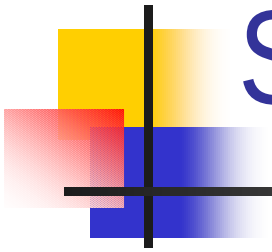
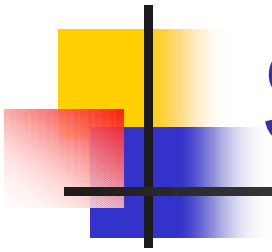
Scanner, Sniffer und Scanlogger



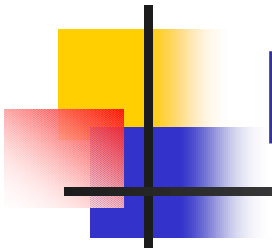
Sniffer

- Sniffer Grundlagen
- Promiscuous Mode
- Ethernet
- Gefahren und Nutzen von Sniffer
- Praxis mit Buttsniff und Sniffit



Sniffer Grundlagen

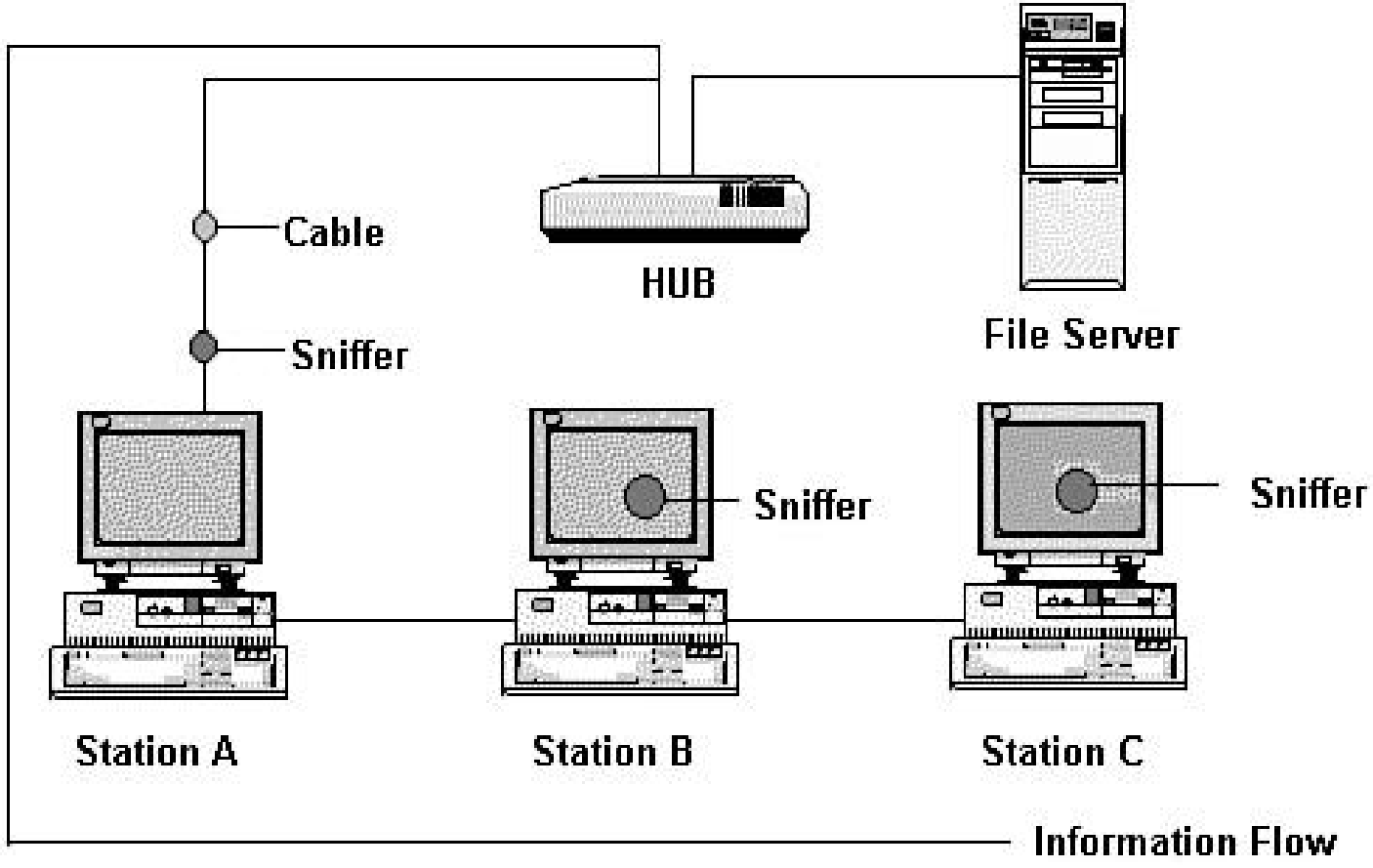
Ein Sniffer ist ein „Device“, ob Software oder Hardware, der den Datenstrom eines Netzwerks aufzeichnen kann. In diesem Netz kann jedes mögliches Protokoll laufen: Ethernet, TCP/IP, IPX oder andere (oder irgendeine Kombination von diesen). Der Zweck des Sniffers ist es, die Netzschnittstelle - in diesem Fall, der Ethernet-Adapter -- in den Promiscuous Mode zu schalten, der zum Erfassen des Netzverkehrs benötigt wird.



Promiscuous Mode

- Im Normal-Modus nimmt ein Netzinterface nur die an sich selbst gerichteten Daten auf.
- Im Promiscuous Modus, kann das Netzinterface, sämtlichen Datenverkehr aufzeichnen, der duch das Ethernet geht. Der Sniffer kann dann auch Pakete aufzeichnen, die nicht für sein Interface gedacht sind.

Ethernet (general broadcast)



Gefahren und Nutzen von Sniffer

- Hinweis: Ein Sniffer kann immer nur den Datenverkehr von einem Netzwerkblock (Segment) aufzeichnen, da die Broadcast Signale nur innerhalb eines Segments übertragen werden (nicht gerouted).
- Da das Internet aus einem wissenschaftliche Netz entwickelt wurde, fanden kryptographische Sicherheitskonzepte kaum Berücksichtigung (alle Passwörter werden im Klartext übertragen).
- Sniffer können eine bedeutende Drohung sein, da sie in der Lage sind **Kennwörter** oder **vertrauliche Informationen** zu erfassen.
- Auf der anderen Seite kann man mit Sniffern Netzwerkfehler lokalisieren und einen generellen Überblik über den Netzwerktraffic bekommen.

Praxis mit Buttsniff

- Der Buttsniff Server wird via
- Buttsniff.exe –i <Netzwerkinterface> <port> gestartet.
- Z.B. buttsniff –i 0 77
- Jetzt kann man von einem beliebigen Client mit telnet <Servername> <Port>
- zum Buttsniff-Server connecten.
- Z.B. telnet b38 77
- Danach kann man im Interaktive Modus sniffen.

Praxis mit Sniffit (1)

- **Filter/Text-Mode**
- Optionen, die SniffIt mit übergeben werden muessen (mind. eine davon):
- -t <IP nr/name> sniffe Pakete, die an <IP> gehen
- -s <IP nr/name> sniffe Pakete, die von <IP> kommen
- Mit "@" werden Wildcards gesetzt. Beispiele hierfür:
- -t 199.145.@ oder
- -t @ (alle)
- -F <device> Netzwerkinterface angeben
- -d Dump mode -> zeigt alles in Byte an
- -a wie -d, zeigt die Ausgabe aber als ASCII an
- -x erweiterte Ausgabe (z.B. SEQ,ACK,etc)
- -P protocol Standard: TCP Weitere: IP,ICMP,UDP
- -p <port> loggt pestimmte Ports. 0 ist default. 0 = alle Ports
- -l <length> Default = 0, 0 = alles.

Praxis mit Sniffit (2)

- **Interactive-Mode:**

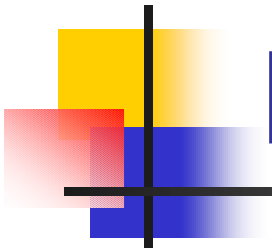
- -> Interactive-Mode (Parameter: -i)
- -> Eingabe von Herkunft- und Ziel-IP
- -> Eingabe von Herkunft- und Ziel-Port
- Optionen für den Interactive Mode:
- -D <device> Alles geniffte wir an dieses Device gesendet
- -L <loglevel> gibt das Loglevel an:
- 1 : Raw level
- 10,12 : Normal level

Tools für Sniffit

- Tools: Touch of Death
- Dieses tool wird als Modul mit in SniffIt eingebunden.
- Eine aktive Verbindung, die gerade gesnifft wird auswählen
- und TOD auswählen -> Verbindungsende-Signale werden gesendet
- Weitere Sniffer für UNIX/Linux:
- tcpdump (Konsole)
- Etherreal (X11)
- Linsniff
- Linux_sniffer

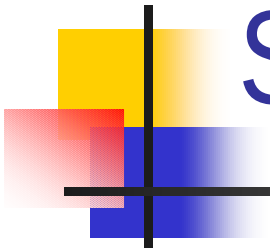
Praxis mit linux_sniffer

- Linux_sniffer macht einen TCP-IP dump
- Mit der Eingabe linux_sniffer > dump wird der dump in die Datei dump geschrieben.
- Mit linux_sniffer | grep “PASS\|USER“ kann man z.B. nach PASS und USER suchen.



Praxis mit Linsniff

- Linsniff starten.
- Unter /tmp/.sniff.log wird eine Log Datei angelegt, die die Passwöter von den Standardports enthält.



Scanner

- Scanner Grundlagen
- Port-Scanner: Hackerangriff oder Systemwerkzeug ?
- Praxis mit fastscan und nmap

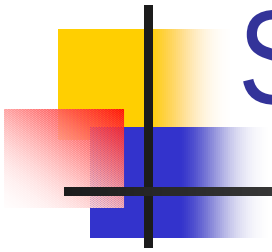
Paxis - NMAP (1)

- **Scan types:**

- -sT tcp connect() port scan
- -sS tcp SYN stealth port scan (must be root)
- -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (only works against UNIX).
- -sP ping "scan". Find which hosts on specified network(s) are up but don't port scan them
- -sU UDP port scan, must be r00t
- -sR RPC scan (use in addition to other TCP and/or UDP scan type(s)
- -b <ftp_relay_host> ftp "bounce attack" port scan

Praxis - NMAP (2)

- **Options (none are required, most can be combined):**
- -f use tiny fragmented packets for SYN, FIN, Xmas, or NULL scan.
- -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
- -O Use TCP/IP fingerprinting to guess what OS the remote host is running
- -p <range> ports: ex: '-p 23' will only try port 23 of the host(s)
- '-p 20-30,63000-' scans 20-30 and 63000-65535. default: 1-102
- -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> Timig policy
- -F fast scan. Only scans ports in /etc/services, a la strobe(1).
- -I Get identd (rfc 1413) info on listening TCP processes.
- -n Don't DNS resolve anything unless we have to (makes ping scans faster)
- -R Try to resolve all hosts, even down ones (can take a lot of time)
- -o <logfile> Output scan logs to <logfile> in human readable.
- -m <logfile> Output scan logs to <logfile> in machine parseable format.
- -i <inputfile> Grab IP numbers or hostnames from file. Use '-' for stdin
- -g <portnumber> Sets the source port used for scans.20 and 53 are good cho
- -S <your_IP> If you want to specify the source address of SYN or FYN scan.
- -v Verbose. Its use is recommended. Use twice for greater effect.
- -e <devicename>. Send packets on interface <devicename> (eth0,ppp0,etc.).



Scanlogger

- Scanlogger Grundlagen
- Was kann man mit den Informationen anfangen ?
- Utilities
- Beispiel SockScan und FakeBO
- Scanlog-Deamon Grundlagen