The Network Protocol Analysis Technique in Snort

Qing-Xiu Wu

Department of Network Engineering, Hainan College of Software Technology, QiongHai, Hainan, 571400, P.R.China

Abstract

Network protocol analysis is a network sniffer to capture data for further analysis and understanding of the technical means necessary packets. Network sniffing is intercepted by packet assembly binary format of the original message content. In order to obtain the information contained. Required based on TCP / IP protocol stack protocol specification. Again to restore the data packets at protocol format and content in each protocol layer. Actual data transferred, as well as the application tier.

© 2012 Published by Elsevier B.V. Selection and/or peer-review under responsibility of Garry Lee

Keywords: Ubuntu; sudo; Software Package

1. Introduction

Open source Snort is the intrusion detection system by far the most famous and most widely used. When make detection pretreatment data library rules and rules of the match in the past one, Network traffic is enormous, Which large amounts of data is normal data, this will waste a lot of match time.

2. Principle Of Network Protocol Analysis Technology

Network protocol analysis refers to the binary format of data packets transmitted on the network are resolved to restore all the network protocol information and the transfer of technology[1].

While the computer is transferring data across the network, through a packet encapsulation process, before transferring data at the application layer, respectively, TCP or UDP protocol header encapsulation, IP protocol header and link layer protocol header (Such as the Ethernet protocol header), If the application layer data exceeds maximum length of the IP packets and link layer, then take a slice or subcontracting policy, split into multiple packets, And then transmitted over a network link. When the network transmits data when it reaches the target host, target host of the network stack will carry out the inverse operation of packet-unpacking process, Like a "peeled onion" stripped as link-layer protocol header,
IP protocol header, TCP or UDP header, and then slice or sub-restructuring, and submit the application layer data to network service or application for processing.

Network protocol analysis and host of principles similar to the process of unpacking, need to be resolve from the bottom up-by-layer network protocols, while an IP fragment and TCP sessions of restructuring, but the difference is that key intended only receive packets on the host gets the application-layer data it contains, transport layer, network layer and link layer information content on don't care, while network protocol analysis needs analysis and save all header fields of the information on the various network layers, as well as the highest level of application layer data content, and make them available to users to understand the full range of network packet information.

Network protocol analysis needs to identify the type of network protocol used by different network levels, then known as network protocols, according to the corresponding standard protocol specification, packet analysis. Network protocol analysis of the typical process involves the following steps:[2][3]

1) First, the network sniffer received raw data is in binary packet link layer transmission, most cases are Ethernet data frame;

2) Structure analysis of Ethernet data frame, positioning the frame head the field structure, according to the Type field to determine if network-layer protocol frame type is in most cases IP protocols (0800), network layer data and extracting data frames containing content;

3) Further to analyze the IP packet, if the Fragment bit set, then an IP fragment restructuring, under IP Protocol in the protocol header field, determines the transport layer protocol type, typically are TCP (6) or UDP (17), and extracts the IP transport layer data in the packet contents;

4) Continue to identify specific TCP or UDP destination port of application layer protocols such as HTTP, FTP, Telnet, and other protocol packets, and splicing the TCP or UDP packets of recombinant, have the application layer protocol-specific application of interactive content;

5) According to the corresponding application layer protocol consolidating data recovery, are actual data transfer.

For an unknown network protocols, such as the custom protocols used by a number of new malicious code, or some protocols use encryption to protect, for example, very difficult for protocol analysis, binary reverse engineering of requires analysts with high technical competence to determine the format of these agreements.

3. Network Protocol Analysis Technology

Open source network sniffer software, such as Tcpdump, Wireshark, and Snort has a corresponding source implementations. Source code for the network protocol decoding with Snort, the network protocol analysis technology enablers are elaborated in detail.

Snort source packages for network protocol analysis in decode.c and decode.h are the main file. Analysis on Network Protocol Packet is the most important data structures, Snort process object—packets are defined, which includes content can be roughly divided into three categories: *Pkth,*PKT indicating that the original data field (respectively the pointer to the packet header and packet); After the agreement to hold the current data packet analysis information fields, due to the need to support multiple protocols, more complex, such as EtherHdr*eh (Ethernet protocol header), TCP/IP/Ethernet/ARP protocol first; literacy, indicate a variety of Protocol message header checksum error, the status of the current packet, identifies the need to call the preprocessing of types.
Network protocol analysis in Snort process as shown in Fig1, entrance is from the libpcap library pcap_loop function ProcessPacket() defined callback function, namely libpcap network sniffer intercept every packet in packet processing function that is called after, Snort by implementing each network layer protocol parsing and processing routines, to complete the packet network protocol analysis, Key processes are as follows:

1. Analysis of Ethernet data frame

   Resolved before considering an Ethernet data frame format diagram, in the frame header includes the 6-byte MAC address for the purposes of, 6 bytes of the source MAC address, 2 bytes of the upper class code, type code for the 0x0800, ARP protocol such as IP protocol type code of 0x0806. Snort dismantling of Ethernet packets corresponding to this data structure is as follows:

   ```c
   typedef struct _EtherHdr
   {
       u_int8_t ether_dst[6];
       u_int8_t ether_src[6];
       u_int16_t ether_type;
   }EtherHdr;
   ```

   Snort decode.c DecodeEthPkt unpack process is as follows:

   Pretreatment, unpacking some preliminary processing before, such as an Ethernet packet count increasing to determine the header length is greater than a predefined maximum header length, is not less than a predefined minimum header length, if it is found in error for error handling.

   Unpacking, Current packages are assigned to Packet data structure pointer in EH (EtherHdr-a pointer).

   Resolution, with switch statement on Ethernet baotou in the specifies of upper application type code (Ethernet package data structure in the of ether_type) for select, calls corresponding of processing function, for example, through judgment are of type for IP agreement, so on will package of pointer back moved 14 byte (Ethernet baotou of length), removed Ethernet baotou, indicates baotou length of variable minus 14 zhihou as parameter calls on IP package analysis of function, Address of Ethernet packets to this resolution.

2. Parse IP packet

   Analysis of IP packet and Ethernet frame processing such data, there is also a structure corresponding IP packet header information, version number and header length as a byte of data storage, of displacement also shared a byte for flags and segments, and did not involve IP package options and padding (in their Packet data structures).

   ```c
   typedef struct _IPHdr
   {
u_int8_t ip_verhl; //version and head length
u_int8_t ip_tos; //type of service
u_int16_t ip_len; //length
u_int16_t ip_id; //identification
u_int16_t ip_off; //offset
u_int8_t ip_ttl; //TTL
u_int8_t ip_proto; //protocol(TCP/UDP/ICMP)
u_int16_t ip_csum; //checksum
struct in_addr ip_src; //source ip address
struct in_addr ip_dst; //dest ip address
}IPHdr;

Parse IP packet DecodeIP process and analysis of an Ethernet frame is similar, is as follows:

Pretreatment, such as IP packet count increasing, judging packet header is less than the minimum length of the IP packet header, as well as some judgments about IPv6 macro processing.

Unpacking, address assignment of the packets to the Packet data in the IP header IPH (IPHdr pointer type), there is also the necessary processing, such as the version of the judgment (in order to support IPv6) packages are malformation or not, is needed for the checksum calculation as well as a number of flags, such as whether there are additional options need to be addressed.

Resolution, TCP, for example, when judge the upper layer protocol is TCP packets, the first TCP counters add up, until resolve TCP processing routines: TCP module moves backwards through the get pointer to the packet IP header length, that is, given a TCP header and content[4][5].

(3) analysis of TCP packet
TCP packet format, also have structures corresponding in Snort, TCP options are Packet data structure, as follows:

typedef struct _TCPHdr
{
    u_int16_t th_sport; //source port
    u_int16_t th_dport; //dest port
    u_int32_t th_seq; //sequence number
    u_int32_t th_ack; //acknowledgement number
    u_int8_t th_offx2; //offset and reserved
    u_int8_t th_flags; //flags
    u_int16_t th_win; //window
    u_int16_t th_sum; //checksum
    u_int16_t th_upr; //urgent pointer
}TCPHdr;

TCP packet analysis process DecodeTCP and is divided into three parts, is as follows:

Pretreatment, Testing of information such as the length of the package. TCP checksum calculations need to introduce a pseudo header, so the Snort has two structures as defined in the function's local variables appear. Also need to make analysis and processing of the packet length is wrong.

Unpacking, Assigns the pointer for the current packet to Packet teph, after which there are some on the legality of packet processing, and flags in the TCP packet processing, options, and process[6].

Resolution, Just move the pointer back TCP header length and assigned to the appropriate pointer, later referred to the appropriate program of data processing, is the port number of the distinction between upper-layer protocols, such as the corresponding HTTP port 80, port 23 Telnet, port 21 for FTP.
4. Summary

Analysis of network sniffer and Protocol is network protocol as well as other initiative such as password cracking relies on the underlying technology. At the same time, Defender, network sniffer and Protocol Analyzer is able to successfully find in-depth analysis of network attacks and the basic skills it must master.

References