Network Intrusion Detection System

Na, Yun Seok

10 Jun 2017

1 README 간단 요약

컴파일은 -std=gnu99 와 -lpcap 옵션을 주고 컴파일을 하면 됩니다. 또는 install.sh는 p3폴더 위에 nids라는 바이너리를 만듭니다. 실행은 두 가지 모드로 가능한데 ./sudo nids rule_file로 실행 하면 pcap_lookupdev로 자동으로 attach를 하고 ./sudo nids rule_file interfac를 는 interface 를 붙여서 실행시킵니다.

2 Struct

1. sniff_ethernet : 이더넷 헤더 구조체(tcpdump.org 참고)
2. sniff_ip : IP 헤더 구조체(tcpdump.org 참고)
3. sniff_tcp : TCP 헤더 구조체(tcpdump.org 참고)
4. sniff_udp : UDP 헤더 구조체(자체 제작)
5. rule : 한 개의 snort rule을 저장하는 구조체

3 Function

3.1 main

각종 libpcap 관련 세팅과 parser 실행 후 network 관련 함수를 실행 시켜주는 역할

3.2 parser

Rule파일의 입력 받아 이를 줄 단위로 memory에 입력시키는 함수

3.3 ruleCounter

Rule파일 내의 줄 수를 카운트하여 몇 개의 룰을 입력받는지 체크하는 함수. 룰의 입력을 dynamic allocation으로 처리하기 때문에 사전에 룰의 개수를 카운트 하기 위한 용도

3.4 displayRule

Debug용도로 Rule파일로부터 struct rule이 잘 형성되었는지 체크하기 위해 사용하는 함수

3.5 optionAdder

룰에서 IP주소, 프로토콜, 포트번호를 파싱하고 ()안에 있는 각종 옵션을 struct rule에 입력하기 위한 용도로 만든 함수. strtok에 delimiter를 semi colon으로 넣어서 만들었습니다.

3.6 ip_parse

IS521 정보보호실습에서 Markus 학생이 Worm을 제작할 때 사용한 함수. string 형식의 IP주소를 uint32로 바꾸어 줍니다.

3.7 ip_tostring

IS521 정보보호실습에서 Markus 학생이 Worm을 제작할 때 사용한 함수. uint32 형식의 IP주소를 string으로 바꾸어 줍니다.

3.8 digit

정수 형식의 데이터의 자리수를 구해주는 함수. 각종 룰을 parsing할 때 필요가 있어서 제작했습니다.

3.9 packetAnalysis

Packet을 입력받아 Packet을 demultiplexing하고 입력받은 snort rule과 비교해서 결과를 출력해주는 함수. packet한 개를 받으면 rule1-rule2-rule3-.... 순서로 rule을 체크해서 snort alert에 걸리는지 확인합니다. 그 후 snort rule에 match되면 결과를 출력합니다.

3.10 print_payload

Payload를 정제해주는 함수. Payload를 받아서 preprocessing을 한 후 실제로 출력해주는 함수인 print_hex_ascii_line으로 보냅니다.

3.11 print_hex_ascii_line

print_payload함수에서 온 정보를 바탕으로 실제로 화면에 payload를 출력해주는 함수

3.12 isIP

SNORT rule의 IP와 입력받은 패킷의 IP를 비교해서 해당 SNORT rule이 관리하는 IP범위인지 체크해주는 함수. 기본적으로 CIDR가 8의 배수 일때만 동작합니다.

3.13 isPORT

SNORT rule의 Port와 입력받은 패킷의 Port를 비교해서 해당 SNORT rule이 관리하는 Port범위인지 체크해주는 함수. any인 경우 무조건 true를 반환. range 형태의 경우 대소비교 후 결과를 반환. single port가 입력된 경우 해당 single port와 같지 않으면 false 같으면 true를 반호나. static port형태로 여러 port가 콤마와 함께 입력된 경우 해당 리스트와 비교 후 결과 값을 반환.

4 Global Variable

1. number_of_rules : 몇 개의 SNORT 규칙이 입력 되었는가? 입력 받는 파일의 줄 수로 계산
2. raw_rule : 입력 받는 파일을 줄 단위로 잘라서 입력한 변수
3. number_of_options : 각 규칙에 몇 개의 option이 있는지 체크하는 함수 각 줄의 semicolon수로 계산
4. ruler : 실제 rule의 구조체

5 Rule Parser

먼저 rule_file을 여는 것을 시도하고 열리지 않으면 error를 출력하고 종료합니다. 그 후 줄 단위로 입력을 처리하고 한 줄을 한 개의 규칙 구조체(Rule struct)으로 바꾸어 줍니다. 이 과정에서 SNORT rule을 준수하는지 아닌지 확인하고 이를 struct 내의 valid라는 변수로 저장합니다.

6 Output

먼저 패킷을 받으면 몇 번째 패킷인지 초록색으로 출력 해줍니다. 패킷이 들어오면 SNORT Rule과 매칭하고 매치가 되면 패킷의 정보와 message, highlight matched field를 출력합니다. IP와 Port는 정보는 빨간색으로 highlight하지 않았습니다.(왜냐하면 SNORT RULE에 매치되는 경우에만 출력하므로) 또한 http_request와 content가 매치되는 경우 payload를 highlight하지 않고 별도의 메세지를 더 출력하도록 하였습니다. Message는 magenta 색깔로 출력됩니다.

7 Reference

http://www.tcpdump.org/pcap.html 과 http://www.tcpdump.org/sniffex.c를 기반으로 작성하였습니다. 특히 payload 출력부분과 몇몇 구조체 설계는 완전히 동일함을 말씀드립니다. 또한 IP주소 변환에 IS521수업(정보보호 실습)에서 공유 된 함수를 사용하였습니다.