Project 2, Part 3

cse461

Serving Dynamic Content

- Serve information about the current state of the router using your web-server
- Due next Friday, 11/14

Required Information

- Number of times the router rebooted
- MAC addresses of hosts sending packets on LAN
- Total number of broadcast packets seen on the LAN by the router
- Average bandwidth utilization over (1) last minute, (2) last hour, (3) last 6 hours

Number of Reboots (one of many strategies)

- Maintain number of reboots in a file
- Assume your server starts at system boot time
  - Download script for this from the project page
- Read /proc/uptime in web-server, update number of reboots if uptime < last uptime
- Serve the string "Num Reboots: " + file contents

For last three reqs use pcap

- MAC addresses of hosts sending packets on LAN
- Total number of broadcast packets seen on the LAN by the router
- Average bandwidth utilization over (1) last minute, (2) last hour, (3) last 6 hours

libpcap (packet capture library)

How to use pcap

1. Determine which device(s) to sniff on
2. Tell pcap which device(s) to sniff on
3. Create, compile, and apply sniffing filters
4. Tell pcap to enter an execution loop
5. Receive packets via a callback function
6. Close pcap session

Which devices to sniff on

- Use ifconfig to determine device names
- Compile and run pcap_list_devs.c
  - (Download from the project page)
- Or use ‘**all**’ as device name to capture packets on all interfaces

Initializing pcap

- **pcap_open_live** (char* device, int snaplen, int promisc, int to_ms, char* ebuf)
  - device : device name (from prior step)
  - snaplen : max number of bytes to capture
  - promisc : promiscuous mode (1: capture packets not destined for nor generated by this host, or 0: to not do so)
  - to_ms : ms internal read waits before timing out (see pcap_dispatch() execution loop explanation)
- Returns a **handle** used for all further interactions with pcap

Setting up pcap filters

- pcap uses tcpdump filter syntax!
  - e.g. ‘port 23’ or ‘ip’
- pcap_compile : compile a string filter into internal pcap representation
- pcap_setfilter : enable a filter on a pcap handle

Execution loops

- pcap_next : capture just one packet
- pcap_loop : capture # of packets
- pcap_dispatch : capture until internal read times out
- For this project use pcap_loop

Close pcap session

- Always release pcap allocated memory
  - pcap_freecode : frees filter-related allocations
  - pcap_close : releases a pcap handle

X-Compiling pcap programs

- Create binary on attu using:

  ```
  /homes/iws/ivan/bin/gcc [filename.c] /homes/iws/ivan/libpcap.a
  ```

- Move binary to router
- Run binary on router

pcap resources

- man pcap
- Read the tutorial:
  - http://www.tcpdump.org/pcap.htm
- Look at example source code:
  - http://www.tcpdump.org/sniffex.c

Extra Credit

- Serving other dynamic content
    - Frequency table of observed ports
    - Traffic statistics per observed IP
    - Periodic sampling of traffic flows
    - Other?