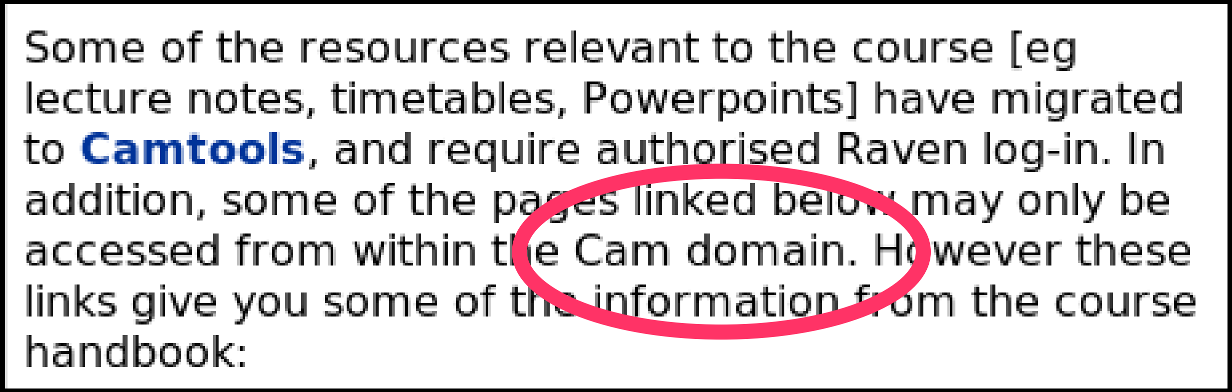
What is this CAM domain thing anyway?

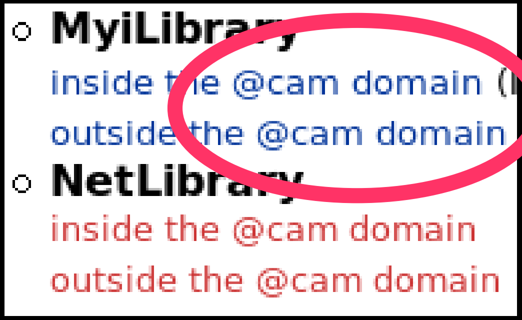
Bob Franklin
Jon Warbrick

University of Cambridge Computing Service

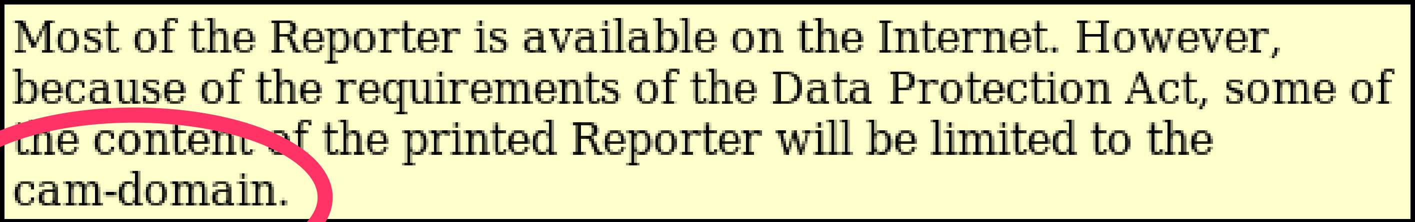
“... may only be accessed from within the CAM domain ...”



Some of the resources relevant to the course [eg lecture notes, timetables, Powerpoints] have migrated to **Camtools**, and require authorised Raven log-in. In addition, some of the pages linked below may only be accessed from within the Cam domain. However these links give you some of the information from the course handbook:



- **MyiLibrary**
  inside the @cam domain (
  outside the @cam domain
- **NetLibrary**
  inside the @cam domain
  outside the @cam domain



Most of the Reporter is available on the Internet. However, because of the requirements of the Data Protection Act, some of the content of the printed Reporter will be limited to the cam-domain.

What *do* we mean?

- Computer 'Connected to the CUDN'?
- Or perhaps called '<something>.cam.ac.uk'?
- Either way, inside the University
  - but perhaps not (VPDN)
  - is that OK too?
- This is 'Location Based' access control
- Why?

Too restrictive

- Especially when we get it wrong!
- What about people working from
  - home
  - conference
  - Internet Cafe
  - wireles network
- Trying to restrict people, not computers

Too lax

- All of these give 'outsiders' access to "University" computers:
  - 'Public' terminal rooms
  - JRS/Eduroam
  - Lapwing Tickets
  - Conference Bedrooms
  - Open proxies

eduroam
JANET Roaming

UCS

Information Strategy

- Contrary to the 'University Information Strategy':

> “7 (iv) Individual identifier access. Access to information and/or data should become person dependent [...] thus allowing them to access and manipulate information wherever they are by identifying themselves to the system, and not be dependent on a particular location or network”.

http://www.admin.cam.ac.uk/reporter/2004-05/weekly/5975/6.html

Alternatives

- Identify the user, not the computer
- Realistically, this means passwords
- For web applications we have Raven
  - but that only works for HTTP+Browser
  - could issue other passwords

Sometimes, a location-based approach is unavoidable

Over to Bob...

Address blocks on the CUDN

A temporary reference:

http://www-uxsup.csx.cam.ac.uk/~jw35/docs/temporary-nets-in-cam.html

IP Addresses and Apache

- 'Allow from' and 'Deny from'
- 'Order' directive:
  - 'deny,allow'
  - 'allow,deny'
- Must be in a <Location> or <Directory> block

```
<Directory /srv/www/WWW/secret>
  Order Allow,Deny
  Allow from 127.0.0.1
  Allow from 128.232
  Allow from 129.169
  Allow from 131.111
  Allow from 192.18.195
  Allow from 193.60.80
  Allow from 193.60.81
  Allow from 193.60.82
  Allow from 193.60.83
  Allow from 193.60.85
  Allow from 193.60.86
  Allow from 193.60.87
  Allow from 193.60.88
  Allow from 193.60.89
  Allow from 193.60.90
  Allow from 193.60.91
  Allow from 193.60.92
  Allow from 193.60.93
  Allow from 193.60.94
  Allow from 193.60.95
  Allow from 193.63.252
  Allow from 193.63.253
  Allow from 172.16.0.0/13
  Allow from 172.24.0.0/14
  Allow from 172.28.0.0/15
  Allow from 172.30.0.0/16
  Allow from 2001:630:200::/48
</Directory>
```

one “word”

“localhost” address

implicitly “193.60.80.0/24”, etc

explicit CIDR notation – same as 172.16.0.0/255.248.0.0

Tomcat

```
<Valve
className="org.apache.catalina.valves.RemoteAddrValve"
allow="^128\.232\., ^129\.169\., ^131\.111\., ^192\.18\.195\.,
^193\.60\.8[0-3]\., ^193\.60\.8[5-9]\., ^193\.60\.9[0-5]\.,
^193\.63\.252\., ^193\.63\.253\., ^172\.1[6789]\., ^172\.2[0-9]\.,
^172\.30\., ^2001:630:200:, 127\.0\.0\.1">
```

“^128\.232\.” = any address starting “128.232.”, etc.

Warning – ENTIRELY UNTESTED!!

UCS

Mailscanner

```
From: 127.0.0.1                 no
From: 128.232.                  no
From: 192.168.                  no
From: 131.111.                  no
From: 192.18.195.               no
From: /^193\.60\.8[0-3]\./      no
From: /^193\.60\.8[5-9]\./      no
From: /^193\.60\.9[0-5]\./      no
From: 193.63.252.               no
From: 193.63.253.               no
From: /^172\.1[6789]\./         no
From: /^172\.2[0-9]\./          no
From: 172.30.                   no
```

Mailscanner needs the '.'

OpenLDAP

Netmask follows a 'non-standard' '%'

```
access to *
        by      peername.ip=127.0.0.1 read
        by      peername.ip=128.232.0.0%255.255.0.0 read
        by      peername.ip=129.169.0.0%255.255.0.0 read
        by      peername.ip=131.111.0.0%255.255.0.0 read
        by      peername.ip=192.18.195.0%255.255.255.0 read
        by      peername.ip=193.60.80.0%255.255.252.0 read
        by      peername.ip=193.60.85.0%255.255.255.0 read
        by      peername.ip=193.60.86.0%255.255.254.0 read
        by      peername.ip=193.60.88.0%255.255.248.0 read
        by      peername.ip=193.63.252.0%255.255.254.0 read
        by      peername.ip=172.16.0.0%255.248.0.0 read
        by      peername.ip=172.24.0.0%255.252.0.0 read
        by      peername.ip=172.28.0.0%255.254.0.0 read
        by      peername.ip=172.30.0.0%255.255.0.0 read
        by      * none
```

How about using names?

- If addresses are too hard, how about names?
- Need an address-to-name lookup service
- DNS – the “Domain Name Service”
  - The Internet 'phone book'
  - A distributed database
- But note: separate address-to-name and name-to-address lookup tables (and others)

cam.ac.uk == CUDN

- As a matter of policy (with very occasional exceptions):
  - cam.ac.uk host names correspond to CUDN addresses
  - CUDN addresses correspond to cam.ac.uk host names
- Might not apply to CNAMEs, MX records, etc

Problems with using host names

- DNS entries broken or missing
  - even cam.ac.uk ones (recent registrations)
- DNS servers broken or slow
- Anyone can claim that their address has a cam.ac.uk host name
  - so have to do double-reverse lookups
- You need a DNS server

Apache again, by name

```
<Directory /srv/www/WWW/secret>
  Order Allow,Deny
  Allow from cam.ac.uk
  Allow from 127.0.0.1
</Directory>
```

Apache, but perhaps not other programs, only matches complete components so this *will* match www.cam.ac.uk but *will not* match www.overheadcam.ac.uk

Tomcat

```
<Valve
className="org.apache.catalina.valves.RemoteHostValve"
allow="\.cam\.ac\.uk$">
```

It looks as if you can't mix names and addresses

OpenLDAP

```
access to *
      by      domain.subtree=cam.ac.uk read
      by      peername.ip=127.0.0.1 read
      by      * none
```

Except that this may not work – many OpenLDAP packages are not compiled with DNS lookups enabled, and it may not do double-reverse lookups anyway

Matching options

- cam.ac.uk – implicit whole-component suffix
- .cam.ac.uk - explicit whole-component suffix
- *.cam.ac.uk – simple wildcard
- \.cam\.ac\.uk$ - regular expression
- .*\.cam.\ac\.uk – anchored regular expression
- ... and probably more

In Summary:

- Control access based on people wherever possible
- If you can't, consider using host names
- If you can't do that, you could use IP addresses
- It's all much harder than it looks!

If you have been, thanks for listening

Any questions?

These slides available at

http://www-uxsup.csx.cam.ac.uk/~jw35/courses/techlink/cudn/