Hiding on an Ethernet

Richard Clayton

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Security Seminar

28th February 2006

Summary

- Traceability outline
- Ethernet basics
- ARP poisoning
- Deliberate hardware collisions
- Experimental results
- Software firewalls
- NATs and hotspots

Traceability

- Record IP address (unspoofed!) and time
- Regional Registries indicate owning ISP
- ISP accounting gives usage at specific time
- Within an enterprise (or a home) then IP address allocations may be static (easy!) or recorded in NAT (occasionally) or DHCP (more often) log files
- Hence can lock up (or educate) offender

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 and 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
```

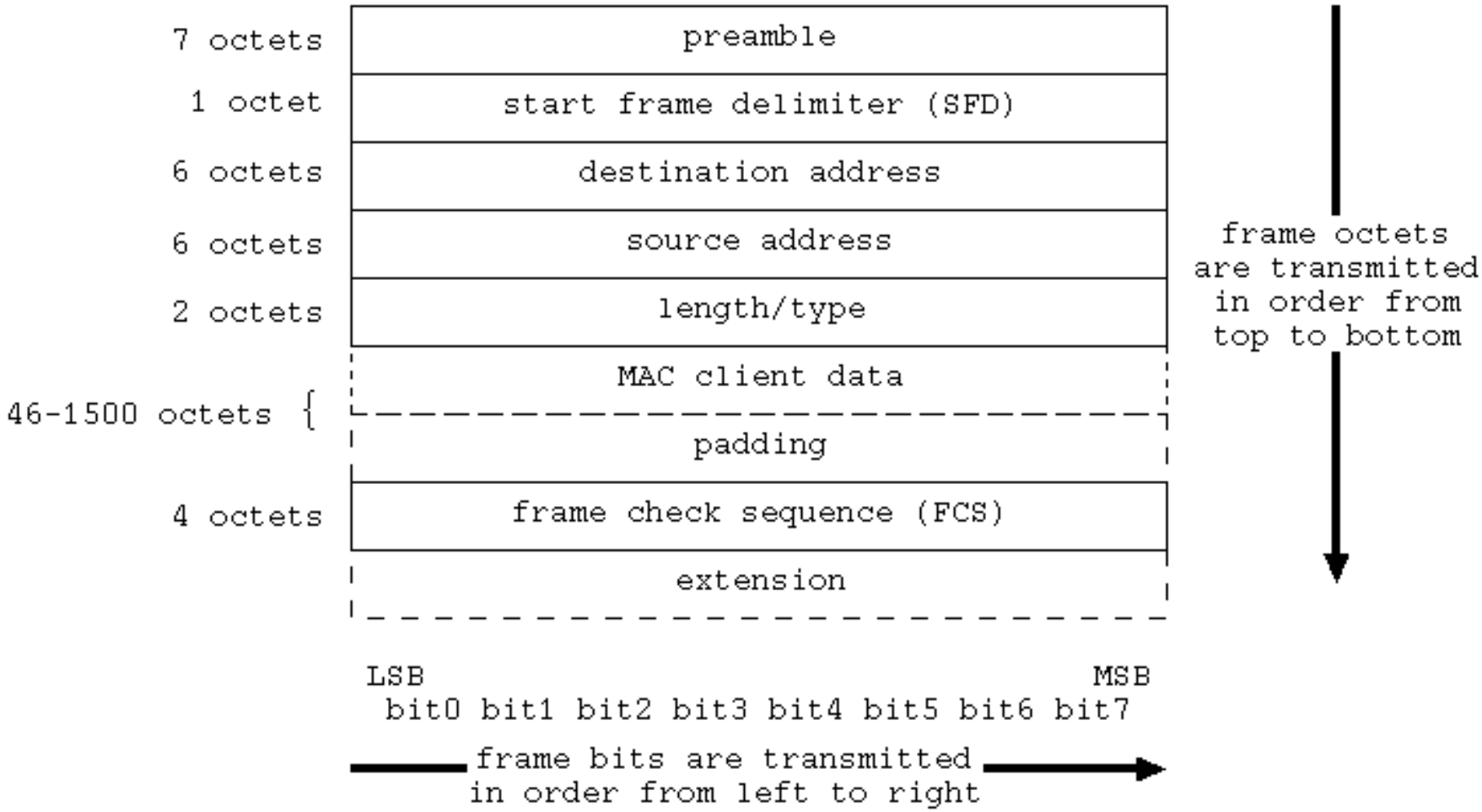
So when we do third packet of handshake we are rebuffed

```
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 (10 Mbit/s)



Collisions

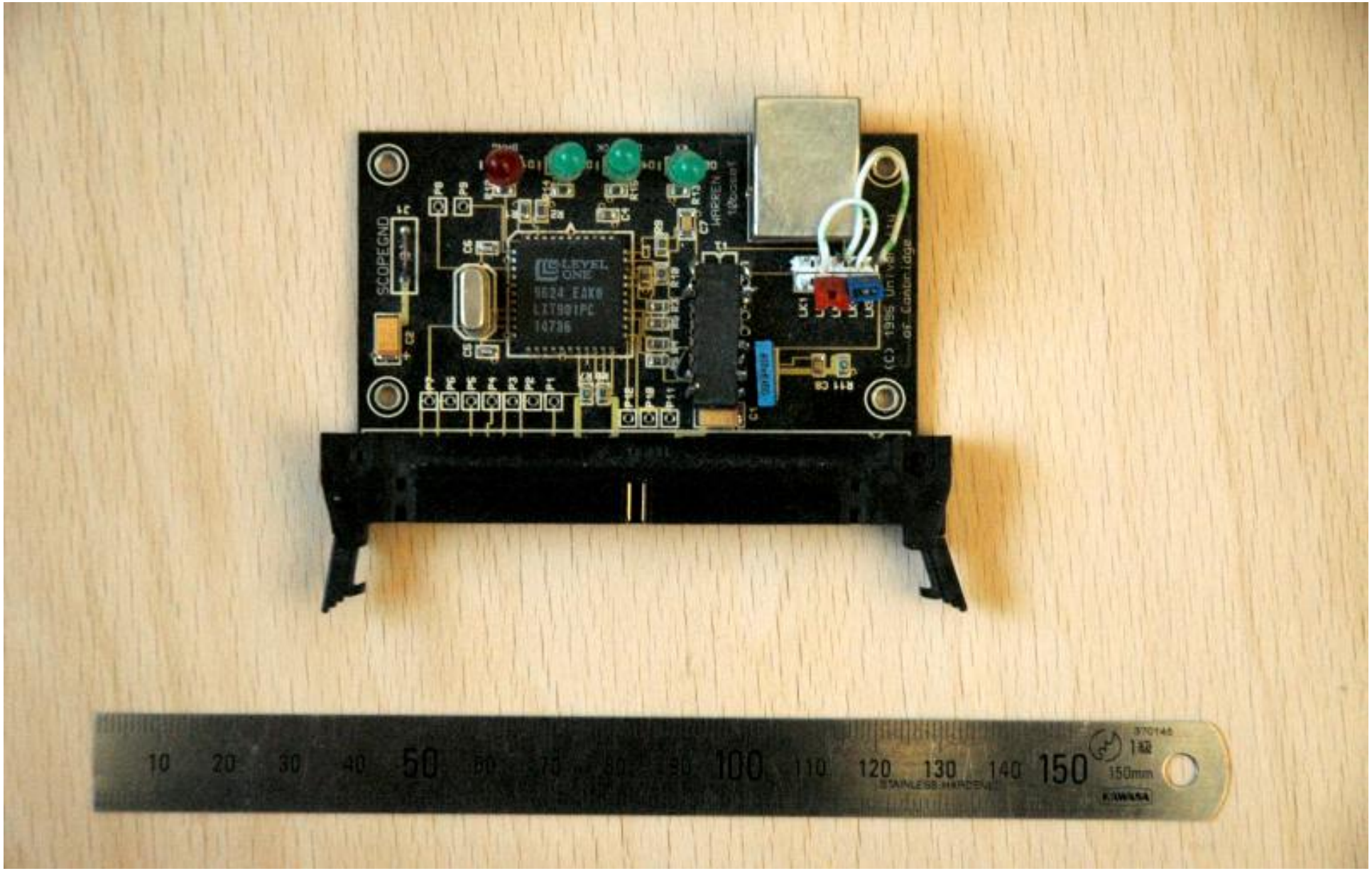
- If two stations start sending at the same time they detect the “collision”
  - perhaps not immediately, broadcast domain may be split across 4 bridges (5 segments)
- They then send a jamming signal
  - this makes sure that the other station notices
- & “truncated binary exponential backoff”

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

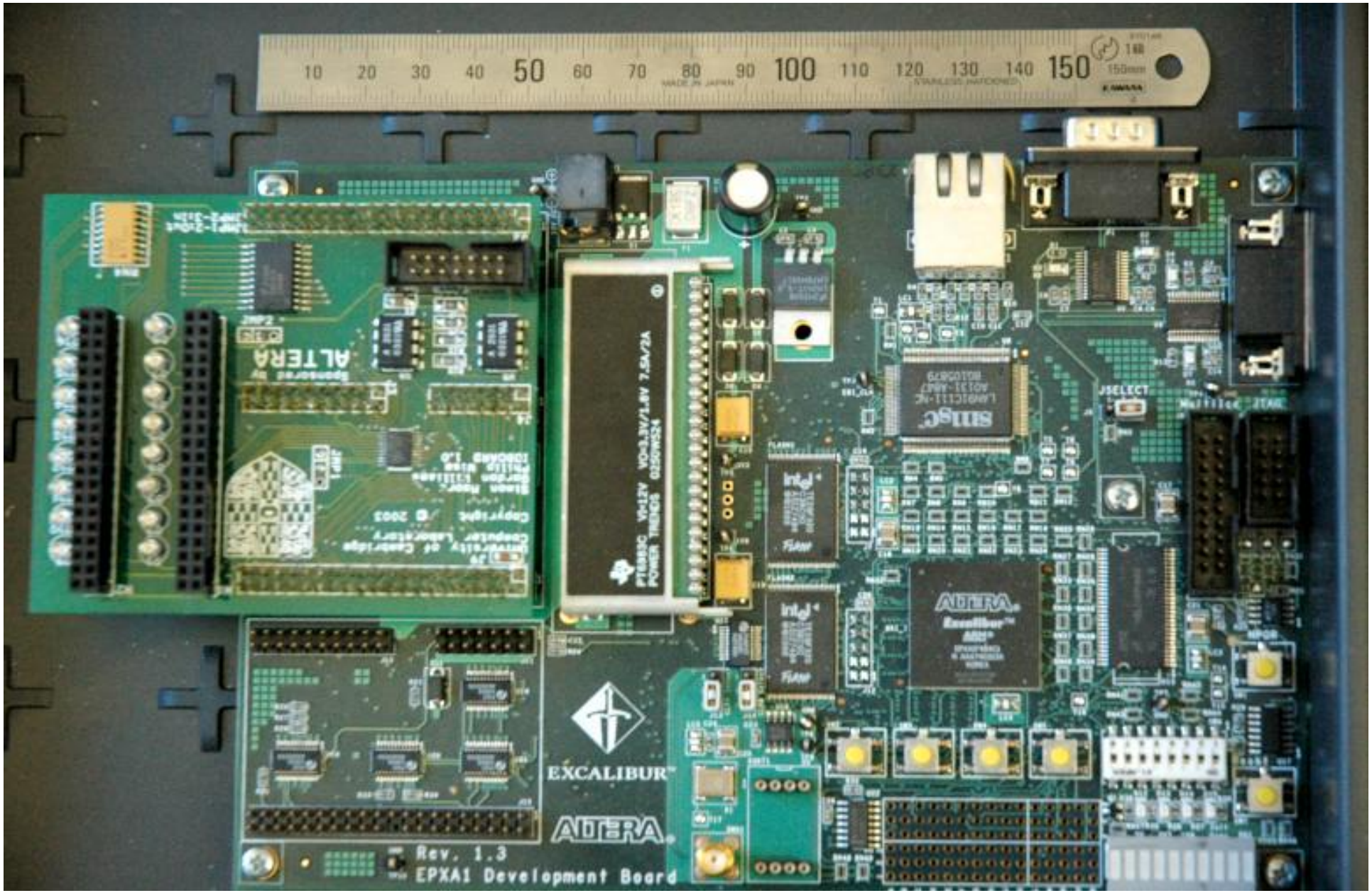
Deliberate collision

- 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…

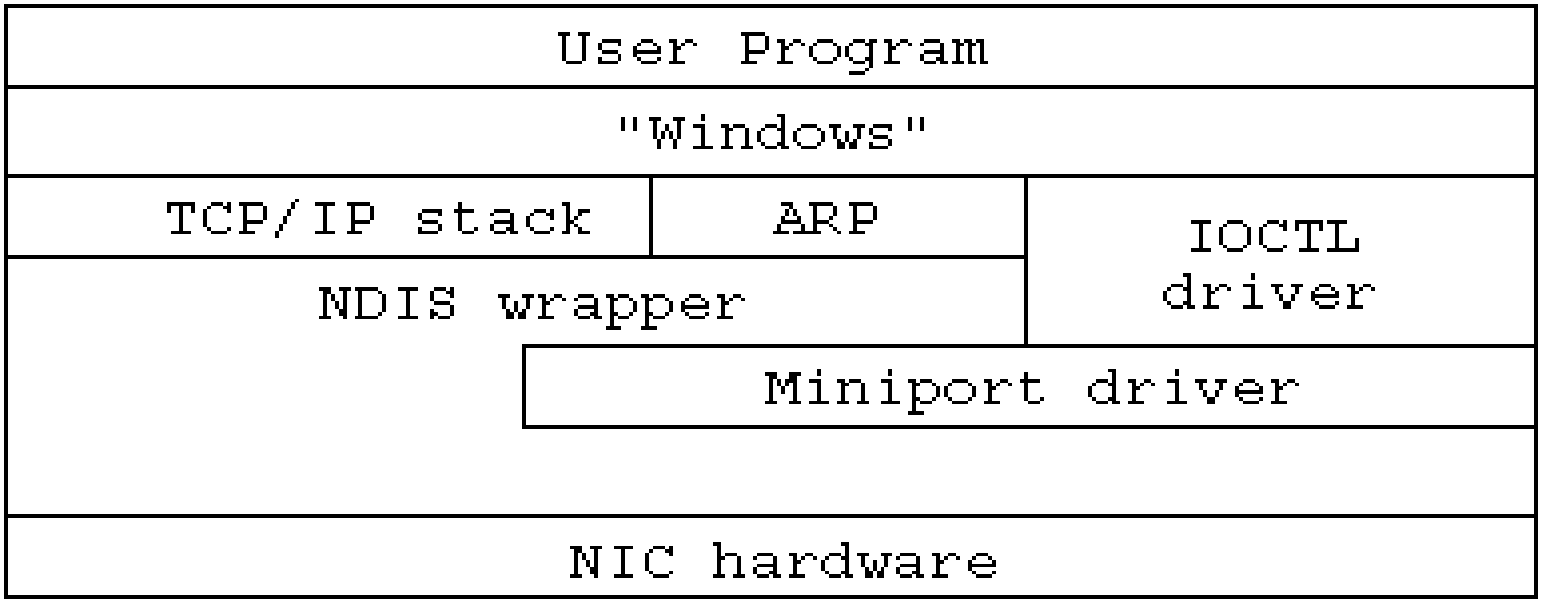
Ethernet PHY (1996 vintage)



FPGA & ARM (2005 vintage)



Windows CE architecture



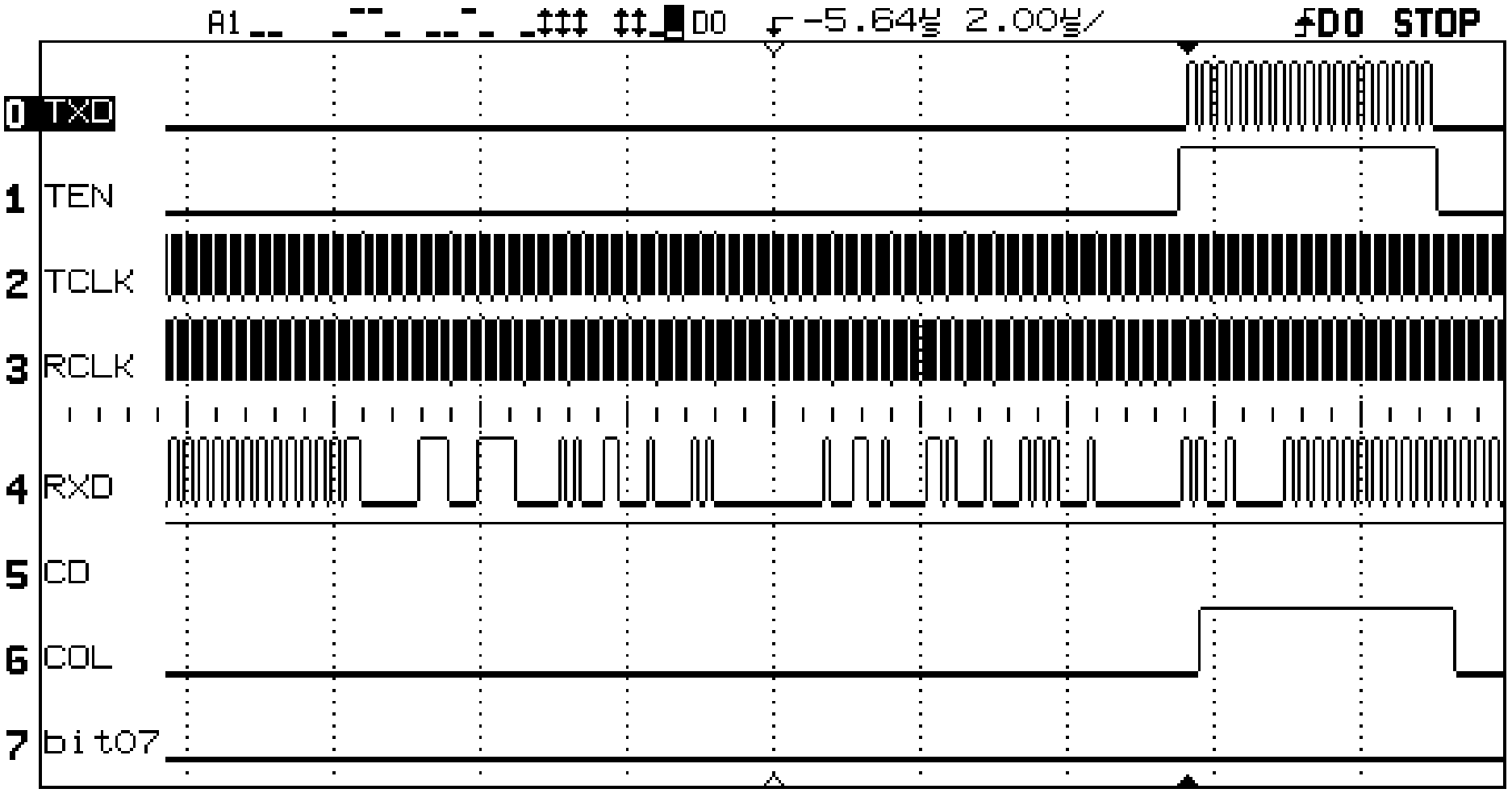
- Had to implement a “connectionless Miniport driver”, an IOCTL device and a user-mode program
  - plus improvements needed existing interrupt handling

Experiment

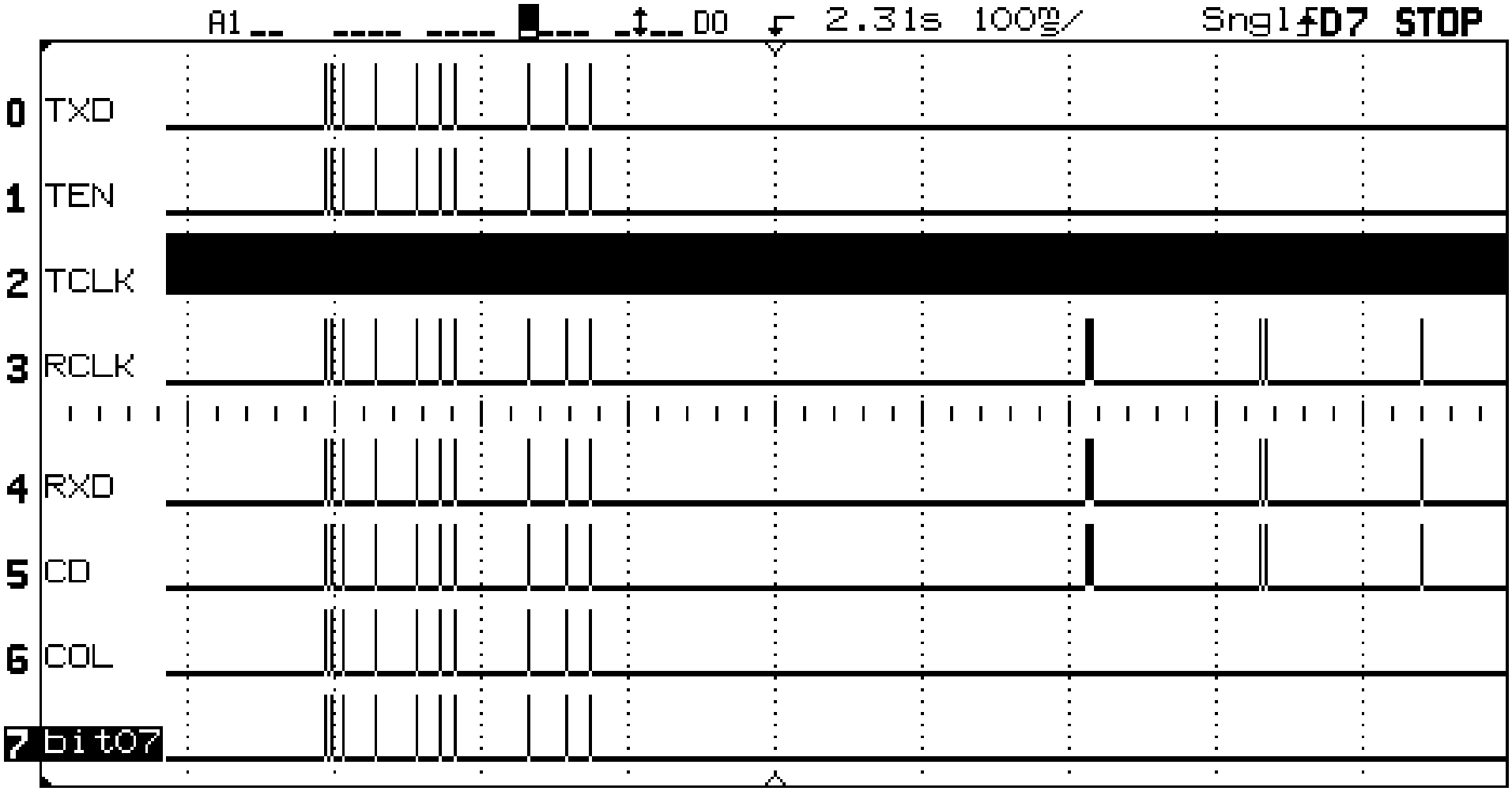
- Run program to send email to server
- Whilst sending, arrange for real owner of the identity to be collided with
- Capture lovely traces on oscilloscope to persuade PhD examiners it was real
- Examine whether or not the spoofed machine notices the collisions

Experimental set-up

One collision



Many collisions

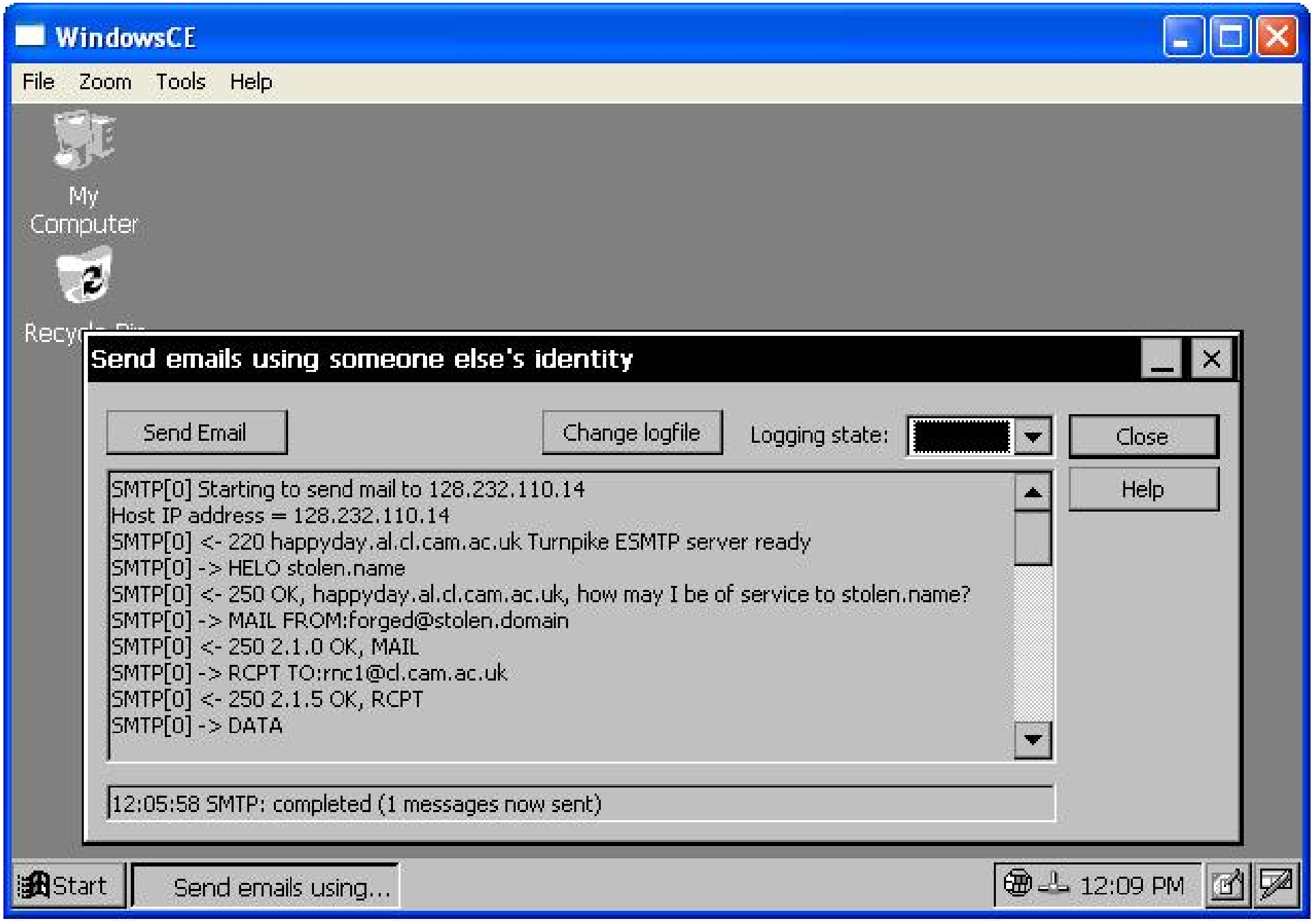


Timing

- 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


WindowsCE
File Zoom Tools Help
My Computer
Send emails using someone else's identity
Send Email
Change logfile
Logging state:
Close
Help
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)
Start
Send emails using...
12:09 PM

```
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

Distributed NAT

- Idea from Steve Bellovin (PhD examiner!)
- If everyone behind a NAT uses the same IP address then the NAT does not have to keep state! Avoids single point of failure and would simplify multi-homing.
- Merely some tedious details to work out to deal with legacy equipment (that expects ARP to work!) …

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!

Further reading

UCAM-CL-TR-653

Anonymity and Traceability in Cyberspace

http://www.cl.cam.ac.uk/~rnc1/