Università degli studi di Salerno
Facoltà di SS.MM.FF.NN

NETWORK SNIFFERS

Autori: Carlo Olivieri, Giuseppe Monaco, Riccardo Nardelli.

Cosa è uno Sniffer?

- Uno sniffer è un "*programma*" che richiede di ricevere pacchetti non destinati a se

Usato per :

- Monitorare il funzionamento e le performance della rete
- Visualizzare dati altrui

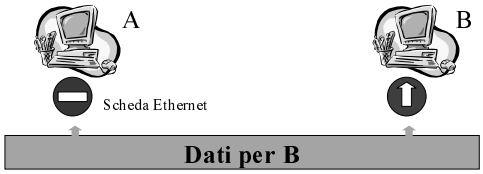
Cosa è uno Sniffer?

- Nella comunicazione tra Alice e Bob si inserisce un intruso che "ascolta" la loro conversazione

Alice ti amo!

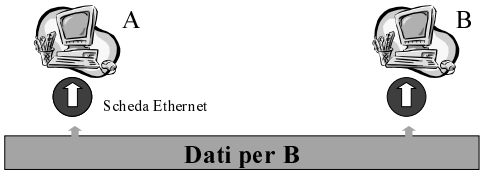
Sniffer

- Normalmente solo l'interfaccia del destinatario passa i dati allo strato superiore



Sniffer

- Richiesta scheda in modalità *promiscua*

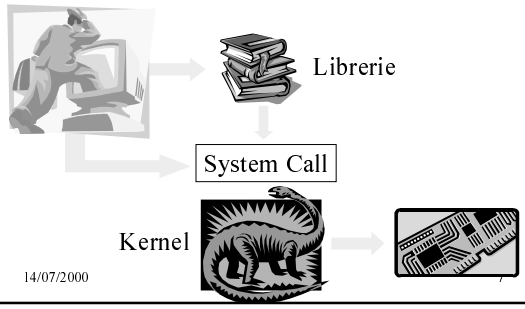


Sniffer e Shared Media

- I canali condivisi agevolano gli sniffer
- In una Ethernet ogni macchina può essere uno sniffer

Sniffer e S.O.

Settare la modalità promiscua

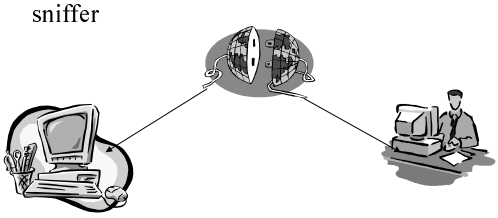


Sniffer e TCP/IP

- TCP/IP resiste ad attacchi nucleari
- Si basa sulla cooperazione
- Non offre meccanismi di protezione dei dati
- La cifratura è lasciata alle applicazioni

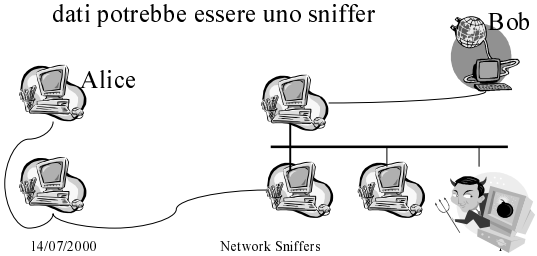
Sniffer e TCP/IP

- Ogni router in Internet potrebbe essere uno sniffer



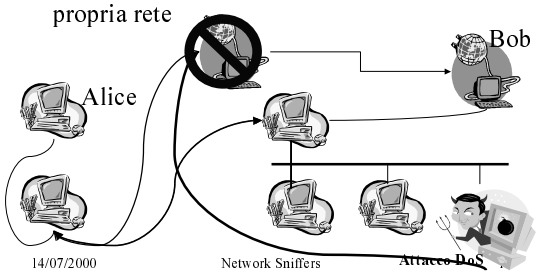
Sniffer e TCP/IP

- Ogni macchina in una rete su cui passano i dati potrebbe essere uno sniffer



Sniffer e TCP/IP

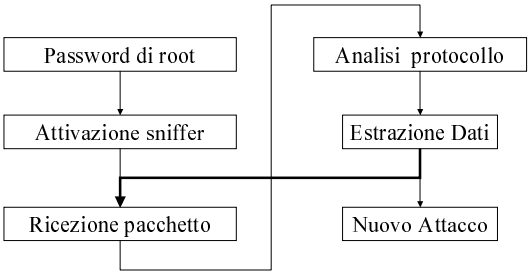
- Un hacker per sniffare può tentare di costringere la vittima a usare un router sulla propria rete



Sniffer e TCP/IP

- Non esiste un punto da cui è possibile spiare tutto

Come opera un sniffer

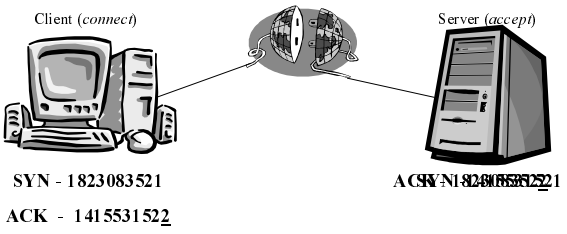


Tipi di attacchi

- Scoprire password e codici

  Sfruttare le nuove password per violare sistemi e attaccare da diversi punti

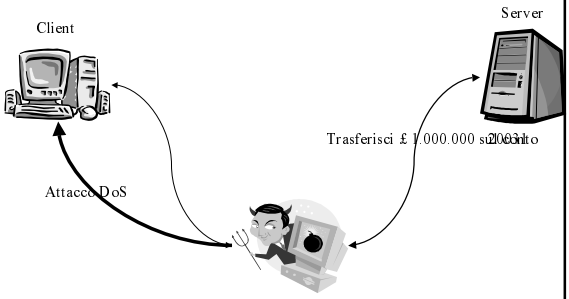
Three-Way Handshake



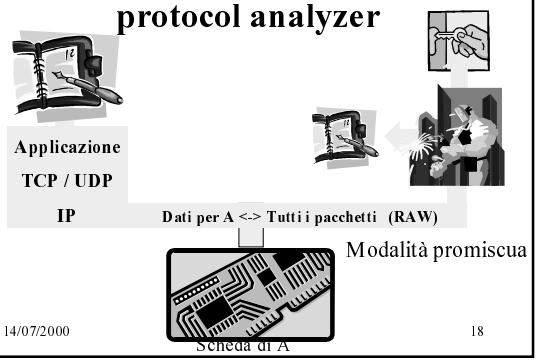
Connessione TCP effettuata

Controllo restituito alle applicazioni

Attacco a previsione del numero di sequenza



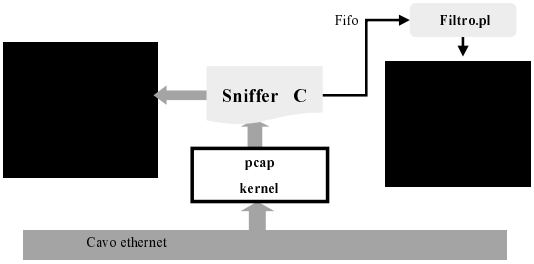
Uno Sniffer è un **protocol analyzer**



Un "*semplice*" sniffer

- La libreria pcap (Packet Capture) è stata sviluppata per rendere agevole la cattura dei pacchetti (Sviluppata a Berkeley)
- E' un interfaccia indipendente dal dispositivo
- Funzioni usate nella realizzazione dello sniffer :
  - pcap_open_live
    si aggancia al dispositivo per la cattura dei pacchetti
  - pcap_next
    Puntatore al prossimo pacchetto
  - pcap_datalink
    Intero indicante il tipo di interfaccia

Funzionamento del nostro sniffer



Esempio di funzionamento in una sessione telnet

Utente

Sniffer

Altri dettagli non riportati

Come funziona (Produttore in C)?

- La funzione pcap_next restituisce un puntatore al primo byte del pacchetto

Header Ethernet	Header IP	Header TCP	Dati Applicazione

- Nel caso di frame Ethernet la lunghezza del header è di 14 byte. Gli ultimi 2 sono 0x0800 se si trasporta un datagramma IP

Header IP

4 - ver	4 - length	TOS	16 bit tot length
16 bit ID		3bit flag	13 fragment offset
8 bit TTL	8 bit protocol	16 bit crc	
32 bit – IP Origine			
32 bit – IP Destinazione			

Header TCP

16 bit – porta sorgente		16 bit - porta destinazione
32 – bit sequence number		
32 bit ack number		
4 bit h –len	SYN,ACK,PSH,RST …	16 windows size
TCP – checksum		16 bit urg ptr

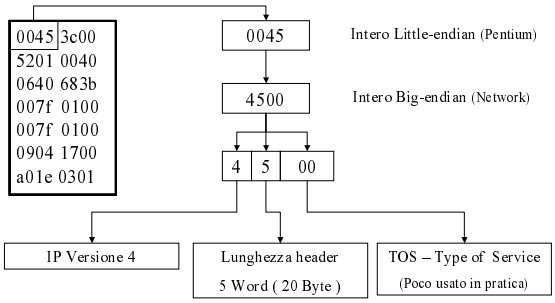
Analisi del datagramma IP

- Il campo protocollo specifica se è contenuto un datagramma UDP (17) o TCP (6)
- Se non contiene TCP o UDP viene scartato
- Le informazioni della connessione vengono stampate a video
- Eventuali dati inviati alla fifo

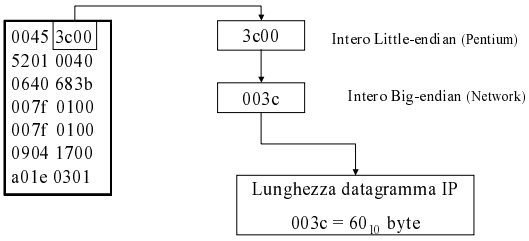
Come funziona (Consumatore in Perl)?

- Utilizza un Hash ***connessioni*** per "mappare" la corrispondenza connessione-dati relativi
- Utilizzo della system call select per I/O non bloccante
- Visualizza un elenco delle connessioni attive
- Scelta della connessione da spiare
- Visualizzazione dati in tempo reale

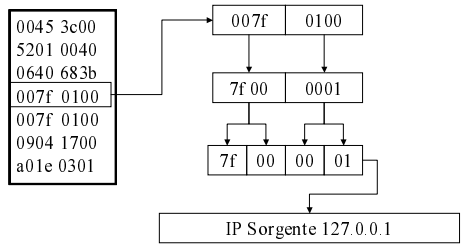
Analizzare il protocollo ?



Analizzare il protocollo ?



Analizzare il protocollo ?



Come interpretare i dati in ambiente Linux

- *dati = pcap_next;*
- *p_ip = (struct p_iphdr *) &dati[offset];*

I campi di p_ip vengono avvalorati in maniera corretta perché la definizione della struttura tiene già conto dell' architettura della macchina (big-endian/little-endian)

Stesso discorso per p_tcp e p_udp

Lettura dei dati dalla fifo

- Nella fifo è specificato :
  - Nome : nome univoco connessione
  - Numero byte di dati (-1 se FIN bit è 1)
  - Dati : Byte di dati grezzi
- Usando l'hash è immediato passare da una connessione ad un'altra

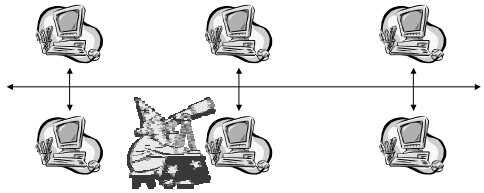
Osservazione

- Invece della fifo si potrebbero usare i socket (Visualizzazione dei dati in remoto)
- Un' altra versione scrive i dati direttamente nei file (uno per ciascun lato della connessione).

LA CATTURA DEI PACCHETTI

Packet Filter

- E' un "Agent" nello spazio kernel
- Il suo compito è quello di catturare i pacchetti



Un Po' Di Storia

- Nel lontano 1980 due università, CMU e Stanford, sviluppano il CSPF
- Il CSPF viene implementato su un PDP-11
- Il lavoro svolto in queste università è la pietra miliare verso un brillante futuro

I Derivati Del CSPF

- NIT (Network Interface Tap) della Sun
- Ultrix Packet Filter sui DEC
- Snoop sulle SGI (Silicon Graphic)

Ma nel 1992 presso i laboratori di Berkeley…

Viene sviluppato…

BSD Packet Filter (BPF)

- Sviluppato nel 1992 a Berkeley
- Offre dei miglioramenti rispetto a CSPF
- Un filtro basato sui registri macchina
- Un modello a buffer non condivisi

BSD Packet Filter

BPF ha in sé due componenti:

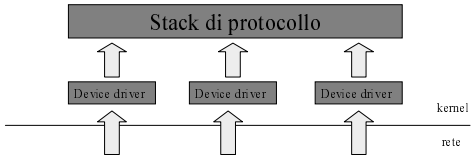
- Il Network Tap
- Il Filtro (Packet Filtering)

Il Network Tap

- Colleziona copie dei pacchetti provenienti dal device driver di rete
- In seguito è il filtro a decidere se la copia del pacchetto verrà accettata o rifiutata

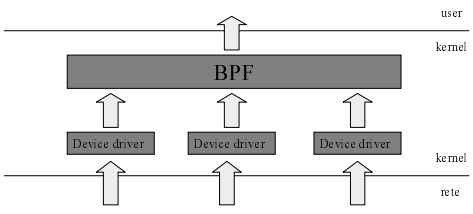
Il Network Tap

Quando un pacchetto arriva all'interfaccia di rete, normalmente il device driver lo passa allo stack di protocollo



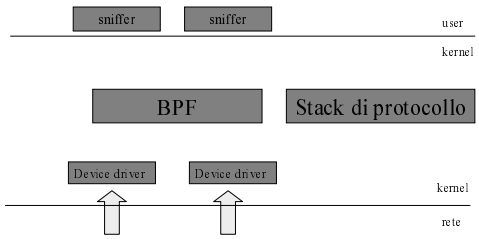
Il Network Tap

Ciò non avviene quando il BPF è attivo



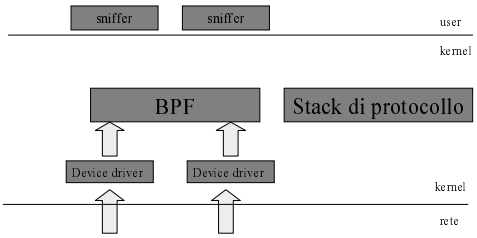
Il Network Tap

Infatti non appena arriva un pacchetto…



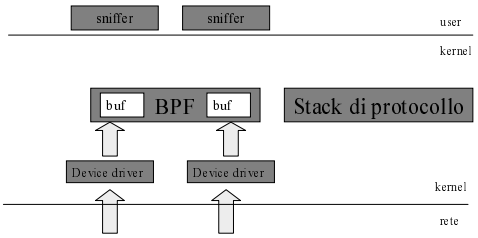
Il Network Tap

Il pacchetto viene prima consegnato al BPF



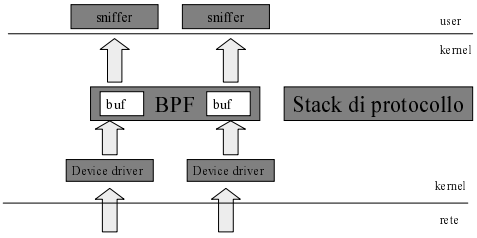
Il Network Tap

Se il filtro accetta il pacchetto, il BPF lo copia in un buffer



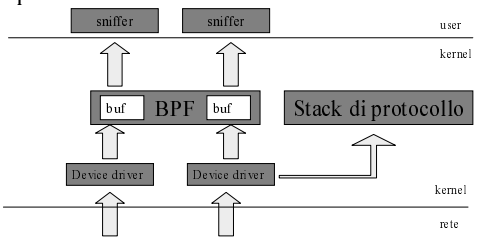
Il Network Tap

Il buffer viene passato alle applicazioni



Il Network Tap

Il pacchetto viene passato allo stack di protocollo



Packet Filtering

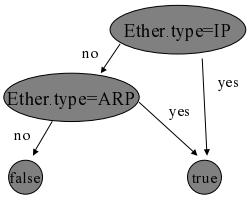
- Svolge un grosso lavoro di esame dei pacchetti
- Filtra i pacchetti “in loco” (DMA)
- Un cattivo filtro può incidere sulle prestazioni totali del nostro “agent”

Packet Filtering

- E’ una funzione booleana sui pacchetti
- Viene implementato usando un ***grafo diretto aciclico di controllo flusso***
- Ricorda l’ultima analisi (parsing) fatta perché il grafo può essere ogni volta riorganizzato

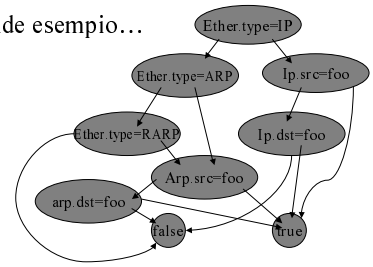
Packet Filtering

- Un piccolo esempio…



Packet Filtering

- Un grande esempio…



Packet Filtering

- Il BPF deve essere indipendente dal protocollo
- I pacchetti sono solo visti come array di byte
- Questa scelta deriva direttamente dal CSPF

Packet Filtering

- Il BPF deve essere generico
- Il set di istruzioni utilizzato deve essere abbastanza ricco da prevedere tutti i possibili usi
- Sufficiente potenza di calcolo aritmetico (ALU)
- Metodi di indirizzamento convenzionali

Packet Filtering

- I riferimenti ai dati nei pacchetti devono essere minimizzati
- Una situazione tipica è quella di confrontare un dato pacchetto con più insiemi di valori
- Il campo relativo al pacchetto viene messo in cache all'interno di un registro e quindi confrontato con l'insieme dei valori

Packet Filtering

- I registri della macchina astratta dovrebbero risiedere all'interno dei registri di una macchina reale
- I registri macchina sono sicuramente più veloci
- Ma il loro numero è limitato

Siti importanti

- http://www.lbl.gov
- ftp://ftp.ee.lbl.gov

Alla ricerca dello SNIFFER!

Scovarlo all'interno di un sistema?

Si dice che sia impossibile……..

……ma non è del tutto vero perché…..

……..ci sono delle soluzioni!!

Come mai uno Sniffer è qui?

Dopo essersi guadagnato il primo accesso, l'intruso…..

…installa uno SNIFFER…

….e poi lo SNIFFER…….

…come per magia cattura password e legge il traffico di rete!!

Ma come facciamo a scoprire se è attivo??

I METODI.

- Metodo PING
- Metodo ARP
- Metodo DNS
- Metodo Source-Route
- Metodo DECOY
- Metodo Host

Metodo PING.

- Sfruttamento della sicurezza del TCP/IP
- Sfruttamento di un indirizzo MAC differente per la spedizione dei pacchetti
- Trasmissione di un “ICMP echo request” (Ping) alla macchina sospetta
- Attesa di una risposta e analisi di essa

Analisi della risposta

NEGATIVA

NESSUNO SNIFFER

POSITIVA

SNIFFER ATTIVO

Metodo ARP

- Stessa idea del metodo Ping
- Sfruttamento della tecnica ARP
- Trasmissione di ARP ad un indirizzo non broadcast(alla macchina sospetta)
- Risposta positiva nel solo caso in cui il pacchetto sia stato sniffato
- La macchina che risponde è in una situazione promiscua.(Probabile sospetta!)

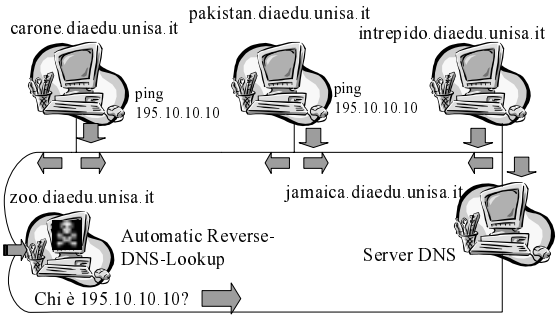
Metodo DNS

•Scovamento della modalità promiscua attraverso il traffico DNS che essa genera

•Ping circolare su un indirizzo inesistente

•Monitoraggio del “Reverse-DNS-Lookup” in arrivo sul server DNS

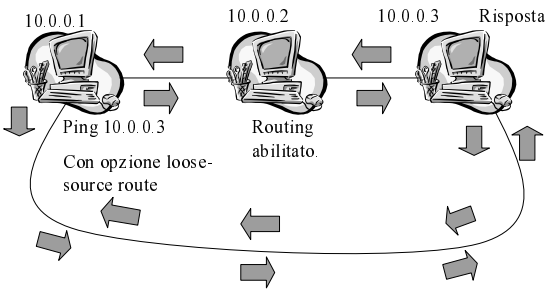
Esempio



Metodo source-route

- Configurazione delle informazioni source-route all’interno dell header dell’IP
- Creazione di un pacchetto ping con una rotta loose-source per forzarlo a passare attraverso un macchina prima di arrivare a destinazione
- Attesa di una risposta e analisi di essa

Funzionamento: 1 caso



Analisi della risposta

Confronto fra il campo TTL del pacchetto che era partito e il pacchetto arrivato come risposta

I campi TTL non corrispondono → **L'host 10.0.0.3 non ha sniffato il pacchetto!**

I campi TTL corrispondono → **L'host 10.0.0.3 ha uno SNIFFER attivo!!**

Metodo Decoy

- Creazione di 2 macchine: un client ed un server (meglio se virtuale)
- Lanciare dal client una richiesta di login sul server tramite protocolli "plaintext"(telnet, POP, IMAP ecc..) facendosi quindi sniffare la password
- Attendere che l'hacker provi a loggarsi con le informazioni appena sniffate, ed il gioco è fatto!

Metodo host (Su un sistema UNIX).

- E' basato sul fatto di scoprire tramite diversi tentativi la presenza di uno sniffer sulla macchina
- Si cerca di capire se l'interfaccia di rete si trova in modalità promiscua

Cosa fanno gli "intrusers" quando riescono a loggarsi…

•Installano un "Rootkit" ovvero un pacchetto comprendente un set di "Cracking tools"

• Attivano uno Sniffer

•Effettuano il logout dal sistema

Contenuto del "Rootkit".

•Versioni modificate di "ls" e "du"

•Versioni modificate di "ps" e "netstat"

•Versioni modificate di "ifconfig" e "login"

•Un programma denominato "fix" che maschera time stamps, permessi, grandezze e checksum

Come scovare…

•Possiamo rilevare i programmi modificati ls, du, ps, netstat ricercandoli attraverso le stringhe "/dev/ptypp", "/dev/ptyqq", "/dev/ptyrr", che non dovrebbero essere presenti nelle versioni originali

•Confrontare l'originale con il falso sospetto usando il tool "diff" di UNIX

…oppure….

•Possiamo sperare che non sia stato installato nessun rootkit e lanciare il comando ifconfig –a. Potremmo scoprire che l'interfaccia di rete è in modalità promiscua!

•Usare un programma denominato "CPM" (check promiscuos mode)

I più famosi…

I più famosi sniffer per UNIX sono 2:

"Es" e "Sunsniffer" ma……

….se il comando ps è stato modificato, non possiamo vederli!!

Qualche speranza può darcela il comando "top"

Come comportarsi se si scova uno sniffer attivo..

•Niente panico! Vietato lo shutdown del sistema

•Fare lo shutdown dell'interfaccia Ethernet con ifconfig –a down

•Eseguire un dump completo a livello 0

•Cercare di vedere quali sono gli agomenti di linea di comando dello sniffer: essi sono i files di output dello sniffer