PASSIVE ATTACK 실습

환경 구성

Linux 환경 갖추기

- vmware workstation 16 player

https://www.vmware.com/kr/products/workstation-player/workstation-player-evaluation.html

- Ubuntu Desktop 64 bit 20.04 LTS

https://releases.ubuntu.com/focal/

- 그 외
  - MacOS
  - 타 hypervisor
  - aws 등 cloud
  - 타 서버

SCAPY

- 트래픽 처리를 통한 보안 테스트 및 ethical hacking
- 패킷 캡처, 분석, 가공
- IDS/IPS 테스트 등
- 설치

```
$ sudo apt install python3-scapy
```

실행

- CLI

  $ scapy

  - ls() : Displays all the protocols supported by scapy
  - lsc() : Displays the list of commands and functions supported by scapy
  - conf : Displays all configuration options
  - help() : Displays help on a specific command, for example, help(sniff)
  - show(): Displays the details of a specific packet, for example, apacket.show()

CLI

패킷 생성 및 보기

- 패킷 구조 : Ether() / IP() / TCP() / Data
  - 쓰지 않으면 default 가 쓰임 (ex. IP default dst : 127.0.01)

```
>>> p3=IP(dst='google.com') / TCP (dport=80)
>>> p3.show()
>>> p3.show
>>> p3.summary()
>>> ls (p3)  # 이렇게 쓸 수도
>>> ls (IP ) # 이렇게 쓸 수도
>>> p2 = Ether() / IP (dst="google.com") /ICMP() /"ABCD"
>>> p2.show()
```

SENDING PACKETS

- send( ) : layer-3 packets
- sendp( ) : layer-2 packets
  - arguments
    - iface : interface to send packets
    - Inter : interval
    - loop : 1 or 0
    - packet

    # root 권한 필요 => 나갔다가 sudo scapy 로 실행, p3, p2 정의 다시

    >>> send (p3)

    >>> sendp (p2)
- Send and receive : sr(), srp()

  >>> srp (p2)

  >>> sr (p3)

RECEIVE 결과 확인

- 결과 보기

```
>>> r2= srp (p2)
>>> r2[0].show()
```

- 1개 결과 받기 : sr1(), srp1()

```
>>> r2= srp1 (p2)
>>> r2.show()
```

PACKET-SNIFFING

- Broadcast 되는 packet을 sniffing 함
  - 자신의 iface를 알아야 함 : 아래 ens33 을 대체
  - 콘솔에서
    $ ifconfig
  - scapy에서

```
>>> pkts = sniff ( iface= "ens33", count=3 )
>>> pkts[0].summary()
>>> pkts[0].show()
```

- parameter
  - count: Number of packets to capture, but 0 means infinity
  - iface: Interface to sniff; sniff for packets only on this interface
  - prn: Function to run on each packet
  - store: Whether to store or discard the sniffed packets; set to 0 when we only need to monitor them
  - timeout: Stops sniffing after a given time; the default value is none
  - filter: Takes BPF syntax filters to filter sniffing

PRN

- prn : packet 처리함수
  - 브라우저에서 http://academy.kitri.re.kr
  - scapy에서
    - 각 패킷의 summary 출력

```
>>> packets = sniff (filter ="tcp", iface="ens33", prn=
lambda x : x.summary() )
>>> packets[-1].show( )
```

SPRINTF

- sprintf : field 단위 출력 가능, 자유로운 형태의 출력
  - %IP.src% 처럼 분류.필드명을 줌

```
>>> packets=sniff (filter="tcp", iface="ens33",
prn=lambda x: x.sprintf (" {IP:%IP.src% -> %IP.dst%\n}" ) )
```

FILTER

- BPF (Berkeley Packet Filter)
  - 포맷 : src|dst　host|port|net
    - 예: "Src host 192.168.1.0"
    - 예: "Dst port 80"
  - Protocol :ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, TCP 또는 UDP
    - filter="tcp and (src port 80 or dst port 80)"

    >>> packets = sniff (filter ="tcp src port 80", iface="ens33", prn= lambda x : x.summary())

- Lfilter　(lambda filter)
  - Lambda function이 true인 경우
    - 예 : lfilter= lambda s: TCP in s

HTTP

- Import 필요

```
>>> from scapy.layers.http import *
```

- http 응답만
  - Lfilter = lambda f : f.haslayer(HTTPResponse)
- 응답만 출력
  - Prn = lambda x : x[HTTPResponse].show()
- 모아서

```
>>> packets = sniff (lfilter =lambda f : f.haslayer(HTTPResponse) , iface="ens33", prn= lambda x : x[HTTPResponse].show()  )
```

- raw http 출력
  - import 필요

```
>>> from scapy.all import *  # Raw 사용을 위해 필요
```

  - x[Raw].show() 를 사용

```
>>>  packets = sniff (lfilter =lambda f : f.haslayer(HTTPResponse) , iface="ens33", prn= lambda x : x[Raw].show()  )
```

과제1

- http://academy.kitri.re.kr 사이트의 입력 id, pw를 sniff하여 출력하는 cli 커맨드 작성
  - Id는 자기 영문 이니셜 : ex) 최대선 -> dschoi
  - 패스워드는 자기학번

- Cli 명령 및 실행 결과 부분을 캡처 => 1.jpg
  - 필요시 1-1.jpg, 1-2.jpg 이렇게 여러 장으로 나누어도 됨

숭실대학교 Soongsil University

과제 제출 방법

- 01.zip 파일에 결과 파일들 묶어서 제출