Netfilter string module example usage

stackoverflow.com/questions/13401096/netfilter-string-module-example-usage

Ask Question

Can anybody point me to some examples in using the xt_string module with netfilter or provide a example. What I am trying to do is to write netfilter module that will drop packets that contain a certain string in the skb->data field.

I initially tried simply `strnstr(skb->data, "mystring", strlen("mystring"))` but this seem to be incorrect approach to this problem (and it does not seem to be working as i dont see any packets being dropped).

Thanks in advance

asked Nov 15 '12 at 15:47

SneakyMummin
453923

3 Answers 3

up vote 5 down vote accepted

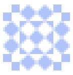
If you mean using iptables string match in user-space, here is one example:

```
iptables -I INPUT 1 -p tcp --dport 80 -m string --string "domain.com" --algo kmp -j
DROP
```

Or if you mean in kernel space, you can use textsearch API which provides KMP/BM/FSM algorithms, the following example is from kernel source lib/textsearch.c:

```
int pos;
struct ts_config *conf;
struct ts_state state;
const char *pattern = "chicken";
const char *example = "We dance the funky chicken";
conf = textsearch_prepare("kmp", pattern, strlen(pattern),
                             GFP_KERNEL, TS_AUTOLOAD);
if (IS_ERR(conf)) {
    err = PTR_ERR(conf);
    goto errout;
}
pos = textsearch_find_continuous(conf, &state, example, strlen(example));
if (pos != UINT_MAX)
    panic("Oh my god, dancing chickens at %d\n", pos);
textsearch_destroy(conf);
```

answered Nov 16 '12 at 5:12



Cong Wang
1,633811

up vote 2 down vote
what you are looking for may be this one, "skb_find_text". It uses the infra in linux mentioned by @Cong Wang. You can also find some examples in the kernel codes.

answered Oct 18 '15 at 10:34

thlin
2113

up vote -1 down vote
here after a source code of netfilter. it's a module to drop received ICMP ECHO

you can use this code to help you to develop your module. You have just to get data from skb and then check it.

```
#define __KERNEL__
#define MODULE
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/slab.h>
#include <linux/list.h>
#include <linux/netfilter_ipv4.h>
#include <linux/ip.h>
#include <linux/icmp.h>
#include <linux/netdevice.h>
#include <linux/netfilter.h>

#include <linux/skbuff.h>
#include <linux/string.h>
#include <linux/inet.h>

MODULE_LICENSE("GPL");


static struct nf_hook_ops netfilter_ops_in;/* IP PRE ROUTING */
static struct nf_hook_ops netfilter_ops_out; /* NF_IP_POST_ROUTING */
struct sk_buff *sock_buff;
struct iphdr *ip_header;
struct net_device *dev;
char *in_face = "eth0";
char *out_face = "eth1";

void log_ip(int sadd,int dadd)
{
    int b1,b2,b3,b4;
    b1 = 255 & sadd;
    b2 = (0xff00 & sadd) >> 8;
    b3 = (0xff0000 & sadd) >> 16;
    b4 = (0xff000000 &sadd) >>24;
    printk("SrcIP: %d.%d.%d.%d",b1,b2,b3,b4);

    b1 = 255 & dadd;
    b2 = (0xff00 & dadd) >> 8;
    b3 = (0xff0000 & dadd) >> 16;
    b4 = (0xff000000 & dadd) >>24;

    printk("  DstIP: %d.%d.%d.%d",b1,b2,b3,b4);
}

unsigned int main_hook(unsigned int hooknum,
                        const struct sk_buff *skb,
                        const struct net_device *in,
                        const struct net_device *out,
                        int(*okfn)(struct sk_buff*))
{
    struct icmphdr* icmp;
    sock_buff = skb_copy(skb,GFP_ATOMIC);
    ip_header = (struct iphdr*)(sock_buff->network_header);
    //ip_header = ip_hdr(sock_buff);

    icmp = (struct icmphdr*) ((char*)ip_header + sizeof(struct iphdr));
    //icmp = icmp_hdr(skb); /* do not return a good value in all cases*/
    log_ip(ip_header->saddr,ip_header->daddr);
    printk("  Dev:%s\n",sock_buff->dev);

    if (icmp->type == ICMP_ECHO)
    {
        printk("ICMP ECHO received and droped\n");
        return NF_DROP;
    }
    return NF_ACCEPT;
}

int init_module(void)
{
    netfilter_ops_in.hook       = main_hook;
    netfilter_ops_in.pf         = PF_INET;
    netfilter_ops_in.hooknum    = NF_INET_PRE_ROUTING; /*NF_INET_PRE_ROUTING;*/
    netfilter_ops_in.priority   = NF_IP_PRI_FIRST;

    nf_register_hook(&netfilter_ops_in);


    printk(KERN_INFO "sw: init_module() called\n");
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "sw: cleanup_module() called\n");
    nf_unregister_hook(&netfilter_ops_in);
    //nf_unregister_hook(&netfilter_ops_out);
    printk(KERN_INFO "sw: hook unregisted, quit called\n");
}
```

answered Nov 15 '12 at 15:57

MOHAMED
18.9k31102181

Your Answer

Sign up or log in

Sign up using Google
Sign up using Facebook
Sign up using Email and Password

Post as a guest

By clicking "Post Your Answer", you acknowledge that you have read our updated terms of service, privacy policy and cookie policy, and that your continued use of the website is subject to these policies.

Not the answer you're looking for? Browse other questions tagged c linux-kernel netfilter or ask your own question.

lang-c