

EX 1013

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so

that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software

interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new

versions
of the General Public License from time to time. Such new versions
will
be similar in spirit to the present version, but may differ in detail
to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and
"any
later version", you have the option of following the terms and
conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number
of
this License, you may choose any version ever published by the Free
Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the
author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we
sometimes
make exceptions for this. Our decision will be guided by the two
goals
of preserving the free status of all derivatives of our free software
and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT
WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER
EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK
AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY
SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper

mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program
`Gnomovision' (which makes passes at compilers) written by James
Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
#
# p0f - passive OS fingerprinting
# (c) <lcamtuf@tpi.pl>
#

CC      = gcc
CLIBS   = -lpcap
SUNLIBS = -lsocket -lnsl -D_SUN_=1
STRIP   = strip
CFLAGS  = -O3 -Wall
FILE    = p0f
VERSION = 1.7

DISTR0  = p0f.c Makefile README COPYING tcp.h p0f.fp

all: $(FILE) strip

$(FILE): p0f.c
        $(CC) $(CFLAGS) -DVER=\"$(VERSION)\" -o $@ $< $(CLIBS) \
        `uname|egrep -i 'sunos|solar' >/dev/null && echo \"$(SUNLIBS)\"`

strip:
        strip $(FILE) || true

clean:
        rm -f core *.o $(FILE)
        rm -rf p0f-$(VERSION)

tgz: clean
        mkdir -m 755 p0f-$(VERSION)
        cp $(DISTR0) p0f-$(VERSION)/
        chmod 644 p0f-$(VERSION)/*
        tar cfvz /$(FILE).tgz p0f-$(VERSION)
        chmod 644 /$(FILE).tgz
        rm -rf p0f-$(VERSION)

publish: tgz
        scp /p0f.tgz lcamtuf@dione.ids.pl:public_html/p0f.tgz
        scp /p0f.tgz lcamtuf@dione.ids.pl:public_html/p0f-$(VERSION).tgz
        rm -f /p0f.tgz
```

```
/*
```

```
p0f - passive OS fingerprinting
```

```
-----  
(c) <lcamtuf@tpi.pl>
```

```
The p0f utility and related utilities are free software; you can  
redistribute it and/or modify it under the terms of the GNU Library  
General Public License as published by the Free Software Foundation;  
either version 2 of the License, or (at your option) any later  
version.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS
```

```
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY,
```

```
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT  
SHALL
```

```
MICHAL ZALEWSKI, OR ANY OTHER CONTRIBUTORS BE LIABLE FOR ANY CLAIM,  
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT  
OR
```

```
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE  
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

```
*/
```

```
#include <stdio.h>  
#include <string.h>  
#include <sys/types.h>  
#include <netinet/in.h>  
#include <pcap.h>  
#include <arpa/inet.h>  
#include <signal.h>  
#include <unistd.h>
```

```
#include "tcp.h"  
#define MAXFPS 1000  
#define FPBUF 120  
#define INBUF 1024  
#define TTLDW 30
```

```
#ifndef VER  
# define VER "(?)"  
#endif /* !VER */
```

```
extern char *optarg;  
extern int optind;
```

```
char fps[MAXFPS][FPBUF];  
int wss, wscale, mss, nop, ttl, df, sok,tmp,header_len=14,dupa;
```

```

int verbose=0,sp,dp;
struct in_addr sip,dip;
struct bpf_program flt;
pcap_t *pt;

void die_nicely() {
    pcap_close(pt);
    exit(0);
}

void lookup(void);

void parse(u_char *blabla, struct pcap_pkthdr *pph, u_char *packet) {
    struct iphdr *iph;
    struct tcphdr *tcph;
    int ilen=0, hlen=0, off, olen;
    dupa=0;

    if (pph->len < header_len+sizeof(struct iphdr)+sizeof(struct
tcphdr)) {
        return;
    }
    // Rare tropical disease ugly dirty obfuscated hack ;>
    iph=(struct iphdr*) (packet);
    if ((iph->ihl>>4)!=4 || iph->protocol!=IPPROTO_TCP)
        iph=(struct iphdr*)(packet+header_len);
    if ((iph->ihl>>4)!=4 || iph->protocol!=IPPROTO_TCP) {
        int a,b;
        iph=(struct iphdr*) (packet);
        // Change ihl byteorder, endian detection ;)
        a=iph->ihl&15;b=(iph->ihl>>4)&15;iph->ihl=a*16+b;
        if ((iph->ihl>>4)!=4 || iph->protocol!=IPPROTO_TCP)
            iph=(struct iphdr*)(packet+header_len);
        if ((iph->ihl>>4)!=4 || iph->protocol!=IPPROTO_TCP) {
            return;
        }
    }
}

ttl=iph->ttl;

off=ntohs(iph->off);
df=((off&IP_DF)!=0);
sip.s_addr=iph->saddr;
dip.s_addr=iph->daddr;
ilen= ( (iph->ihl&0x0f) );

switch (ilen) {
    case 5: /* no options */
        tcph=(struct tcphdr*)(iph+1);

```

```

    break;
default: /* parse ipoptions */
    if ((header_len+(ilen<<2)+sizeof(struct tcphdr)) > pph->len) {
        return;
    }
    tcph=(struct tcphdr *)(packet+header_len+(ilen<<2));
    break;
}

off=tcph->th_flags;
if (!(off&TH_SYN)) return;
if ((off&TH_ACK)) return;

wscale=-1;
mss=0;
nop=0;
sok=0;

hlen=(tcph->th_off)*4;

{
    void* opt_ptr;
    int opt;
    opt_ptr=(void*)tcph+sizeof(struct tcphdr);
    while (dupa<hlen) {
        opt=(int)((u_char*)(opt_ptr+dupa));
        dupa+=1;
        switch(opt) {
            case TCPOPT_EOL:
                dupa=100000; break; // Abandon ship!
            case TCPOPT_NOP:
                nop=1;
                break;
            case TCPOPT_SACKOK:
                sok=1;
                dupa++;
                break;
            // Long options....
            case TCPOPT_MAXSEG:
                dupa++;
                mss=EXTRACT_16BITS(opt_ptr+dupa);
                dupa+=2;
                break;
            case TCPOPT_WSCALE:
                olen=(int)*((char*)opt_ptr+dupa)-2; dupa++;
                if (olen<0) olen=0;
                wscale=(int)*((u_char*)opt_ptr+dupa);
                dupa+=olen;
                break;
            case TCPOPT_TIMESTAMP:

```

```

        olen=(int)*((char*)opt_ptr+dupa)-2; dupa++;
        if (olen<0) olen=0;
        dupa+=olen;
        break;
    default:
        olen=(int)*((char*)opt_ptr+dupa)-2; dupa++;
        if (olen<0) olen=0;
        dupa+=olen;
        break;
    }
}
}
}
#endif
#if BYTE_ORDER == LITTLE_ENDIAN
    sp=htons(tcph->th_sport);
    dp=htons(tcph->th_dport);
    wss=htons(tcph->th_win);
#else
    sp=tcph->th_sport;
    dp=tcph->th_dport;
    wss=tcph->th_win;
#endif
    lookup();
    return;
}

void lookup(void) {
    int i=0,got=0,down=0;
    int origw=wscale;
    char buf[INBUF],*p;
    char* plonked="\n";
    plonk:
    for (down=0;down<TTLDW;down++) {
        i=0;
        sprintf(buf,"%d:%d:%d:%d:%d:%d:
%d:",wss,ttl+down,mss,df,wscale,sok,nop);
        while (fps[i][0]) {
            if (!strncmp(buf, fps[i], strlen(buf))) {
                got=1;
                p=strrchr(fps[i],':')+1;
                if (strchr(p, '\n')) p[strlen(p)-1]=0;
                printf("%s [%d hops]: %s%s",inet_ntoa(sip),down+1,p,plonked);
                if (verbose) {
                    printf(" + %s:%d ->",inet_ntoa(sip),sp);
                    printf(" %s:%d\n", inet_ntoa(dip),dp);
                }
                break;
            }
            i++;
        }
    }
}

```

```

    if (got) break;
}
if (!got) if (wscale==-1) { plonked=" *\n";wscale=0; goto plonk; }
if (!got) printf("%s: UNKNOWN [%d:%d:%d:%d:%d:%d:%d].\n",
    inet_ntoa(sip), wss, ttl, mss, df, origw, sok, nop);
fflush(0);
}

```

```
int fips;
```

```

void load_fprints(char *filename) {
    FILE *x;
    int i=0;
    char *p;
    x=fopen(filename, "r");
    if (!x) x=fopen("p0f.fp", "r");
    if (!x) {
        fprintf(stderr, "No OS fingerprint database (%s) found. Dumb mode
on.\n",
            filename);
        return;
    }
    while (fgets(fps[i],FPBUF-1,x)) {
        if ((p=strchr(fps[i], '#')) *p=0;
            if (fps[i][0]) i++;
        }
        fips=i;
        fclose(x);
    }
}

```

```
char *ifa,*rul;
```

```

void usage(char* what) {
    fprintf(stderr,"p0f: %s\n",what);
    fprintf(stderr,"usage: p0f [ -f file ] [ -i device ] [ -s file ]
[ -v ][ 'filter rule' ]\n");
    fprintf(stderr, " -f file    read fingerprint information from
file\n");
    fprintf(stderr, " -i device  read packets from device\n");
    fprintf(stderr, " -s file    read packets from file\n");
    fprintf(stderr, " -v        verbose mode\n");
    exit(1);
}

```

```

int main(int argc, char *argv[]) {
    char errbuf[PCAP_ERRBUF_SIZE];
    char *filename = NULL, *inputfile = NULL;
    int r, s = 0;
}

```

```

while ((r = getopt(argc, argv, "f:i:s:v")) != -1) {
    switch (r) {
        case 'f':
            filename = optarg;
            break;
        case 'i':
            ifa = optarg;
            break;
        case 's':
            s = 1;
            inputfile = optarg;
            break;
        case 'v':
            verbose = 1;
            break;
        default:
            usage("Unknown option.");
    }
}

/* set a reasonable default fingerprint file */
if (!filename || !*filename)
    filename = "/etc/p0f.fp";

/* anything left after getopt'ing is a rule */
if (argv[optind] && *(argv[optind]))
    rul = argv[optind];

if (!ifa) ifa=pcap_lookupdev(errbuf);
if (!ifa) { ifa="lo"; }

fprintf(stderr, "p0f: passive os fingerprinting ver. " VER " by
<lcamtuf@tpi.pl>\n");

if (s && inputfile && *inputfile) {
    if ((pt=pcap_open_offline(inputfile, errbuf))==NULL) {
        fprintf(stderr, "pcap_open_offline failed: %s\n", errbuf);
        exit(1);
    }
} else {
    if ((pt=pcap_open_live(ifa,100,1,100,errbuf))==NULL) {
        fprintf(stderr, "pcap_open_live failed: %s\n", errbuf);
        exit(1);
    }
}

signal(SIGINT,&die_nicely);
signal(SIGTERM,&die_nicely);
load_fprints(filename);

```

```
if (pcap_compile(pt, &flt, rul?rul:"", 1, 0)) {
    if (rul) {
        pcap_perror(pt,"pcap_compile");
        exit(1);
    }
}

if (!rul) rul="all";
fprintf(stderr,"p0f: file: '%s', %d fprints, iface: '%s', rule:
'%s'.\n",filename,fips,ifa,rul);

pcap_setfilter(pt, &flt);

pcap_loop(pt,-1,(pcap_handler)&parse,(void*)0L);
return 0; //not reached;>
}
```

```
#
# p0f - passive OS fingerprinting
#
# Valid entry describes the way server starts TCP handshake (first
# SYN).
# Important options are: window size (wss), maximum segment size
# (mss),
# don't fragment flag (DF), window scaling (wscale), sackOK flag, nop
# flag, and initial time to live (TTL) ;)
#
# How can you determine initial ttl? Well, usually it's first power of
# 2
# bigger than TTL returned in scan. So, for example, if you get TTL 55
# in
# fingerprint returned by p0f, initial TTL will be usually 64... NOTE:
# it's better to overestimate initial TTL than underestimate it ;)
#
# There are some brain-damaged devices, like network printers etc,
# that
# have stupid initial TTLs like 60, but who cares, if HP LaserJet
# wants to
# visit your server, you have to think again about your life ;)
#
# Format:
#
# www:ttt:mmm:D:W:S:N:OS Description
#
# www - window size
# ttt - time to live
# mmm - maximum segment size
# D - don't fragment flag (0=unset, 1=set)
# W - window scaling (-1=not present, other=value)
# S - sackOK flag (0=unset, 1=set)
# N - nop flag (0=unset, 1=set)
#
31072:64:3884:1:0:1:1:Linux 2.2.12-20 (RH 6.1)
512:64:1460:0:0:0:0:Linux 2.0.38
32120:64:1460:1:0:1:1:Linux 2.2.14 or Cobalt Linux 2.2.12C3
16384:64:1460:1:0:0:0:FreeBSD 4.0-STABLE, 3.2-RELEASE
8760:64:1460:1:0:0:0:Solaris 2.6 (2)
9140:255:9140:1:0:0:0:Solaris 2.6 (sunsite)
49152:64:1460:0:0:0:0:IRIX 6.5 / 6.4
8760:255:1460:1:0:0:0:Solaris 2.6 or 2.7
8192:128:1460:1:0:0:0:Windows NT 4.0
8192:128:1460:1:0:1:1:Windows 9x (1)
8192:128:536:1:0:1:1:Windows 9x (2)
2144:64:536:1:0:1:1:Windows 9x (4)
16384:128:1460:1:0:1:1:Windows 2000
32120:32:1460:1:0:1:1:Linux 2.2.13
```

8192:32:1460:1:0:0:0:Windows NT 4.0
5840:128:536:1:0:1:1:Windows 95 (3)
16060:64:1460:1:0:1:1:Debian/Caldera Linux 2.2.x (check)
8760:255:1380:1:0:0:0:Solaris 2.7
8192:128:1456:1:0:1:1:Linux 2.2.13
32768:64:1432:0:0:0:0:??? (PlusGSM, InterNetia proxy)
16384:255:1460:1:0:0:1:FreeBSD 2.2.6-RELEASE
8192:64:1460:1:0:0:1:BSDI BSD/OS 3.1
16384:64:1460:0:0:0:1:NetBSD 1.3/i386
24820:64:1460:1:0:0:0:SCO UnixWare 7.0.1
32768:64:1460:1:0:0:0:HP-UX B.10.01 A 9000/712
16384:64:512:0:0:0:0:AIX 3.2, 4.2 - 4.3
32768:64:1460:1:0:0:1:Digital UNIX V4.0E
32694:255:536:0:0:0:0:3Com HiPer ARC, System V4.2.32
4128:255:556:0:0:0:0:Cisco 1750 IOS 12.0(5), Cisco 2500 IOS 11.3(1)
4128:255:556:0:0:0:0:Cisco 3600 IOS Version 12.0(7)
4288:255:1460:0:-1:0:0:0:Cisco 3620 IOS 11.2(17)P
512:64:0:0:-1:0:0:0:Linux 2.0.35 - 2.0.37
8192:128:1460:1:-1:1:0:0:Windows NT
32120:64:1460:1:190:1:1:Linux 2.2.16
32696:64:536:0:0:1:1:SCO UnixWare 7.1.0 x86
24820:64:1460:1:0:0:1:SCO UnixWare 7.1.0 x86
32120:58:1460:0:-1:0:0:0:Linux 2.0.38
65535:128:1368:1:-1:0:0:0:BorderManager 3.5
33580:255:1460:1:-1:0:0:0:Solaris 7
8192:128:25443:1:-1:1:1:Microsoft NT 4.0 Server SP5
8192:64:1460:1:-1:0:0:0:AXCENT Raptor Firewall Windows NT 4.0/SP3
8192:32:1456:1:-1:0:0:0:Windows 95 (?)
16384:64:0:0:-1:0:0:0:ULTRIX V4.5 (Rev. 47)
16384:64:512:0:0:0:1:OpenBSD 2.6
32768:128:1460:1:-1:0:0:0:Novell NetWare 4.11
16384:64:1460:1:0:0:1:FreeBSD 2.2.8-RELEASE
4288:255:536:0:-1:0:0:0:Cisco 1600 IOS 11.2(15)P
4096:32:1024:0:245:0:0:Alcatel (Xylan) OmniStack 5024
4288:255:536:0:-1:0:0:0:Cisco 2500 IOS 11.2(5)P or Cisco 4500 IOS
11.1(7)
2144:255:536:0:-1:0:0:0:Cisco IGS 3000 IOS 11.x(16), 2500 IOS 11.2(3)P
4128:255:1460:0:-1:0:0:0:Cisco 2611 IOS 11.3(2)XA4
61440:64:1460:0:-1:0:0:0:IRIX 6.3
61440:64:512:0:-1:0:0:0:IRIX 5.3 / 4.0.5F
4128:255:1460:0:-1:0:0:0:Cisco C2600 IOS 12.0(5)T1
31856:64:1460:1:0:1:1:Linux 2.4.0-test1
4096:30:1024:0:245:0:0:Alcatel (Xylan) OmniStack 5024 v3.4.5
4096:30:1024:0:-1:0:0:0:Chorus MiX V.3.2 r4.1.5 COMP-386
4128:255:1460:0:-1:0:0:0:Cisco 4500 IOS 12.0(9), 3640 12.1(2), 3620
12.0(8) or 11.3(11a)
32120:64:1460:1:101:1:1:Linux 2.2.15
32120:64:1460:0:-1:0:0:0:Linux 2.0.33
512:64:1460:0:52:0:0:0:Linux 2.0.33

--=--
p.0.f
--=--

"Dr. Jekyll had something to Hyde"

passive OS fingerprinting tool
version 1.7 <lcamtuf@tpi.pl>

-= buffer0verfl0w security team -=

<http://lcamtuf.hack.pl/p0f.tgz>

Special thanks to:

- * Lance Spitzner for whitepaper on passive OS fingerprinting:
<http://www.enteract.com/~lspitz/finger.html>
- * tf8 for initial piece of libpcap support and packet parsing
- * teso/security.is/b0f/#hax for ideas and testing
- * Jeremy Weatherford, Chris Wilson and Szilveszter Adam for portability testing/patches, bugfixes and ideas,
- * other BUGTRAQ readers for OS fingerprints and useful patches
- * other people involved (or not) in this project
- * very, very special thanks to el- :*

Background:

- * What is passive OS fingerprinting?

Passive OS fingerprinting technique bases on information coming from remote host when it establishes connection to our system.

Captured

packets contains enough information to determine OS – and, unlike active scanners (nmap, queS0) – without sending anything to this host.

If you're looking for more information, read Spitzner's text at:
<http://www.enteract.com/~lspitz/finger.html>

- * How it works?

Well, there are some TCP/IP flag settings specific for given systems.

Usually initial TTL (8 bits), window size (16 bits), maximum segment size

(16 bits), don't fragment flag (1 bit), sackOK option (1 bit), nop option

(1 bit) and window scaling option (8 bits) combined together gives

unique,
51-bit signature for every system.

* What are main advantages?

Passive OS fingerprinting can be done on huge portions of input data
– eg.
information gathered on firewall, proxy, routing device or Internet
server,
without causing any network activity. You can launch passive OS
detection
software on such machine and leave it for days, weeks or months,
collecting
really interesting statistical and – *erm* – just interesting
information.
What's really funny – packet filtering firewalls, network address
translation and so on are transparent to p0f-alike software, so
you're able
to obtain information about systems behind the firewall. Also, such
software
can determine distance between remote host and your system, allowing
you to
generate network structure maps for firewalled/structural networks.
And
all without sending _any_ packet. Just think about IDS systems.

Limitations

Proxy firewalls and other high-level proxy devices are not
transparent to
any tcp fingerprinting software. It applies to p0f, as well.

In order to obtain information required for fingerprinting, you have
to
receive at least one SYN packet initializing TCP connection to your
machine or network. Note: you don't have to respond to particular
SYN.
Of course, it's impossible to perform any kind of OS detection
without
receiving any information.

It is possible to perform fingerprinting on alive TCP connection or
even when connection is initialized from your network. But these
techniques
are less reliable (as in many implementations some parameters are
copied
from first SYN packet, so if connection has been initialized from
our
network, fingerprinting won't be successful; also, some parameters
like

window size are constant for initial TCP/IP packet, but changing rapidly later).

Why our bubble gum is better?

There is another passive OS detection utility, called 'siphon'. It's pretty good piece of proof-of-concept software, but it isn't perfect. Well, p0f isn't perfect for sure, but has several improvements:

- it's single-threaded and pretty clean,
- works properly on Linuxes (siphon has a problem with bpf on 2.2), as well as on BSD systems and SunOS/Solaris,
- has pretty large and detailed fingerprints database,
- uses more information for fingerprinting (26 extra bits),
- it's more accurate,
- you can define your own filtering rules in the tcpdump flavour: p0f 'src host 1.2.3.4' or p0f 'gateway 1.2.3.4 and port 80', and listening interface (using option -i).

What more? Dunno :) Simply, check it out.

Not working!

Probably p0f isn't working well on every platform in the world; first

of all, you'll need libpcap 0.4 or newer; sometimes pcap.h is placed in /usr/include/pcap instead of /usr/include/ (eg. in broken RH 6.1 package).

In this case, simply issue:

```
ln -s /usr/include/pcap/pcap.h /usr/include/  
ln -s /usr/include/pcap/net/bsf.h /usr/include/net/
```

NOTE: if p0f recognized system incorrectly or cannot recognize it at all, please send OS signature and system description to author. Thanks :)

Tested platforms:

- NetBSD
- FreeBSD

- OpenBSD
- Linux 2.0/2.2
- Solaris 2.6-2.7

Requires: libpcap 0.4 or newer; GNU cc 2.7.x or newer; GNU make 3.7x;

GNU egrep (for proper Makefile processing)

Files:

/etc/p0f.fp or ./p0f.fp - OS fingerprints database. Format is described inside:

```
# Valid entry describes the way server starts TCP handshake (first SYN).
# Important options are: window size (wss), maximum segment size (mss),
# don't fragment flag (DF), window scaling (wscale), sackOK flag,
nop
# flag, and initial time to live (TTL) ;)
#
# How can you determine initial ttl? Well, usually it's first power
of 2
# bigger than TTL returned in scan. So, for example, if you get TTL
55 in
# fingerprint returned by p0f, initial TTL will be usually 64...
NOTE:
# it's better to overestimate initial TTL than underestimate it ;)
#
# There are some brain-damaged devices, like network printers etc,
that
# have stupid initial TTLs like 60, but who cares, if HP LaserJet
wants to
# visit your server, you have to think again about your life ;)
#
# Format:
#
# www:ttt:mmm:D:W:S:N:OS Description
#
# www - window size
# ttt - time to live
# mmm - maximum segment size
# D   - don't fragment flag (0=unset, 1=set)
# W   - window scaling (-1=not present, other=value)
# S   - sackOK flag (0=unset, 1=set)
# N   - nop flag (0=unset, 1=set)
```

What to do?

- COLORFUL INTERFACE :))))

License, disclaimer:

The p0f utility and related utilities are free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS

OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL

MICHAL ZALEWSKI, OR ANY OTHER CONTRIBUTORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR

OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-- Michal Zalewski <lcamtuf@tpi.pl>

```

#include <sys/types.h>

#ifdef _SUN_
typedef      unsigned char u_int8_t;
#endif

#define TCPOPT_EOL           0
#define TCPOPT_NOP           1
#define TCPOPT_MAXSEG        2
#define TCPOPT_WSCALE        3
#define TCPOPT_SACKOK        4
#define TCPOPT_TIMESTAMP     8

#define EXTRACT_16BITS(p) \
    ((u_short)*((u_char *) (p) + 0) << 8 | \
     (u_short)*((u_char *) (p) + 1))

#define IP_DF    0x4000 /* dont fragment flag */
#define IP_MF    0x2000 /* more fragments flag */

#define TH_FIN    0x01
#define TH_SYN    0x02
#define TH_RST    0x04
#define TH_PUSH   0x08
#define TH_ACK    0x10
#define TH_URG    0x20

struct iphdr {
    u_char  ihl;
    u_char  tos;          /* type of service */
    short   tot_len;       /* total length */
    u_short id;           /* identification */
    short   off;           /* fragment offset field */
    u_char  ttl;          /* time to live */
    u_char  protocol;     /* protocol */
    u_short check;        /* checksum */
    u_long   saddr;
    u_long   daddr;       /* source and dest address */
};

typedef u_long  tcp_seq;

struct tcphdr {
    u_short th_sport;      /* source port */
    u_short th_dport;      /* destination port */
    tcp_seq th_seq;        /* sequence number */
    tcp_seq th_ack;        /* acknowledgement number */
#ifdef BYTE_ORDER == LITTLE_ENDIAN
    u_int8_t th_x2:4;      /* (unused) */
    u_int8_t th_off:4;     /* data offset */

```

```
#else /* __BYTE_ORDER == __BIG_ENDIAN */
    u_int8_t th_off:4;          /* data offset */
    u_int8_t th_x2:4;          /* (unused) */
#endif
    u_char  th_flags;
    u_short th_win;             /* window */
    u_short th_sum;             /* checksum */
    u_short th_urp;             /* urgent pointer */
};
```