

# SniffJoke

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



vecna @ delirandom | winstonsmith | s0ftpj

**ESC 2009**

TCP, kernel delirium, sniffing, privacy,  
insicurezza & paura, filosofie di sicurezza.

# SniffJoke eviscerated

- 1, Come funziona uno **sniffer**,  
*Come ricostruisce i flussi TCP,*  
*Perché si basa su delle assunzioni,*  
*E se le facessimo crollare ?*
- 2, Le **intercettazioni** per l'uomo della strada  
*il reale effetto di sicurezza,*  
*e di insicurezza ?*
- 3, **SniffJoke, effetti, obiettivi, significato**

[vecna@delirandom.net](mailto:vecna@delirandom.net) <http://www.delirandom.net>  
progetto winston smith - <http://winstonsmith.info>

# Come funziona uno sniffer ?

In una rete, una macchina (A) comunica tramite il gateway (G). Se esiste almeno una macchina con una scheda di rete in promisc, anch'essa riceverà i pacchetti delle comunicazioni tra A e G.

il gateway, legittimo o illegittimo, riceve il pacchetto a lui è destinato senza che nulla sia stato alterato.

Questo si contestualizza nelle LAN casalinghe (promisc, mitm), in quelle aziendali (monitor mode), in quelle degli ISP (monitor mode tra switch e gateway).

# Cosa raccoglie uno sniffer ?

Pacchetti, una lista continua.

```
struct packet
{
    struct ether_header ethhdr; (14 byte)
    struct iphdr ip; (20 byte)
    struct tcphdr tcp; (20 byte)
    unsigned char data[1442];
};
```

Ogni struttura contiene i dati del protocollo associato:

**ethhdr**, campi ethernet, definiscono gli indirizzi fisici di mittente-destinatario  
**ip**, indirizzi ip, lunghezza del dato, checksum  
**tcp**, servizi, sequence number, flag, checksum

# Come si riconoscono i flussi TCP ?

```
IP remote:1025 > localweb:80: S 2653389673:2653389673(0) win 65535 <mss
16344,nop,wscale 3,nop,nop,timestamp 237170260 0,sackOK,eol>
IP localweb:80 > remote:1025: S 224414191:224414191(0) ack 2653389674 win
65535 <mss 16344,nop,wscale 3,nop,nop,timestamp 237170260
237170260,sackOK,eol>
IP remote:1025 > localweb:80: . ack 1 win 65535
<nop,nop,timestamp 237170260 237170260>
IP localweb:80 > remote:1025: . ack 1 win 65535
<nop,nop,timestamp 237170260 237170260>
IP remote:1025 > localweb:80: P 1:18(17) ack 1 win 65535
<nop,nop,timestamp 237170355 237170260>
IP localweb:80 > remote:1025: . ack 18 win 65535
<nop,nop,timestamp 237170355 237170355>
IP remote:1025 > localweb:80: P 18:20(2) ack 1 win 65535
<nop,nop,timestamp 237170362 237170355>
IP localweb:80 > remote:1025: . ack 20 win 65535
<nop,nop,timestamp 237170362 237170362>
IP localweb:80 > remote:1025: P 1:155(154) ack 20 win 65535
<nop,nop,timestamp 237170362 237170362>
IP remote:1025 > localweb:80: . ack 155 win 65535
<nop,nop,timestamp 237170362 237170362>
IP localweb:80 > remote:1025: F 155:155(0) ack 20 win 65535
<nop,nop,timestamp 237170362 237170362>
IP remote:1025 > localweb:80: . ack 156 win 65535
<nop,nop,timestamp 237170362 237170362>
IP remote:1025 > localweb:80: F 20:20(0) ack 156 win 65535
<nop,nop,timestamp 237170362 237170362>
IP localweb:80 > remote:1025: . ack 21 win 65535
<nop,nop,timestamp 237170362 237170362>
```

# Cosa contengono i flussi TCP ?

**SYN**

**SYN ACK**

**ACK**

**GET / HTTP 1.1/**

**ACK (del dato CLIENT)**

**HTTP/1.0 403 Forbidden**

**Date: Thu, 18 Jun 2009 19:39:09 GMT**

**Server: BProxy 0.1**

**Content-Length: 25**

**Content-Type: application/json**

**{"error": "403 Forbidden"}**

**ACK (del dato SERVER)**

**FIN**

**ACK (del FIN)**

**FIN**

**ACK**

# Come si ricostruiscono i flussi ?

Si legge il pacchetto

Si verificano la coerenza delle dimensioni del pacchetto

Si verifica la tupla **ipsrc:portasrc** – **ipdst:portadst**

Si legge lo stato della tupla e si associa il flag del pacchetto

Se ci sono dati, si appende alla sessione.

# pacchetti inaspettati ?

Se gli sniffer non fanno tutti i check necessari, potrebbe succedere che uno sniffer accetti un pacchetto scartato dal server!

Ottenendo una desincronizzazione tra la sessione che sta avvenendo sul server e quella che sta seguendo lo sniffer...



# pacchetti inaspettati ?

CLIENT

-> SYN  
<- SYN/ACK  
-> ACK  
-> PSH

<- ACK  
-> PSH

SNIFFJOKE

bruteforce  
<- SYN/ACK  
-> ACK  
-> PSH falso  
-> PSH vero  
-> RST falso  
-> FIN falso

-> PSH falso  
-> PSH vero  
-> PSH falso  
-> PSH  
vero/invalido

SNIFFER

-> SYN  
<- SYN/ACK  
-> ACK  
-> PSH falso  
-> PSH vero  
-> RST  
(chiusura ?)

<- ACK vero ?  
-> ...

SERVER

-> SYN  
<- SYN/ACK  
-> ACK  
--! PSH falso  
-> PSH (vero)  
--! RST falso

<- ACK  
-> PSH

# Macrotipi d'attacco

I pacchetti possono essere:

non accettati dal server, accettati dallo sniffer (checksum/IPopt/TCPopt).

non accettati dallo sniffer, accettati dal server (IPopt/TCPopt).

essere ricevuti dallo sniffer, e non raggiungere mai il server (TTL).

race condition ?

Cosa possono causare ?

Chiusura prematura della sessione,  
Cancellarne delle parti,  
Accettare dati fasulli,  
Crashare!

La cosa bella...

E' che questi attacchi  
esistono da 11 anni!

Phrack 54-10

**Defeating Sniffers and Intrusion  
Detection Systems**

Thomas H. Ptacek, Timothy N. Newsham:  
**Insertion, evasion, and denial of  
service, Eluding network intrusion  
detection**

# I grossi problemi implementativi

Il problema di questi software, è che non si può fare il "classico packet forging". (es, creo IP+TCP+dato, invio su SOCK\_RAW).

Perché per il kernel queste connessioni non sono legittime (lo si nota con un SYN scan):

```
scanner -> SYN -> server
server -> SYN+ACK | RST+ACK -> scanner
client -> RST+ACK -> server
```

# I grossi problemi implementativi

E poi l'utente dovrebbe in modo trasparente riuscire a manipolare il proprio traffico, quindi il software deve essere trasparente.

L'applicativo (firefox, telnet, irsii, ...) si collega tramite il kernel.

il kernel invia i pacchetti (come sempre)

il software interviene, li modifica, li blocca, ne mette di nuovi, ecc...

# I grossi problemi implementativi

Per manipolare i pacchetti inviati dal kernel è necessario programmare in kernel space.

Cosa che va bene a 20 anni, ma non era piu' il tempo perché impazzissi davanti a kernel panic ...

Come sviluppare un software in user space in grado di manipolare i pacchetti in uscita dall'host dopo che il kernel li ha fatti passare nei suoi meandri ?

# Innova 1.0

framework basato su plugins,

utilizza iptables/iproute/MARK per:  
mettere in blocco i dati in uscita  
sniffarli localmente  
generarli senza MARK affinché escano.  
bloccare i pacchetti in entrata  
rigirarli su localhost

In pratica:  
sangue & merda.

# SniffJoke 0.1/0.2

Si ispirava ad innova,

era un plugin per ulogd,

potava solo mandare pacchetti in ritardo rispetto alla sessione reale.

Superava ethereal, non wireshark, non libnids.



# SniffJoke 0.3

Si ispirava al MOCA :)

Un client VPN è anche il software che manipola il traffico dopo che è transitato per il gateway.

SniffJoke 0.3 fa la stessa cosa, e poi inoltra al default gateway.

Puo' bloccare i pacchetti, e quindi le opzioni di attacco aumentano...

# SniffJoke 0.3

Bruteforce del TTL per ogni host,

Utilizzo randomico della sequenza di attacchi,

Utilizzo randomico delle variabili nei singoli hack,

Iniezione di IP options sui pacchetti legittimi (anche se il risultato non è testato! le implementazioni degli stack TCP/IP differiscono tra tutti gli OS!)

# TODO: SniffJoke 0.4 ~ 1.0

Sulla mailing list di wireshark è seguito un thread.

In sintesi "non viene definito un bug, non vogliono entrare in una code war".

<http://en.roolz.org/trafscrambler.html>  
per MACosX cit: "This project was spawned because of my laziness to port sniffjoke to OSX and my interest in writing LKM for OSX."

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

<http://www.delirandom.net/20090707/sniffjoke-in-transizic>

# TODO: SniffJoke 0.4 ~ 1.0

Cache della distanza di HOP

hack in relazione a servizi/server

l'hack definitivo

funzionamento lato server

payload fasulli con coerenza a layer 5

# SniffJoke 0.3

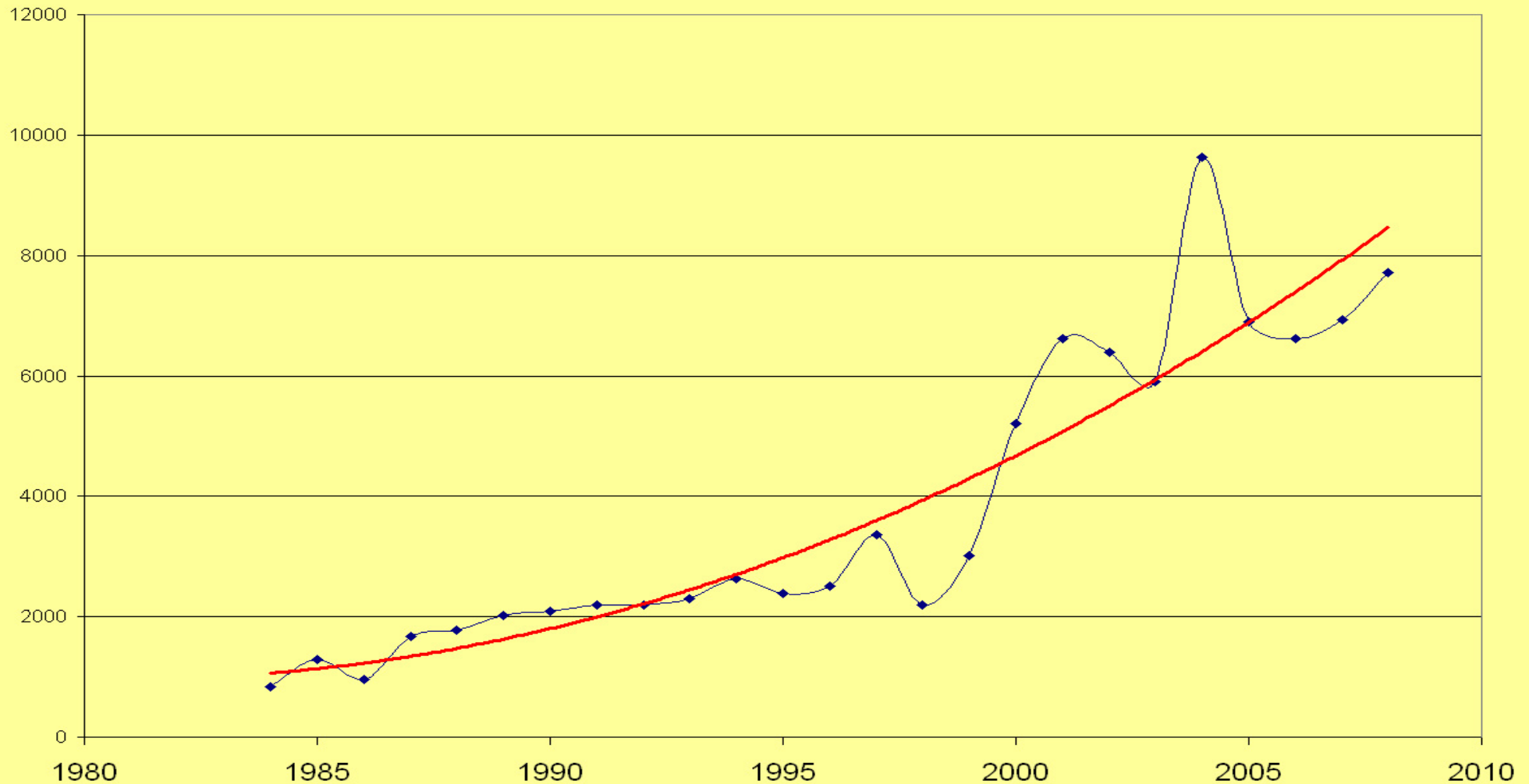


With Sniff Joke

# Perché, parte 1

La "percezione della paura" è una problematica sociale che HA FATTO CARRIERA

**Occorrenza della parola "Paura" su Repubblica dal 1984**



# Perché, parte 1

Ed anche il "suggerimento" della  
sicurezza...

**Occorrenze della parola "Sicurezza" su Repubblica dal 1984 ad oggi**



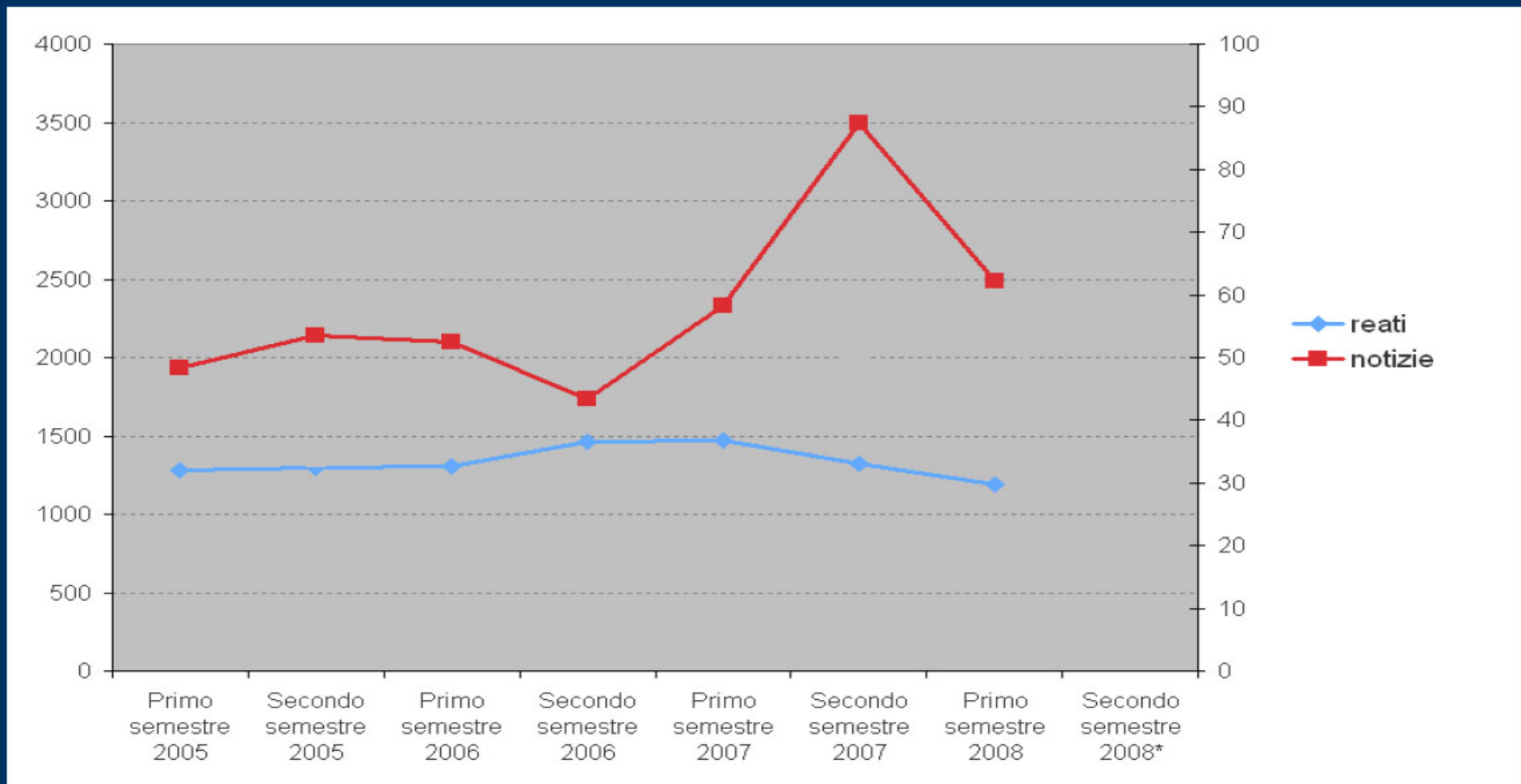
# Perché, parte 1

Inutile dire che la scelta è arbitraria

La percezione della sicurezza tra comunicazione e realtà



## Trend di notizie e reati (7 semestri)





# Perché, parte 2

La necessità di SENTIRSI SICURO (spontanea o indotta, pensateci voi) da parte del cittadino è aumentata.

Nell'immaginario collettivo "ci vogliono più controlli". Perché ? boh. per via di hollywood forse, o perché si è semplicemente ignoranti in termini di scienza della sicurezza. del resto si parla di cittadini, giornalisti, politici... non di security engineer.

Quindi si sogna che, **controllando Internet** come avviene con il telefono, **"SI STIA PIU' AL SICURO"**

# HAAAAHAHAHAHAHAHAHAHAHAHAHAH, Perché (è impossibile) ?

La rete Internet funziona con meccanismi  
totalmente diversi da quelli ai quali,  
la maggioranza della cittadinanza, è  
abituata (a immaginare).

Dove tutti erano utenti allo stesso  
modo...

Qui chiunque puo' installare software

Dove solo lo stato aveva la possibilità

Ora i privati concedono questa  
possibilità

Dove le leggi erano statali, e bastavano

La rete è globale

# securiry-enforcement, lawful-interception,

Devono di conseguenza funzionare  
diversamente !

1) Il controllo totale serve solo per  
gli sprovveduti

1 bis) crea uno squilibrio di potere

2) non risolve problemi, ne crea di  
nuovi

3) non è possibile

4) molti civili lo vorrebbero!!!!!!one

# e SniffJoke ?

Dovrebbe impattare poco!

Chi fa analisi manuali, su di un target, puo' riconoscere che c'e' del traffico fasullo (plausibile solo per una macchina)

Durante un'indagine, si ha uno spettro di visione molto piu' ampio che gli stream TCP

Aiuta a proteggersi da trojan e da sniffer massivi (LAN, ISP, o remota)

Serve per diffondere i contenuti di questa presentazione

# SniffJoke e effetti

Consente l'occultamento, l'offuscamento, della connessione.

Gli hack non so neppure quando fanno effetto :P

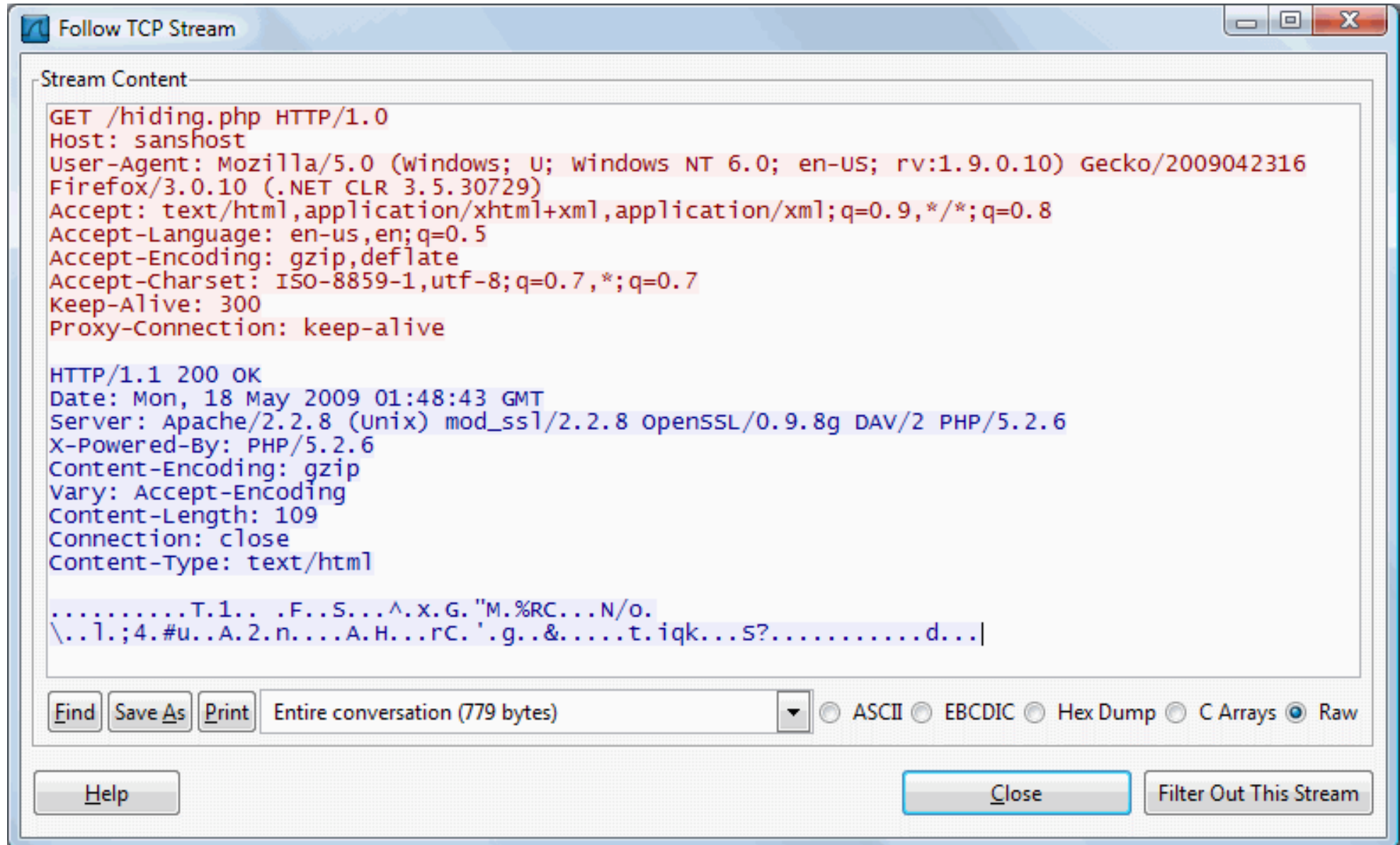
Si basano sull'immaginare quali parti di codice possono essere state eliminate per ottimizzare l'analisi del traffico.

Sul formare connessioni che siano "plausibili", ma che tocchino quelle parti di RFC non definite con esattezza.

# SniffJoke con #HACKSDEBUG

```
Session[0]: local:33103 -> 66.135.60.177:25 puppet 9711 TTL exp 0 wrk 255
** [fake data] (lo:33103 66.135.60.177:25 #2) id 32636 exp:16 wrk:17 len 52-113[77] data 25 {01000}
** [fake SEQ] (lo:33103 66.135.60.177:25 #2) id 32643 exp:16 wrk:17 len 52-111[75] data 23 {00000}
** [fake FIN/RST] (lo:33103 66.135.60.177:25 #2) id 32638 exp:16 wrk:17 len 52-88[52] data 0 {01001}
** [zero window] (lo:33103 66.135.60.177:25 #2) id 32634 exp:16 wrk:17 len 52-88[52] data 0 {10011}
** [valid RST bad SEQ] (lo:33103 66.135.60.177:25 #2) id 32636 exp:16 wrk:17 len 52-88[52] data 0 {01001}
** [fake SYN] (lo:25 66.135.60.177:33103 #2) id 32640 exp:16 wrk:17 len 52-88[52] data 0 {10000}
** [fake data] (lo:33103 66.135.60.177:25 #3) id 32639 exp:16 wrk:17 len 64-119[83] data 19 {01000}
** [fake SEQ] (lo:33103 66.135.60.177:25 #3) id 32644 exp:16 wrk:17 len 64-116[80] data 16 {00000}
** [fake FIN/RST] (lo:33103 66.135.60.177:25 #3) id 32640 exp:16 wrk:17 len 64-100[64] data 0 {01010}
** [zero window] (lo:33103 66.135.60.177:25 #3) id 32635 exp:16 wrk:17 len 64-100[64] data 0 {10011}
** [valid RST bad SEQ] (lo:33103 66.135.60.177:25 #3) id 32643 exp:16 wrk:17 len 64-100[64] data 0 {01001}
** [fake SYN] (lo:33103 66.135.60.177:25 #3) id 32640 exp:16 wrk:17 len 64-100[64] data 0 {10000}
```

# Wireshark – follow TCP stream



The screenshot shows the 'Follow TCP Stream' window in Wireshark. The window title is 'Follow TCP Stream'. The main content area displays the following text:

```
Stream Content
GET /hiding.php HTTP/1.0
Host: sanshost
User-Agent: Mozilla/5.0 (windows; U; windows NT 6.0; en-US; rv:1.9.0.10) Gecko/2009042316
Firefox/3.0.10 (.NET CLR 3.5.30729)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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
Proxy-Connection: keep-alive

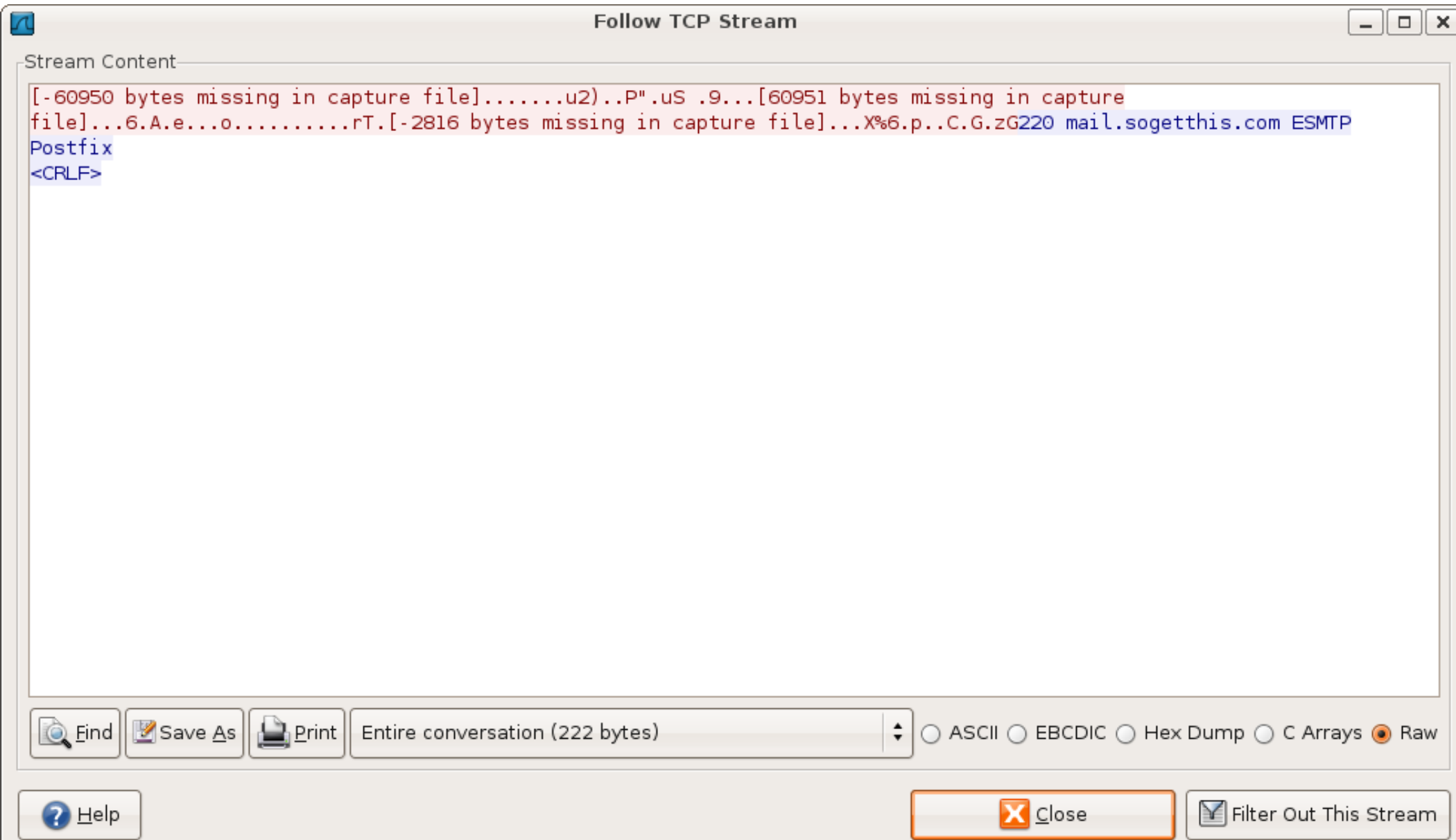
HTTP/1.1 200 OK
Date: Mon, 18 May 2009 01:48:43 GMT
Server: Apache/2.2.8 (Unix) mod_ssl/2.2.8 OpenSSL/0.9.8g DAV/2 PHP/5.2.6
X-Powered-By: PHP/5.2.6
Content-Encoding: gzip
Vary: Accept-Encoding
Content-Length: 109
Connection: close
Content-Type: text/html

.....T.1.. .F..S...^..x.G."M.%RC...N/o.
\..l.;4.#u..A.2.n....A.H...rC.'g..&.....t.iqk...s?.....d...|
```

At the bottom of the window, there are several controls:

- Buttons: Find, Save As, Print
- Dropdown menu: Entire conversation (779 bytes)
- Radio buttons: ASCII, EBCDIC, Hex Dump, C Arrays, Raw (selected)
- Buttons: Help, Close, Filter Out This Stream

# Wireshark mail + sniffjoke



The image shows a screenshot of the 'Follow TCP Stream' window in Wireshark. The window title is 'Follow TCP Stream'. The main content area displays 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 ESMTP  
Postfix  
<CRLF>
```

Below the main content area, there is a toolbar with the following buttons: Find, Save As, Print, and 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 three buttons: 'Help', 'Close', and 'Filter Out This Stream'.



# (senza) SniffJoke, xplico











http://localhost:71/webs/resBody/30

**Google** nature  Cerca immagini Cerca nel Web [Ricerca avanzata](#) [Preferenze](#)  
Protezione SafeSearch media attivata

**Immagini** Mostra: Immagini di tutte le dimensioni | Qualsiasi contenuto | Risultati 1 - 20 di circa **132.000.000** (0,03 secondi)

[Natura](#) Scopri su Focus tutte le novità su ambiente, **natura** e tanto altro!  
[www.Focus.it](http://www.Focus.it)

[Hot New Cars](#) Link sponsorizzati  
2 Fast Cars  
Good Gas Cars  
[www.mycomputer.com](http://www.mycomputer.com)

 <p>in mezzo alla <b>natura</b> 1024 x 768 - 513k - jpg <a href="http://blog.libero.it">blog.libero.it</a></p>	 <p>Galleria 1024 x 768 - 176k - jpg <a href="http://www.european-webzine.eu">www.european-webzine.eu</a></p>	 <p>Perche' anche la <b>natura</b> ha un suo ... 1600 x 1200 - 1302k - jpg <a href="http://www.enature.it">www.enature.it</a></p>	 <p>... 1024 x 768 - 153k - jpg <a href="http://blog.libero.it">blog.libero.it</a></p>	 <p><b>Nature</b> 1024 x 768 - 369k - jpg <a href="http://www.myspace.com">www.myspace.com</a> [ <a href="#">Altre risultati da photobucket.com</a> ]</p>
 <p>1024x768, centro, wallpaper 1024 x 768 - 127k - jpg <a href="http://www.es.it">www.es.it</a></p>	 <p><b>nature</b>-orange-sydney.jpg 800 x 600 - 86k - jpg <a href="http://www.le-conscience.com">www.le-conscience.com</a></p>	 <p>Mother <b>Nature</b> ... 500 x 375 - 36k <a href="http://www.the-nature.com">www.the-nature.com</a></p>	 <p>... Earth <b>Nature</b> 500 x 375 - 123k - jpg <a href="http://www.flickr.com">www.flickr.com</a></p>	 <p><b>Nature</b> 550 x 400 - 36k - jpg <a href="http://www.design-studio.com">www.design-studio.com</a></p>

Done

FoxyProxy: Disabled 127.0.0.1

# (con) SniffJoke, xplico

 http://images.google.it/images?gbv=2&hl=it&q=nature&sa=N&start=20&ndsp=20

[Web](#) [Immagini](#) [Maps](#) [News](#) [Video](#) [Gmail](#) [altro](#) 

# Disservizi, considerazioni, ecc...

Crea un tunnel VPN in modo automatico (per ora funziona solo su Linux, forse per 0.4, o per la 1.0, MacOSX/BSD)

Cambia default gateway = le regole di iptables legate ad un'interfaccia cessano di funzionare.

Eventuali altri tunnel VPN non vengono considerati, perché trova il default gateway solo in ethernet (e fa packet forging ethernet)

# Disservizi, considerazioni, ecc...

All'inizio di una sessione TCP, effettua l'equivalente di un tcptraceroute per conoscere la distanza in HOP con il server da contattare.

Tiene una cache per ogni avvio, versione 0.4/1.0, avrà il dump su file  
trivia: come fareste a supportare diverse network locations ?

Manca la configurazione per porte TCP  
Manca l'injection a payload fake layer 5

# GUI & distro

Siccome sono convinto che software per la privacy e la difesa online abbiano senso se vengono utilizzati da utenti non esperti, i due modi provati sono stati:

1) cercare di far qualcosa facilmente packettizabile, così godendo dei repository di riferimento per diffondersi.

2) una GUI e dei default il piu' possibile autonomi. la GUI è stata fatta con libswill, tramite un webserver locale che stampa HTML come fosse un vecchio CGI.

# o'rly !?

Pagina del progetto  
<http://www.delirandom.net/sniffjoke>

progetto su github:  
<http://github.com/vecna/sniffjoke/tree/master>

software di Gianluca Costa, che ha analizzato gli  
effetti di SniffJoke su questo e altri sniffer  
<http://www.xplico.com>

Defeating sniffer & IDS  
<http://www.phrack.org/issues.html?issue=54&id=10>

anti anti sniffer patch, marginalmente correlato  
<http://www.s0ftpj.org/bfi/online/bfi9/BFi09-14>

Discussione sulla mailing list WireShark riguardo a Sj  
<http://www.mail-archive.com/wireshark-dev@wireshark.org/msg13465.html>