Code a network packet sniffer in PHP

PHP By Silver Moon On Nov 24, 2011 10 Comments 3 1 Tweet 2

Example of a packet sniffer is Wireshark. Packet sniffers pick up the packets going in and out of a system and analyse them and present them to user for further analysis.

Network Forensics

www.npulsetech.com

Real Forensics for a 10 Gig World Ultrafast Packet Capture & Analysis

In this post we are going to code a simple packet sniffer in php. The basic theory this packet sniffer is that , raw packets can sniff without much effort if they are put into receiving mode. This will work only on a Linux system , since we cannot create raw sockets on windows.

So the steps are :

1. Create a raw socket
2. Receive on it.
3. See what you received.

Code

```
<?php
/**
        Packet sniffer in PHP
        Will run only on Linux
        Needs root privileges , so use sudo !!
*/

error_reporting(~E_ALL);

//Create a RAW socket
$socket = socket_create(AF_INET , SOCK_RAW , SOL_TCP);
if($socket)
{
        echo "Starting sniffing...\n";
        while(true)
        {
```

Google™ Custom Search

SEARCH

SevOne PERFORMANCE VISIBILITY MONITOR DATACENTERS SERVERS, SWITCHES, LOAD BALANCES & MORE ALL-IN-ONE SOLUTION SOLVE PROBLEMS TODAY! DOWNLOAD

Download 10 Free Linux Ebooks

Related Posts

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        //Start receiving on the raw socket
        socket_recv ( $socket , &$buf , 65536 , 0 );

                //Process the packet
        process_packet($buf);
        }
}

//Some error - check that you used sudo !!
else
{
        $error_code = socket_last_error();
        $error_message = socket_strerror($error_code);

        echo "Could not create socket : [$error_code] $error_message";
}

/**
        Process the captured packet.
*/
function process_packet($packet)
{
        //IP Header
        $ip_header_fmt = 'Cip_ver_len/'
        .'Ctos/'
        .'ntot_len/'
        .'nidentification/'
        .'nfrag_off/'
        .'Cttl/'
        .'Cprotocol/nheader_checksum/Nsource_add/Ndest_add/';

        //Unpack the IP header
        $ip_header = unpack($ip_header_fmt , $packet);

        if($ip_header['protocol'] == '6')
        {
        print_tcp_packet($packet);
        }
}

/*
  Process a TCP Packet :)
*/
function print_tcp_packet($packet)
{
```


Output buffering in php and apache

Socket programming with streams in php

Run php code online

Convert simplexml object to array in php

PHP get adsense earnings and reports

PHP get DNS Nameserver , Cname , MX , A records of a domain

PHP strtotime in 64 bit environment

Be a Fan

Binarytides

Like

5,338 people like Binarytides.

Facebook social plugin

Circle us

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
$ip_header_fmt = 'Cip_ver_len/'
.'Ctos/'
.'ntot_len/';

$p = unpack($ip_header_fmt , $packet);
$ip_len = ($p['ip_ver_len'] & 0x0F);

if($ip_len == 5)
{

        //IP Header format for unpack
        $ip_header_fmt = 'Cip_ver_len/'
        .'Ctos/'
        .'ntot_len/'
        .'nidentification/'
        .'nfrag_off/'
        .'Cttl/'
        .'Cprotocol/'
        .'nip_checksum/'
        .'Nsource_add/'
        .'Ndest_add/';
}
else if ($ip_len == 6)
{
        //IP Header format for unpack
        $ip_header_fmt = 'Cip_ver_len/'
        .'Ctos/'
        .'ntot_len/'
        .'nidentification/'
        .'nfrag_off/'
        .'Cttl/'
        .'Cprotocol/'
        .'nip_checksum/'
        .'Nsource_add/'
        .'Ndest_add/'
        .'Noptions_padding/';
}

$tcp_header_fmt = 'nsource_port/'
.'ndest_port/'
.'Nsequence_number/'
.'Nacknowledgement_number/'
.'Coffset_reserved/';

//total packet unpack format
```

BT

BinaryTides

google.com/+binarytides

How to guides, latest news and other interesting stuff on Linux, Ubuntu and more

Follow +1

+ 6,916

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
$total_packet = $ip_header_fmt.$tcp_header_fmt.'H*data';

$p = unpack($total_packet , $packet);
$tcp_header_len = ($p['offset_reserved'] >> 4);

if($tcp_header_len == 5)
{
        //TCP Header Format for unpack
        $tcp_header_fmt = 'nsource_port/'
        .'ndest_port/'
        .'Nsequence_number/'
        .'Nacknowledgement_number/'
        .'Coffset_reserved/'
        .'Ctcp_flags/'
        .'nwindow_size/'
        .'nchecksum/'
        .'nurgent_pointer/';
}
else if($tcp_header_len == 6)
{
        //TCP Header Format for unpack
        $tcp_header_fmt = 'nsource_port/'
        .'ndest_port/'
        .'Nsequence_number/'
        .'Nacknowledgement_number/'
        .'Coffset_reserved/'
        .'Ctcp_flags/'
        .'nwindow_size/'
        .'nchecksum/'
        .'nurgent_pointer/'
        .'Ntcp_options_padding/';
}

//total packet unpack format
$total_packet = $ip_header_fmt.$tcp_header_fmt.'H*data';

//unpack the packet finally
$packet = unpack($total_packet , $packet);

//prepare the unpacked data
$sniff = array(

        'ip_header' => array(
                'ip_ver' => ($packet['ip_ver_len'] >> 4) ,
                'ip_len' => ($packet['ip_ver_len'] & 0x0F) ,
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
                        'tos' => $packet['tos'] ,
                        'tot_len' => $packet['tot_len'] ,
                        'identification' => $packet['identification'] ,
                        'frag_off' => $packet['frag_off'] ,
                        'ttl' => $packet['ttl'] ,
                        'protocol' => $packet['protocol'] ,
                        'checksum' => $packet['ip_checksum'] ,
                        'source_add' => long2ip($packet['source_add']) ,
                        'dest_add' => long2ip($packet['dest_add']) ,
                ) ,

                'tcp_header' => array(
                        'source_port' => $packet['source_port'] ,
                        'dest_port' => $packet['dest_port'] ,
                        'sequence_number' => $packet['sequence_number'] ,
                        'acknowledgement_number' => $packet['acknowledgement_number'] ,
                        'tcp_header_length' => ($packet['offset_reserved'] >> 4) ,

                        'tcp_flags' => array(
                                'cwr' => (($packet['tcp_flags'] & 0x80) >> 7) ,
                                'ecn' => (($packet['tcp_flags'] & 0x40) >> 6) ,
                                'urgent' => (($packet['tcp_flags'] & 0x20) >> 5 ) ,
                                'ack' => (($packet['tcp_flags'] & 0x10) >>4) ,
                                'push' => (($packet['tcp_flags'] & 0x08)>>3) ,
                                'reset' => (($packet['tcp_flags'] & 0x04)>>2) ,
                                'syn' => (($packet['tcp_flags'] & 0x02)>>1) ,
                                'fin' => (($packet['tcp_flags'] & 0x01)) ,
                        ) ,

                        'window_size' => $packet['window_size'] ,
                        'checksum' => $packet['checksum'] . ' [0x'.dechex($packet['checksum']).']',
                ) ,

                'data' => hex_to_str($packet['data'])
        );

        //print the unpacked data
        print_r($sniff);
}

/*
        idea taken from http://ditio.net/2008/11/04/php-string-to-hex-and-hex-to-string-functions/
        modified a bit to show non alphanumeric characters as dot.
*/
function hex_to_str($hex)
```

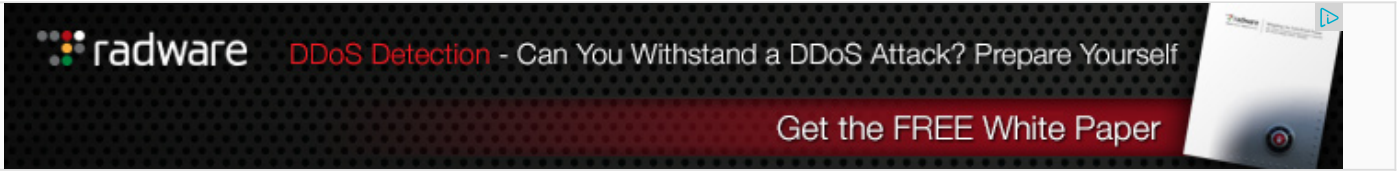
open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
{
    $string='';

    for ($i=0; $i < strlen($hex)-1; $i+=2)
    {
        $d = hexdec($hex[$i].$hex[$i+1]);

        //Show only if number of alphabet
        if( ($d >= 48 and $d <= 57) or ($d >= 65 and $d <= 90) or ($d >= 97 and $d <= 122) )
        {
                $string .= chr(hexdec($hex[$i].$hex[$i+1]));
        }
        else
        {
                $string .= '.';
        }
    }

    return $string;
}
```



The script needs to run with root privileges. So on ubuntu you can run like this :
sudo php sniffer.php

Analysis

This is what creates a raw socket

```
$socket = socket_create(AF_INET , SOCK_RAW , SOL_TCP);
```

SOCK_RAW means raw.

No start receiving on the raw socket.

```
while(true)
        {
        //Start receiving on the raw socket
        socket_recv ( $socket , &$buf , 65536 , 0 );

                //Process the packet
        process_packet($buf);
```

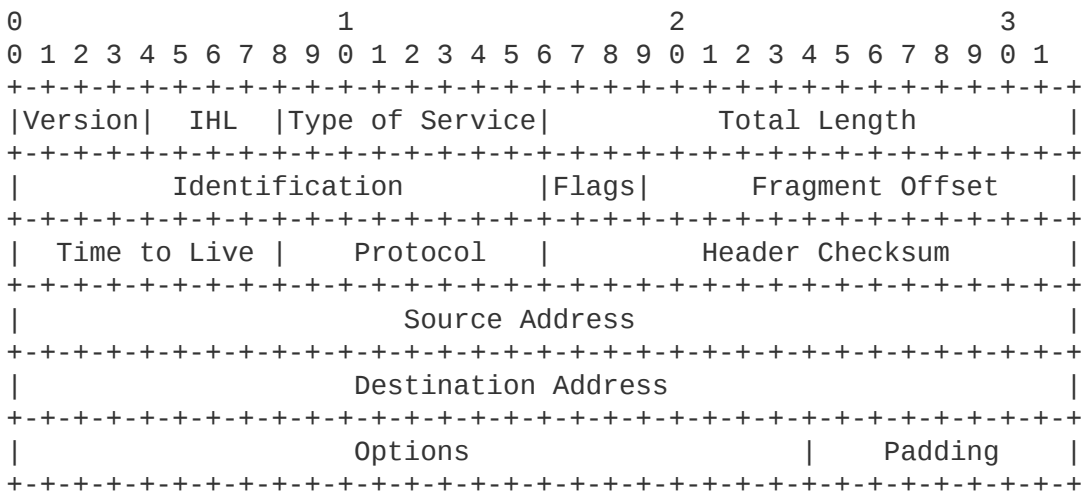
open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        }
```

The socket_recv receives some passing by packet in $buf. Then process_packet will analyse the packet which is in $buf.

The structure of a typical TCP packet is like this : IP Header + TCP Header + Data
So the packet needs to be broken down into the relevant parts.

IP Header
According to RFC 791 the IP headers looks like this :



The IHL (IP Header Length) contains the length of the ip header / 4. So if its value is 5 then header length is 20 bytes. Generally header length is 5 and options and padding is not there. If options and padding is present then length would be 6. This has to be checked first before moving to the next part that is TCP header.

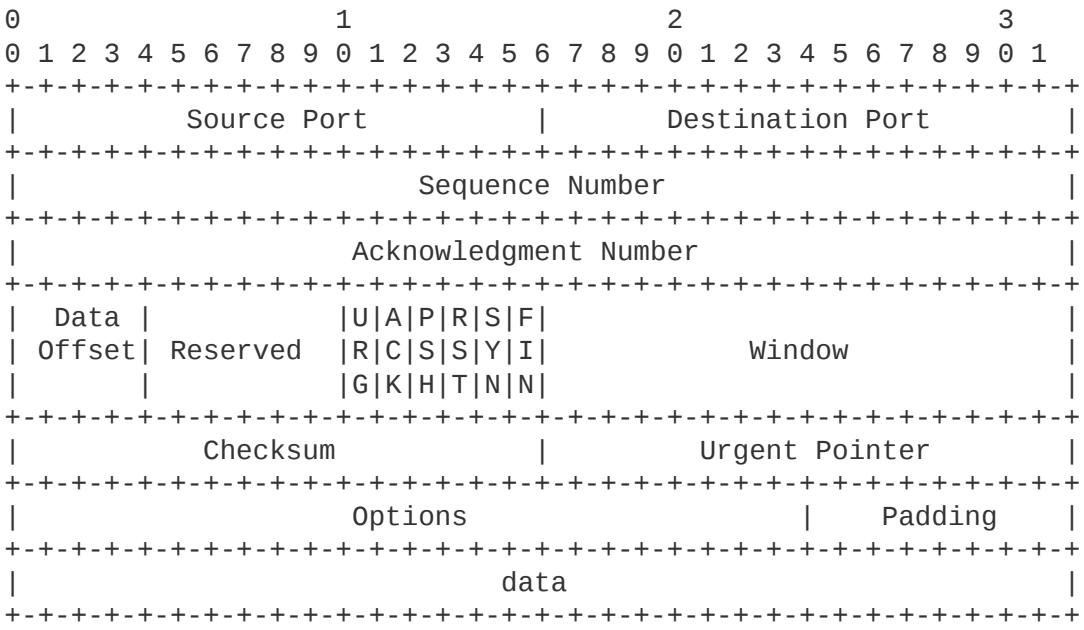
Since the $buf contains data in binary format , it needs to be unpacked using the unpack function of php. The unpack string should be:

```
//IP Header format for unpack
                $ip_header_fmt = 'Cip_ver_len/'
                .'Ctos/'
                .'ntot_len/'
                .'nidentification/'
                .'nfrag_off/'
                .'Cttl/'
                .'Cprotocol/'
                .'nip_checksum/'
                .'Nsource_add/'
                .'Ndest_add/';
```

For TCP packets the value of protocol is 6.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

TCP headers look like this :



The TCP header also has an options and padding part. The TCP header length is present in the Data Offset part and needs to be multiplied by 4 to get the length in bytes.

They can be unpacked with this string :

```
//TCP Header Format for unpack
                $tcp_header_fmt = 'nsource_port/'
                .'ndest_port/'
                .'Nsequence_number/'
                .'Nacknowledgement_number/'
                .'Coffset_reserved/'
                .'Ctcp_flags/'
                .'nwindow_size/'
                .'nchecksum/'
                .'nurgent_pointer/';
```

The print_tcp_packet breaks down the tcp packet into its parts and saves them in a array with proper key names for analysis.

```
//total packet unpack format
        $total_packet = $ip_header_fmt.$tcp_header_fmt.'H*data';

        //unpack the packet finally
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
$packet = unpack($total_packet , $packet);

//prepare the unpacked data
$sniff = array(

        'ip_header' => array(
                'ip_ver' => ($packet['ip_ver_len'] >> 4) ,
                'ip_len' => ($packet['ip_ver_len'] & 0x0F) ,
                'tos' => $packet['tos'] ,
                'tot_len' => $packet['tot_len'] ,
                'identification' => $packet['identification'] ,
                'frag_off' => $packet['frag_off'] ,
                'ttl' => $packet['ttl'] ,
                'protocol' => $packet['protocol'] ,
                'checksum' => $packet['ip_checksum'] ,
                'source_add' => long2ip($packet['source_add']) ,
                'dest_add' => long2ip($packet['dest_add']) ,
        ) ,

        'tcp_header' => array(
                'source_port' => $packet['source_port'] ,
                'dest_port' => $packet['dest_port'] ,
                'sequence_number' => $packet['sequence_number'] ,
                'acknowledgement_number' => $packet['acknowledgement_number'] ,
                'tcp_header_length' => ($packet['offset_reserved'] >> 4) ,

                'tcp_flags' => array(
                        'cwr' => (($packet['tcp_flags'] & 0x80) >> 7) ,
                        'ecn' => (($packet['tcp_flags'] & 0x40) >> 6) ,
                        'urgent' => (($packet['tcp_flags'] & 0x20) >> 5 ) ,
                        'ack' => (($packet['tcp_flags'] & 0x10) >>4) ,
                        'push' => (($packet['tcp_flags'] & 0x08)>>3) ,
                        'reset' => (($packet['tcp_flags'] & 0x04)>>2) ,
                        'syn' => (($packet['tcp_flags'] & 0x02)>>1) ,
                        'fin' => (($packet['tcp_flags'] & 0x01)) ,
                ) ,

                'window_size' => $packet['window_size'] ,
                'checksum' => $packet['checksum'] . ' [0x'.dechex($packet['checksum']).']',
        ) ,

        'data' => hex_to_str($packet['data'])
);
```

Similar piece of code can be written for UDP packets , ICMP packets and so on.

The C version of the same code can be found here : http://www.binarytides.com/blog/packet-sniffer-code-in-c-using-linux-

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

sockets-bsd/.

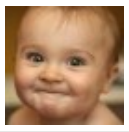
Last Updated On : 4th August 2012

php socket programming

Subscribe to get updates delivered to your inbox Enter email to subscribe Subscribe

Related Posts

- Code a network packet sniffer in python for Linux
- Packet Sniffer Code in C using Winsock
- Code a packet sniffer in C with winpcap
- Packet Sniffer Code in C using Linux Sockets (BSD) – Part 2
- Code raw sockets in C on Linux



About **Silver Moon**

Php developer, blogger and Linux enthusiast. He can be reached at m00n.silv3r@gmail.com. Or find him on Google+

10 Comments BinaryTides Log

Sort by Best Share Favorit

Join the discussion…

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Jerome • a month ago

Hi Silver moon,

I tried your PHP sniffer for TCP connections and it works fine.
I would like to get a PHP sniffer to be able to get IGMP messages (not data) : like IGMP join, leave, issued on 224.0.0.22 but other IGMP messages as well.

So I modified your code this way :

```
$prot = getprotobyname('igmp');
echo "protocole : $prot\n";
$socket = socket_create(AF_INET , SOCK_RAW , $prot);
$address = '224.0.0.22';
$tab_mcast = array("group" => $address, "interface" => 0);
socket_set_option($sock, getprotobyname('ip'), MCAST_JOIN_GROUP, $tab_mcast);

if($socket)
{
echo "Starting sniffing...\n";
while(true)
```

see more

Reply • Share ›

shubham • a year ago

Call-time pass-by-reference has been removed

Reply • Share ›

smrtl • 2 years ago

Hello sir,

Your code hold my attention so i tried it. While encountering no problem to start it and get the "Starting sniffing...\r output I did not manage to capture any packet. Am I understanding the thing wrongly if I think this sniffer should be able to capture the packets sent and received when a page is loaded in a browser on my machine ?
I am wondering if the socket may be bound to the wrong interface (like lo0) thus receiving nothing.
Any idea ?

Thanks for sharing this

Reply • Share ›

Silver Moon Mod ➔ smrtl • 2 years ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

yes, the sniffer will capture packets that are received by the browser.
the program should be run with root privileges (sudo on ubuntu for example).
without root it will not be able to capture packets, because it uses raw sockets which require root privilege
if the interface is correct, the program should pickup packets.

• Reply • Share ›

smrtl ➚ Silver Moon • 2 years ago

yes the script ran with root privileges through sudo (that's why i got the start sniffing output and no errors). My question is how do I select the interface ? How do I know with interface the script reads from ?
The problem is not a privilege thing or something related to PHP i think because after compiling ar running the same script in C (posted on this blog too) I got the same result. That is everything is s up correctly but no packet is captured...

(note: i'm trying that on an OS X box; PHP 5.3.10 PHP 5.3.10 with Suhosin-Patch (cli))

• Reply • Share ›

Silver Moon Mod ➚ smrtl • 2 years ago

selecting a particular device/interface might not be possible in php.

in c it can be done like this :
setsockopt(sock_raw , SOL_SOCKET , SO_BINDTODEVICE , "eth0" , strlen("eth0")+ 1);

but the documentation at
http://www.php.net/manual/en/f...
does not define any such option as SO_BINDTODEVICE

you have to experiment to find whether it can be made to work or not.
I have not tested the code on os x.

• Reply • Share ›

smrtl ➚ Silver Moon • 2 years ago

Thanks a lot for your quick answer and for the SO_BINDTODEVICE explanation. It gave me access to a brand new field of google results :)

FYI, the SO_BINDTODEVICE option is not defined on *BSD systems (as Darwin/OS X). There is actually no way (unless the socket is multicast) to determine the interface it is bound to. This seems to be made automatically based on the routing table.

But the reason for getting nothing does not come from a wrong interface problem....

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Quoting a post from http://stackoverflow.com/quest... :
"
FreeBSD takes another approach. It *never* passes TCP or UDP packets to raw sockets. Such packets need to be read directly at the datalink layer by using libraries like libpcap or the bpf API. It also *never* passes any fragmented datagram. Each datagram has to be completeley reassembled before it is passed to a raw socket.
FreeBSD passes to a raw socket:
a) every IP datagram with a protocol field that is not registered in the kernel

see more

• Reply • Share ›

Silver Moon Mod → smrtl • 2 years ago
thats good information.
then maybe libpcap has to be tried.

• Reply • Share ›

Ventre • 2 years ago
Thanks for posting this!

I am trying to run it, but can't create the socket. How do you specify to run it as sudo to view/run the script in a browser (I have it saved in: htdocs/sniff/sniffer.php)? Your suggestion of "sudo php sniffer.php" from the commar line results in a command not found msg.

Could you please let me know how you are running this awesome script?

Thanks!

• Reply • Share ›

Silver Moon Mod → Ventre • 2 years ago
you need php cli to be installed.
Check it with the command "php -v"
It should show something like this :

PHP 5.3.5-1ubuntu7.4 with Suhosin-Patch (cli) (built: Dec 13 2011 18:30:11)
Copyright (c) 1997-2009 The PHP Group
Zend Engine v2.3.0, Copyright (c) 1998-2010 Zend Technologies
with Xdebug v2.1.0, Copyright (c) 2002-2010, by Derick Rethans

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

If it says that command not found then its not installed.

Also this code will work only on Linux.

When running the script provide the full path like :

sudo php /var/htdocs/sniff/sniffer.php

or something similar

• Reply • Share ›

Subscribe Add Disqus to your site DISQ

Copyright © 2014 BinaryTides

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com