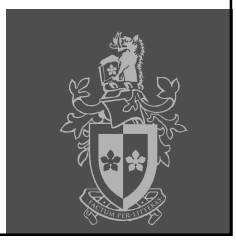
SWIN BUR NE

CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Upgrading the Remote Unix Lab Environment (RULE)

Gareth Edwards
Joanne Ly
Huu Ngoc Duy Nguyen

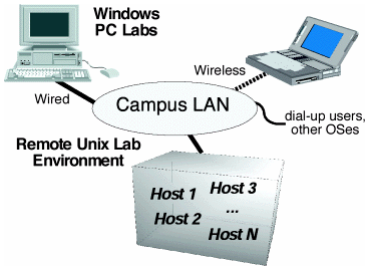


Overview

- RULE Background
- Project Aim
- Proposed Changes to RULE
- Description of RULE Prototype
- Further Work

RULE Background

- Remote Unix Learning Environment
  - Multiple virtual Unix-like hosts
  - Uses FreeBSD's jail functionality
- Used in HET436: Broadband Multimedia Networks lab exercises
  - E.g. Setting up proxy servers



RULE Background (Continued)

- Disadvantages
  - The jail functionality does not allow students to have full control of networking functionalities
    - No access to the kernel
    - Simplistic lab exercises in comparison to subject contents
    - More suitable for usage in a first year subject
  - Resource management mechanism
    - Does not ensure processes from one jail host does not starve processes from another jail host

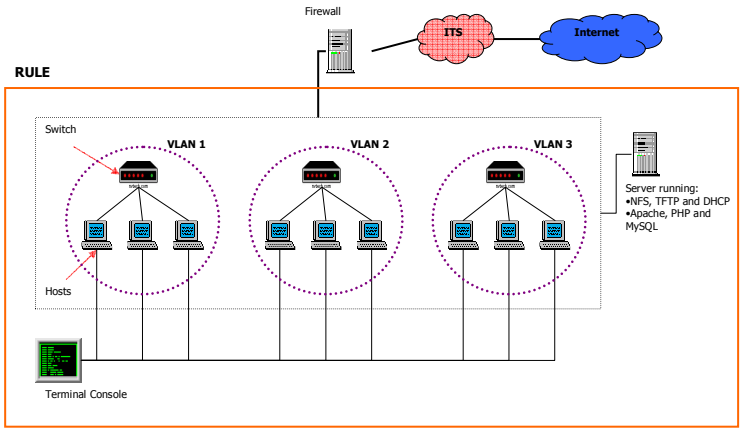
Project Aims

- Allow each student their own dedicated FreeBSD host(s)/client(s)
  - Students able to modify and rebuild kernel
- Provide mechanisms to restore hosts to a working configuration
  - Controlled via a web interface
- Implement remote hardware resetting

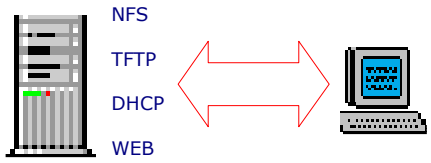
RULE Prototype Description

- Each lab group is allocated an 802.1q VLAN
- VLAN connected to a server (through a switch) running:
  - NFS, TFTP and DHCP for remote, diskless booting
  - Apache, PHP and MySQL for web interface
- Hosts connected to a terminal console server
  - Reused from old RULE system

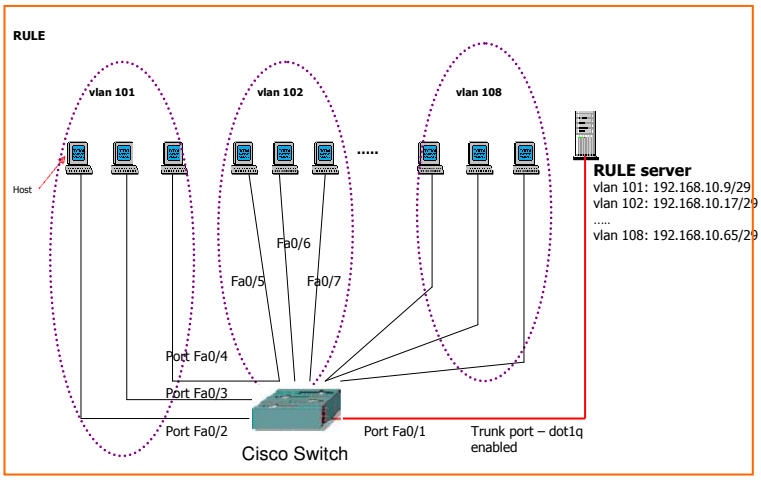
RULE Prototype Description (continued)



RULE Server



VLAN



VLAN (continued)

- FreeBSD supports multiple pseudo interfaces (VLANs).
- Assign multiple IP addresses, subnet masks and VLAN tags to one Network Interface
- Used a Cisco switch
  - 802.1q VLAN setup
  - Turn off Spanning-Tree Protocol to avoid PXE boot timeout

Terminal Console Server

- Remote console access to RULE hosts
- Reused from old RULE system
  - Supports multi-port serial cards
  - Free “conserver” software package

Conserver Configuration

- Define serial ports to be monitored and the speed
- Ensure boot loader uses the serial port
  - `boot.config`
- Allow logins over the console port
  - `/etc/ttys`

Remote booting of hosts

- Implemented using Pre-boot eXecution Environment® (PXE)
  - Used to boot diskless machines through a network connection
  - DHCP assigns static IP address to host machine, identified by its MAC address
  - DHCP points to functional `pxeboot` executable
  - TFTP transfer of `pxeboot` to host's local memory
  - Host NFS-mounts a nominated file system

DHCP Configuration

- Contents of `dhcpd.conf`

```
host client2 {
        #Assign a static IP to the client
        hardware ethernet 00:40:63:cb:56:e5;
        fixed-address 192.168.10.15;
        #Assign a hostname to the client
        send host-name "client2";

        #TFTP Server
        next-server 192.168.10.1;
        #Copy of PXEBOOT on TFTP Server Location
        filename "client2/boot/pxeboot";
        #Root path to find 'boot' folder
        option root-path "/diskless/client2/";
}
```

Booting Scenarios

- The student:
  - Has modified their kernel and would like to boot straight off the HDD
  - Needs to perform recovery operations
    - Boot a GENERIC kernel
    - Mount a compressed RAM image of a “known working” file system exported from a NFS server
  - Wants to boot a GENERIC kernel
    - But have their file system on the HDD automatically mounted

Modifying the PXE Boot Process

- Standard process needs to be modified
  - Each time `dhcpd.conf` is changed, need to restart DHCP
  - Will disrupt other students
- Better solution:
  - Control via combination of different `loader.conf` files and softlinks

loader.conf

- Contains boot information
  - the kernel to be booted
  - the file system to be mounted
  - where to get each one from
    - NFS server
    - Host HDD
- Can edit this file without requiring any processes to restart

Soft links

- `dhcpd.conf` specifies location of host boot folder
  - Paths remain static for each individual client
  - Can use soft links to point from these locations to a file structure on the NFS server
- Links removed and added depending on how the user requires each client to boot

```
client1 -> rulehost/fbsd410_inst
client2 -> rulehost/fbsd54_ram
client8 -> rulehost/fbsd54_inst
client9 -> rulehost/fbsd54_ram
```

PXE Booting: Scenario 1

- Booting straight off the HDD
- Remove link to host's boot folder
- PXE boot fails
  - Unable to find boot folder
  - Proceeds to boot of HDD

PXE Booting: Scenario 2

- Boot a GENERIC kernel
- Mount a compressed RAM image of a file system
- Host goes through general PXE boot process
  - Contents of `loader.conf` file in `boot` folder

```
mfsroot_load="YES"
mfsroot_type="mfs_root"
mfsroot_name="/boot/mfsroot"
vfs.root.mountfrom="ufs:/dev/md0c"
init_path="/sbin/init"
```

PXE Booting: Scenario 3

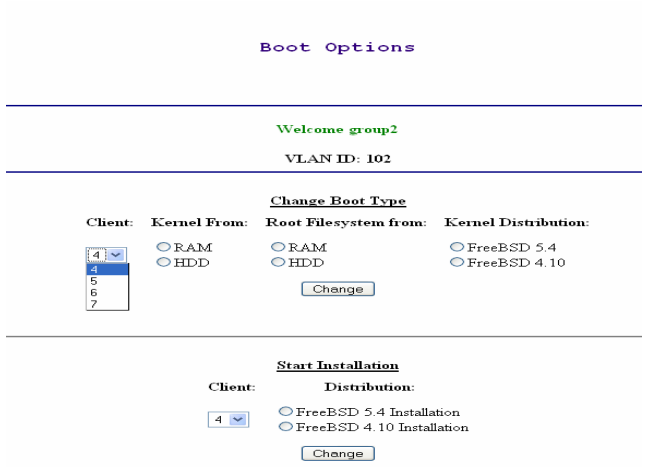
- Boot a GENERIC kernel
- Mount file system off HDD
- Host goes through general PXE boot process
  - Contents of loader.conf file in boot folder

```
vfs.root.mountfrom="ufs:/dev/ad0s1a"
```

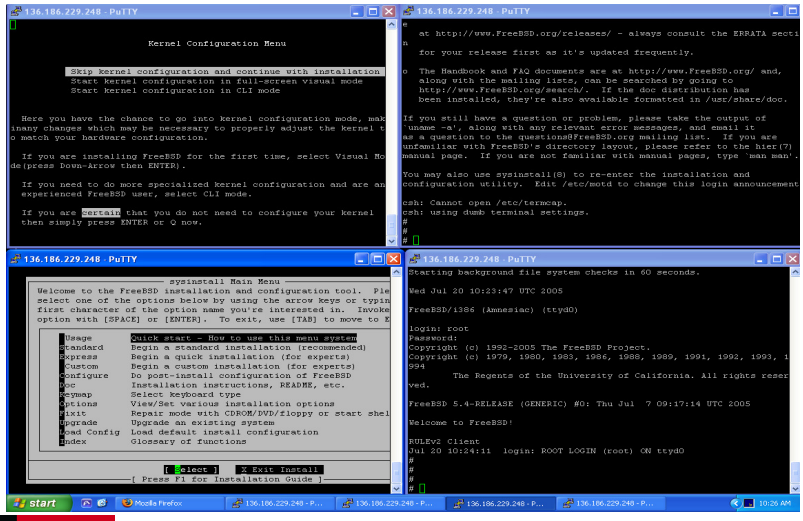

Web Interface

- Implemented using PHP and MySQL
- MySQL database
  - Consists of 3 tables
  - Table 1: Stores lab groups' passwords
  - Table 2: Stores a list of which VLANs are assigned to a lab group
  - Table 3: Stores which hosts are assigned to a VLAN

Web Interface



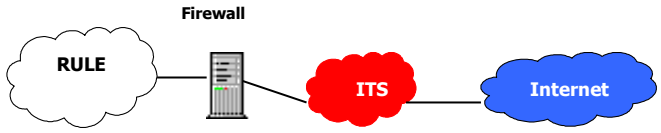
Console Interface



Hardware Reset

- Enable physical control of remote host
- Parallel port of NFS sever connected to motherboard power header via simple circuit
- Server application causes power pin to be tied to ground for specified amount of time
  - Defined by type of reset

Firewall



- IP Firewall (IPFW)
- NATD
- DUMMYNET

Firewall (continued)

- Rebuild the kernel
- Enable IP Firewall (IPFW), NATD, DUMMYNET at start-up
- Define IPFW rules at: `/etc/ipfw.rules`

Dummynet

- Fair share of bandwidth utilisation between RULE hosts
- Limit the inbound and outbound bandwidth of each RULE host to 200Kbit/s
- Can be included in `/etc/ipfw.rules` file

Further Development

- Booting different operating systems
- Booting different types of motherboards
- Administration web interface
  - To reallocate hosts to VLANs
  - To reallocate VLANs to lab groups
  - To update group passwords