Sniffer Detector - a Prototype

GSAL

IBM Zurich Research Laboratory

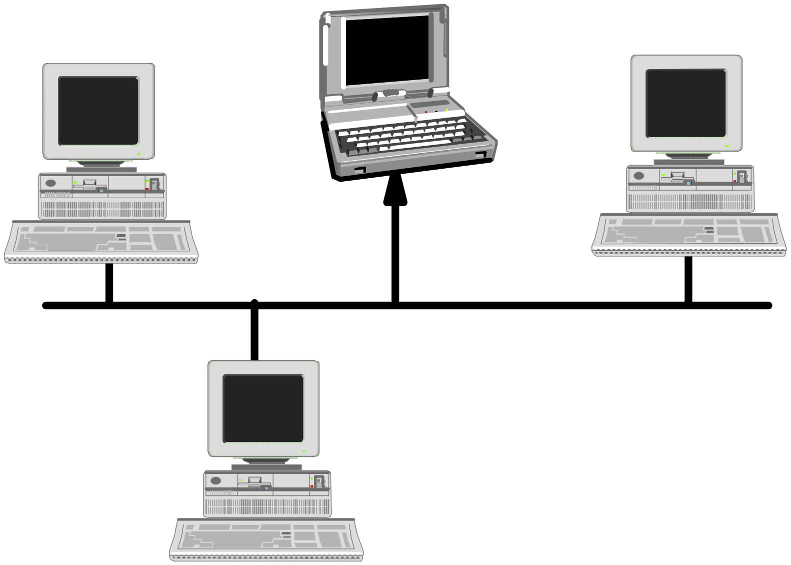
http://w3.zurich.ibm.com/Projects/gsal

gsal@zurich.ibm.com

Threat

A sniffer

- listens to every packet transmitted on the network,
- is almost invisible from a network point of view, and
- can retrieve or reconstruct sensitive information passing on the wire.



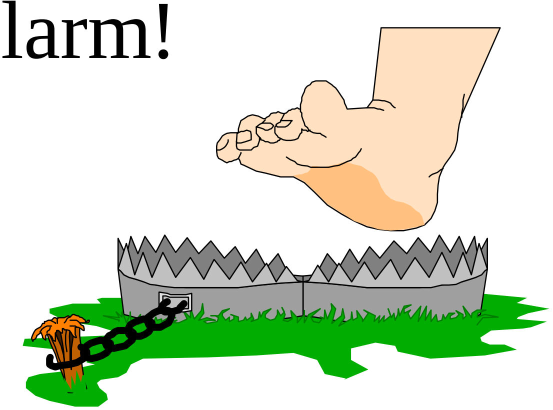
An attacker can use the captured information to break into a system!

Sniffer Detector - Concept

It is difficult to detect a sniffer directly.

→ Wait for the owner of the sniffer to exploit the information he has collected.

- Generate sessions (information baits).
- Wait for the intruder to re-use the transmitted information.
- Detect it and trigger an alarm!

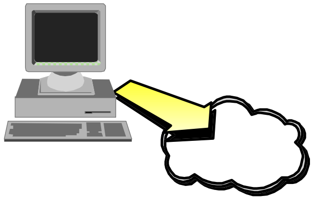


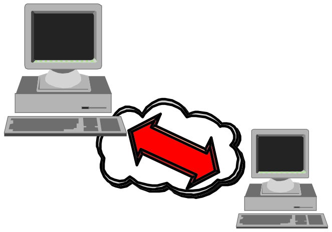
Options - Packet Generation

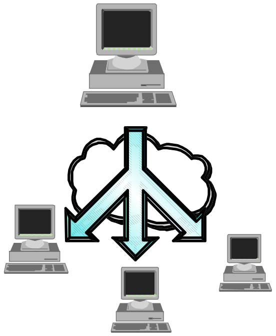
There are essentially two ways to generate the sessions:

	Packet forging	Real connections
Complexity	Need to forge all packets, from lowest layer and up	Simple scripts
TCP	Pseudo-random sequence numbers	Natural numbers, diversity of real stack behavior
Simulation	Complete protocol	User interactions only
Resistance to attack	Attackproof: Do not bother of incoming packets	May be the target of stack attacks (SYN flood, Hijacking, ...)

Options - Architecture





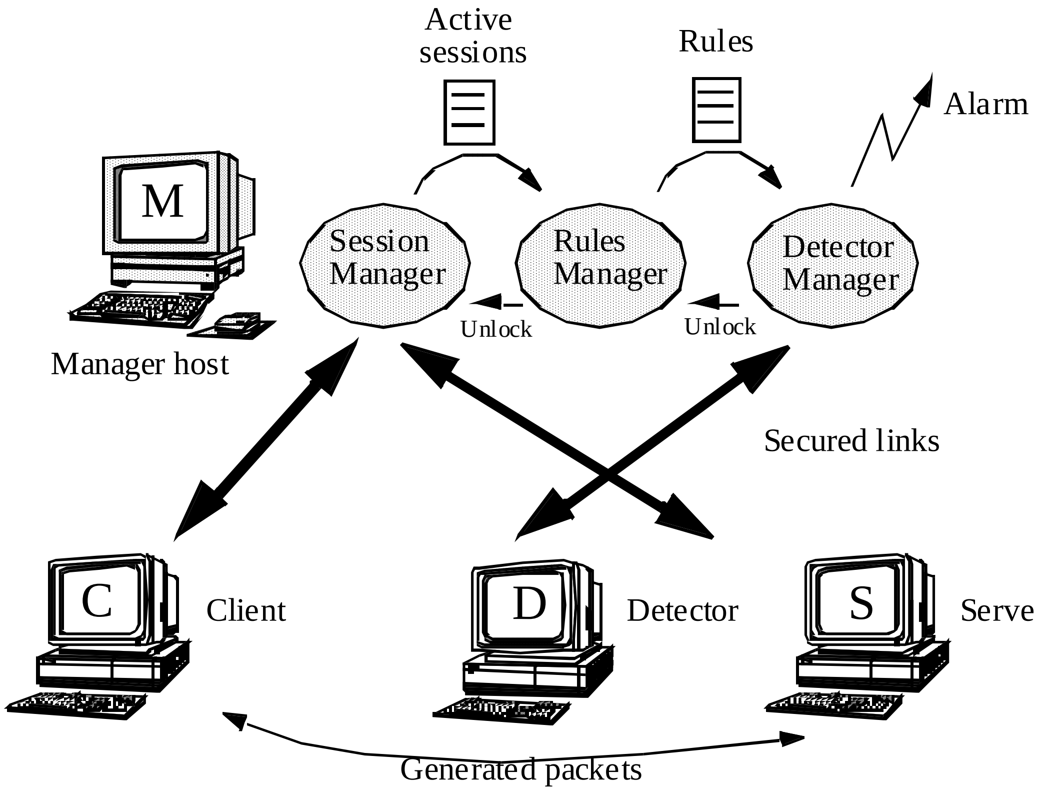


	+	-
Single host	Simple IPC. Low cost.	No packet routing. Difficult to monitor a large network.
Two hosts, one master and one slave, local detection	Packet routing through the network.	Complex synchronisation. No correlation.
Multi hosts, one manager, many clients/servers/probes	Full packet routing through the network. Sniffer localization. Add redundancy. Scalable. Messages correlation.	Interhost secure communication channel. Expensive. Complex.

Solution Choice

Multi-host :

- One Manager
- Many Clients/Servers (both physical and logical)
- One or more Probe(s)



Real connections are established ⟶ more convincing

Temporary configuration ⟶ protection against intrusion

Rules, Events & Alarms

1) For each session generated

→ Rules are transmitted to the Probe:

❑ Look_host: *IP_address*

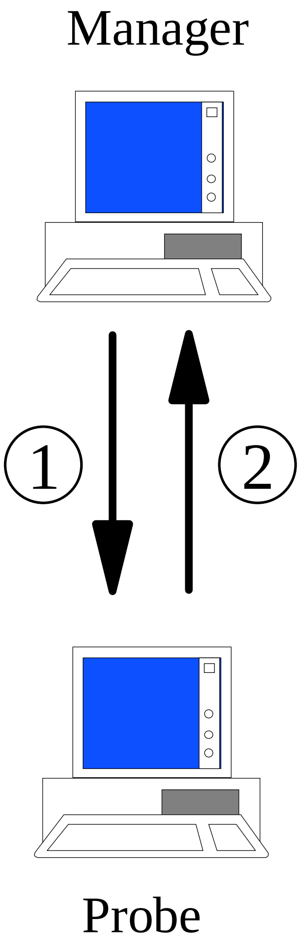
❑ Look_protocol: *telnet, login, passwd*

2) The probe sees a "hot" packet

→ Events are reported to the manager:

Hot: *IP_SRC[src_port], IP_DST[dst_port]*

☑ Look_host: *IP_DST*

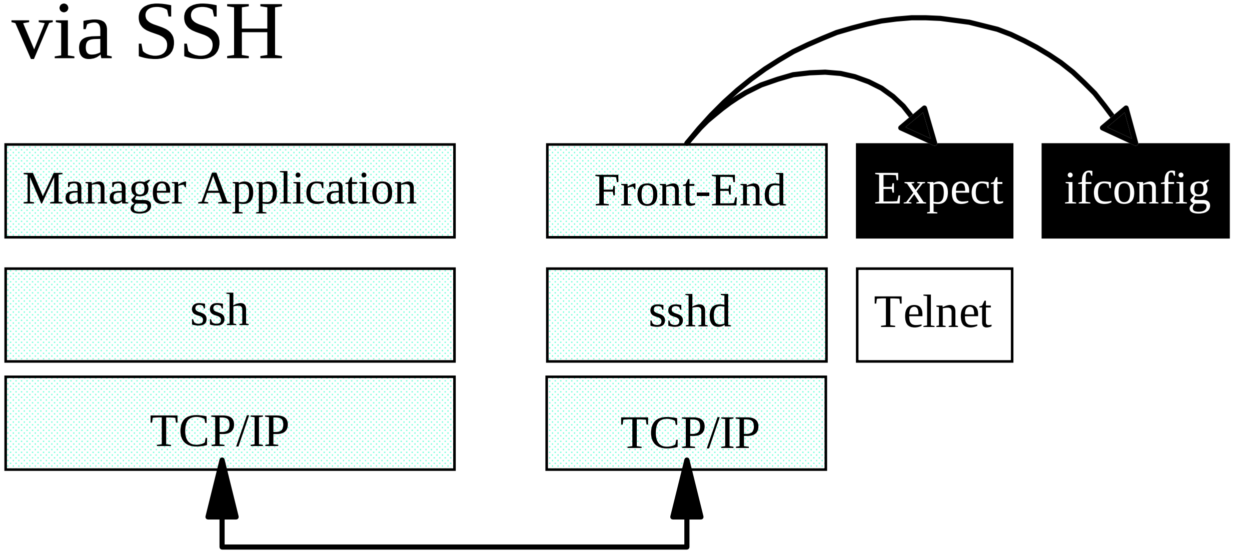


The manager differentiates between *session generated events* and *real attacks*:

→ ALARM to the system administrator

Implementation Aspects (1)

- Manager sends requests to front-ends
- Secure connections via SSH

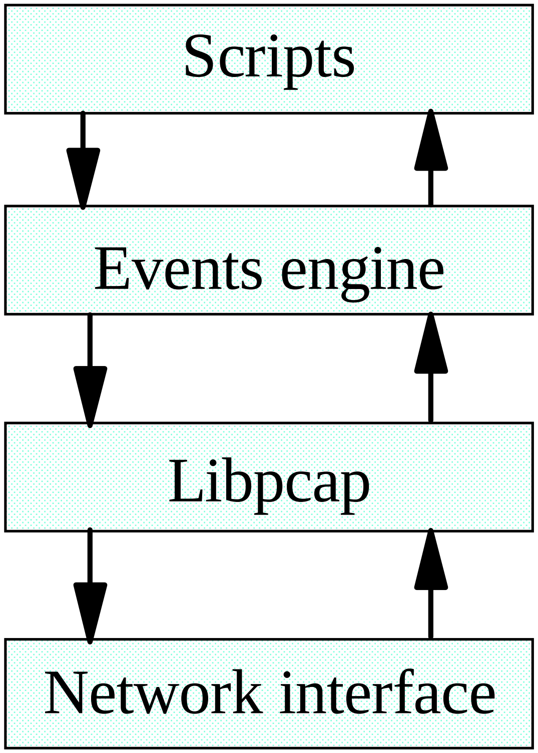


Prototype:

- Generates Telnet and FTP sessions
- Logins and Passwords as *baits*

Implementation Aspects (2)

- "Commercial" Detector:
  Bro, Lawrence Berkeley Lab (Network Research Group)
- Very good conception, easily upgradable, free

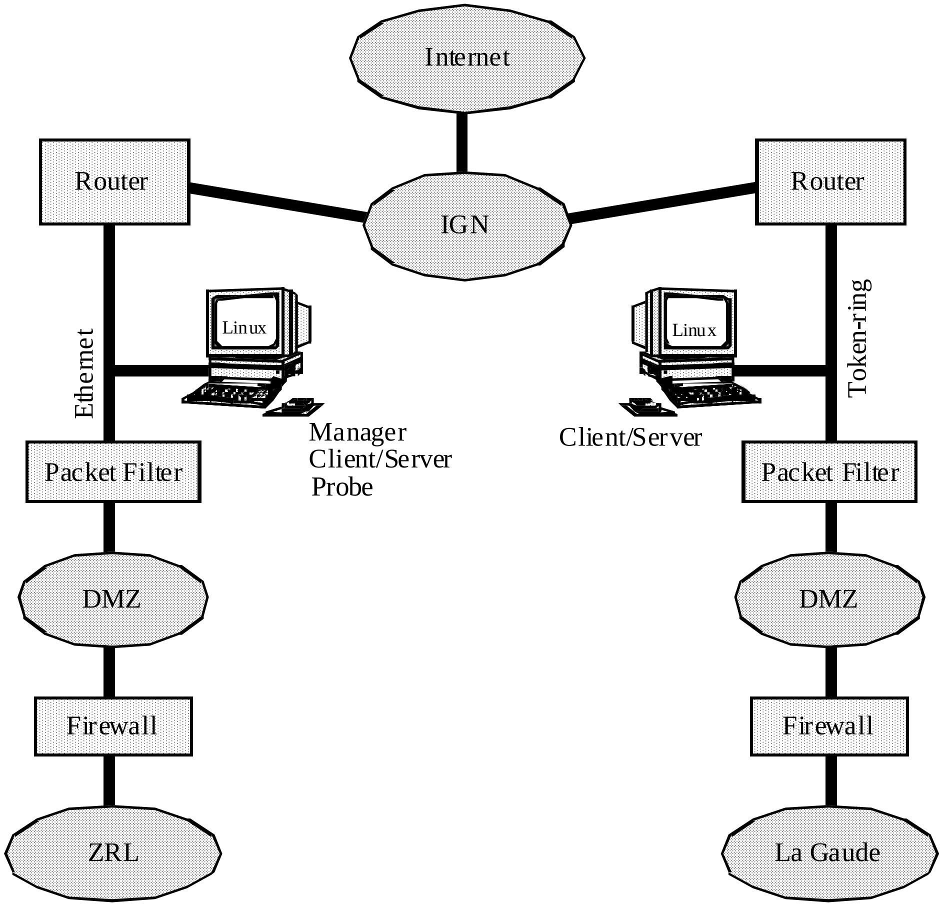


Detects:

- Any TCP or UDP packet with a *hot* IP address.
- Any Telnet or FTP session showing a *hot* login and password **pair.**

Real Environment Test

- Working prototype tested between Zurich (Switzerland) and La Gaude (France).
- Telnet and FTP sessions only.
- So far no sniffer detected!



Conclusions

- We have validated the sniffer detection concept.
- The Sniffer Detector is a new component for the intrusion detection toolbox.

Interested in a Prototype Installation?

What you need:

- Linux-host(s) with IP-aliasing enabled (Bro, Expect, ssh)
- A couple of free IP addresses
- Ethernet segment for the Probe

We welcome remote sites to
further test our Sniffer Detector!

Please send e-mail to gsal@zurich.ibm.com