Traceability

Richard Clayton

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Check Point Course

11 September 2009

Outline

- TCP/IP refresher
- When IP addresses don’t work
- When IP addresses do work
- Steps to finding the source
- When IP addresses are not enough
- Hiding on ADSL
- Hiding on a LAN
  - Fancy (FPGA)
  - Simple (Firewalls)

Further reading

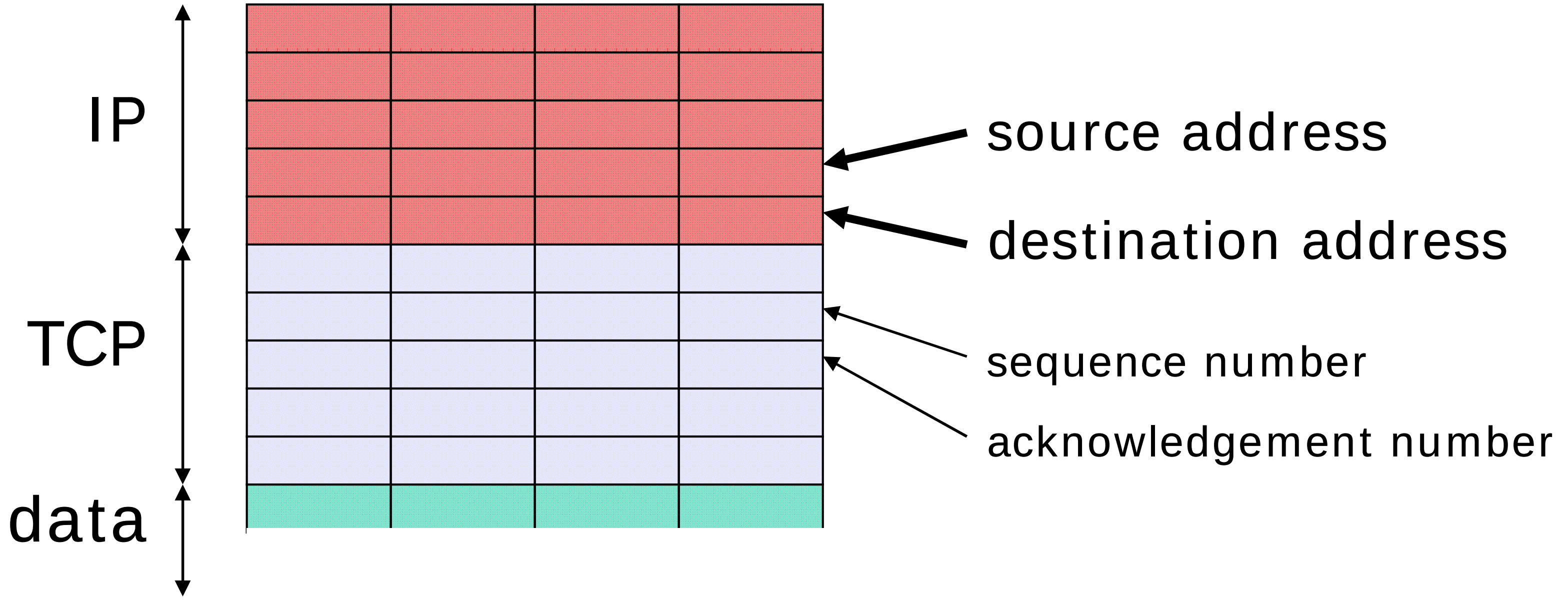
`http://www.linx.net/noncore/bcp/traceability-bcp.html`

written by UK ISP industry;
edited by Richard Clayton

`http://www.cl.cam.ac.uk/~rnc1/thesis.pdf`

UCAM-CL-TR-653
Richard Clayton

(Almost) all you need to know about TCP/IP



Are IP addresses valid ?

- Destination address is always valid
- Source address is valid for 2-way traffic
- Can send single bad packets with 1-way traffic
- Can do denial of service with 1-way traffic
- Filters can be useful in ensuring validity; but beware of source routing

- Also, can spoof addresses if the stack is poorly written and can predict responses...

Spoofing

- 3-way handshake

-->	SYN	client offset
<--	SYN-ACK	server offset
-->	ACK	

- If offset (and other info) is predictable don’t need to see the return traffic to have a successful conversation
- Described by Morris (85) and CERT (95)
- Fix by making sequence numbers **random** and perhaps by suitable packet filtering at borders

Who “owns” an address ?

- Regional registries issue numbers
  ARIN, APNIC, LACNIC, AfriNIC & RIPE
- ISPs reallocate within their blocks
- Hence “whois” will yield owner
- Reverse DNS should also yield name (but is unreliable and inconsistent):

  eg: for 100.101.102.103:
  103.102.101.100.in-addr.arpa

If the owner is unclear ?

- Traceroute may give a clue

```
  5    59 ms    61 ms    64 ms
                  tele-border-12-168.router.demon.net
  6    65 ms    66 ms    63 ms  linx.u-net.net
  7    64 ms    61 ms    63 ms  194.119.177.228
  8   179 ms    66 ms    62 ms  213.2.253.5
  9    62 ms    61 ms    63 ms  212.188.191.1
 10     *        *        *     Request timed out.
```

- ie: try to identify upstream providers

Identifying dial-up users

- Dynamic IP is commonplace
- RADIUS logs connect and disconnect
- Hence from time + IP can deduce account
- Various “gotchas”
  - UDP means logs incomplete
  - timestamps may be inaccurate
  - timezone may be unclear
  - logs are large and only kept short-term…
  - ... but EU Data Retention Directive has fixed that

Identifying ADSL users

- Customer supplies username & password
- DSLAM creates PVC to “Home Gateway” (BT)
- BT asks ISP (part of username) if login is OK
- ISP says yea/nay and provides IP address
- Traceability is from IP address to customer a/c

Except it may not work...

- Link back to physical copper is held by Home Gateway, & does not necessarily keep logs
  - no binding of credentials and line identifier

More practical problems

- RADIUS and IP allocation may be done by different organisations, hence have to chase around to get all necessary data

AND there’s problems caused in the logging:

- Timestamp may be rubbish (as may timezone)
- Name of remote machine may have been recorded but not its IP address
  - NB: the bad guys control their own DNS!
  - hence deducing the IP address to determine ownership is problematic

More complications

- Network Address Translation
  - may be part of a firewall, or router solution
  - used to preserve IP address space
  - used to hide network architecture
  - unlikely to be logged

- DHCP
  - dynamic allocation of addresses
  - logging can also be problematic

Mobile IP providers

- Data phones and Internet “dongles” for laptops mean millions of new TCP/IP users
- BUT providers cannot obtain huge blocks of IP address space (IPv4 will soon be exhausted)
- So they are using NAT, with many (hundreds) of users sharing the same IP address
- Hence need to provide IP address + timestamp (& timezone) PLUS port number
- Existing security logging often inadequate
- AND not addressed by Data Retention Directive

Authenticity

- Logs need to be authentic & correctly timed
- DNS needs to be trustworthy
- IP Allocations need to be documented
- Machines need to be secure
- Staff need to be trustworthy

  nightmare scenarios :

  chasing a sysadmin or ISP staff

Review

- 2-way traffic makes an IP address trustworthy
- Registries and traceroute will locate ISP
- ISP logging will locate the account
- Account details will reveal user
- CLI will reveal dial-up user
- Local records (NAT/DHCP) will reveal a LAN user
  - BUT the last hop may not lead you to exactly the right person, especially if looking for a skilled adversary who can “frame” an innocent bystander

“Practical anonymity”

- Steal a password
- Use a free account and withhold your CLI
- Use a pre-paid WAP phone
- Use a cybercafe
- Use someone else’s WiFi
- Multiple jurisdictions will slow tracing down
  - Though perhaps avoid the USA
- NB: Best Practice is far from universal
- or you could just go into work and use the LAN

Traceability on LANs

- A LAN is a broadcast medium
- Hard to locate senders
  - big practical problem for DHCP on NTv4
  - but bridges know direction, and switches know more
  - can fingerprint the analog properties of NICs!
- Naïve to think MAC addresses are fixed
- Possible to steal MAC & IP addresses
  - may be prevented by switch architecture
  - genuine owners must be switched off
    OR subject to DoS

Ethernet basics

- Unswitched Ethernet is a broadcast medium
- By convention one ignores packets without the correct MAC address
- ARP is used to map IP addresses to MACs

  Y broadcast: who has IPx, tell IPy

  X reply to MACy: IPx is at MACx

  results cached for a short period (20 mins)

ARP poisoning

- Send ARP packets to two endpoints

  X→B: `I am IP-A and my MAC is MAC-X`

  X→A: `I am IP-B and my MAC is MAC-X`

- X now “man-in-the-middle” twixt A and B
- NB: works on switched Ethernets as well
- Modern switches detect this!
  - or you can run **`arpwatch`**

Simple identity theft

- Borrow someone else’s IP address
  - – if IP address is in use then “gratuitous ARP” (sent by machine that has been rebooted to flush caches)
  - – if not in use then will be caught by logging at MAC level (sysadmins often collect MACs for machine identification)

Complex identity theft

- Borrow IP address <u>and</u> MAC address
  - if real owner isn't present then will work just fine! Investigators will have to resort to CCTV footage, building entry records or holes in the record of activity of your machine
  - if real owner is present then will need to sniff traffic (easy) and do something about their TCP resets...

TCP resets

Start to talk to a mail server

```
1028 > smtp [SYN]      Seq=0 Ack=0 Win=32768 MSS=1460
smtp > 1028 [SYN, ACK] Seq=0 Ack=1 Win=17520 MSS=1460
```

But real owner of identity sends reset to the mail server

```
1028 > smtp [RST]      Seq=1 Ack=4087568586 Win=0
```

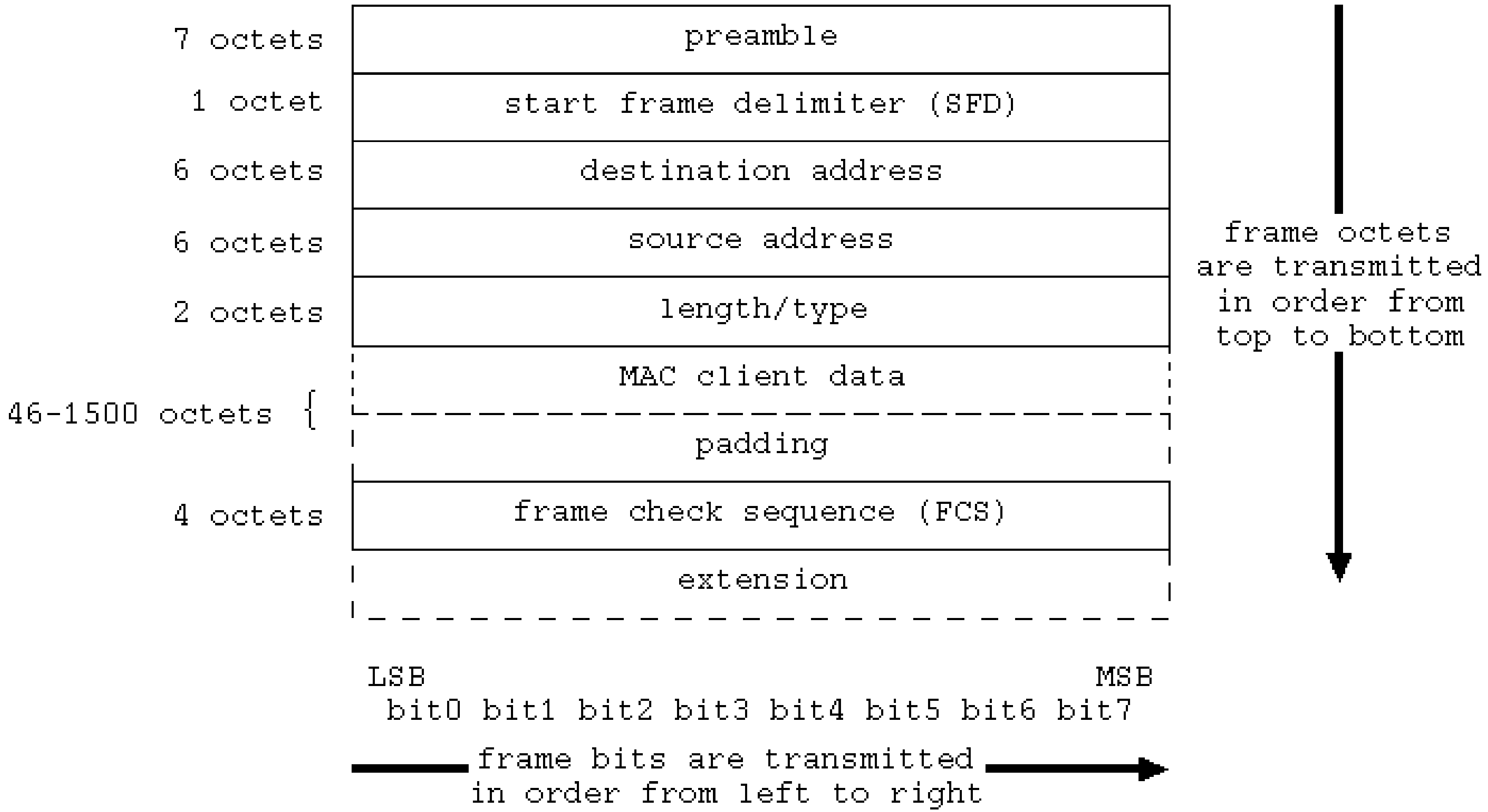
So when we do third packet of handshake we are rebuffed

```
1028 > smtp [ACK]      Seq=1 Ack=1 Win=32768
smtp > 1028 [RST]      Seq=1 Ack=207398712 Win=0
```

Preventing TCP resets

- What if we were to prevent the true owner of the IP (& MAC) address from sending out their reset ? Identity theft will then be successful (and CCTV footage won’t help!)
- Traditionally done by “blue screening”
- My innovation is to consider deliberate packet level collisions to prevent sending…

Ethernet packet format (10Mbit/s)



7 octets	preamble
1 octet	start frame delimiter (SFD)
6 octets	destination address
6 octets	source address
2 octets	length/type
46-1500 octets {	MAC client data
	padding
4 octets	frame check sequence (FCS)
	extension

frame octets are transmitted in order from top to bottom

LSB MSB

bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7

frame bits are transmitted in order from left to right

Collisions

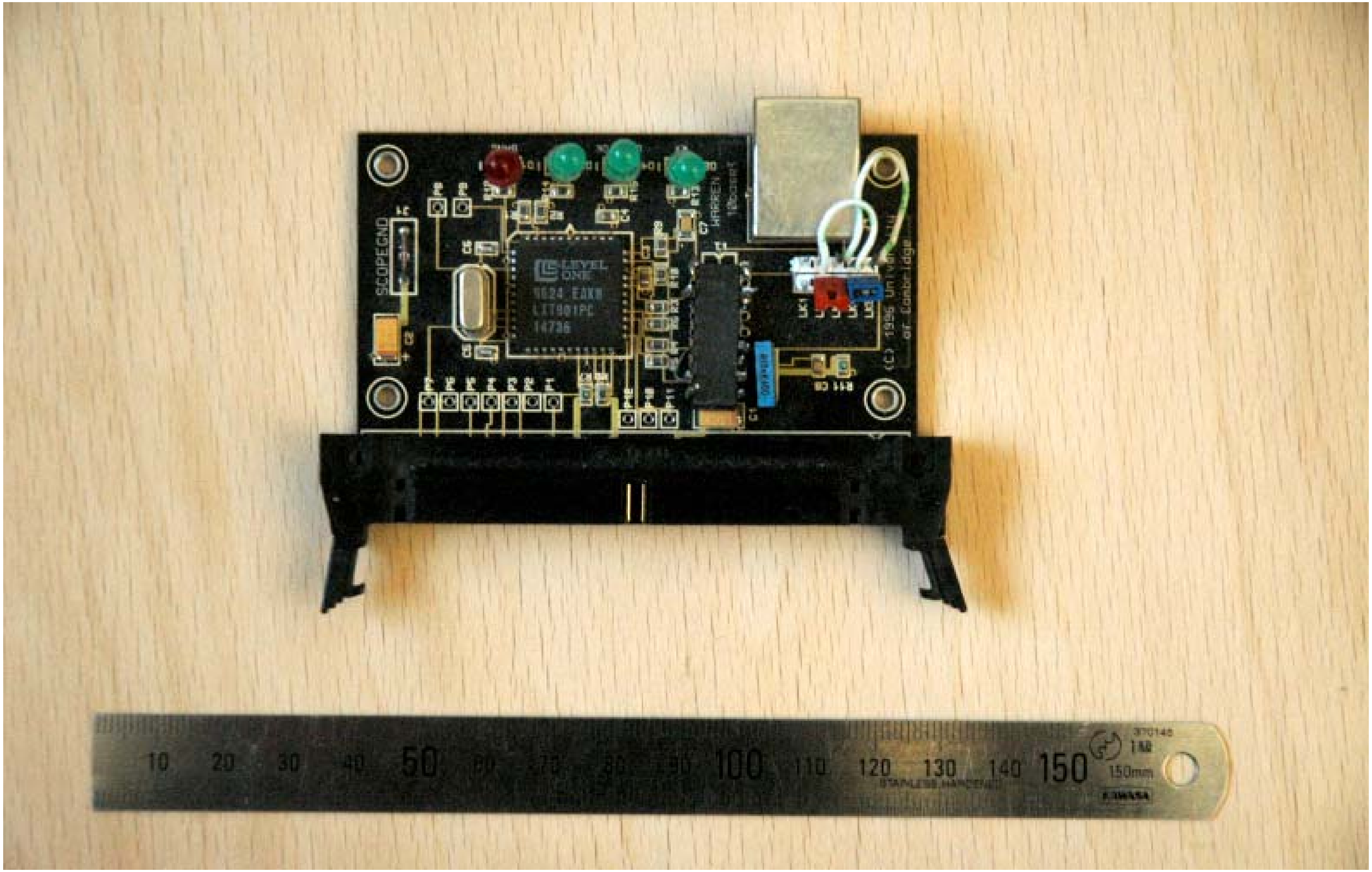
- If two stations start sending at the same time they detect the “collision”
  - perhaps not immediately, broadcast domain may be split across 4 bridges (5 segments)
- They then send a jamming signal
  - this makes sure that the other station notices
- & “truncated binary exponential backoff”

  [0, 2^{n}-1] * 1/20,000 second (*n* = min(N, 10))

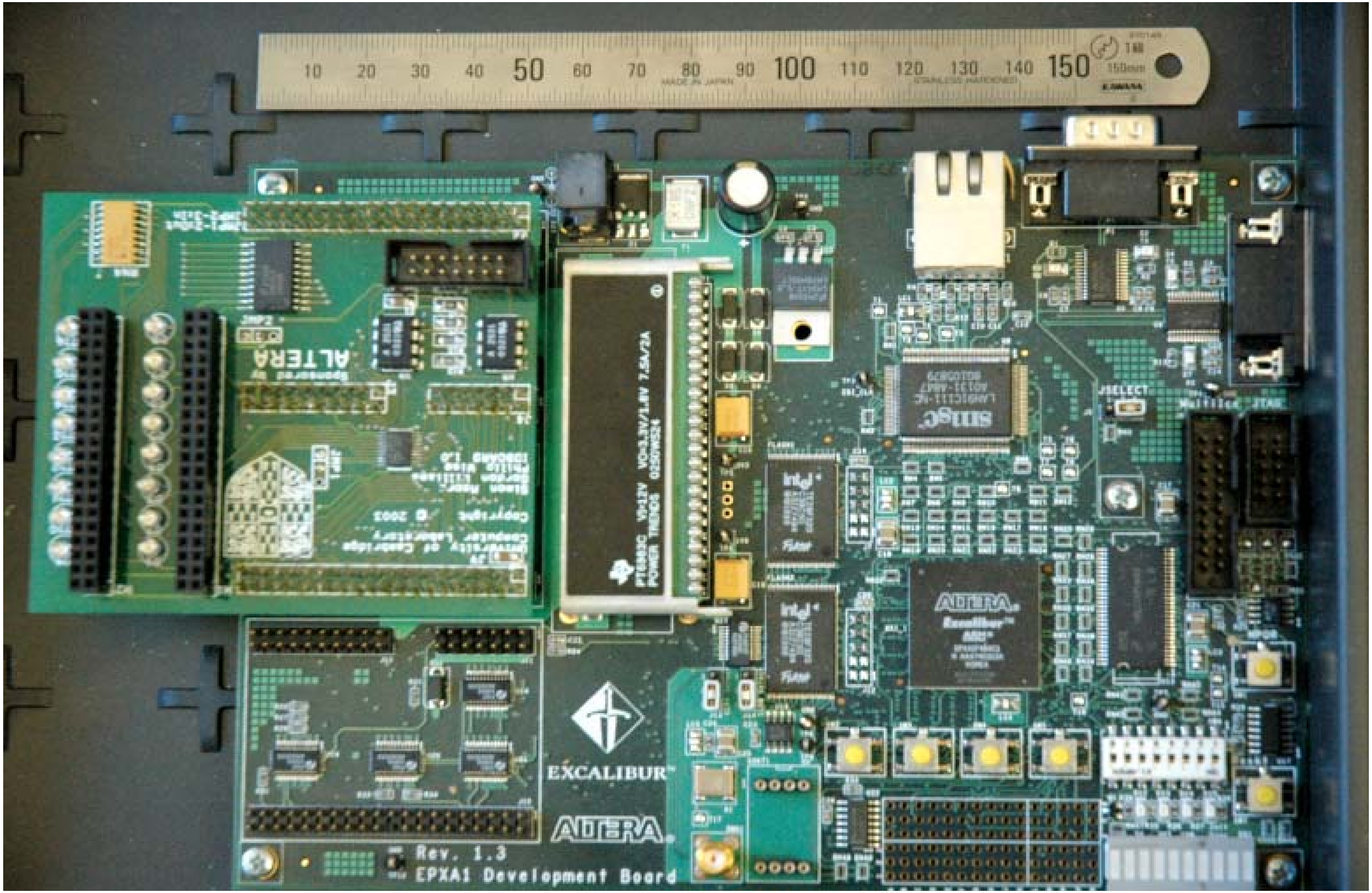
Deliberate collision

- Collision is not “late” until 512 bits sent
  - ie 64 bytes (hence data padded to 46 bytes)
- So (provided not 5 segments away) plenty of time to spot the sending address and deliberately send a jamming signal!
- Ethernet system design means that you need some hardware...

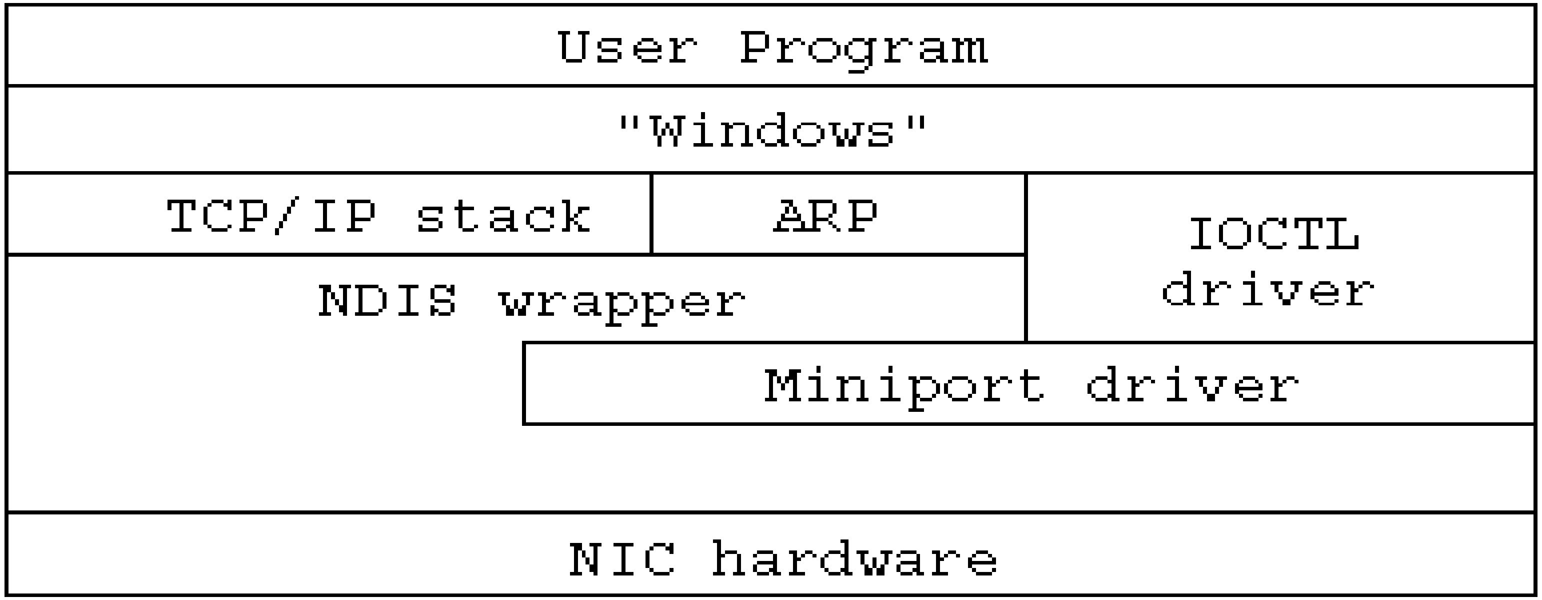
Ethernet PHY (1996 vintage)



FPGA & ARM (2005 vintage)



Windows CE architecture



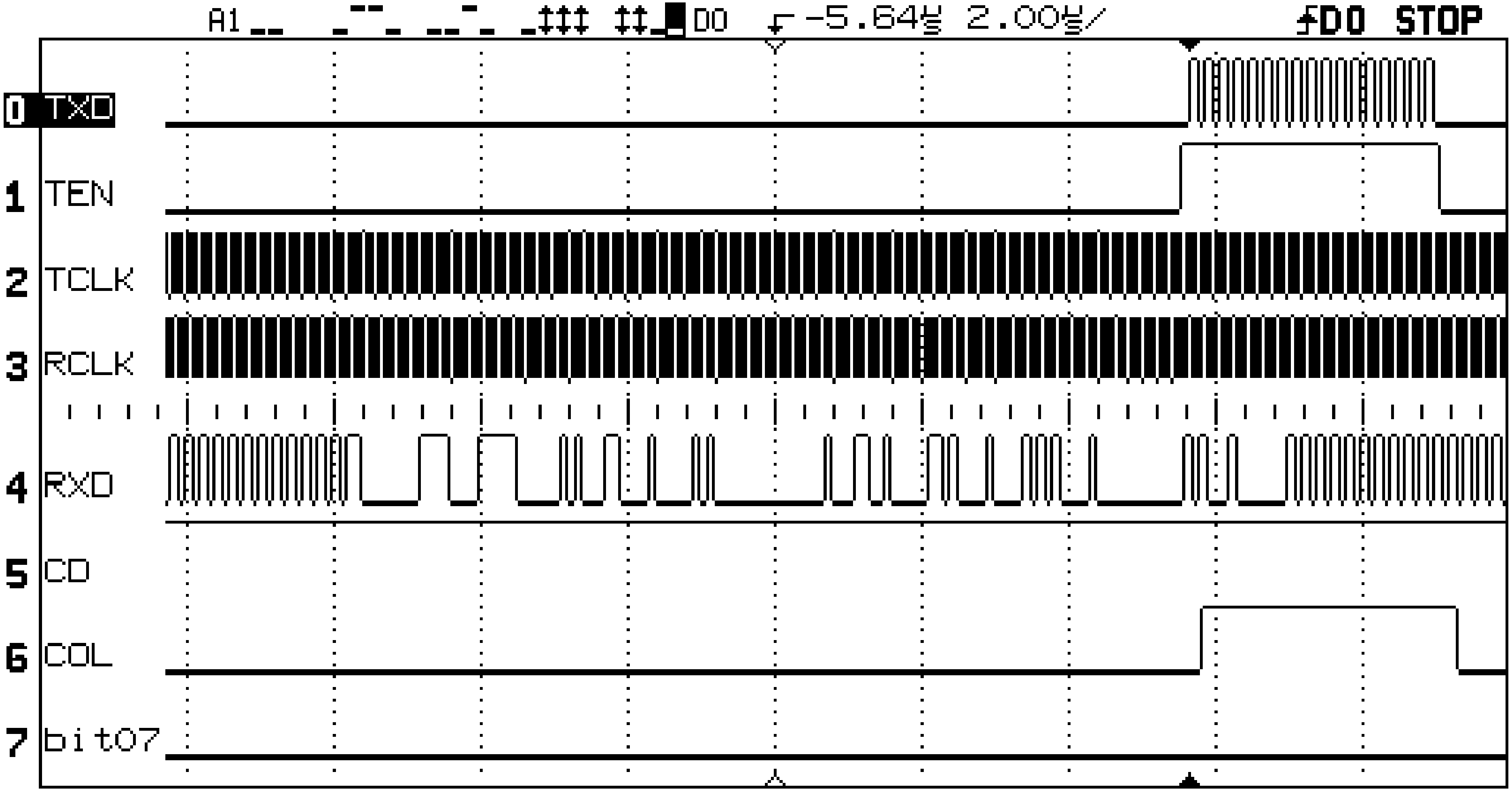
- Had to implement a “connectionless Miniport driver”, an IOCTL device and a user-mode program
  - plus improvements needed existing interrupt handling

Experiment

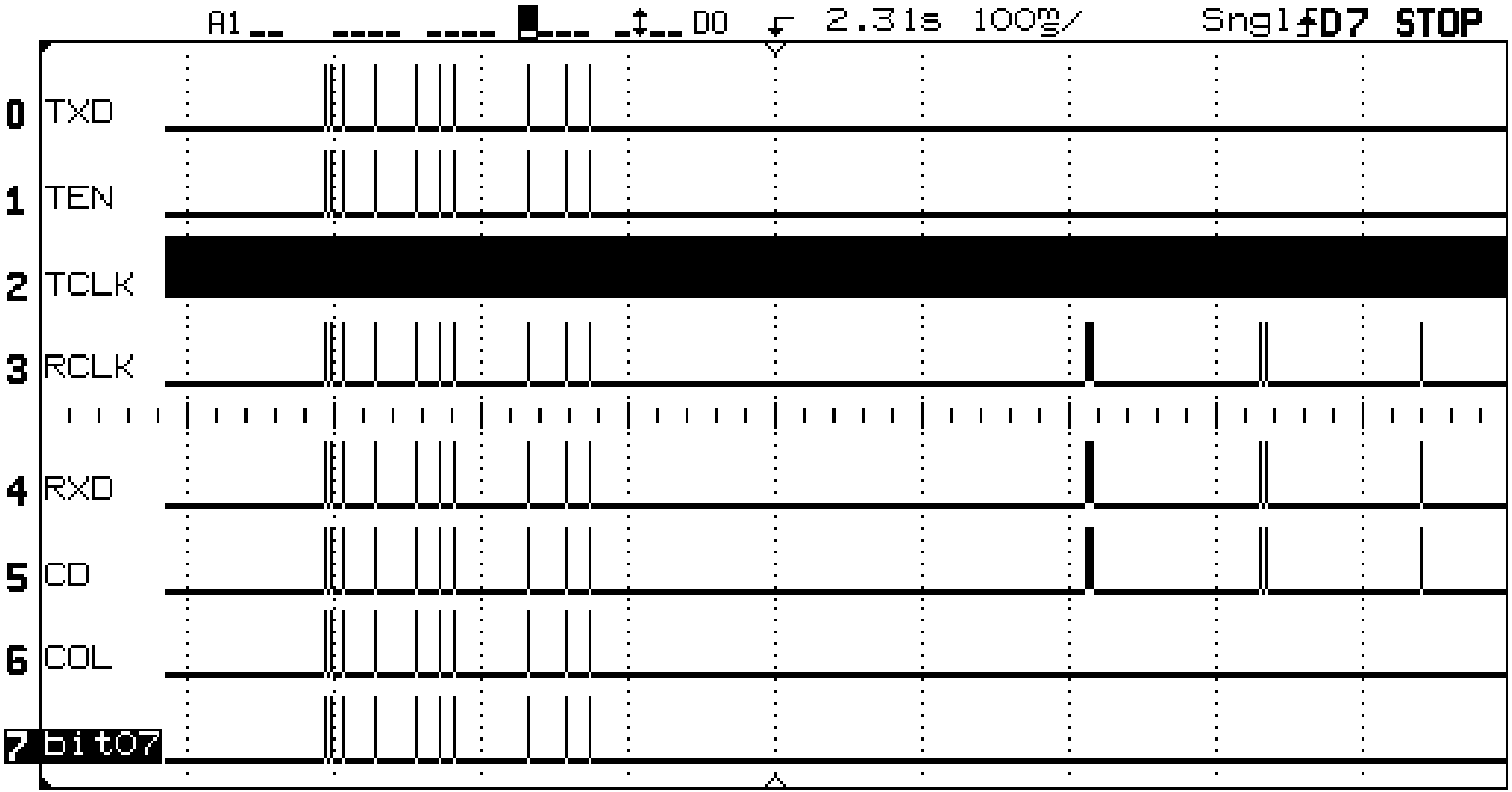
- Run program to send email to server
- Whilst sending, arrange for real owner of the identity to be collided with
- Capture lovely traces on oscilloscope to persuade PhD examiners it was real
- Examine whether or not the spoofed machine notices the collisions

Experimental set-up

One collision



Many collisions

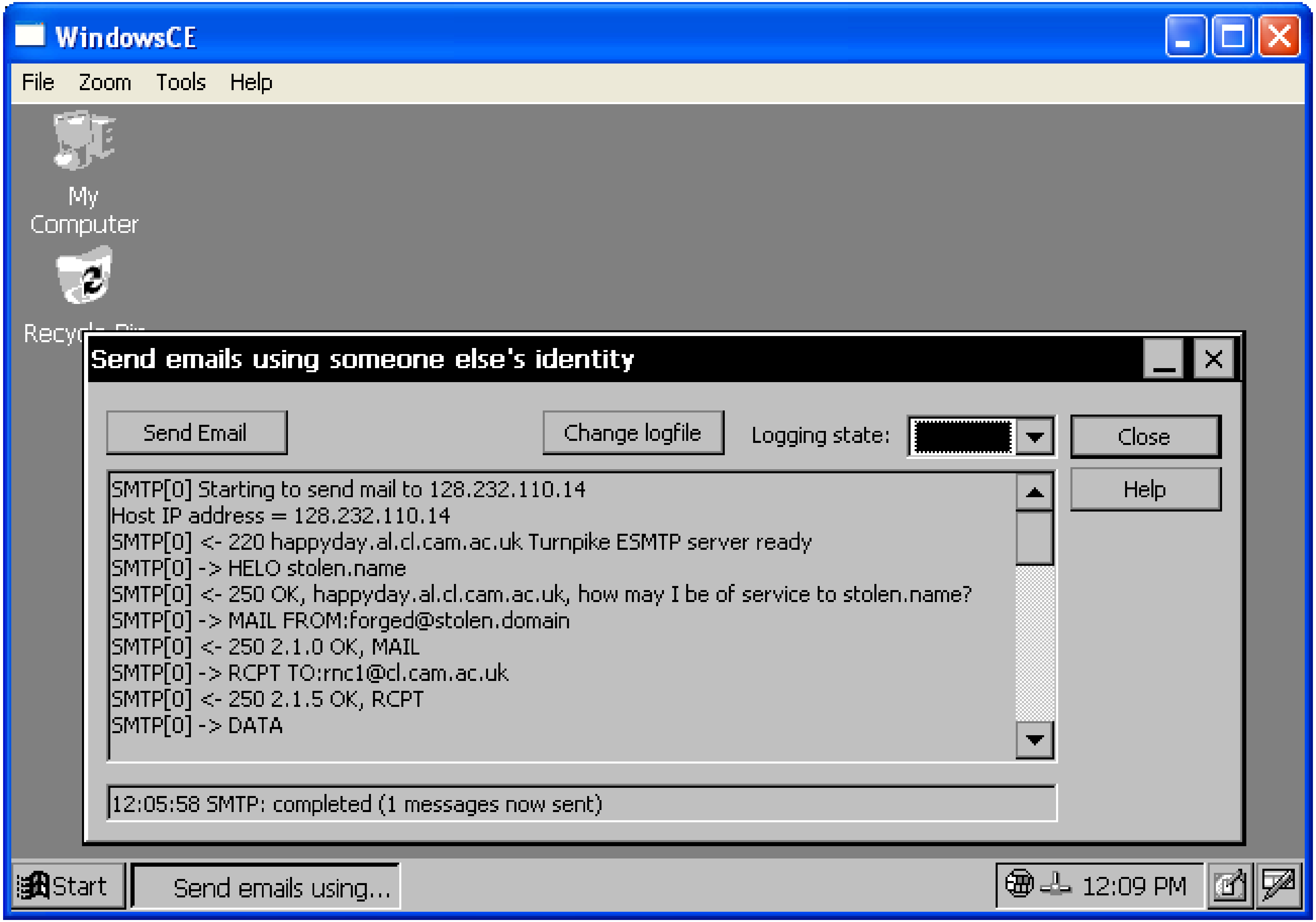


Timing

- Hardware collisions only occupy 200ms
  - my card gave up at N=10
- After that higher protocol levels take over
  - TCP will depend on Round Trip Time (etc)
  - UDP protocols vary considerably
  - RSTs will not generally be resent

Limited detection

- If machine idle then identity theft invisible
- If machine active then immediate effect on **scp** transfers ("stalled" reported after 5 sec)
- Timeouts typically 20 seconds or more (sometimes as much as a minute)
- Was taking my 166 MHz design about 7 seconds to send a short email



```
SMTP[0] Starting to send mail to 128.232.110.14
Host IP address = 128.232.110.14
SMTP[0] <- 220 happyday.al.cl.cam.ac.uk Turnpike ESMTP server ready
SMTP[0] -> HELO stolen.name
SMTP[0] <- 250 OK, happyday.al.cl.cam.ac.uk, how may I be of service to stolen.name?
SMTP[0] -> MAIL FROM:forged@stolen.domain
SMTP[0] <- 250 2.1.0 OK, MAIL
SMTP[0] -> RCPT TO:rnc1@cl.cam.ac.uk
SMTP[0] <- 250 2.1.5 OK, RCPT
SMTP[0] -> DATA
```

```
12:05:58 SMTP: completed (1 messages now sent)
```

```
Return-Path: <forged@stolen.domain>
Received: from stolen.name ([192.168.1.2]) by
   happyday.al.cl.cam.ac.uk
with SMTP id <tqRzmTABiDxCBA16@happyday.al.cl.cam.ac.uk>
   for <rnc1@cl.cam.ac.uk> ; Thu, 30 Jun 2005 19:22:57 +0100
Message-ID: <demo1@stolen.domain>
Date: Thu, 30 Jun 2005 19:22:02 +0100
From: Impersonated User <forged@stolen.domain>
To: Richard Clayton <rnc1@cl.cam.ac.uk>
Subject: Demonstration email #1
MIME-Version: 1.0

This email actually came from 192.168.1.4
However, not only has it been forged to appear to
have come from <forged@stolen.domain> but also the
Traceability information in the Received header field
has been recorded by the (honest) recipient
to be 192.168.1.2

This would mislead an investigator into examining
the wrong machine....
```

Software firewalls

- Encountered an unexpected difficulty generating dumps of RST packets when identity was stolen
- Eventually found that “ZoneAlarm” was discarding incoming SYN/ACK (and other segments) for an unknown connection
- Microsoft XP firewall does the same!

Stealth mode: an urban myth

- Bastion firewalls try and hide machines
  - slow down the hackers by obscuring detail
- Copied by “software firewalls”
  - despite them serving a different purpose
- Shields Up! made “stealth mode” a virtue
  - assumes that attackers probe and then pounce
  - assumes attackers are single threaded

Wireless hotspots

- Airports (etc) charge for wireless access
- Hence can borrow the identity of nearby Windows XP user – firewall on “to be safe”
- Economic analysis interesting : no incentive on software firewall maker to apply fix
- Airport could (probably) spot the subterfuge by analysis of port number usage etc
  - cf: counting hosts behind a NAT

Robert in India

- Could see backbone wireless AP but not those meant to be used by customers
- Spoofed the IP address and MAC of an AP
- Identified gateway address (eventually)
- Ensured did not send RSTs or ICMPs

```
net.inet.tcp.blackhole = 2
net.inet.udp.blackhole = 1
```

- Bob's your uncle! ☺

Take homes

- Ethernet addressing works through convention and cooperation
- Switched networks reduce opportunities for identity theft – but 802.11 brings them right back again
- Firewalls don’t always make you safer!

Traceability

Richard Clayton

http://www.lightbluetouchpaper.org

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Check Point Course

11 September 2009