WIRESHARK BASICS AND NETWORK ATTACKS

Hobo | Chukar | jolly? | TcP

Introduction

- Wireshark
  - What does it do?
  - When should it be used?
  - Along the way:
    - TCP/IP
    - Network Attacks
    - How to use Wireshark
    - Misc. programming/security tibdits

What is Wireshark?

- Wireshark – Network Protocol Analyzer
  - Used for network troubleshooting, analysis, development, and *hacking*
  - Allows users to see everything going on across a network*
    - The challenge becomes sorting trivial and relevant data
  - Other tools
    - Tcpdump- predecessor
    - Tshark – cli equivalent
  - Can read live traffic or can analyze pcap files
    - Pcap – *packetcapture* file
      - File created from libpcap library (allows us to read packet info)
  - Where in the attack lifecycle would we use this tool?
    - What information can it give us?
    - How could we use that information?

Setup

- Wireshark
  - Network interface needs to be in promiscuous mode to view all packets on a LAN
    - “ifconfig <interface name> promisc”
  - Need run as root
    - “sudo wireshark &”
      - “&” after a command gives you back the $ shell prompt
  - Start packet capture
    - Choose interface
    - Watch packets fly

A Step Back

- What is actually happening?
  - In promiscuous mode: Interface passes traffic to cpu rather than just the frames
    - Get to see everything within the packets
  - Broadcast traffic
- How do we manage to view a particular person's traffic?
  - MiTM attacks*

Network Attack overview

- TcP slides rock
  - So let's look at them (briefly)
- Types
  - DNS poisoning, XSS, other app-specific vulns
  - Session hijacking, port scanning, SYN floods
  - Route changes, ICMP bombs,
  - ARP poisoning*, dDoS

Man in the Middle

- A word of Warning
- Spoofing
  - Can't I just pretend to be someone else?
  - But wait! ARP!
    - Purpose – to map out and connect machines and their IP addresses
      - MAC/IP addr. Pair
      - What is a MAC Adress? (nothing to do with ole' Stevie Jobs)
        - A unique identifier assigned to a network interface for physical network communication layer
    - Typical conversation
      - "if your IP address is w.x.y.z, send me your MAC address"
        - All computers receive request, and the *correct* computer replies
  - a connection between two users (famous Alice, Bob, Trudy example)

Man in the Middle

- But wait…ARP!
  - Trust model is…well, it's not good
    - No accountability for computer responses. Does not (cannot) authenticate RARP*
  - Easy to spoof
    - Race condition
      - Heh?
    - Flood Arp tables with incorrect info (e.g. "Hey, I'm the router! Forward all outbound packets to me!")
    - Refined spoofing between two parties
    - is it really that easy
      - Well, yes and no

Back to Wireshark

- Once you've captured your packets…
  - What am I supposed to do with 18,000 packets?
  - Filter options
    - Operators: ! == || &&
    - By source/destination
      - "ip.<src,dst>==w.x.y.z"
    - "frame contains <string>"
      - Search a particular string within a packet (very useful, a personal favorite)
    - The wireshark "Analyze" tab
      - Lots of stuff
      - For web traffic: Analyze->follow TCP/UDP stream gives you're the packet content in ASCII (and other formats)

Wireshark and You

- Stuff TODO:
  - Chain together filter options
    - "(ip.src==10.105.225.100) && !(ip.dst==70.136.12.158)"
      - Looks for all traffic from 10.105.225.100 unless the destination IP is 70.136.12.158
  - Read Packet Content
    - What do the packets look like (follow the TCP stream)
    - What ports are typically used when http traffic is unencrypted? Encrypted?
    - What are some protocols you've never seen before? What do they do?
  - MiTM
    - Remember the warning
    - Set up
      - ????
    - Then you can filter all data to/from source
    - Profit

Enough Talk, you Hobo!

- Demonstration