**Workshops on Programming Logic and Computer Networks**

**Dr AIT Bangalore**

Day 01: Programing logic –I

| Sess # | Theory content |
| --- | --- |
| S1 | Registration and Keynote address |
| S2 | Write logic for a simple requirement and compare logic arrived at by participants and analyze. |
| S3 | Write logic for requirements involving matrix operations and simple numerical methods. |
| S4 | Examine the different solutions arrived, and how these can be improved. Compare results, analyze and discuss. |

Requirements:

Pre-requisite: None

Other requirements: Notepad and pen. **Computers to be avoided**.

Day 02 : Programing logic –II

| Sess # | Session focus |
| --- | --- |
| S1 | Write logic for simple requirements involving strings. Some of these could even be to build the logic implemented in some of the standard library functions |
| S2 | Work out exercises dealing with file manipulations. Compare the results, Discuss refinements to improve the results and comparative analysis. |
| S3 | Team Exercises: Exercises as per participants’ background. Mimic some of the practices in a software development life cycle. |
| S4 | Open discussion and Closing Remarks |

Requirements:

Pre-requisite: None

Other requirements: Notepad and pen. **Computers to be avoided**.

Day 03 :Basics of Networking

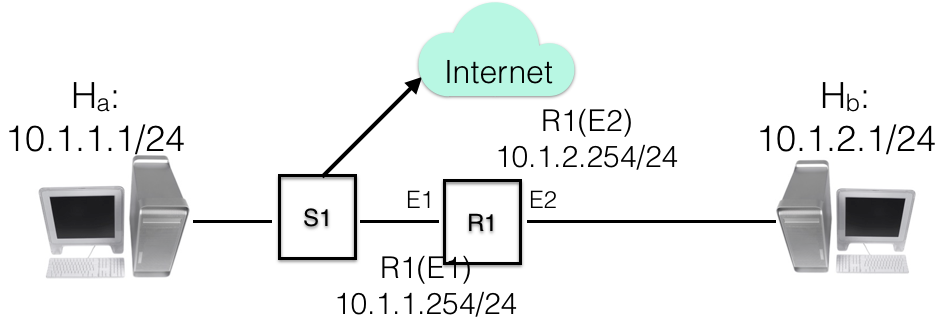
| Sess # | Theory content | Hands on Learning |
| --- | --- | --- |
| S1 | Understanding basic networking diagnostics and debugging tools e.g.  nc, wget, curl, ssh, wireshark | Wireshark: capture and display filters, protocol dissection, TCP sessions, packets export, opening existing capture files etc.  tcpdump: command line options for wireshark  nc: tcp and udp communication  wget: offline browser, web crawling, parameters passing, header passings etc.  ssh: auto login, shell access, proxy access, .tunneling |
| S2 | Understanding TCP/IP stack 4-layers | Analysing layer 2 (Ethernet), Layer 3(IP), Layer 4 (TCP/UDP) and Layer 7 (HTTP) |
| S3 | Understanding IP addressing and subnets | Subnet creation, hierarchical addressing, supernetting.  Analysing unreachability with subnet mismatch. |
| S4 | Communication across subnets. What is required for machine in one subnet to reach a machine in another subnet.  Longest prefix matching | Creating basic subnets  Creating Hierarchical subnets  Creating overlapping subnets with longest prefix matching. |

**Requirements:**

Pre-requisite: None

**Setup requirement per team**.

* 3 systems on a network with internet connectivity.
* One system to act as router with two network interfaces (can be either another Ethernet interface, or USB to ethernet adaptor) .
* R1 and Hb to run Ubuntu 6.04 LTS.
* USB to Ethernet adaptor
* Ethernet patch cords (RJ45 cables).



Day 04 : Basics of Network Programming

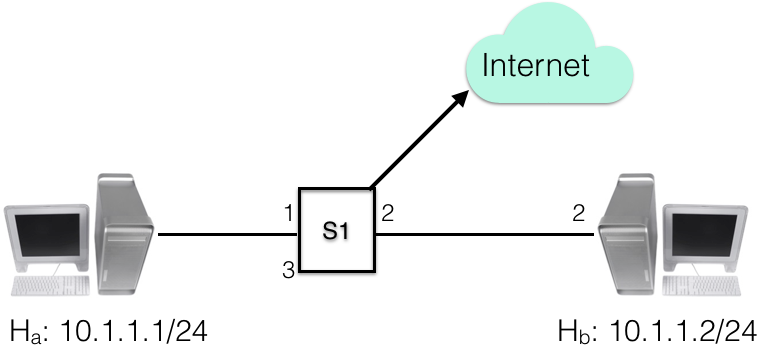
| Sess # | Theory content | Hands on Learning |
| --- | --- | --- |
| S1 | Understanding basics of Inter process communication. Understanding socket calls for client side and server side. Difference socket API flow between client side and server side. | Learning of TCP programming with simple socket calls.  Writing simple client side program and simple server side programs. |
| S2 | Analysing basic errors in socket programming. Importance of network byte ordering, buffer initialisation, partial read, connection close and errors | Learning of programming practices to ensure network byte ordering, avoiding buffer initiation and its impacts.  Handling partial reads |
| S3 | Handling of multiple concurrent clients with single threaded server program  Understanding send success even without data going to other side and TCP buffering | Learning client side established connection even without server doing an accept of connection.  Learning buffering of data  Learning impact of listen() for clients exceeding acceptable queue size |
| S4 | Understanding select() call to handle multiple concurrent calls. | Experiential learning of blocked sockets  Learning the difference of readable, writable and erroneous state sockets. |

**Requirements**:

Pre-requisite: Basic programming skills

**Setup requirement per team**:

2 systems on a network with internet connectivity



Day 05: Basics of web security and IEEE publicationss

| Sess # | Theory content | Hands on Learning |
| --- | --- | --- |
| S1 | Understanding HTTPS protocol  Understanding certificate Mgmt.  Understand HTTP mixed content and its implication.  Understand lock icon and its various forms to understand security risks | Generate a self signed certificate and deploy.  Analyze browser error with certificate mismatch.  Experience lock icon various forms and behaviour with active and passive mixed content |
| S2 | Overview of basic MITM (Man In The Middle) attack.  Understand ARP spoofing to implement MITM.  Understand CSP (Content Security Policy).  Understanding HSTS protocol. | Use arpspoof to implement MITM.  Eavesdrop on the communication between two users using MITM setup and analyse data using wireshark.  Use CSP to enable/disable XS and other security policies |
| S3 | IEEE SWEBOK  IEEE publication process | Open discussion with IEEE staff |
| S4 | Closing ceremony | Valedictory address |

**Requirements**:

Pre-requisite: Basics of Networking

**Setup requirement per team**.

* 3 systems on a network with internet connectivity.
* One system to act as MITM Attacker with two network interfaces (can be either another Ethernet interface, or USB to ethernet adaptor).
* Attacker system run Ubuntu 6.04 LTS with package **dsniff**, and **sslstrip**
* USB to Ethernet adaptor, and Ethernet patch cords (RJ45 cables).

