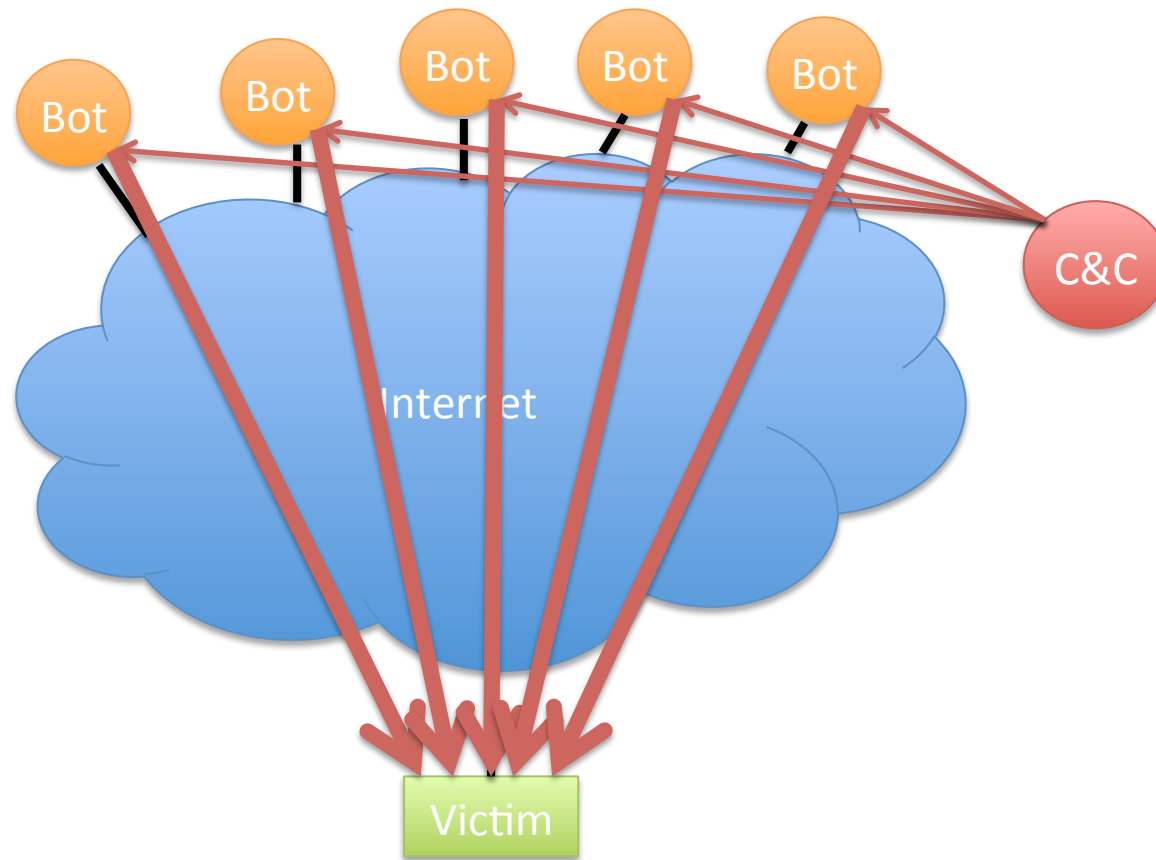
A FireEye report found that 38 percent of organisations in India were exposed to targeted advanced persistent attacks in the first half of 2015, a 23 percent increase from the previous report.

"Geopolitical tensions and digitization in the region have steadily ratcheted up in recent months, and cyber activity reflects this," the security firm said.

Main Goals for Today

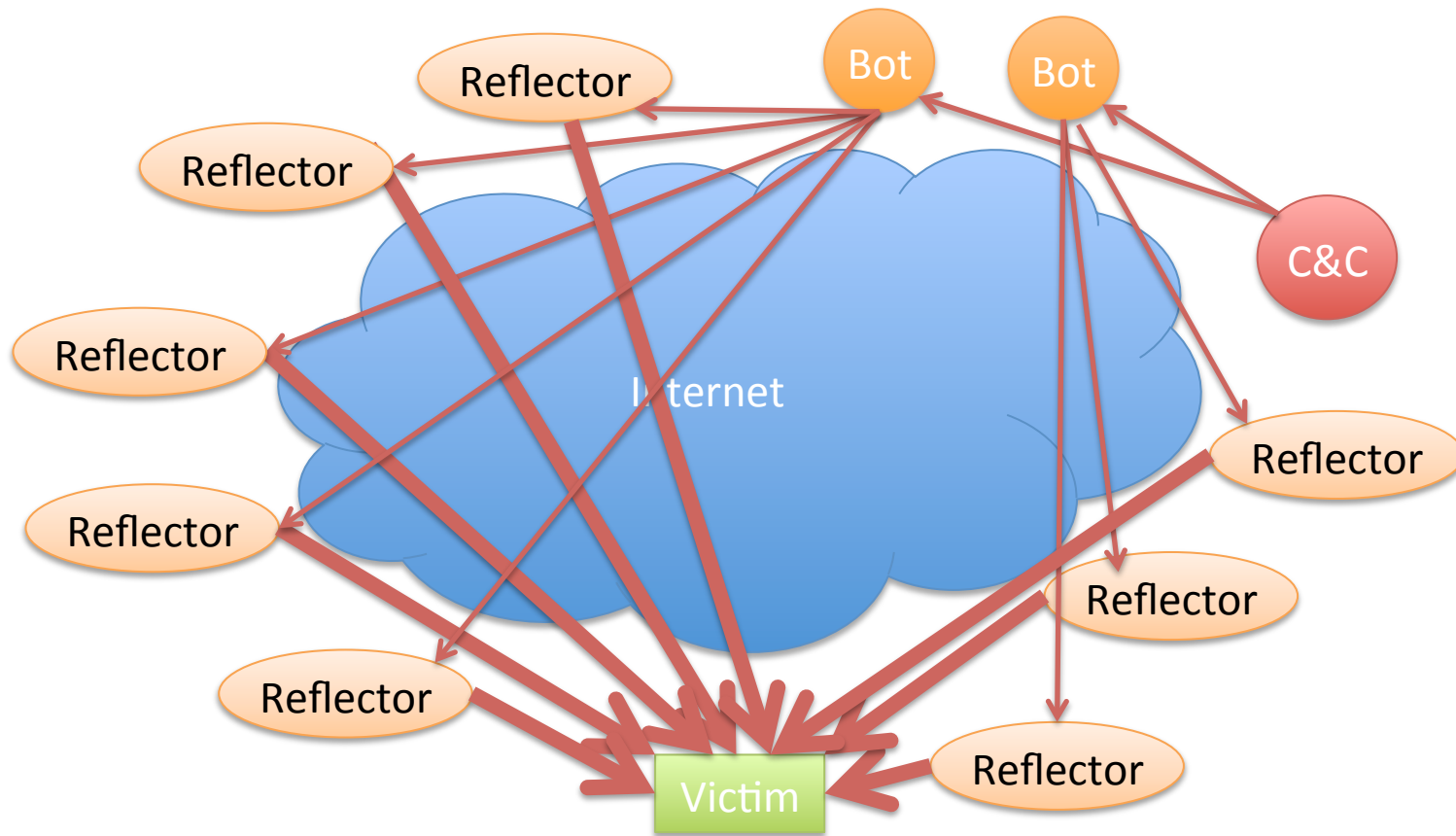
- Distributed Denial of Service.
- Start NIDS.

Basic Setup of a DDOS Botnet



Illustrative only: practical attacks will have many more bots

Reflection Attacks

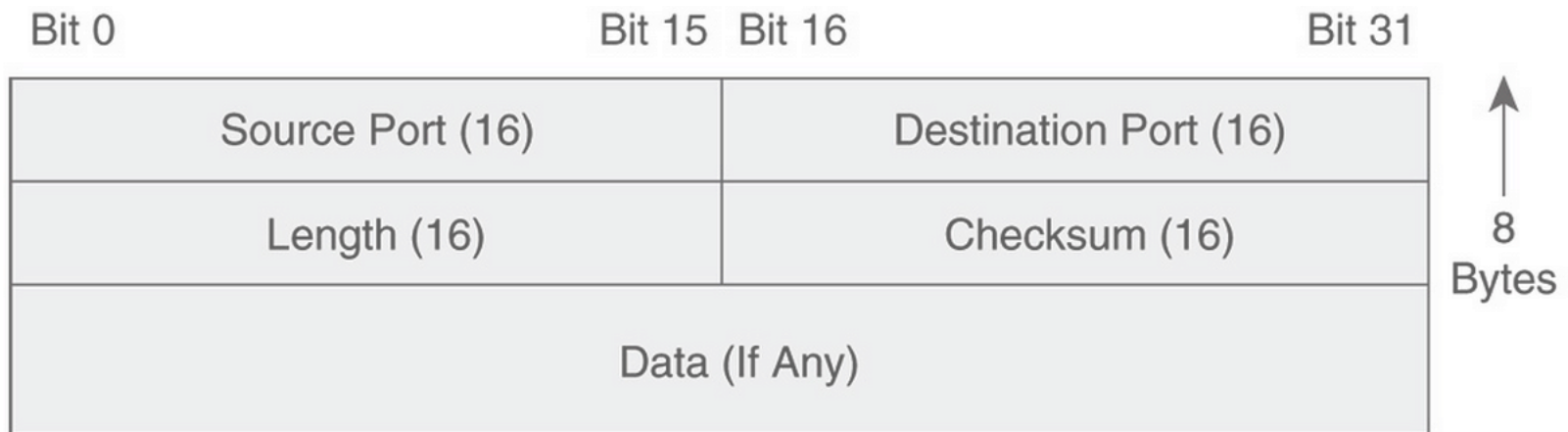


Illustrative only: practical attacks will have many more bots/reflectors

What Will Work as a Reflector?

- Any TCP host (send SA or R in response to S)
- ICMP (eg echo response to echo request)
- DNS – especially with recursion
 - Issue on campus recently
 - Let's look at this in more detail

UDP

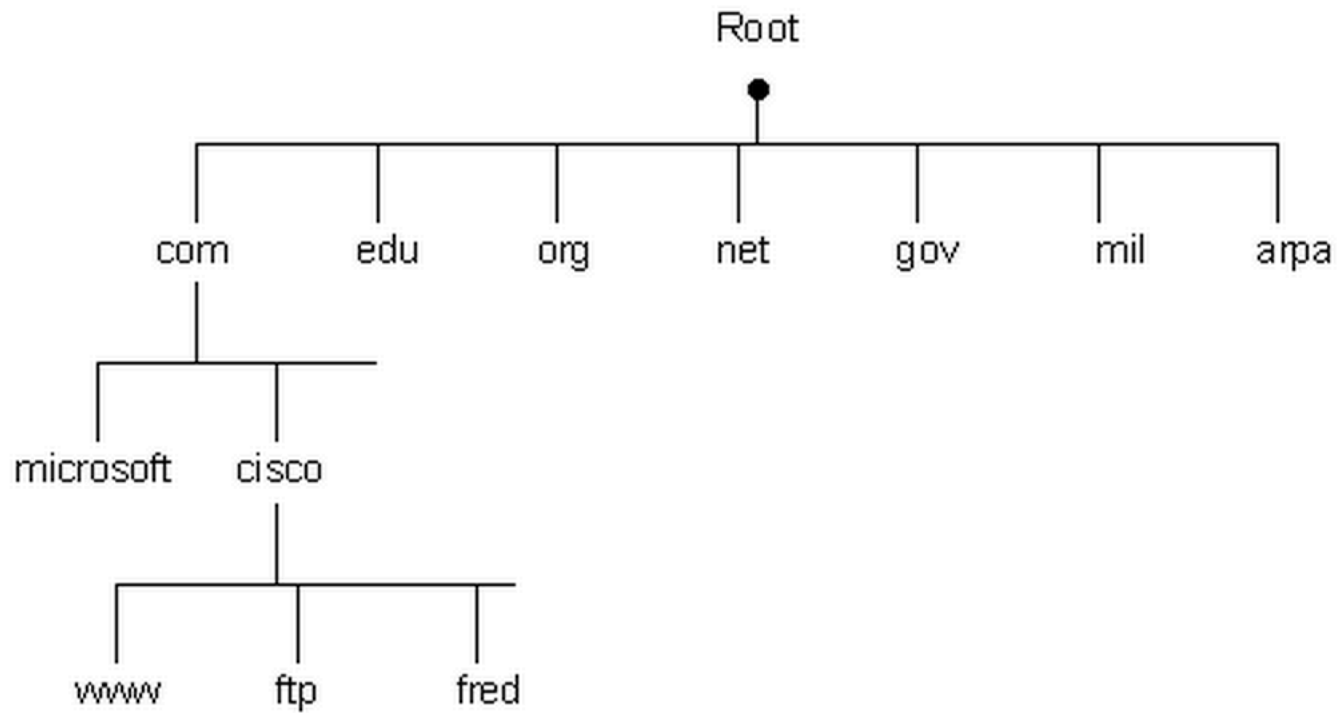


No Sequence Or Acknowledgment Fields

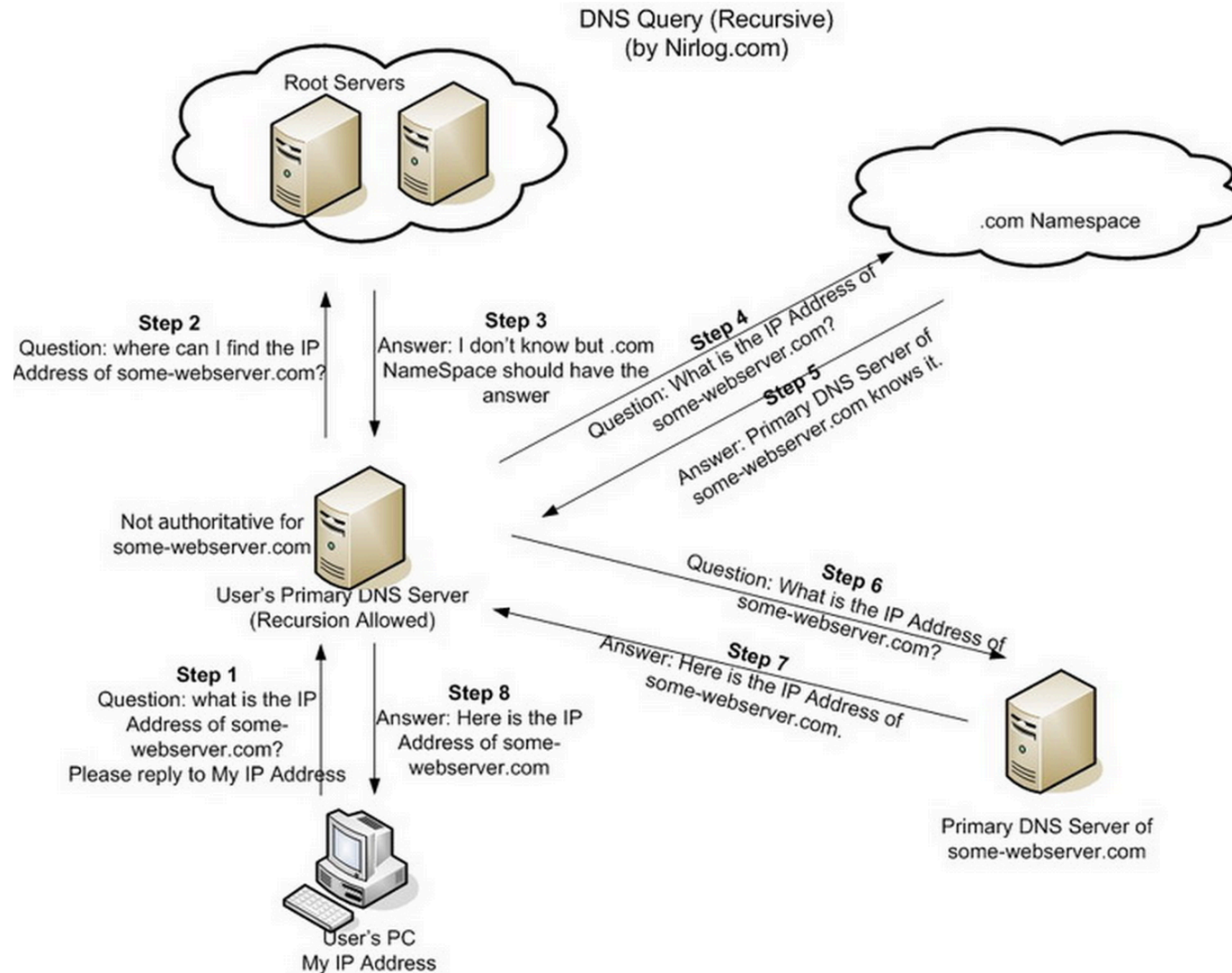
Domain Name Service

- Global Internet service to map names to IP addresses.
- Part of core TCP/IP suite of protocols
 - RFC 882 (1983) updated by RFC 1034 (1987)
 - Replaced manually maintained “hosts.txt” of all Internet connected computer’s IP addresses.
- Let’s do it
 - unplug from fw demo
 - Turn on wireless
 - dig www.nytimes.com

The DNS Hierarchical Name Tree

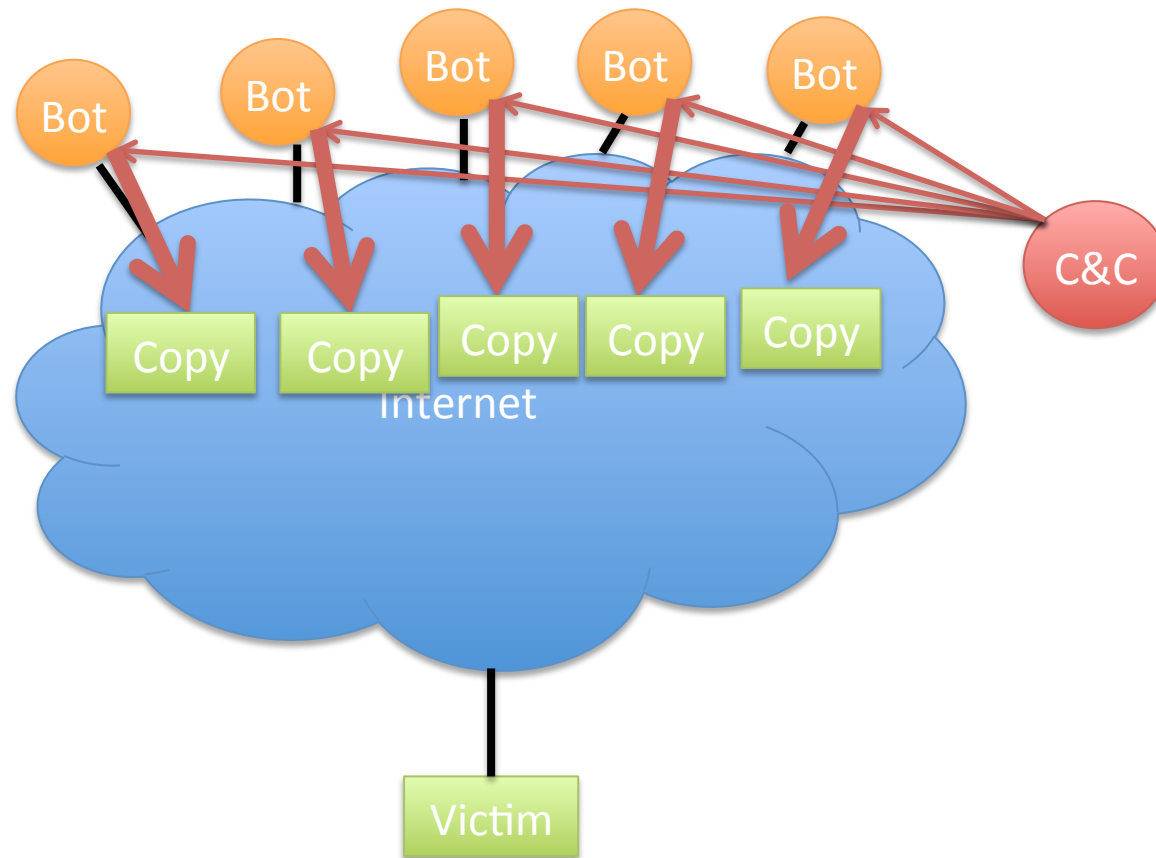


How a DNS Query Works

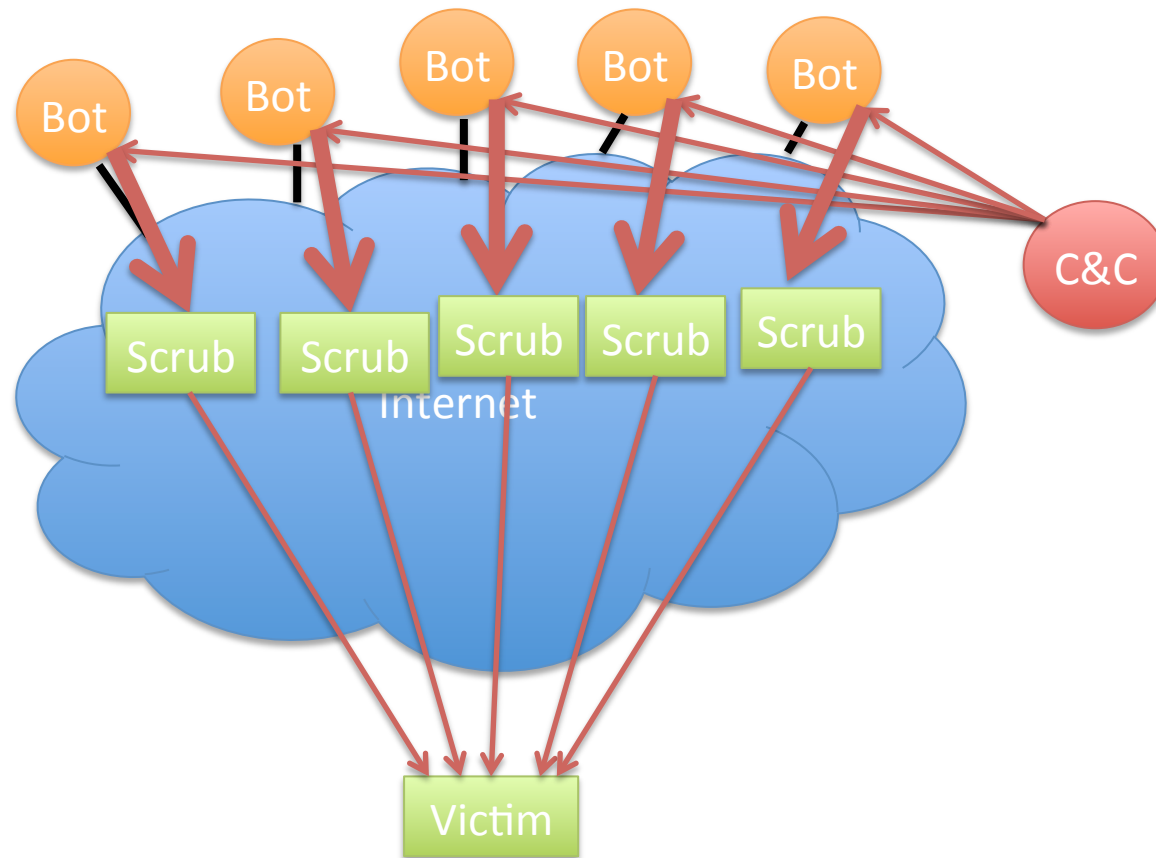


Credit: <http://securitytnt.com/dns-amplification-attack/>

DDOS Defense: Content Distribution



DDOS Defense: Distributed Scrubbing



Egress Filtering

- Can have many purposes, but in DDOS case:
 - Don't let spoofed packets out of our network

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.

Simple Example

- If we see a long string of 0x90 in the middle of a network packet, what should we think?

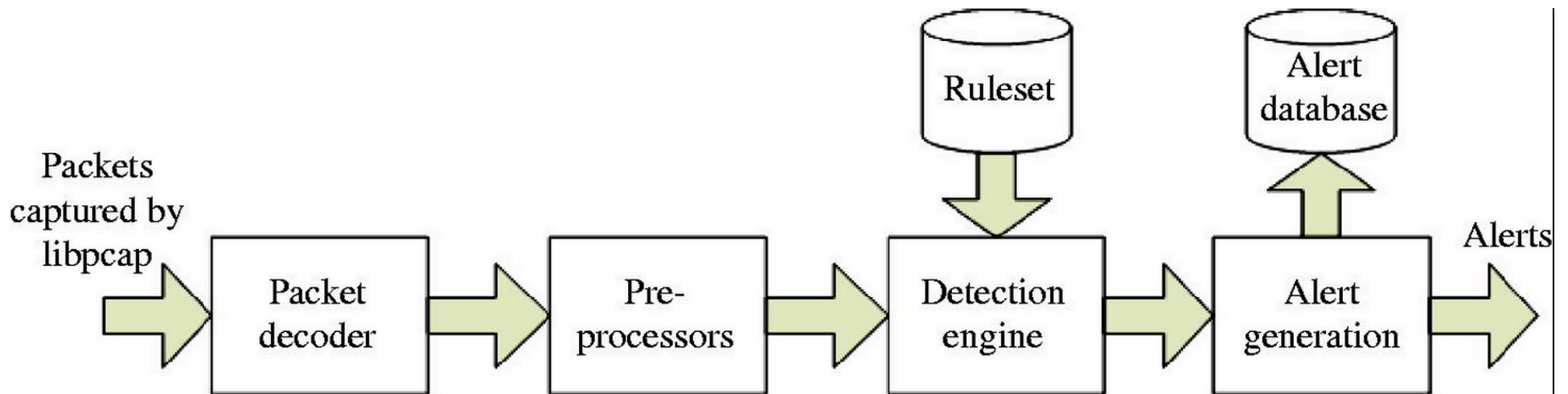
Example NIDS Rule

- 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

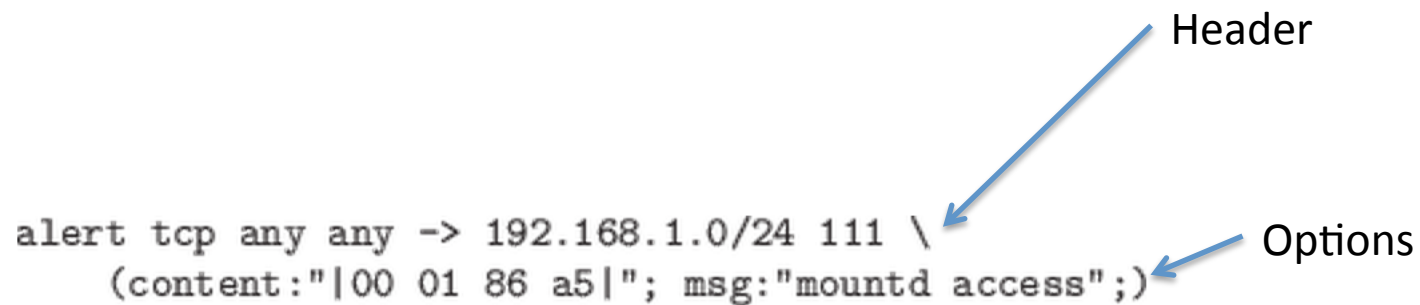
- Heberlein et al NSM – 1989
- Wheelgroup NetRanger - 1995
- Snort – 1998
- Intruvert – 2000
- FireEye – 2004 (but really 2007)
- Focus on Snort here, as conveniently accessible.

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 second line of the rule.

Figure: Sample Snort Rule

Snort Detection Engine Data Structure

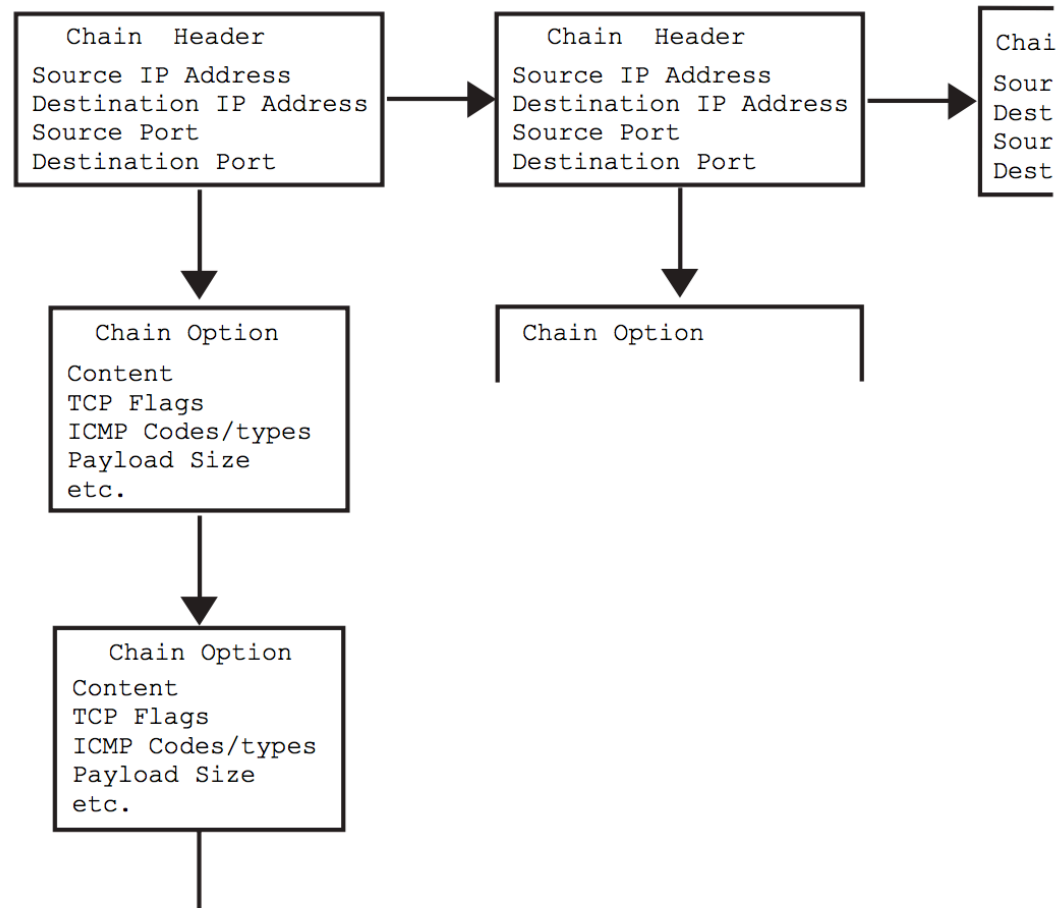


Figure 3: Rule Chain logical structure.

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

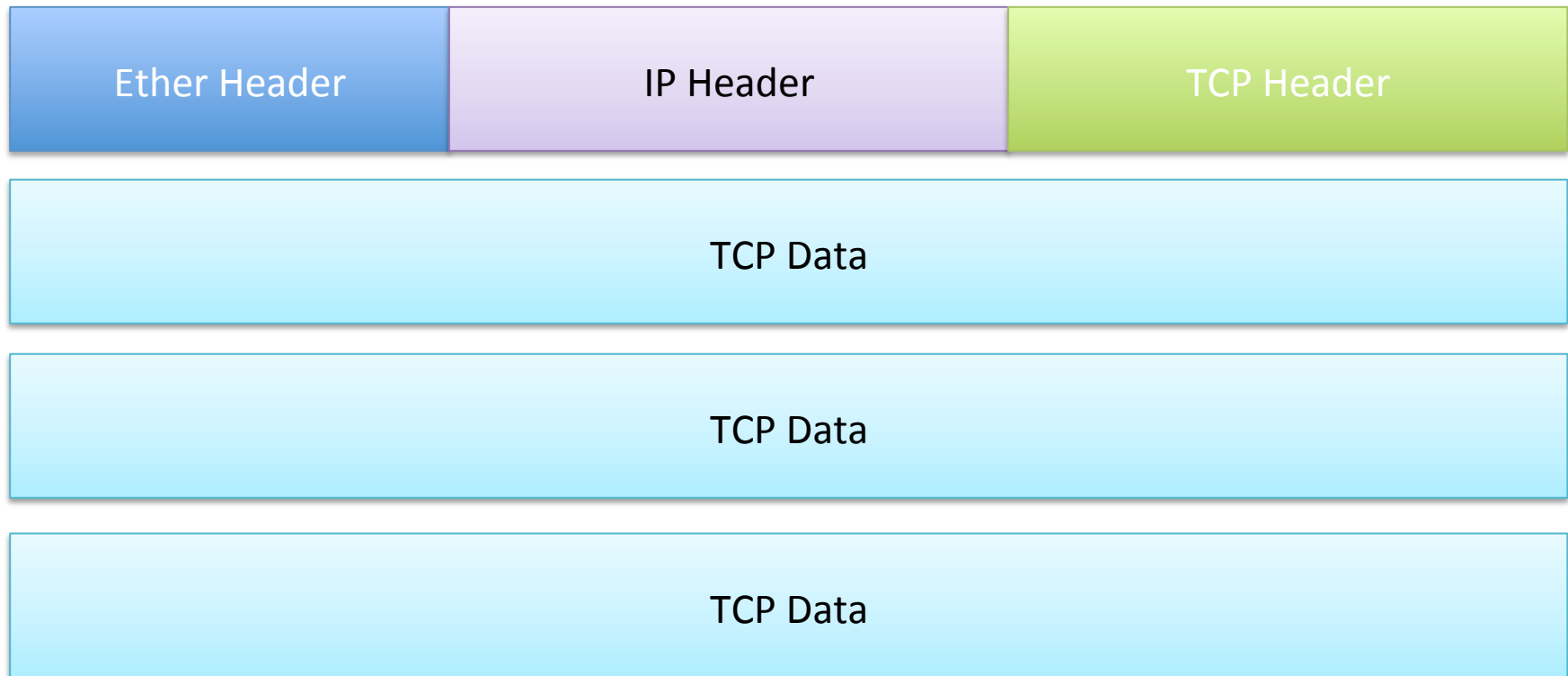
```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
(msg:"BROWSER-IE IE5 compatibility mode use after free attempt";
flow:to_client,established; file_data; content:"meta http-equiv=|22|X-UA-
Compatible|22| content=|22|IE=5|22|"; fast_pattern:only;
content:".runtimeStyle.setExpression";
content:"document.body.innerHTML"; 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: 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

