

Network Security 3: Hosts

i.g.batten@bham.ac.uk

Hosts hold Assets

- So we need to make sure that only authorised users can access the hosts, and that those authorised users can only do what they are authorised to do.
- And we need to have a way to see when bad things happen, and to investigate when bad things do happen.

Log Files

- Most sites do not do logging properly.
- Important that logs be:
 - complete
 - accurate
 - trustworthy
 - secure

Don't Filter!

- Disk space is incredibly cheap
- Log files may be useful in a forensic situation, and the most trivial stuff may matter
- Save everything you can, and worry about filtering to make it easy to read, not to make the volumes smaller

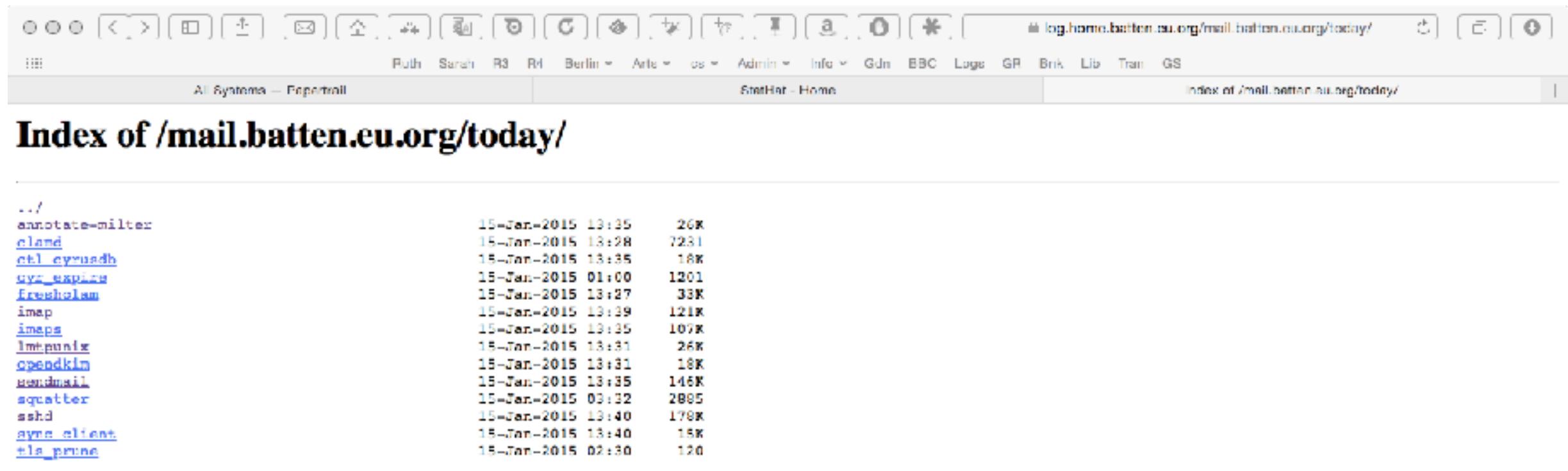
Get it away!

- An attacker can manipulate local log files
- An attack may find it harder to manipulate remote log files, especially if the machine does nothing but accept log entries (papertrail, for example).
- Some people even run the most vital logging straight to a printer in the data centre

Practical Logging

- Let's look at home-brew (syslog-ng) and commercial (papertrailapp).

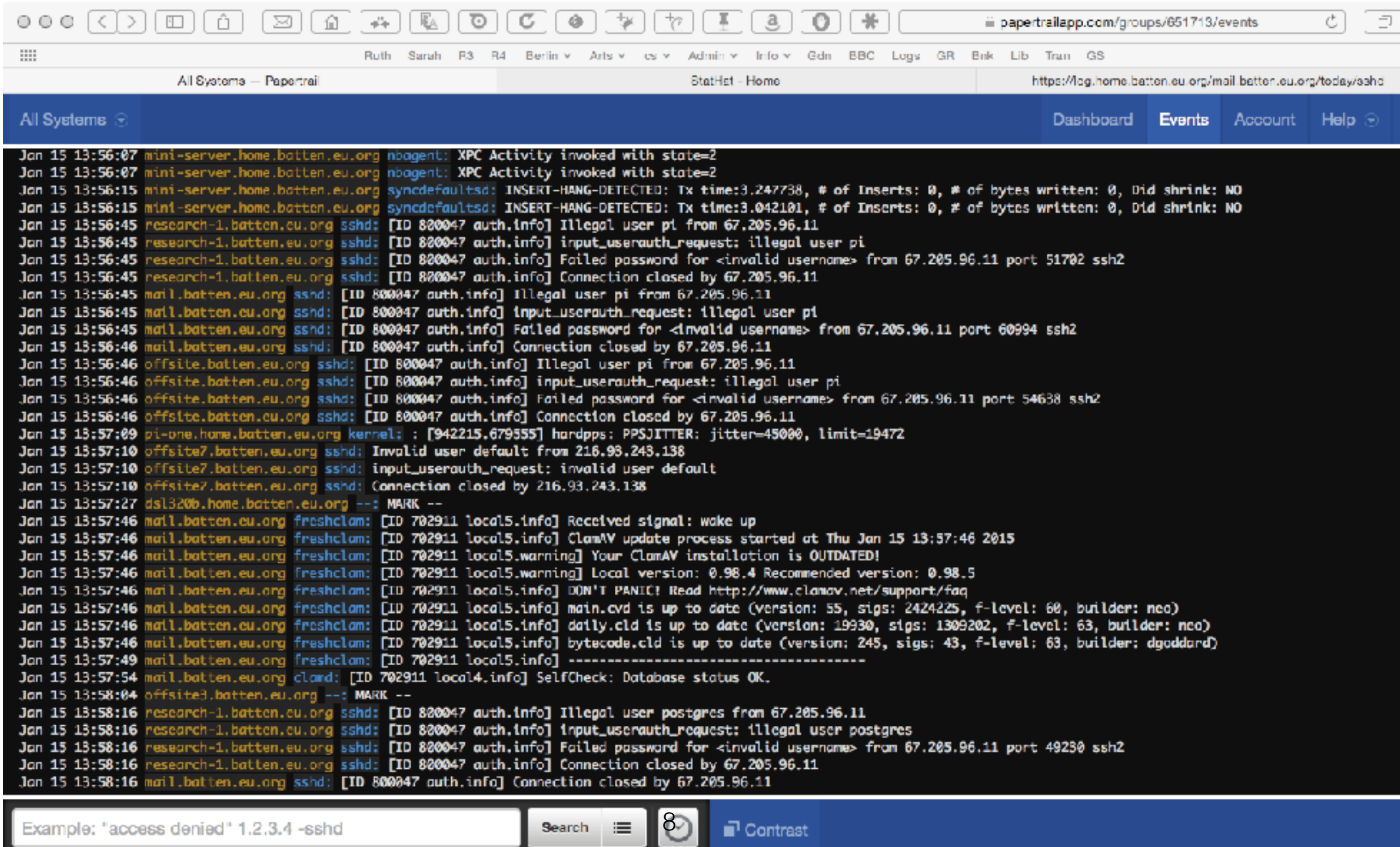
syslog-ng



The screenshot shows a web browser window with the address bar displaying `log.home.batten.eu.org/mail.batten.eu.org/today/`. The browser's address bar also shows the URL `log.home.batten.eu.org/mail.batten.eu.org/today/`. The page title is "Index of /mail.batten.eu.org/today/". The page content is a directory listing of files and directories, including `../`, `annotate-milter`, `clamd`, `ctl_cyrusdh`, `cyr_expire`, `freshclam`, `imap`, `imaps`, `lmtpunix`, `opendkim`, `sendmail`, `squatter`, `sshd`, `sync_client`, and `tls_prune`. Each entry is followed by its last modified date and time, and its size in bytes.

../			
annotate-milter	15-Jan-2015	13:35	26K
clamd	15-Jan-2015	13:28	7231
ctl_cyrusdh	15-Jan-2015	13:35	18K
cyr_expire	15-Jan-2015	01:00	1201
freshclam	15-Jan-2015	13:27	33K
imap	15-Jan-2015	13:39	121K
imaps	15-Jan-2015	13:35	107K
lmtpunix	15-Jan-2015	13:31	26K
opendkim	15-Jan-2015	13:31	18K
sendmail	15-Jan-2015	13:35	146K
squatter	15-Jan-2015	03:32	2885
sshd	15-Jan-2015	13:40	178K
sync_client	15-Jan-2015	13:40	18K
tls_prune	15-Jan-2015	02:30	120

papertrail.com



Patching

- Vulnerabilities are found all the time
- Often in security-sensitive code (they're less interesting in other code, after all, so people aren't looking for them)
- That means regular patching of “exposed” machines
 - If the company policies make this hard, because of configuration management, fix the policies

Linux, for example

- Weekly report:

Reading package lists...

Reading package lists...

Building dependency tree...

Reading state information...

The following packages have been kept back:

fake-hwclock

The following packages will be upgraded:

curl file libcurl3 libcurl3-gnutls libcurl4-openssl-dev libevent-2.0-5

libmagic1 python-rpi.gpio python3-rpi.gpio

9 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.

Problems with patching

- You have code which relies on a bug, and the patch fixes a bug and breaks code
- The patch itself is broken
- Patching doesn't work correctly and breaks the machine

The solution is...patching

- The solution to problems with patching is to patch more often
 - Problems rapidly show themselves
 - Each patch set is smaller
- Alternative is old machines that everyone is frightened of and don't dare update

Custom Software

- A typical problem for Linux is needing magic kernels (eg “server” options on a “desktop” system).

```
-NO_HZ_COMMON y  
HZ_PERIODIC n -> y  
NO_HZ y -> n  
NO_HZ_IDLE y -> n  
+NTP_PPS y
```

- The solution is to build a process for compiling kernels from new source with the required options.

```
Linux pi-one 3.12.36+ #1 PREEMPT-igb Sun Jan 18 19:35:05 GMT 2015 armv6l GNU/Linux  
Linux pi-two 3.12.36+ #737 PREEMPT Wed Jan 14 19:40:07 GMT 2015 armv6l GNU/Linux  
Linux pi-three 3.12.36+ #737 PREEMPT Wed Jan 14 19:40:07 GMT 2015 armv6l GNU/Linux
```

Service Minimisation

- There is no point in running services you do not need
- Attackers can attack anything that is running; they do not helpfully avoid trying the things you've forgotten are there
- So you have the problem of running machines with the bare minimum of services.

Why minimise?

- Services, in this context, are pieces of software accessible to a user who is not logged on.
 - “Accessible” means “can have bytes sent to”: it may not matter if they have not authenticated.
- Every service has security problems, either known or unknown.
- Reducing the number of services reduces the “attack surface”: the number of places an attacker can try.

Services

- SMTP listener for incoming email
- IMAP listener for mail storage
- SSH listener for login
- MySQL/Postgres/etc listener for databases
- DNS listener for nameservers
- Etc

Options

- Don't install the service
- Don't run the service at all
- Run the service, but have it only available locally
- Run the service, but protect it from the network
- Run the service, exposed to the network



Don't install the service / Don't run the service

- If a machine is not a fileserver, you don't need the fileserving software
- If a machine is not a webserver, you don't need the web service
- Don't install it

Run locally

- Sometimes you need service X to support service Y
- An example would be needing MySQL or some other database to support a network management product
- The database is only needed by local services, so can listen on a local socket (AF_UNIX, “named pipe”) or on 127.0.0.1.

Protect from network

- Sometimes this means running a service that is only needed locally, but where the software always wants to listen more widely. Use a firewall.
- Alternatively, if it only needs to listen to **some** sources, don't listen more widely:
 - syslog servers should only listen to machines they are logging for
 - May require VLANs

Run publicly

- If you must. If you must.

Tools to find services

- nmap and other port mapping tools
- netstat -a and other “what’s listening” tools
- lsof/pfiles to look at what a process is listening on
- ps -ef to look at processes (ps -ax on palaeolithic Unixes)
- Windows equivalents are available

Example

```
[igb@offsite7 ~]$ netstat -a | grep LISTEN
tcp        0      0 offsite7.batten.eu.o:domain *:.*      LISTEN
tcp        0      0 localhost.localdomai:domain *:.*      LISTEN
tcp        0      0 *:ssh      *:.*      LISTEN
tcp        0      0 localhost.localdomain:rndc *:.*      LISTEN
tcp        0      0 localhost.localdomain:smux *:.*      LISTEN
tcp        0      0 *:mysql    *:.*      LISTEN
tcp        0      0 localhost.localdomain:http *:.*      LISTEN
tcp        0      0 *:domain   *:.*      LISTEN
tcp        0      0 *:ssh      *:.*      LISTEN
tcp        0      0 *:https    *:.*      LISTEN
unix  2      [ ACC ]     STREAM  LISTENING   2151442158 //var/syslog-ng.ctl
unix  2      [ ACC ]     STREAM  LISTENING   2151441266 @/com/ubuntu/upstart
unix  2      [ ACC ]     STREAM  LISTENING   2151442725 /var/lib/mysql/mysql.sock
unix  2      [ ACC ]     STREAM  LISTENING   2151442135 /dev/log
[igb@offsite7 ~]$ sudo lsof | grep :mysql
mysqld    787    mysql    10u      IPv4        2151442724      0t0      TCP *:mysql (LISTEN)
[igb@offsite7 ~]$ ps -ef | grep 787
mysql    787    685    6 Jan11 ?          14:04:53 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql
--user=mysql --log-error=/var/log/mysqld.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/
mysql.sock
igb      9522  9490   0 20:53 pts/0    00:00:00 grep 787
[igb@offsite7 ~]$
```

Example

```
[igb@offsite7 ~]$ netstat -a | grep LISTEN
tcp        0      0 offsite7.batten.eu.o:domain *:.*          LISTEN
tcp        0      0 localhost.localdomai:domain *:.*          LISTEN
tcp        0      0 *:ssh      *:.*          LISTEN
tcp        0      0 localhost.localdomain:rndc *:.*          LISTEN
tcp        0      0 localhost.localdomain:smux *:.*          LISTEN
tcp        0      0 *:mysql    *:.*          LISTEN
tcp        0      0 localhost.localdomain:http *:.*          LISTEN
tcp        0      0 *:domain   *:.*          LISTEN
tcp        0      0 *:ssh      *:.*          LISTEN
tcp        0      0 *:https    *:.*          LISTEN
unix  2      [ ACC ]     STREAM    LISTENING   2151442158 //var/syslog-ng.ctl
unix  2      [ ACC ]     STREAM    LISTENING   2151441266 @/com/ubuntu/upstart
unix  2      [ ACC ]     STREAM    LISTENING   2151442725 /var/lib/mysql/mysql.sock
unix  2      [ ACC ]     STREAM    LISTENING   2151442135 /dev/log
[igb@offsite7 ~]$ sudo lsof | grep :mysql
mysqld    787    mysql    10u      IPv4        2151442724      0t0      TCP *:mysql (LISTEN)
[igb@offsite7 ~]$ ps -ef | grep 787
mysql    787    685    6 Jan11 ?        14:04:53 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql
--user=mysql --log-error=/var/log/mysqld.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/
mysql.sock
igb      9522   9490    0 20:53 pts/0    00:00:00 grep 787
[igb@offsite7 ~]$
```

```
ians-macbook-air:clocks igb$ telnet offsite7 mysql
Trying 64.188.45.237...
Connected to offsite7.batten.eu.org.
Escape character is '^]'.
4
5.1.73(5rbq4NCVg%~o3MhFeI
```


Example

```
ians-macbook-air:clocks igb$ telnet offsite7 mysql
Trying 64.188.45.237...
Connected to offsite7.batten.eu.org.
Escape character is '^]'.
4
5.1.73(5rbq4NCVg%~o3MhFeI
```

Example

Why are we running mysql? This machine is running cacti, a network monitoring package that needs a mysql data, but is itself accessed only over https.

```
ians-macbook-air:clocks igb$ telnet offsite7 mysql
Trying 64.188.45.237...
Connected to offsite7.batten.eu.org.
Escape character is '^]'.
4
5.1.73(5rbq4NCVg%~o3MhFeI
```

Example

Why are we running mysql? This machine is running cacti, a network monitoring package that needs a mysql data, but is itself accessed only over https.

```
ians-macbook-air:clocks igb$ telnet offsite7 mysql
Trying 64.188.45.237...
Connected to offsite7.batten.eu.org.
Escape character is '^]'.
4
5.1.73(5rbq4NCVg%~o3MhFeI
```

add **bind-address = 127.0.0.1** to /etc/my.cnf and restart

Example

Why are we running mysql? This machine is running cacti, a network monitoring package that needs a mysql data, but is itself accessed only over https.

```
ians-macbook-air:clocks igb$ telnet offsite7 mysql
Trying 64.188.45.237...
Connected to offsite7.batten.eu.org.
Escape character is '^]'.
4
5.1.73(5rbq4NCVg%~o3MhFeI
```

add **bind-address = 127.0.0.1** to /etc/my.cnf and restart

```
[igb@offsite7 ~]$ sudo vi /etc/my.cnf
[igb@offsite7 ~]$ sudo service mysqld restart
Stopping mysqld:
Starting mysqld:
[igb@offsite7 ~]$
```

```
[ OK ]
[ OK ]
```

Example (cont'd)

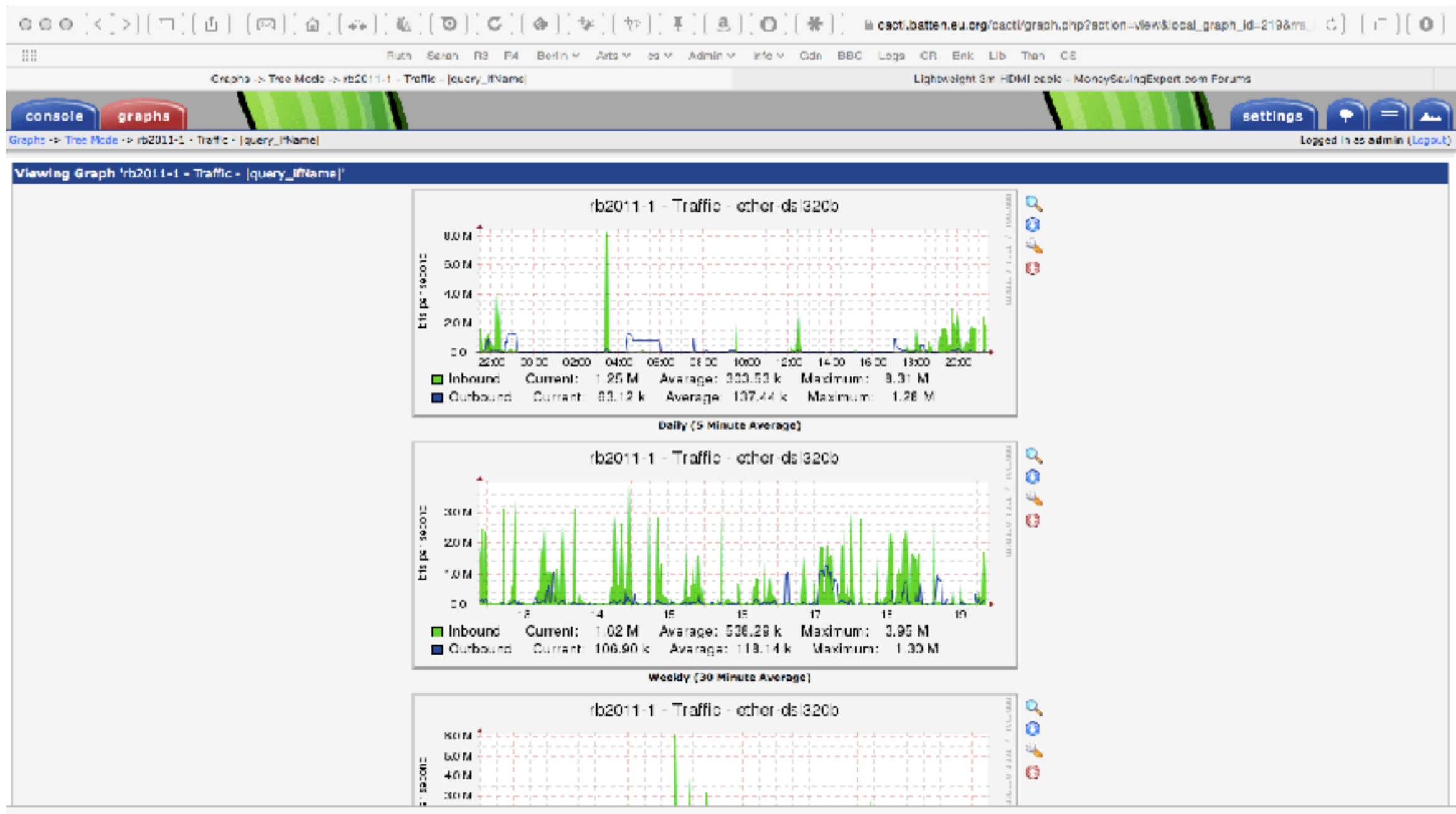
```
[igb@offsite7 ~]$ ps -ef | grep mysql
root      9765      1  0 21:00 pts/0    00:00:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --socket=/var/lib/
mysql/mysql.sock --pid-file=/var/run/mysqld/mysqld.pid --basedir=/usr --user=mysql
mysql     9870   9765  0 21:00 pts/0    00:00:00 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql
--log-error=/var/log/mysqld.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/mysql.sock
igb       9914   9490  0 21:01 pts/0    00:00:00 grep mysql
[igb@offsite7 ~]$ sudo lsof -p 9870 | grep TCP
mysqld 9870 mysql 10u IPv4 2350892539 0t0 TCP localhost.localdomain:mysql (LISTEN)
[igb@offsite7 ~]$ netstat -a | grep mysql
tcp        0      0 localhost.localdomain:mysql *:*          LISTEN
unix  2      [ ACC ]     STREAM  LISTENING   2350892540 /var/lib/mysql/mysql.sock
[igb@offsite7 ~]$
```

Example (cont'd)

```
[igb@offsite7 ~]$ ps -ef | grep mysql
root      9765      1  0 21:00 pts/0    00:00:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --socket=/var/lib.
mysql/mysql.sock --pid-file=/var/run/mysqld/mysqld.pid --basedir=/usr --user=mysql
mysql     9870   9765  0 21:00 pts/0    00:00:00 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql
--log-error=/var/log/mysqld.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/mysql.sock
igb       9914   9490  0 21:01 pts/0    00:00:00 grep mysql
[igb@offsite7 ~]$ sudo lsof -p 9870 | grep TCP
mysqld  9870 mysql  10u  IPv4  2350892539      0t0      TCP localhost.localdomain:mysql (LISTEN)
[igb@offsite7 ~]$ netstat -a | grep mysql
tcp        0      0 localhost.localdomain:mysql *:*          LISTEN
unix  2      [ ACC ]     STREAM  LISTENING   2350892540 /var/lib/mysql/mysql.sock
[igb@offsite7 ~]$
```

```
ians-macbook-air:~ igb$ telnet offsite7.batten.eu.org mysql
Trying 64.188.45.237...
telnet: connect to address 64.188.45.237: Connection refused
Trying 2607:f2e0:10f:14:4321:4321:5e6:9ee6...
telnet: connect to address 2607:f2e0:10f:14:4321:4321:5e6:9ee6: Connection refused
telnet: Unable to connect to remote host
ians-macbook-air:~ igb$
```

And to test...



Summary

- We are running cacti, a service which needs mysql
- Mysql was installed so it was listening to the internet generally, but was only needed locally
- Mysql modified to only listen locally
- Cacti still works
- This is a real example: I found it while doing the slides!