INTRODUCTORY SESSION ON SOCKET PROGRAMMING

SUBHOJEET MUKHERJEE

CSU DATABASE AND SECURITY RESEARCH GROUP
WHO AM I

• Phd student at Database and Security Group
• Advisors
  • Dr. Indrajit Ray
  • Dr. Indrakshi Ray
  • Dr. Adele Howe (Master’s)
• Focus
  • In-Vehicle Network Security
  • Access Control Systems
  • Cyber Security Concept Extraction
  • Personalization of Security

RECITATION FACTS

Did You KNOW ……

• Our recitations will be held on
  • W : 12:12:50 pm
  • F : 9-9:50 and 11-11:50 pm
  • Room: 315 CSB
• You can use your laptops in-lab.
• Stop me anytime I go into the Usain Bolt mode.
• Ask billions of questions.

RECITATION TOPICS

• Socket Programming
• Networking Tools
• Useful tips and tools for projects.
• Hands-on programming.
Mr. Beaver (computer B) wants to chat with Mr. Tortoise (computer T).

- Beaver wants to know the recipe for a perfect martini.

B and T are connected to the same network, say on two CS lab 120 machines.

- Their machines do not have GUI support.
  - So no social network, no chat messengers.

What's worse, they can't even install command line chat tools.

- Beaver has to write his own chat program and share it somehow with Tortoise.

- The program takes input from STDIN and sends it over the network.
A TECHNICAL OVERVIEW OF THE CHAT PROCESS

Beaver and Tortoise prepare for chat
- Tortoise asks
  - What port should I listen on? / What port am I listening on?
- Beaver asks
  - Is machine T alive and listening on port?
  - Beaver types his question → STDIN.
- Program makes a list of system calls via socket API functions.
- Kernel level (TCP/IP) operations are performed.
- Network Interface Card fetches and transfers information on the wires.
- On the receiving end the opposite happens
  - Except some socket functions are triggered on receiving a new message.
RECITATION COVERAGE

• Preparatory Steps
  • A very brief introduction to some networking tools.

• C/C++ Programming
  • A detailed tutorial ..... Just kidding !!
  • MakeFile
  • GNU Debugger

• Socket API
  • All of our projects use the Client/Server paradigm.
  • We will do a lot of programming using sockets and our recitations will be highly focused on this.

• Kernel Level Operations [TCP/IP concepts]
  • In class theory.

• Hardware
  • You can check internet resources if you are interested.

• Communication Concepts
  • Not covered in recitations.
PREPARATORY STEPS
SOME USEFUL NETWORKING TOOLS

• Check if ports are busy
  • Do not use ports having numbers less than 1024.
  • Try using ports within the range 49152 to 65535 [ephemeral ports].
  • Use `netstat` to see if another user program is using the same port.
    • Eg. `netstat -tulpn | grep <port-num>`

• Is the other machine listening on the particular port?
  • `telnet <machine-IP> <port>`
    • “Connected ” means it listens. Ctrl + ] and then type ‘quit’ to exit.
C/C++ TOOLS
MAKEFILE

CC='gcc'/'g++'
COPTS=–Wall -o

socket_program: socket_client.o socket_server.o
  $(CC) $(COPT) socket_program socket_client.o
  socket_server.o socket_program.c
  socket_client.o: socket_client.c socket_client.h
  $(CC) –c $(COPT) socket_client.o socket_client.c
  socket_server.o: socket_server.c socket_server.h
  $(CC) –c $(COPT) socket_server.o socket_server.c

clean:
  rm –f *.o socket_program

• Basic form
  • Target : dependencies
  • <TAB> build command

• Variables
  • Evaluated using $(variable).

• Execution Command
  • Run ‘make’ in the directory containing the makefile.

• Fresh make
  • Type ‘make clean’
  • Type ‘make’

• Much more , even conditional evaluations
  • Online !!
int main()
{
    struct abc *random;
    random->count = 54;
    return 0;
}

• How many times have you done this?
  • And got this.
• Debugging with printf-s and cout-s can become tedious and tricky especially if the code is fairly long.
• With Open Arms welcome the GNU Debugger (GDB).
• Simply compile your code with the –g option.
• Load it into GDB using the command `gdb <your exec binary>`.
• Once into the GDB shell run `run`.
• Your code will execute until it fails.
  • Check the program stack
  • Print temp values
  • Set break points and watch the program flow step by step.
• Check tutorials online
  • http://www.unknownroad.com/rtfm/gdbtut/
  • http://www.yolinux.com/TUTORIALS/GDB-Commands.html

Segmentation Fault (Core Dumped)