Project Specification

Packet Inspection for Malware Detection

Haya Shulman
Fachbereich Informatik
Technische Universität Darmstadt
haya.shulman@gmail.com

Abstract

The goal of this project is to prepare a mini-firewall. The firewall should inspect all inbound packets and print to file packets which match the rules that it receives in an input. The firewall code is a user space implementation, that reads packets from kernel queue, applies its processing and then returns the packets back to the kernel queue.

1 Specification

You should write a user space implementation, that reads inbound packets (sent from some host in the Internet to the firewall host) from the kernel queue, applies its processing and then sends the packets back to the kernel. To retrieve the packets from the kernel queue, you should use `iptables` firewall rules on the host on which you run your implementation; here is nice tutorial on iptables: `http://www.thegeekstuff.com/2010/07/list-and-flush-iptables-rules/`
Defining firewall rules requires root privileges (as so does the code that opens a raw socket to capture network packets).

The firewall implementation called `cap.c` receives in an input an IP address `ip`, a port `p`, and a string 'hello', for instance: `sudo ./cap ip p i hello`

Then it should capture *all* incoming packets from the kernel, and check the source IP address and source port – if they match the input values, then the implementation should check if the input string appears in the payload of that packet. The implementation should print to a file `out.txt` payload of packets which match in the source IP address and source port and contains in the payload the input string, and the number of appearances of the input string. For instance: if the third input parameter is `hello`, and source IP and ports (`ip, p`) match then
`payload: blah blah hello rtuzisd`
`appearances: 1`

The program should stop after printing i packets.

Basic details regarding kernel processing hooks and code that captures packets can be found here:
`http://manpages.ubuntu.com/manpages/precise/en/man3/libipq.3.html`

You need to extend it with the required functionality: (1) inputs (`ip, port`, counter i, `string`), (2) process headers (IP and UDP) to check if the packet matches the requirements, (3) print the payload and appearances to file if the packet matches.

Your implementation should run on Linux kernel 3.6 (or more recent ones) and the deliverable (code) should be written in C programming language.

The code should use a shell to set the firewall rule when it is invoked and after printing i packets, before terminating execution, the code should delete the firewall rules.

The deliverable should be a code, and a document explaing the libraries that should be used for compilation of the code and invocation of the executable.

```
/* Firewall rules */

// For outgoing packets:

iptables -A OUTPUT -p udp -j QUEUE

// For incoming packets:

iptables -A INPUT -p udp -j QUEUE


/* Pseudocode */

process_packets(ip,port,i,string){
j = 0
while (j <= i){
   packet = read_from_kernel_queue()
   (ip_header, udp.header, payload) = process(packet)
   if ((ip_header.src_ip_address == ip or
            ip_header.dst_ip_address == ip) and
       (udp_header.src_port == port or
            udp_header.drc_port == port) and
            (exists(payload,string)>1))

          n = count_appearances(payload,string)
          print to file (payload,n)
          j++
 }
}
```