

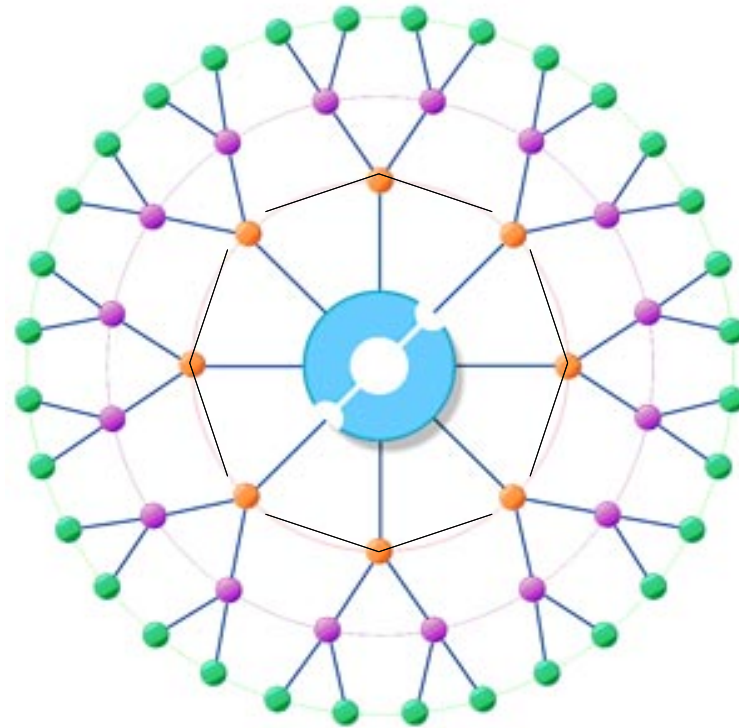
SniffJoke 0.4 alpha 2



SniffJoke 0.4 release, Tour index:

-
-
-
- What's a sniffer
-
- Why we need to protect from them
-
- How does a sniffer work
-
- What's the classical defense ?
-
- Why SniffJoke is different from the other solutions
-
- How does SniffJoke work ?
-

shiny network paint



The users are the **green dots**.

every other dot is a network element.

(when you make traceroute/tracert, you see them)

every network element could dump your traffic (forever ?)

how many sniffers do exist ?

tcpdump -ni eth0 -s 0 -w sniffed.pcap &

ettercap

password collector, man in the middle,

xplico

10 giga pcap dissection

wireshark

every network protocol supported with layer 5 dissector

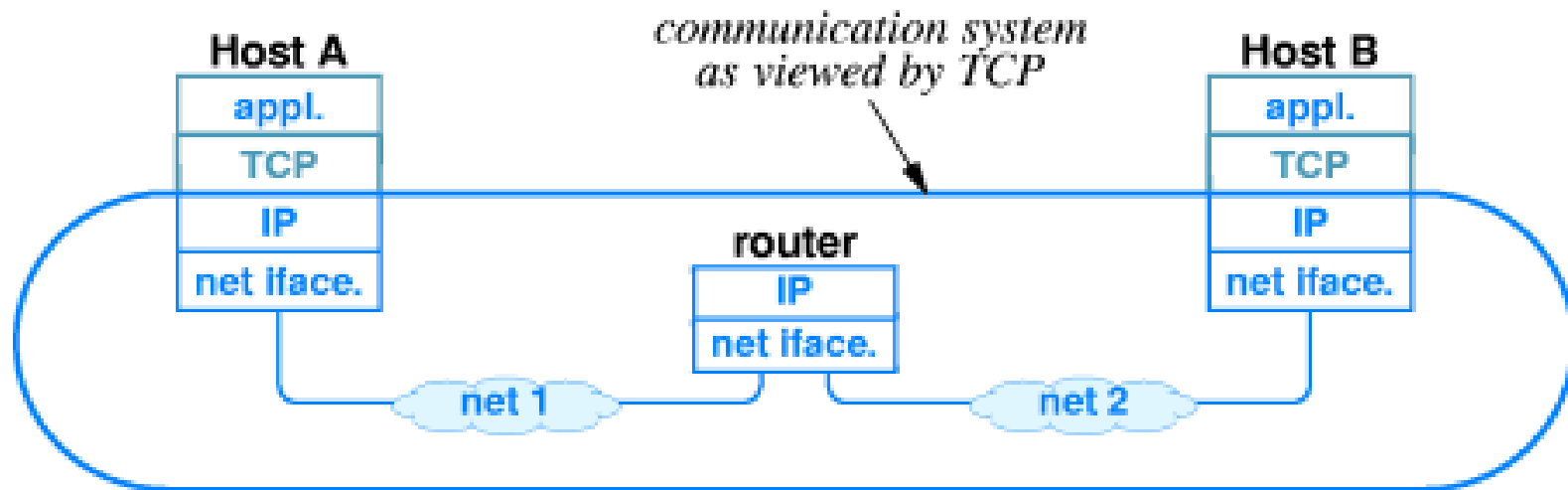
dsniff

command line passive-active-boring-disturbing tool

every kind of sniffer available:

http://www.corriere.it/scienze_e_tecnologie/10_gennaio_20/spie-grazie-a-internet-lavinia-hanay-raj_94283bfa-05db-11df-a1d7-00144f02aabe.shtml

How does a sniffer work ?



Your software receives application layer data.
usually, the network software work on this layer.
(HTTP GET|POST, HELO MAIL FROM ...)

Your socket reads network layer datagram
few software work on this layer (firewall, ping, traceroute)
(struct iphdr *ip = &buffer[1500], ...);

Your interface handle physical layer buzz.
sniffers live here!

Collateral damage

(if) your life transits & grows on the network...

if your transactions are *vulnerable to interception*,
these should be actively manipulated

your e-life follows the same rule.

Common misconception: why an attacker
should care about me ?

targeted attack

you have a smart (evil|good) attacker interested in you

untargeted attack

one of your internet profile is interesting for a smart (evil|good)
attacker

Why sniffing is the easiest way to obtain data ?

a seized hard disk require a law operation or a stealing operation.

a web mail require cooperation between the attacker and the provider.
since few months of existence every large provider receives a lot of official and illegal requests.

Physical interception tool like keyloggers, hidden mics and so on..
require a study of the target and a precise organization.
instead the sniffing is standard :)

Internet interception should obtain data for a long time working in **passive** mode, **unamanged** and **silently**.

How does an interception appears ?

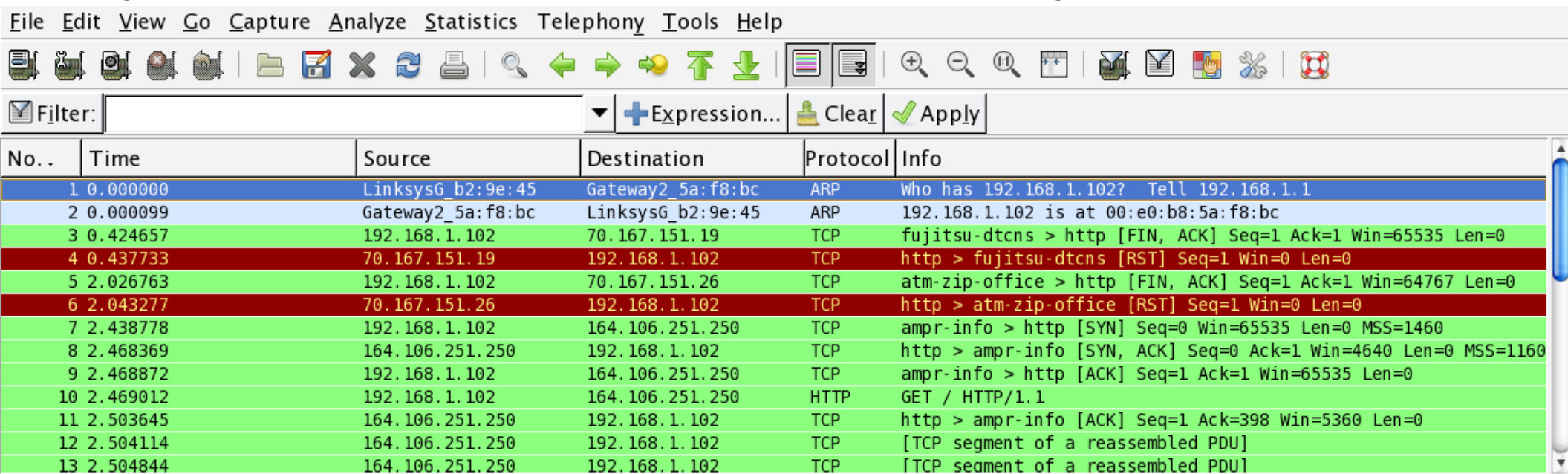
18:51:40.030995 IP (tos 0x0, ttl 64, id 9228, offset 0, flags [none], proto UDP (17), length 99) 172.16.1.4.48386 > 151.71.210.41.46198: UDP, length 71

18:51:40.035150 IP (tos 0x0, ttl 113, id 21297, offset 0, flags [none], proto UDP (17), length 267) 75.76.143.27.61146 > 172.16.1.4.48386: UDP, length 239

18:51:40.052764 IP (tos 0x0, ttl 48, id 39611, offset 0, flags [none], proto UDP (17), length 267) 24.22.20.169.63157 > 172.16.1.4.48386: UDP, length 239

18:51:40.057088 IP (tos 0x0, ttl 64, id 8262, offset 0, flags [none], proto UDP (17), length 99) 172.16.1.4.48386 > 76.202.53.212.34622: UDP, length 71

18:51:40.082275 IP (tos 0x0, ttl 64, id 38503, offset 0, flags [none], proto UDP (17), length 99) 172.16.1.4.48386 > 75.24.111.235.61831:UDP, length 71

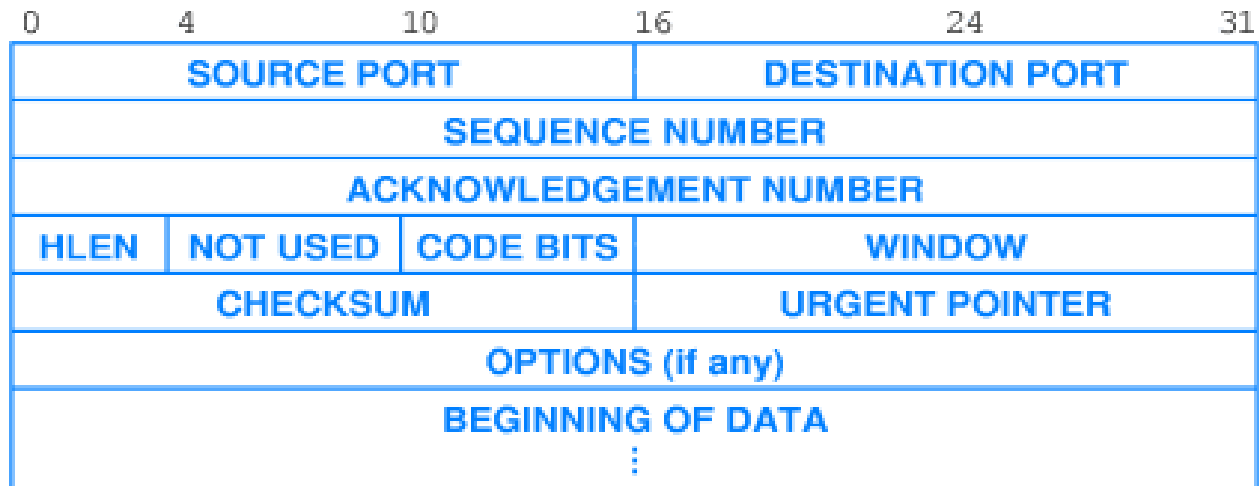


The screenshot shows the Wireshark interface with a list of captured packets. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Help), a toolbar with various icons, and a filter bar. The packet list table is as follows:

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	LinksysG_b2:9e:45	Gateway2_5a:f8:bc	ARP	Who has 192.168.1.102? Tell 192.168.1.1
2	0.000099	Gateway2_5a:f8:bc	LinksysG_b2:9e:45	ARP	192.168.1.102 is at 00:e0:b8:5a:f8:bc
3	0.424657	192.168.1.102	70.167.151.19	TCP	fujitsu-dtcns > http [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.437733	70.167.151.19	192.168.1.102	TCP	http > fujitsu-dtcns [RST] Seq=1 Win=0 Len=0
5	2.026763	192.168.1.102	70.167.151.26	TCP	atm-zip-office > http [FIN, ACK] Seq=1 Ack=1 Win=64767 Len=0
6	2.043277	70.167.151.26	192.168.1.102	TCP	http > atm-zip-office [RST] Seq=1 Win=0 Len=0
7	2.438778	192.168.1.102	164.106.251.250	TCP	ampr-info > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
8	2.468369	164.106.251.250	192.168.1.102	TCP	http > ampr-info [SYN, ACK] Seq=0 Ack=1 Win=4640 Len=0 MSS=1160
9	2.468872	192.168.1.102	164.106.251.250	TCP	ampr-info > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
10	2.469012	192.168.1.102	164.106.251.250	HTTP	GET / HTTP/1.1
11	2.503645	164.106.251.250	192.168.1.102	TCP	http > ampr-info [ACK] Seq=1 Ack=398 Win=5360 Len=0
12	2.504114	164.106.251.250	192.168.1.102	TCP	[TCP segment of a reassembled PDU]
13	2.504844	164.106.251.250	192.168.1.102	TCP	[TCP segment of a reassembled PDU]

A single packet

Layer ethernet
Layer IP
Layer TCP:

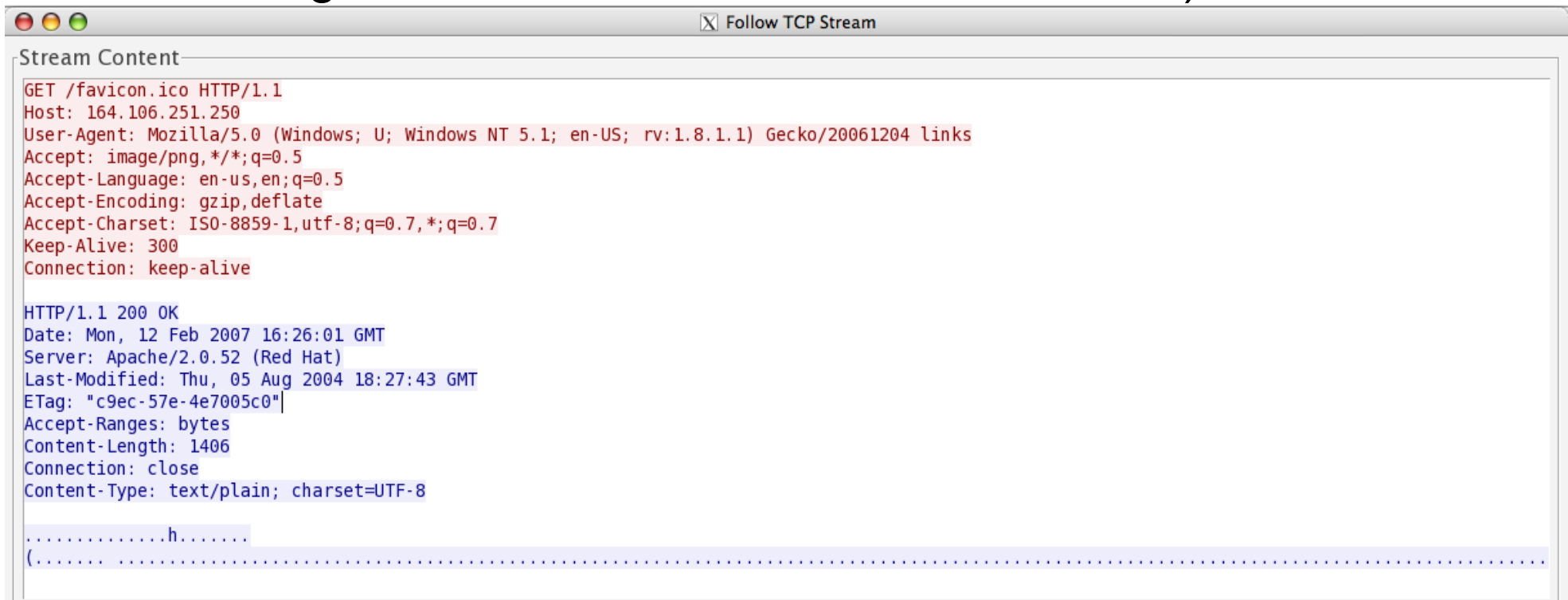


Application layer
Session layer
Presentation layer

How an interception appear

The packet stream is assemble in a data flow,
the data flow is analyzed on the in analyzed by the applicative
meaning (is an email, a web navigation, contains image or
attachment...)

*A complete reassembly of the sniffer transaction require
every kind of plugin the client supports too (ajax, flash, non
common image format, cache is used in the chain...)*



```
Stream Content
GET /favicon.ico HTTP/1.1
Host: 164.106.251.250
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.1) Gecko/20061204 links
Accept: image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*,q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Date: Mon, 12 Feb 2007 16:26:01 GMT
Server: Apache/2.0.52 (Red Hat)
Last-Modified: Thu, 05 Aug 2004 18:27:43 GMT
ETag: "c9ec-57e-4e7005c0"
Accept-Ranges: bytes
Content-Length: 1406
Connection: close
Content-Type: text/plain; charset=UTF-8

.....h.....
(.....)
```

TCP/IP reassembly

A TCP/IP session is a bidirectional data exchange.

A TCP guarantee confidentiality in the session. The two peer involved in the transmission, sent the acknowledge for the data received, and not the data only.

Other kind of packet is used in signaling pourpose (open the session, close the session, slow your sending...)

What 's the sniffer view ?

Client send DATA: sending of 1200 byte packet

Server send ACK: I've received your 1200 byte

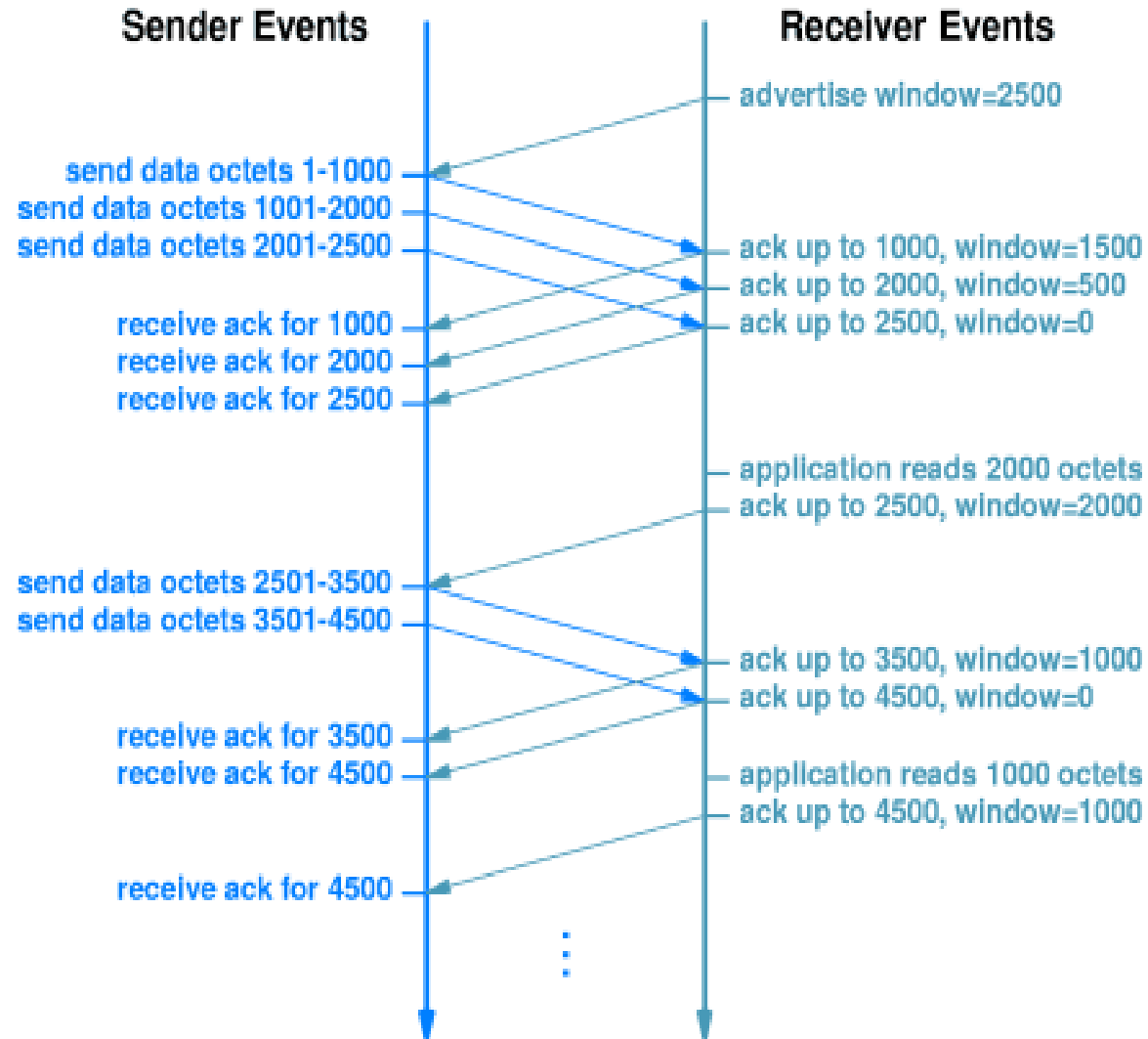
Server send DATA: sending of 40 byte packet

Client send ACK: I've recived your 40 byte

Server signaling: Closing connection where I've received 1200 byte.

Client signaling: Closing connection where I've received 40 byte.

TCP flow appearance:



Reassembly of TCP/IP session: problems

What happen in a sniffer when a session interrupt itself ?

Client send DATA: sending of 1200 byte packet

Server send ACK: I've received your 1200 byte

Not the client goes offline

Server (after 5 seconds): sending of 40 byte packet

Server (after 10 seconds):sending of 40 byte packet

Server (after 20 seconds):sending of 40 byte packet

Server (after 60 seconds): connection closed.

Lo sniffer ha ricevuto questo pacchetto, ma il client no.

Questo mostra la prima discrepanza: quello che lo sniffer sta leggendo non è detto sia stato effettivamente scambiato.

IP routing quirk and interception

A TCP transaction use the IP network in order to reach the remote peer,

The IP network is made over different technology around the world, switch of every age, router of every kind, sat, modem.

for work efficiently, the network protocol work discovering the best path, the appropriate device change the routing on the fly.

If this routing modification happen after the sniffer, no difference are noticed.

if this happen before the sniffer, some sessions will not be intercepted again.

TCP flow quirk: checksum

Will happen an error in the transmission, because the physical media or the software had some kind of bugs or working boundary.

Every packet has a checksum, an algebraic computation over the packet itself. This is used from the remote peer to make a computation of the receiving packet for check corruptions.

This is what a sniffer sees:

Client: sending a 1200 byte packet.

(physical error, packet corrupted)

Server: beside the packet is corrupted, I've dumped the data

Client: (after 5 seconds without the ACK) re-sending of 1200 byte

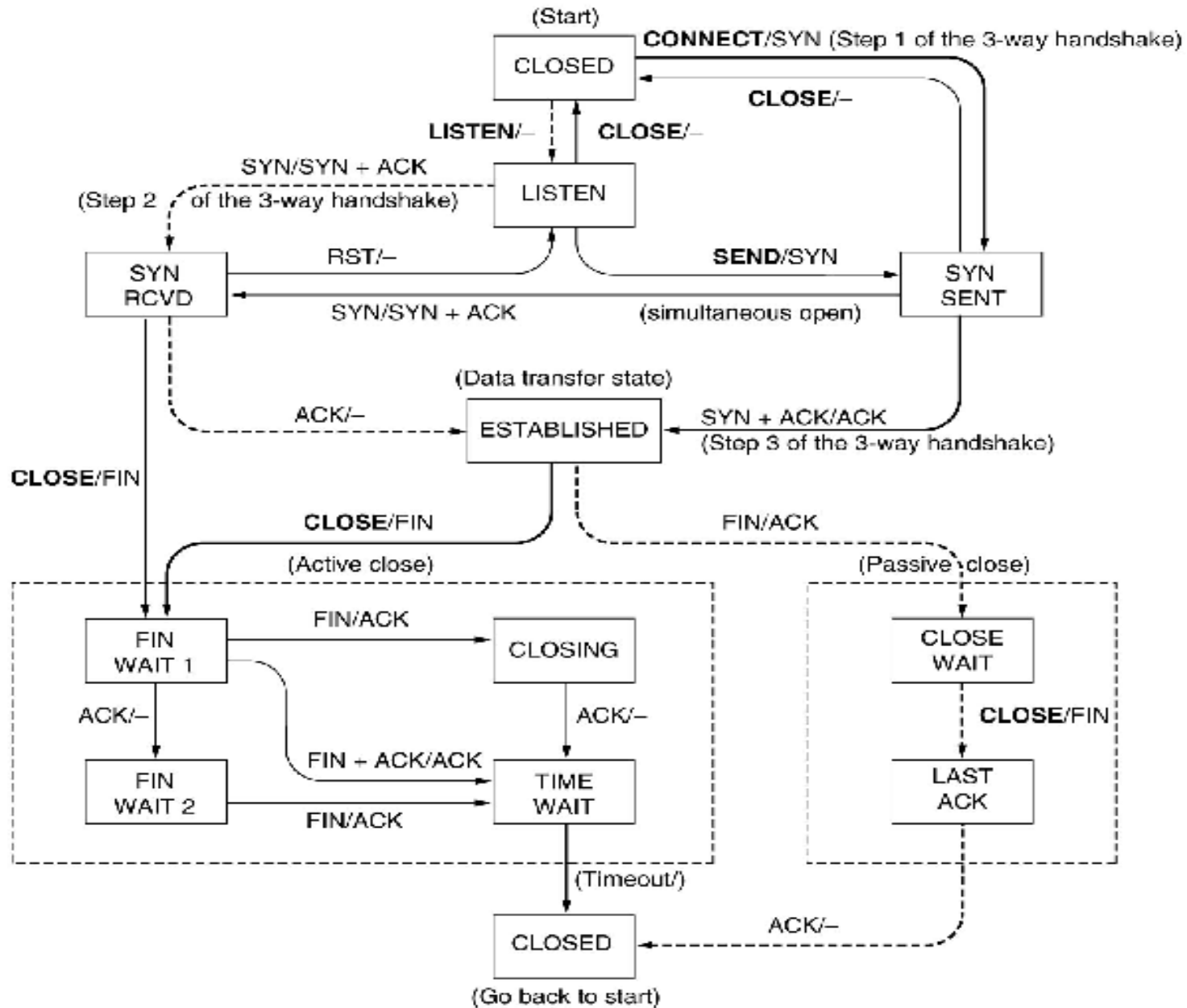
Server ACK: I've received 1200 byte

...

The sniffer had read two packets, but the server accepted only one.

Which should be trusted between ? In the data flow, what packet must be used ?

Complexity of TCP state finite machine



Sniffer complexity

Someone work in the home network, some other in 10gbis/sec

Saving every session ? make pattern matching in order to look specific keyword ? for the patten matching is required registration of entire session, because the pattern should happen at the end of the transmission.

preventive saving involve a timeout analysis, because the TCP had timeout of 30 minutres, 2 hours, 5 days (in different states).

The sniffer don't know the client-server state!

Sniffer reversing

Assumption:

If exist conditions where the sniffer will don't know which packet present to the analyst, **we should exploit them!**

Because could be developed a software that create arbitrarily those conditions:

- 1) when an application start a connection...
- 2) the software send fake packets knowing that will be dumped by the server.
- 3) send the real packet
- 4) manage the connection and back to point 1.

Sniffer reversing, details

The packets could be:

not accepted by the server but accepted by the sniffer (checksum/ipopt/tcpopt),

refused by the sniffer and accepted by the server (IPopt/TCPopt)

received by the sniffer and not from the server (TTL)

And what those packets could do ?

race condition ?

premature closing of the session ?

overwriting segments ?

crash!

Sniffer reversing, how to discover bugs...

don't reverse the sniffer, the power in sniffjoke is to be supported by the kernel!

if you reverse the single sniffer, fail over closed source sniffer

http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c

http://lxr.free-electrons.com/source/net/ipv4/ip_fragment.c

http://lxr.free-electrons.com/source/net/ipv4/ip_output.c

http://lxr.free-electrons.com/source/net/ipv4/ip_options.c

http://lxr.free-electrons.com/source/net/ipv4/ip_options.c

http://lxr.free-electrons.com/source/net/ipv4/tcp_output.c

Anti sniffing appearance:

this is a damaged email:

The screenshot shows a window titled "Follow TCP Stream" with a "Stream Content" pane. The pane displays a corrupted email header with several lines of garbled text in red and blue. The visible text includes: "[-60950 bytes missing in capture file].....u2)..P".uS .9...[60951 bytes missing in capture file]...6.A.e...o.....rT.[-2816 bytes missing in capture file]...X%6.p..C.G.zG220 mail.sogetthis.com ESMTP Postfix <CRLF>". The "mail.sogetthis.com" domain is highlighted in blue. Below the stream content, there are buttons for "Find", "Save As", and "Print", along with a dropdown menu showing "Entire conversation (222 bytes)". To the right of the dropdown are radio buttons for "ASCII", "EBCDIC", "Hex Dump", "C Arrays", and "Raw" (which is selected). At the bottom of the window, there are buttons for "Help", "Close" (highlighted with an orange border), and "Filter Out This Stream".

Follow TCP Stream

Stream Content

```
[-60950 bytes missing in capture file].....u2)..P".uS .9...[60951 bytes missing in capture  
file]...6.A.e...o.....rT.[-2816 bytes missing in capture file]...X%6.p..C.G.zG220 mail.sogetthis.com ESMTP  
Postfix  
<CRLF>
```

Find Save As Print Entire conversation (222 bytes) ASCII EBCDIC Hex Dump C Arrays Raw

Help Close Filter Out This Stream

Implementation problems

You could not forge the entire session, because is handled by the kernel,

You could not use simple socket raw, because your kernel could bring problems, error signaling, receive unexpected packets...

and a sniffjoke should not be coded in kernel!

solution: fake gateway with tun interface
background service and CLI management
solid default, apt-get & immediate running.

./sniffjoke --help

```
./sniffjoke [command] or ./sniffjoke --options:  
--debug [level 1-4] enable debug and set the verbosity [default:1]  
--logfile [file] set a logfile, [default sniffjoke.log]  
--user [username] downgrade privilege to the specified user  
[default:nobody]  
--group [groupname] downgrade privilege to the specified group  
[default:users]  
--chroot-dir [dir] runs chrooted into the specified dir  
[default:disabled]  
--force force restart if sniffjoke service  
--foreground running in foreground  
--version show sniffjoke version  
--help show this help
```

while sniffjoke is running, you should send one of those commands as command line argument:

```
start start sniffjoke hijacking/injection  
stop stop sniffjoke (but remain tunnel interface active)  
stat get statistics about sniffjoke configuration and network  
set start end value set per tcp ports the strongness of injection  
the values are: [heavy|normal|light|none]  
clear alias to "set 1 65535 none"  
showport show TCP ports strongness of injection  
loglevel 0 = normal, 1 = verbose, 2 = debug
```

<http://www.delirandom.net/sniffjoke>

the old and the new...

In Italy we had a large discussion about interception, but none about the effective security value of this investigative tool.

telephony, for the most, should be an investigative tool because:

- . require a specific access in the network center
- . require a lawful support and modality

Internet instead:

- . should be applied in every network hop
- . require free software

is a security thread for the people, not an investigative tool!

everyone sensible about the problem, could protect himself:

- . when had the control in client-server couple, the best way is

cryptology

- . when you had only one peer under your control, sniffjoke could be a solution.

Sniffjoke project

<http://github.com/vecna/sniffjoke>

the github project, supported from evilaliv3 in:

<http://github.com/evilaliv3/sniffjoke>

history:

2001:	libvsk	
2002-2004:	innova	
2006:	sniffjoke 0.1	ulog plugin
2007:	sniffjoke 0.2	ulog plugin
2008:	sniffjoke dead	kernel module
2008:	sniffjoke 0.3	service + local web gui
2010:	sniffjoke 0.4	service + CLI management

Sniffjoke project

<http://en.roolz.org/trafscrambler.html>

“This project was spawned because of my laziness to port sniffjoke to OSX and my interest in writing LKM for OSX.”



Without Sniff Joke



Sniffjoke project



With Sniff Joke

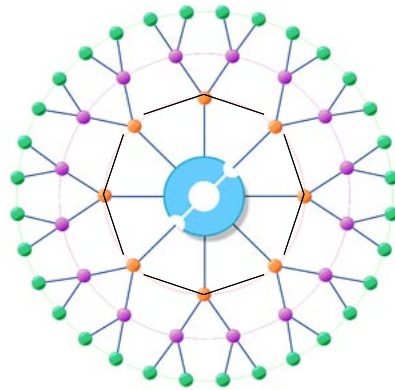
SniffJoke 0.4 alpha 2



SniffJoke 0.4 release, Tour index:

-
-
-
- What's a sniffer
-
- Why we need to protect from them
-
- How does a sniffer work
-
- What's the classical defense ?
-
- Why SniffJoke is different from the other solutions
-
- How does SniffJoke work ?
-

shiny network paint



The users are the **green dots**.

every other dot is a network element.

(when you make traceroute/tracert, you see them)

every network element could dump your traffic (forever ?)

how many sniffers do exist ?

```
# tcpdump -ni eth0 -s 0 -w sniffed.pcap &
```

ettercap

password collector, man in the middle,

xplico

10 giga pcap dissection

wireshark

every network protocol supported with layer 5 dissector

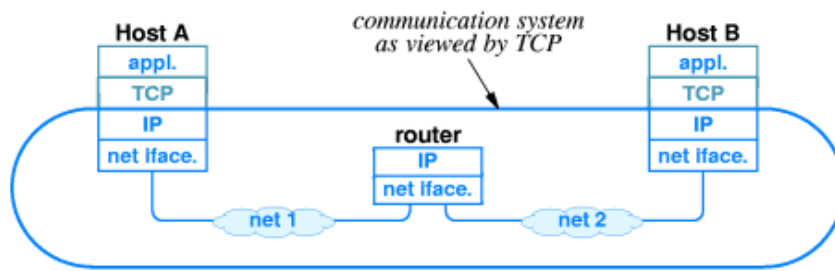
dsniff

command line passive-active-boring-disturbing tool

every kind of sniffer available:

http://www.corriere.it/scienze_e_tecnologie/10_gennab_20/spie-grazie-a-internet-lavinia-hanay-raja_94283bfa-05db-11df-a1d7-00144d2aabe.shtml

How does a sniffer work ?



Your software receives application layer data.
usually, the network software work on this layer.
(HTTP GET|POST, HELO MAIL FROM...)

Your socket reads network layer datagram
few software work on this layer (firewall, ping, traceroute)
(struct iphdr *ip = &buffer[1500], ...);

Your interface handle physical layer buzz.
sniffers live here!

Collateral damage

(if) your life transits & grows on the network...

if your transactions are *vulnerable to interception*,
these should be actively manipulated

your e-life follows the same rule.

**Common misconception: why an attacker
should care about me ?**

targeted attack

you have a smart (evil|good) attacker interested in you

untargeted attack

one of your internet profile is interesting for a smart (evil|good)
attacker

Why sniffing is the easiest way to obtain data ?

a seized hard disk require a law operation or a stealing operation.

a web mail require cooperation between the attacker and the provider. since few months of existence every large provider receives a lot of official and illegal requests.

Physical interception tool like keyloggers, hidden mics and so on.. require a study of the target and a precise organization. instead the sniffing is standard :)

Internet interception should obtain data for a long time working in **passive** mode, **unamanged** and **silently**.

How does an interception appears ?

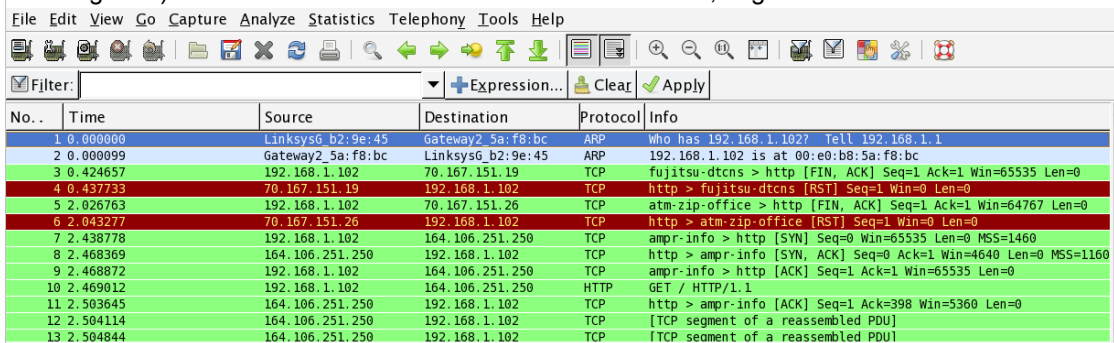
18:51:40.030995 IP (tos 0x0, ttl 64, id 9228, offset 0, flags [none], proto UDP (17), length 99) 172.16.1.4.48386 > 151.71.210.41.46198: UDP, length 71

18:51:40.035150 IP (tos 0x0, ttl 113, id 21297, offset 0, flags [none], proto UDP (17), length 267) 75.76.143.27.61146 > 172.16.1.4.48386: UDP, length 239

18:51:40.052764 IP (tos 0x0, ttl 48, id 39611, offset 0, flags [none], proto UDP (17), length 267) 24.22.20.169.63157 > 172.16.1.4.48386: UDP, length 239

18:51:40.057088 IP (tos 0x0, ttl 64, id 8262, offset 0, flags [none], proto UDP (17), length 99) 172.16.1.4.48386 > 76.202.53.212.34622: UDP, length 71

18:51:40.082275 IP (tos 0x0, ttl 64, id 38503, offset 0, flags [none], proto UDP (17), length 99) 172.16.1.4.48386 > 75.24.111.235.61831: UDP, length 71

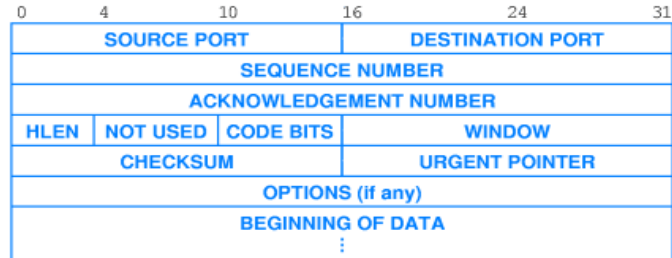


The screenshot shows the Wireshark interface with a packet list table. The table has columns for No., Time, Source, Destination, Protocol, and Info. The packets listed are:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	LinksysG_b2:9e:45	Gateway2_5a:f8:bc	ARP	Who has 192.168.1.102? Tell 192.168.1.1
2	0.000099	Gateway2_5a:f8:bc	LinksysG_b2:9e:45	ARP	192.168.1.102 is at 00:e0:b8:5a:f8:bc
3	0.424657	192.168.1.102	70.167.151.19	TCP	fujitsu-dtcns > http [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.437733	70.167.151.19	192.168.1.102	TCP	http > fujitsu-dtcns [RST] Seq=1 Win=0 Len=0
5	2.026763	192.168.1.102	70.167.151.26	TCP	atm-zip-office > http [FIN, ACK] Seq=1 Ack=1 Win=64767 Len=0
6	2.043277	70.167.151.26	192.168.1.102	TCP	http > atm-zip-office [RST] Seq=1 Win=0 Len=0
7	2.438778	192.168.1.102	164.106.251.250	TCP	ampr-info > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
8	2.468369	164.106.251.250	192.168.1.102	TCP	http > ampr-info [SYN, ACK] Seq=0 Ack=1 Win=4640 Len=0 MSS=1160
9	2.468872	192.168.1.102	164.106.251.250	TCP	ampr-info > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
10	2.469012	192.168.1.102	164.106.251.250	HTTP	GET / HTTP/1.1
11	2.503645	164.106.251.250	192.168.1.102	TCP	http > ampr-info [ACK] Seq=1 Ack=398 Win=5360 Len=0
12	2.504114	164.106.251.250	192.168.1.102	TCP	[TCP segment of a reassembled PDU]
13	2.504844	164.106.251.250	192.168.1.102	TCP	[TCP segment of a reassembled PDU]

A single packet

Layer ethernet
Layer IP
Layer TCP:

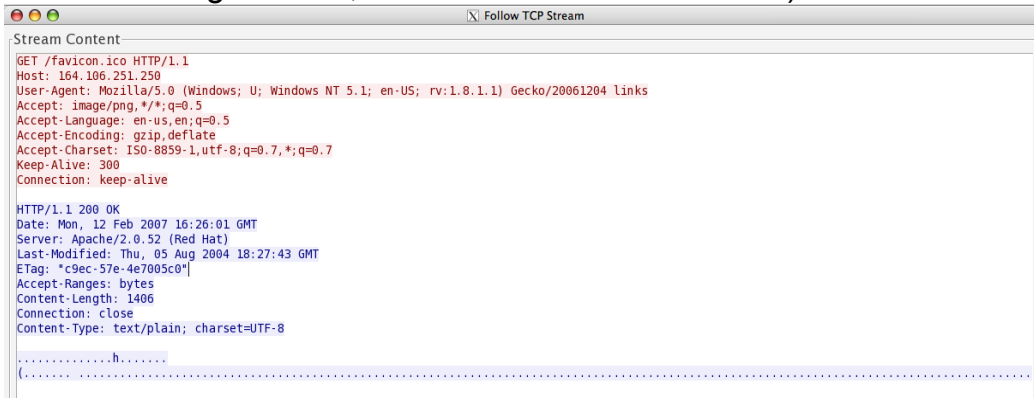


Application layer
Session layer
Presentation layer

How an interception appear

The packet stream is assemble in a data flow,
the data flow is analyzed on the in analyzed bye the applicative
meaning (is an email, a web navigation, contains image or
attachment...)

*A complete reassembly of the sniffer transaction require
every kind of plugin the client supports too (ajax, flash, non
common image format, cache is used in the chain...)*



```
Stream Content
GET /favicon.ico HTTP/1.1
Host: 164.106.251.250
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.1) Gecko/20061204 Links
Accept: image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Date: Mon, 12 Feb 2007 16:26:01 GMT
Server: Apache/2.0.52 (Red Hat)
Last-Modified: Thu, 05 Aug 2004 18:27:43 GMT
ETag: "c9ec-57e-4e7005c0"
Accept-Ranges: bytes
Content-Length: 1406
Connection: close
Content-Type: text/plain; charset=UTF-8

.....h.....
(.....
```

TCP/IP reassembly

A TCP/IP session is a bidirectional data exchange.

A TCP guarantee confidentiality in the session. The two peer involved in the transmission, sent the acknowledge for the data received, and not the data only.

Other kind of packet is used in signaling pourpose (open the session, close the session, slow your sending...)

What's the sniffer view ?

Client send DATA: sending of 1200 byte packet

Server send ACK: I've received your 1200 byte

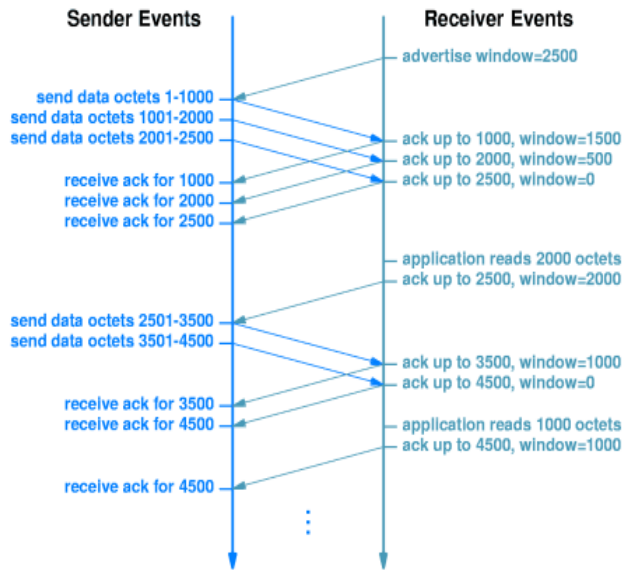
Server send DATA: sending of 40 byte packet

Client send ACK: I've recived your 40 byte

Server signaling: Closing connection where I've received 1200 byte.

Client signaling: Closing connection where I've received 40 byte.

TCP flow appearance:



Reassembly of TCP/IP session: problems

What happen in a sniffer when a session interrupt itself ?

Client send DATA: sending of 1200 byte packet

Server send ACK: I've received your 1200 byte

Not the client goes offline

Server (after 5 seconds): sending of 40 byte packet

Server (after 10 seconds):sending of 40 byte packet

Server (after 20 seconds):sending of 40 byte packet

Server (after 60 seconds): connection closed.

Lo sniffer ha ricevuto questo pacchetto, ma il client no.

Questo mostra la prima discrepanza: quello che lo sniffer sta leggendo non è detto sia stato effettivamente scambiato.

IP routing quirk and interception

A TCP transaction use the IP network in order to reach the remote peer,

The IP network is made over different technology around the world,
switch of every age, router of every kind, sat, modem.

for work efficiently, the network protocol work discovering the best
path, the appropriate device change the routing on the fly.

If this routing modification happen after the sniffer, no difference are
noticed.

if this happen before the sniffer, some sessions will not be intercepted
again.

TCP flow quirk: checksum

Will happen an error in the transmission, because the physical media or the software had some kind of bugs or working boundary.

Every packet has a checksum, an algebraic computation over the packet itself. This is used from the remote peer for make a computation of the receiving packet for check corruptions.

This is what a sniffer see:

Client: sending a 1200 byte packet.

(physical error, packet corrupted)

Server: beside the packet is corrupted, I've dumped the data

Client: (after 5 seconds without the ACK) re-sending of 1200 byte

Server ACK: I've received 1200 byte

...

The sniffer had read two packets, but the server accepted only one. Which should be trusted between ? In the data flow, what packet must be used ?

Sniffer complexity

Someone work in the home network, some other in 10gbis/sec

Saving every session ? make pattern matching in order to look specific keyword ? for the patten matching is required registration of entire session, because the pattern should happen at the end of the transmission.

preventive saving involve a timeout analysis, because the TCP had timeout of 30 minutres, 2 hours, 5 days (in differents states).

The sniffer don't know the client-server state!

Sniffer reversing

Assumption:

If exist conditions where the sniffer will don't know which packet present to the analyst, **we should exploit them!**

Because could be developed a software that create arbitrarily those conditions:

- 1) when an application start a connection...
- 2) the software send fake packets knowing that will be dumped by the server.
- 3) send the real packet
- 4) manage the connection and back to point 1.

Sniffer reversing, details

The packets could be:

not accepted by the server but accepted by the sniffer (checksum/ipopt/tcpopt),

refused by the sniffer and accepted by the server (IPopt/TCPopt)

received by the sniffer and not from the server (TTL)

And what those packets could do ?

race condition ?

premature closing of the session ?

overwriting segments ?

crash!

Sniffer reversing, how to discover bugs...

don't reverse the sniffer, the power in sniffjoke is to be supported by the kernel!

if you reverse the single sniffer, fail over closed source sniffer

http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c
http://lxr.free-electrons.com/source/net/ipv4/ip_fragment.c
http://lxr.free-electrons.com/source/net/ipv4/ip_output.c
http://lxr.free-electrons.com/source/net/ipv4/ip_options.c
http://lxr.free-electrons.com/source/net/ipv4/ip_options.c
http://lxr.free-electrons.com/source/net/ipv4/tcp_output.c

Anti sniffing appearance:

this is a damaged email:

The screenshot shows a window titled "Follow TCP Stream" with a "Stream Content" pane. The pane contains the following text:

```
[-60950 bytes missing in capture file].....u2)..P".us .9...[60951 bytes missing in capture file]...6.A.e...o.....rT.[-2816 bytes missing in capture file]...X%6.p..C.G.zG220 mail.sogetthis.com ESMTF Postfix <CRLF>
```

Below the stream content, there is a toolbar with icons for Find, Save As, and Print. A dropdown menu shows "Entire conversation (222 bytes)". To the right, there are radio buttons for "ASCII", "EBCDIC", "Hex Dump", "C Arrays", and "Raw" (which is selected).

At the bottom of the window, there is a "Help" button on the left, a "Close" button in the center, and a "Filter Out This Stream" button on the right.

Implementation problems

You could not forge the entire session, because is handled by the kernel,

You could not use simple socket raw, because your kernel could bring problems, error signaling, receive unexpected packets...

and a sniffjoke should not be coded in kernel!

**solution: fake gateway with tun interface
background service and CLI management
solid default, apt-get & immediate running.**

./sniffjoke --help

```
./sniffjoke [command] or ./sniffjoke --options:  
--debug [level 1-4] enable debug and set the verbosity [default:1]  
--logfile [file] set a logfile, [default sniffjoke.log]  
--user [username] downgrade privilege to the specified user  
[default:nobody]  
--group [groupname] downgrade privilege to the specified group  
[default:users]  
--chroot-dir [dir] runs chrooted into the specified dir  
[default:disabled]  
--force force restart if sniffjoke service  
--foreground running in foreground  
--version show sniffjoke version  
--help show this help
```

while sniffjoke is running, you should send one of those commands as
command line argument:

```
start start sniffjoke hijacking/injection  
stop stop sniffjoke (but remain tunnel interface active)  
stat get statistics about sniffjoke configuration and network  
set start end value set per tcp ports the strongness of injection  
the values are: [heavy|normal|light|none]  
clear alias to "set 1 65535 none"  
showport show TCP ports strongness of injection  
loglevel 0 = normal, 1 = verbose, 2 = debug
```

<http://www.delirandom.net/sniffjoke>

the old and the new...

In italy we had a large discussion about interception, but none about the effective security value of this investigative tool.

telephony, for the most, should be an investigative tool because:

- . require a specific access in the network center
- . require a lawful support and modality

Internet instead:

- . should be apply in every network hop
- . require free software

is a securtiy threadd for the people, not an investigative tool!

everyone sensible about the problem, could protect himself:

- . when had the control in client-server couple, the best way is cryptography
- . when you had only one peer under your control, sniffjoke could be a solution.

Sniffjoke project

<http://github.com/vecna/sniffjoke>

the github project, supported from evilaliv3 in:

<http://github.com/evilaliv3/sniffjoke>

history:

2001:	libvsk	
2002-2004:	innova	
2006:	sniffjoke 0.1	ulog plugin
2007:	sniffjoke 0.2	ulog plugin
2008:	sniffjoke dead	kernel module
2008:	sniffjoke 0.3	service + local web gui
2010:	sniffjoke 0.4	service + CLI management

Sniffjoke project

<http://en.rootz.org/trafscrambler.html>

“This project was spawned because of my laziness to port sniffjoke to OSX and my interest in writing LKM for OSX.”



