IESoc Course: Network Security Fundamental

Adrian Sai Wah Tam
(swtam9@ie)

April 10, 2006

1 Outline

- How network is different from standalone PC
- How TCP/IP and Ethernet works?
- Attacks
  - Sniffering
  - Sproofing
  - Connection hi-jack
  - Redirect attack
  - DDoS
  - SQL injection

2 How Network is Different from Standalone PC?

- Network is multiple access (usually)
  - So we can sniffer
- Network communication is routed
  - So we can sproof
- Network connection is a state machine (usually)
  - So we can hi-jack
- Network communication is complicated
  - So we can DDoS

3 How networked computers work?

3.1 Bottom layers: From Ethernet to IP packet

- Ethernet: Send and receive, multiple access
  - Strictly speaking, all packets are broadcasted
  - "Packets" in Ethernet is called "frame", with maximum size of 1500 bytes
- Switch: Selective transmission
  - Do not forward packets to unnecessary hosts
  - Better than hub
  - Make hacking harder (but not really)

- TCP/IP: The most widely-used protocol
- How does IP packets get into network?
  - Essential things about IP network:
    1. My address (e.g. 192.168.0.1)
    2. Subnet mask (e.g. 255.255.255.0)
    3. Gateway address (e.g. 192.168.0.254)
  - Derived information:
    1. Network address: (My Address) & (Subnet mask)
    2. Broadcast address: (My Address) | ~(Subnet mask)
  - For any two nodes, whose network addresses are the same, then send directly
  - If the two nodes are not in the same network (network address are not the same), then send through gateway
- Direct sending: ARP, assume node A send to node B
  1. node A send ARP in broadcast mode: "who is B", then wait for reply
  2. node B get the ARP request, send the reply "B is 00:AA:11:BB:22:CC"
  3. node A learnt about B and its MAC address, then it constructs Ethernet frame targetted for B
  4. Send a series of Ethernet frame to B
- Indirect sending: using Gateway
  1. node A knows who is the gateway
  2. node A learn about gateway's MAC address using ARP request-reply
  3. node A construct Ethernet frame, which the IP receiver = B but the Ethernet receiver = gateway
  4. send

3.2 Upper layers: From TCP to Application

- TCP is something stored in IP packet
- TCP is a stream of bytes (e.g. a file), but put into pieces
  - Encapsulation
  - TCP is encapsulated in IP packets, which in turn encapsulated in Ethernet frames
- TCP and UDP have separate port space, with total 65535 ports each
  - One port for one application/program
- TCP start-up: Three way handshake
  1. Receiver (server) *listen* to a port
  2. Sender (client) try to *connect* to receiver
     (a) Sender send SYN, with sender's initial sequence number
     (b) Receiver get the SYN, reply with SYN+ACK, which set the receiver's initial sequence number and echo the sender's initial sequence number
     (c) Sender get the Receiver's SYN+ACK, reply with ACK to echo the receiver's initial sequence number
  3. Sender start sending data to the receiver
  4. Receiver give ACK regularly to tell sender how much data is received successfully
- Data: Provided by something that uses TCP
  - Example: Email sending (SMTP, tcp/25); Email receiving (IMAP, tcp/143); Web (HTTP, tcp/80); File transfer (FTP, tcp/21); BitTorrent (tcp/6881)

  - The application prepares data and use TCP to send to the other computer, where there are another program to handle it

- Example: HTTP
  - Server: listen on port 80, reply with web content
  - Client: Your browser, get the web content and then display it
  - Procedure:
    1. Client:
       `GET /index.html HTTP/1.0`
    2. Server:
       ```
       HTTP/1.0 200 OK
       Date:  Wed, 23 Feb 2005 11:06:26 GMT
       Content-Type:  text/html

       <html>
       ...
       </html>
       ```

4 Attacks

4.1 Sniffering

- To know what people are talking about in the network
- Example: People make ICQ clone by sniffering the network, so we finally understand how ICQ works
- Example: People in Columbia University learn how Skype work by sniffering
- Tool:
  - tcpdump (for all the world except Windows) or windump (for Windows)
  - Ethereal
  - so many other to be listed
- Example:
  ```
  tcpdump -i eth0 host 192.168.0.1 and not port 22
  tcpdump -i any net 192.168.0.0/24 and port 80
  ```
  - For more detail, see man page of tcpdump:
    `man tcpdump`

4.2 Sproofing and Connection Hi-jack

- Tell lies
- Example: Give you wrong IP address of "course.ie.cuhk.edu.hk", so when you read the homepage of "http://course.ie.cuhk.edu.hk/~ieg4321/", I can give you a faked lecture notes
- Example: Give you wrong MAC address of the gateway, so all your packets will be directed to me
- Example: Handle the connection for you so that all data passed by is modified
- Tool:
  - Ettercap (available for both Linux and Windows)
  - tcpdump + Perl
  - or write your own
- Other name: Man-in-the-middle attack
- Example:
  `ettercap -T -q -M ARP /192.168.0.0-255/ //`
  - MITM works for SSL connections as well!

4.3 REDIRECT

- RFC792 specified: ICMP type 5 is redirect
- When a host received ICMP type 5 for a specified packet, it will change the gateway it is used
- Although it is ignored usually nowadays, that's something can help you doing sproofing

4.4 DDoS

- Distributed Denial-of-Service attack
- Try to exhaust your CPU, so that nothing can be done to serve the legitimate customers
- How to do?
  - Find a number of computers (think: virus)
  - Write a small program (think: an IEG3310 homework)
  - The program will loop and send a lot of packet to a victim
  - Many computers do this at the same time = DDoS
- Why it works?
  - Computer need to process network I/O (slow and expensive), so no time to handle running programs
  - If DDoS with SYN flooding, the victim will easily use up all his port or RAM

4.5 SQL injection

- Example PHP code:
  - Login_Page.html:
    ```
    <form action=''login.php''>
    Username:  <input type=text name=''username'' />
    Password:  <input type=text name=''password'' />
    <input type=submit>
    </form>
    ```
  - login.php:
    ```
    ....
    SQL = 'SELECT * FROM ACCOUNTS WHERE USERNAME=''$username'' AND PASSWORD=''$password''';
    ....
    ```
- What if my password is: `ANYTHING'' OR ''1''=''1`
  - Resultant SQL statement:
    `SELECT * FROM ACCOUNTS WHERE USERNAME=''ADRIAN'' AND PASSWORD=''ANYTHING'' OR ''1''=''1''`
  - You can pass through the system without correct password!
- How to prevent?
  - No foolproof way!
  - Write your program carefully!
  - Do input validation
  - *Regular expression* is your friend.