CS155: Computer and Network Security

Programming Project 3 – Spring 2004

Matt Rubens
mrubens@stanford.edu

Project Overview and Setup

Project Overview

1) Use standard network monitoring tools to examine different networking protocols
2) Use a packet capture library to automatically intercept FTP transfers
3) Write a program to perform an injection attack on the RLOGIN protocol

Goals of the assignment

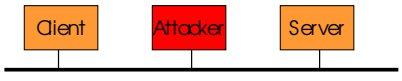
- Get some hands-on networking experience
- Learn how secure different protocols are
- Learn about common attacks on clear-text protocols
- DON'T end up in jail
  - Never test your code outside of the boxes environment!

Setup

- You are given three cow images corresponding to three separate machines on the network
  - Client, server, and attacker
- There are a number of users on the client sending network requests to services on the server
- The attacker (you!) is trying to perform different attacks (the assignment) on the client and server

Setup (2)

- All three boxes are located on the same Ethernet hub
- Ethernet is a broadcast medium
  - Every machine sees every packet, regardless of address!
    - Normally, packets not intended for a host are discarded by the network card
    - But in promiscuous mode all packets are available!



Setup (3)

- To start up the boxes, follow these steps
  - xterm –e ./string &
    - Make sure to use the copy of *string* included with the cow images!
      - Otherwise the attacker will not be to see the network traffic
  - xterm –e (open|closed)box clientcow 10.64.64.64 &
  - xterm –e (open|closed)box servercow 10.64.64.65 &
  - xterm –e (open|closed)box attackcow 10.64.64.66 &
- You must use these exact IP addresses!

Setup (4)

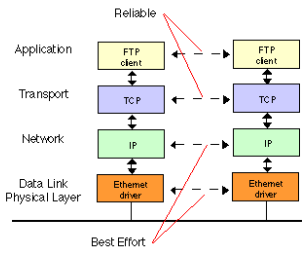
- You are NOT given an account on the client and server machines
  - If you're good you might get one soon!
    - Once you have a password, you can remotely shutdown the client and server with
      - ssh (username)@(ipaddr) /sbin/halt
        - We installed halt as setuid-root (bad idea in general!)
  - But until then, you won't be able to do a clean shutdown on clientcow and servercow
    - So keep a backup of the original images to avoid fscking

Quick TCP/IP Review

TCP/IP Overview

- On this assignment, we are only dealing with protocols that run over TCP/IP
- We assume a basic knowledge on the level of packets and ports
  - If you're not that comfortable with this, stop by office hours

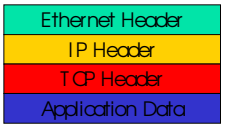
Relevant Network Layers



From http://www.erg.abdn.ac.uk/users/gorry/course/images/ftp-tcp-enet.gif

Cliffs Notes Version

- Each TCP packet that you see is actually a TCP packet wrapped inside of an IP packet wrapped inside of an Ethernet packet.



TCP Flags

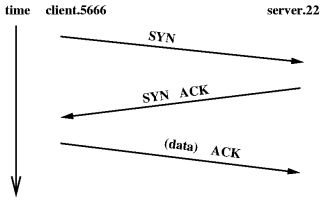
- Synchronize flag (SYN)
  - Used to initiate a TCP connection
- Acknowledgement flag (ACK)
  - Used to confirm received data
- Finish flag (FIN)
  - Used to shut down the connection

TCP Flags (2)

- Push flag (PSH)
  - Do not buffer data on receiver side – send directly to application level
- Urgent flag (URG)
  - Used to signify data with a higher priority than the other traffic
    - I.e Ctrl+C interrupt during an FTP transfer
- Reset flag (RST)
  - Tells receiver to tear down connection immediately

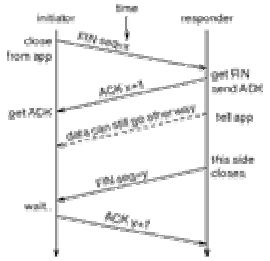
Connection setup

- "Three-way handshake"



From http://www.cs.colorado.edu/~tor/sadocs/tcpip/3way.png

Connection termination



- Either side can initiate termination
  - Note that the first FIN packet may still contain data!

From http://homepages.feis.herts.ac.uk/~cs2_sn2/sn2-img62.png

The actual assignment (finally!)

Phase 1: Sniffing

- Goal: observe network traffic, learn about different protocols
  - Also: gain access to client and server machines in order to make Phases 2 and 3 easier!
- Installed tools (must be run as root):
  - Tcpdump
    - Old faithful, just gives raw packet info
  - Tethereal
    - Like tcpdump, but with more smarts about protocols
  - Tcpflow
    - Focuses on the payload of the packets
    - Great for examining application level data (i.e passwords)!

Tcpdump options

- All three network monitoring tools take similar command line options
  - Can filter packets by address, port, protocol, length, TCP flags, etc.
    - Make sure to read the tcpdump manpage closely!
  - For your submission, we want you to list the options that you used to isolate the packets containing username/password information.

Phase 2: File Eavesdropping

- Manual packet sniffing is an interesting exercise, but programmatically capturing packets is much more powerful
- In this part of the assignment, you will write a program to reconstruct a sniffed FTP file transfer

Libpcap

- Libpcap is a packet capture library written in C
  - It allows you to write code to automate packet sniffing attacks.
- The library is fairly simple to use
  - Pseudocode:

```
while (true) {
  packet = pcap_next();
  // do something with the packet
}
```

- We give you starter code in /home/user/pp3/sniff.c on the attackcow image.

What to do

- Figure out which packets correspond to an FTP file transfer
- Detect when a transfer starts and create a local file to store the data
- Extract data from packets and write them to the file
- Figure out when the transfer completes, close the file, and exit the program

What to do (2)

- The hard part is figuring out how to parse the various layers of headers.
  - You can find the header definitions at:
    - Ethernet: /usr/include/net/ethernet.h
    - IP: /usr/include/netinet/ip.h
    - TCP: /usr/include/netinet/tcp.h
- You'll also need to figure out how FTP data transfers work
  - Using the techniques you learned in Phase 1 might be more productive than poring over protocol docs

Phase 3: Packet Injection

- RLOGIN - allows remote login session
  - Very similar to Telnet
- Does not ask for password if the client machine is mentioned in /etc/hosts.equiv or ~/.rhosts
  - (big convenience.... even bigger vulnerability)
- After authentication - the rest of the traffic is in the clear!
- Uses one TCP channel for communication

Attacks

- Can spoof an entire TCP connection
  - If the spoofed sender is present in /etc/hosts.equiv or ~/.rhosts, server won't ask for password
- Already established session can be hijacked by spurious injections (what you will do)
  - You can run any command on the server with the permissions of the client
    - i.e. /sbin/halt (if halt is setuid-root), rm –rf, etc.

Libnet

- Packet injection library
  - Allows you to modify each and every field of packet
  - Build packets from top to bottom : TCP -> IP -> Ethernet
  - Automatically calculates correct checksums - no need to worry about them
- Starter code is provided for you in /home/user/pp3/inject.c on the attackcow

What to do

- Observe traffic generated by an ongoing rlogin session
  - for each interactive action, 3 packets will be generated
    - client -> server : with the data (for eg: "ls\r\n")
    - server -> client : echo the data - ack the previous packet (also send results of command)
    - client -> server : ack the server packet
- Find out the correct sequence number (and other fields) to put in your malicious packet

What to do (2)

- Other information to take care of :
  - TCP header
    - TCP options - contain timestamps of the packet being acked
    - port numbers
    - window size
  - IP header
    - source/destination IP addresses
    - TOS : type of service
    - IP flags
    - IP ID
  - Ethernet header
    - source/destination Ethernet addresses

What to do (3)

- You might try to figure out a way to get your own rlogin account on servercow
  - Then you could easily test out your injection program

Wrapup

- This whole assignment shouldn't take more than a couple hundred lines of code
  - However, it requires a good understanding of what's happening on the network
  - The programs seem simple, but they can take more time than anticipated (remember pp1?)
  - Enjoy yourself – this is fun stuff!