최종보고서

리눅스 쉘 프로그램을 이용한 서버 취약점 및 스캐닝도구 개발

과 목 명 : 캡스톤 디자인
과 목 명 : The Creator
팀 장 : 허 종 만
팀 원 : 장순현
하상우

지도교수 : 양환석 교수님

요 약 문

1. 연구제목

리눅스 쉘 프로그램을 이용한 서버 취약점분석 및 포트스캔 활용방법 구축

2. 연구 목적 및 필요성

지식정보화 사회에서 정보보안의 중요성을 알리고 그에 따라 최근 화두가 되고 있는 포트 스캔(port scan) 및 서버 취약점분석 툴을 운영하여 서버에서 열려 있는 TCP/UDP 포트를 검색하고 그에 따른 결과 값을 보기 좋게 보고서형식으로 제작에 중점을 두었다.

3. 연구 내용

본 연구에서는 쉘 프로그램을 이용하여 서버의 취약점 분석과 포트스캔, Snort 이용하여 어떤 포트가 열려있고 닫혀있는지 조사하고 자신의 서버의 네트워크 서비스들을 점검하며 해커가 해킹을 하기 위해 정보 수집에서 타깃 서버의 정보를 수집하기 위해 사용되고 있다.

하지만 여러 서버를 한 번에 관리하기가 어렵고 FTP, DNS, Web서버 등 서버들의 취약점에 대비한 패치가 용이하지 않고 Snort 침입탐지 시스템의 각종 설정 및 데이터를 한 번에 관리하기 힘들다는 단점을 보완하기 위해 이러한 주제를 선정하였다.

4. 연구 결과

본 연구에서 개발한 리눅스 쉘 프로그램을 이용한 서버 시스템을 사용자가 손쉽게 관리할 수 있으며, 네트워크에 존재하는 서버들의 취약점에 대한 내용을 보고서로 작성하여 그 결과를 손쉽게 확인하고 대비할 수 있는 장점을 가지고 있다. 그러나 사전에 설치되어야 할 툴들이 있으며 관리할 서버들의 목록은 수동으로 입력해야하는 단점을 가지고 있다.

목 차

1. 연구 계획

1.1. 조원 소개 및 역할 분담

조원	역할
허종만	조장, 개발 담당, 프로젝트 총괄
장순현	자료 종합 담당(ppt), 프로그램 검증 담당
하상우	자료조사(snort, 각종서버), 자료 분석 담당

1.2. 연구의 배경 및 목적

수동적인 방법 보다 도구를 이용하여 웹 서버의 종류나 버전, 디렉터리 정보나 중요 파일 정보가 존재하는지, 웹 서버 자체의 취약점은 무엇인지 검사하는 것이다. 대상 시스템에 HTTP 요청을 보낸 뒤 이에 대한 응답 코드를 받으면, 그 코드를 분석해서, 요청한 페이지가 존재하는지, 서버의 어느 부분이 취약한 지 등을 알 수 있다. 다만 주의해야 할 점은 현행법상 스캐닝만 하여도 공격으로 간주하여 처벌되기 때문에, 개인 테스트 컴퓨터 또는 친구 또는 주위에서 허가한 컴퓨터를 대상으로 스캐닝 하여야 된다.

2. 이론적 배경

2.1.1. 취약점 분석의 의미

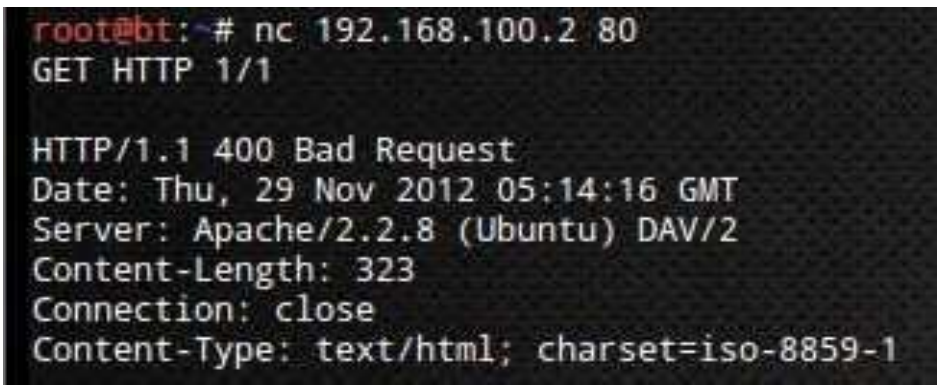
ⅰ) 취약점 분석은 자신의 취약점 데이터베이스(or 지식, 기술, 경험)을 참조해 대상 시스템에 존재하는 취약점을 검색/나열하는 것을 일컫는다. 운영체제마다 사용하는 네트워크 구현 방식이 다르기 때문에 네트워크로 특정 데이터 구현 방식이 다르기 때문에 네트워크로 특정 데이터를 보냈을 때의 반응은 각기 다르게 응답하게 된다. 이런 고유한 형태의 응답은 취약점 스캔/분석의 단계를 통해 운영체제 버전 정보, 패치정보 등을 알아내는 데 사용하는 Fingerprint 역할을 하는데 도움을 준다.
여기에 (무료/사용)취약점 스캐너는 주어진 정보를 활용해 원격 시스템 로그인에 필요한 사용자 자격 증명을 설정하거나, 소프트웨어를 오픈하거나, 서비스의 패치 여부까지 판별해 낼 수 있는 수준이 되었다. 취약점 스캐너는 이렇게 얻은 결과로 시스템에서 탐지된 모든 취약점 보고서를 작성할 수 있으며, 이렇게 발견되어 취약점 목록은 추후 Penetration Test에 활용되기도 한다.

ⅱ) **취약점 스캐너를 이용한 취약점 스캐닝**: 취약점 스캐너는 일반적으로 대량의 네트워크 트래픽(부정한)을 생성하게 된다. 이 때문에 모의해킹을 수행할 때 탐지되지 않아야 하는 테스트가 필요한 경우는 사용하지 않는 것이 좋다. 하지만 숨길 필요가 없는

테스트를 진행 할 때에 이러한 취약점 스캐너는 시스템의 패치 수준과 취약점 식별을 수동으로 조사하는 것보다 시간을 절약할 수 있게 해준다.
자동화된 스캐너를 사용하든 수동으로 스캐닝을 진행하든지 간에 스캐닝은 모의 해킹 단계에서 가장 중요한 단계 중 하나이며, 철저하게 수행할수록 고객에게 가치 있는 결과를 제공할 수 있게 된다는 사실을 명심한다.

iii) **일반적인 수동 스캐닝 방법**

가장 기본적인 수준에서 스캔이 어떻게 수행되는지 생각해본다.
예를 들어, NC(Netcat)을 사용하여 대상시스템을 배너 그래빙(Banner grabbing)하면, 아래와 같은 형태로 정보를 획득할 수 있다.
(배너 그래빙: 원격 네트워크 서비스에 접속하고 반환되는 서비스 배너(확인)문구를 살펴보고 추측한 것으로, 웹, 파일전송, 메일 서버 같은
다양한 네트워크 서비스는 원격으로 접속하자마자 즉시 배너를 반환하거나 특정 명령의 응답으로 배너를 반환하는 성질을 이용한 기술)
- 대상 시스템: Metasploitable v2 (192.168.100.2)
- 스캔 시스템: bt5 r3 (192.168.100.9)



[그림 1] 일반적인 수동 스캐닝 방법

위 결과는, TCP Port 80 웹 서버에 접속하고, 응답으로 원격 서버에서 반환되는 헤더 정보를 확인하기 위해 “Get Http/1.1"을 요청한 결과이다.
반환된 정보를 통해 Tcp Port80을 이용하는 **대상시스템은 Apache/2.2.8을 사용하는 Ubuntu 기반의 웹 서버라는 사실을 알 수 있다.** 이 정보는 향후 모의 해킹을 시도 할 때에 도움이 될 수 있을 뿐만 아니라, 자동화된 취약점 스캐닝을 이용할 때에도 상당한 도움이 될 수 있다.

2.1.2. 웹 서버 취약점 분석

최근 웹 해킹의 증가로 웹 서버 및 웹 응용 프로그램의 보안이 이슈가 되고 있다. 따라서 지신이 운영하는 서버 내 취약성을 점검하기 위해 여러 방법이 사용되고 있지만 그 중에서 nikto라는 웹 취약성프로그램은 사용법이 간단하면서도 관리자가 쉽게 지나칠 수 있는 부분을 지적해 주어 개인적으로 많이 활용하고 있는 프로그램이다.

특히 nikto를 이용하여 추측이 가능한 기본 디렉토리 사용 여부 웹을 통한 각종 설정 파일이나 로그 파일 접근 여부 버전 정보 등 불필요한 정보 제공 여부 불필요하게 허용

되어 있는 메소드(Method)등을 검색할 수 있다.

ex) 취약성 스캔의 예

```
# ./nikto.pl -generic -h www.target.com
(1)Server:Apache/1.3.29(Unix)PHP/4.3.4mod_ssl/2.8.16 OpenSSL/0.9.7c
(2) Retrieved X-Powered-By header: PHP/4.3.4
(3) Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS,
                PATCH, PROPFIND, PROPPATCH,  LOCK, UNLOCK, TRACE
(4) /admin/ - This might be interesting... (GET)
(5) /phpinfo.php - Contains PHP configuration information (GET)
(6) .bash_history - This might be interesting... (GET)
```

(1) apache의 경우 httpd.conf 에 ServerTokens Prod 라고 추가하면 외부에 불필요하게 버전 정보를 노출하지 않습니다.

(2) php를 사용할 경우 php.ini 에서 expose_php = On을 Off로 설정하면 불필요하게 php 버전 정보를 노출하지 않습니다.

(3) 웹 서버 운영 시 일반적으로 GET/HEAD/PUT 이외 다른 메소드를 허용 할 필요가 없습니다. 따라서 아래와 같이 GET(HEAD포함됨), POST 이외 다른 메소드를 거부하시기 바랍니다.

```
<Directory />
  <Limit GET POST>
     Order allow,deny
     Allow from all
   </Limit>
<LimitExcept GET POST>
        Order allow,deny
        deny from all
</LimitExcept>
</Directory>
```

(4) 관리자 모드 디렉토리로 누구나 추측이 가능한 /admin/을 사용하는 것은 좋지 않습니다. 추측이 어려운 다른 이름으로 변경하고 접근 통제를 엄격하게 하시기 바랍니다.

(5) info.php 또는 php.php와 같이 phpinfo()에 접근 가능할 경우 서버 의 환경 설정에 대한 많은 정보를 유출하게 됩니다. 개발이 종료 후 에는 반드시 파일을 삭제하는 것이 좋습니다.

(6) 이외 웹을 통해 sqlnet.log나 .bash_history등에 접근 가능할 경우 공격자에게는 매우 유용한 정보가 되므로 주의하여야 합니다.

2.1.3. Snort구축 및 IDS응용 방법

▶snort 란?

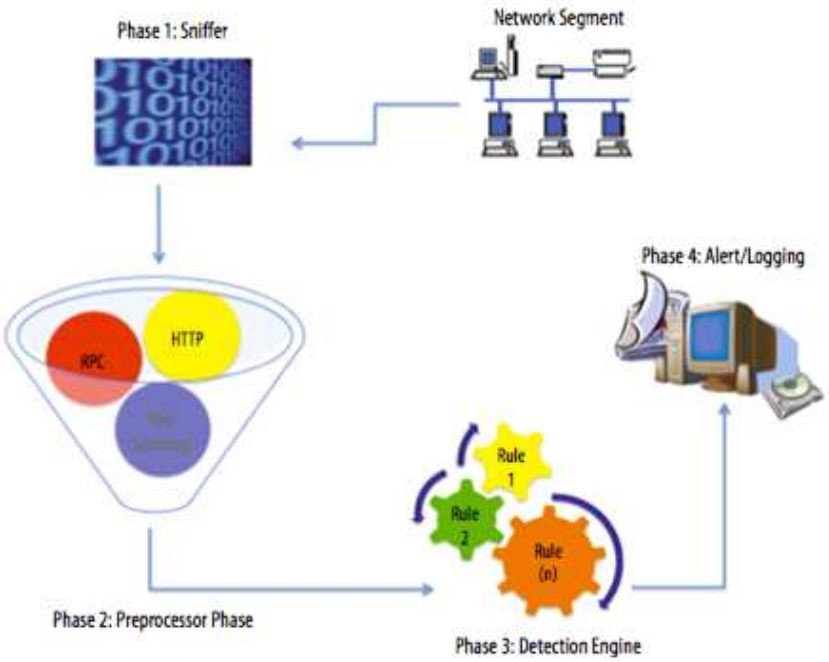
○ 1988년 Sourcefire CTO(최고기술경영자) Martin Roesch 발표
○ 공개 네트워크 침입탐지시스템 (NIPS, Network IDS/IPS)
○ snort 용어는 sniffer and more 라는 말에서 유래
○ 커뮤니티를 통해 지속적인 탐지 Rule 제공
○ Multi-Platform(다양한OS)에서 실행 가능
○ 관리자가 직접 탐지 Rule 설정 가능

▶Snort의 기능(모드)분류

○ 패킷스니퍼(sniffer) : 네트워크의 패킷을 읽어 보여주는 기능, 패킷 데이터의 ID/PASS 도청 가능
○ 패킷로거(logger) : 모니터링 한 패킷을 저장, 로그기록, 트래픽 디버깅에 유용
○ Network IDS : 침입탐지시스템(Intrusion Detection System), 네트워크 트래픽(패킷)분석, 공격탐지기능
○ Snort lnline : 침입방지시스템(Intrusion Prevention System), 패킷분석, 공격차단기능

▶Snort 구조

○ 스니퍼(Sniffer) : Snort IDS를 통과하는 모든 패킷 수집
○ Perprocessor : 효율적인 공격 탐지를 위해 몇가지 플로그인을 먼저 거치며 매칭되는지 확인
○ 탐지엔진 : Rule 기반의 탐지엔진, 사전에 정의된 탐지룰과 매칭되는 지 확인
○ 로깅(출력) : 정책에 따라 로그 기록



[그림 2] Snort의 구조

▶플로그인(Preprocessor)종류

○ Minfrag - 아주 작게 조각난 패킷을 탐지

○ Defrag - 조각난 패킷을 감지하고 IP 패킷을 재조합

○ Stream - TCP Stream을 재조합하여 탐지율 향상

○ HTTP Decode - URL 문자열을 알아보지 못할 경우 snort엔진에서 알아 볼 수 있도록 문자열 변환

○ Portscan - 포트스캔 공격을 탐지하고 로그를 남김

○ Portscan lgnorehost - 포트스캔 detector가 작동하지 않아야 할 IP 리스트 지정

▶침입패턴을 정의하는 Rule 설정

○ 형식 : var <name> <vlaues>

○ 사용예 : var HOME_NET xxx.xxx.xxx.0/24

2.1.4 PortSacn을 사용하여 포트스캐닝

PortScan이란, TCP와UDP를 이용하여 해당 시스템이 포트가 열려 있는지를 스캔하는 것이다. TCP 프로토콜은 기본적으로 3-웨이 핸드셰이킹(3-way Handshaking)을 이용한 스캔을 한다. UDP프로토콜은 포트가 닫혀 있는 경우 공격 대상은 ICMP Unreachable 패킷을 보내지 않는다. 그러나 UDP 패킷은 네트워크를 통해서 전달되는 동안에 라우터나 방화벽에 의해 손실 될 수 있기 때문에 신뢰하기가 어렵다. 기밀 정보 등이 기업 밖으로 유출되는 것을 방지하는 시스템이다.

PortScan의 다양한 종류

- 스텔스 스캔(Stealth Scan)
  Open 스캔처럼 세션을 완전히 성립하지 않고, 공격 대상 시스템의 포트 활성화 여부를 알아내기 때문에 공격 대상 시스템에 로그정보가 남지 않는다.

▶TCP half Open스캔

- 이 스캔 방법은 Open스캔처럼 완전한 세션을 성립하지 않는다.
  동작방식>
  : 먼저 SYN패킷을 보낸다.
  : 포트가 열려 있는 경우, 서버는 SYN+ANK패킷을 보내고
  : 이때 공격자는 즉시 연결을 끊는 RST(Reset)패킷을 보낸다.

  : 포트가 닫혀 있는 경우에는 Open스캔과 같다.

▶FIN, NULL, XMAS 패킷을 이용한 스캔

- FIN, NULL, XMAS패킷을 보내게 되면
: 포트가 열려 있을 경우에는 응답이 없고,
: 포트가 닫혀 있는 경우에만 RST패킷이 되돌아온다.

FIN패킷 = Finsh플래그가 설정된 패킷,
NULL패킷 = 플래그(flag)값을 설정하지 않고 보낸 패킷,
XMAS패킷 = ACK, FIN, RST, SYN, URG 플래그 모두를 설정하여 보낸 패킷

▶ACK패킷을 보내는 방법

- ACK패킷을 모든 포트에 보내게 되면, ACK패킷을 받은 시스템은 이에 대한 RST패킷을 보낸다. 공격자는 이 RST패킷의 TTL값과 윈도우의 크기를 분석한다.
: 포트가 열려 있는 경우 TTL 값은 64이하의 값을 갖고, 윈도우는 0이 아닌 값을 갖는다.

: 포트가 닫혀 있는 경우 TTL 값이 운영체제에 따라 일정하게 큰 값을 가지며, 윈도우의 크기가 0인 RST패킷이 돌아온다.

-이러한 스캔 방법은 너무 많이 알려져서 최근 시스템에는 거의 적용되지 않지만 SYN패킷을 이용한 스캔 방법은 세션을 성립하기 위한 정당한 패킷과 구별할 수 없기 때문에 아직도 유효하며 효과적이다.

▶TCP 단편화(Fragmrntation)방법

- 이 방법은 TCP 헤더의 크기는 20개를 두 개의 패킷으로 나누어 보내는 것이다.

- 첫 번째 패킷에 출발지 IP주소와 도착지 IP주소를,

두 번째 패킷에 스캔 하고자 하는 포트 번호가 있는 부분을 보낸다.

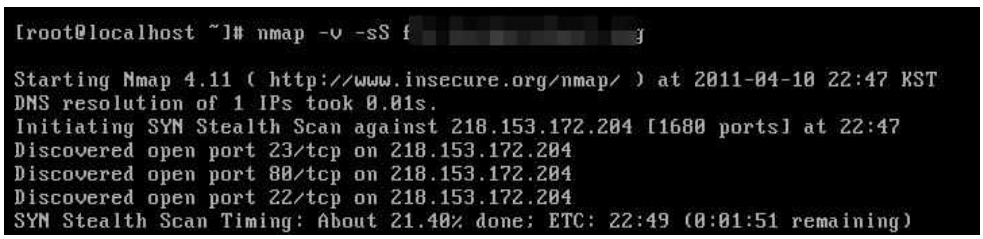
- 첫 번째 패킷은 TCP 포트에 대한 정보가 없기 때문에 방화벽을 통과하고,
  두 번째 패킷은 출발지 주소와 목적지 주소가 없기 때문에 방화벽을 지날 수 있다.

이러한 스텔스 스캔으로 방화벽은 통과할 수 있으나 IDS를 완전히 피할 수 는 없다.
- 그 뿐만 아니라 방화벽과 IDS가 발전함에 따라 스텔스 스캔의 탐지가 가능해졌다.
- 이를 피하기 위해서 시간차를 이용한 스캔이 나오게 되었다.

▶시간차를 이용한 스캔의 두 가지 방법
1. 아주 짧은 시간 동안 많은 패킷을 보내어, 방화벽과 IDS의 처리 용량의 한계를 넘기는 방법
2. 긴 시간 동안에 걸쳐서 패킷을 보내어, 방화벽과 IDS가 그 패턴에 대한 정보를 알 수 없게 하는 방법

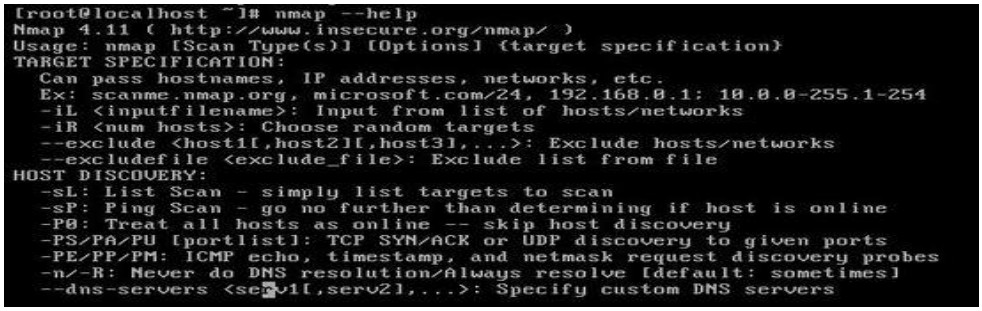
◎ 시간차에 의한 공격의 구분
:Paranoid : 5분이나, 10분 간격으로 패킷을 하나씩 보낸다.
:Sneaky : WAN에서는 15초 단위로, LAN에서는 5초단위로 패킷을 보낸다.
:Polite : 0.4초 단위로 패킷을 보낸다.
:Normal 정상적인 경우이다.
:Aggressive : 호스트에 대한 최대 타임아웃은 5분이며, 패킷당 1.25초 까지 응답을 기다린다.
:Insane : 호스트에 대한 최대 타임아웃은 5분이며, 패킷당 1.25초 까지 응답을 기다린다.
(방화벽과 IDS의 네트워크 카드가 100Mbps 이상이 아니면 이를 탐지하지 못한다.

▶리눅스에서 nmap을 이용한 포트스캔 방법
:간단 설치
yum -y install nmap



[그림 3] 리눅스 nmap을 이용한 포트스캔방법

:설치 후 namp --help을 치면 nmap 에 대한 사용법과, 스캔종류, 옵션을 볼 있다.



[그림 4] 리눅스 nmap을 이용한 스캔종류 및 옵션

▶참고자료 & 사이트

○ Snort를 이용한 IDS구축(KISA자료)

○ 리눅스 IDS 구축

○ 윈도우에 Snort 설치 (2개 - ppt, hwp)

○ 허니넷 (Snirt lnline설명)

→ http://blog.naver.com/oneandonlyme/120053869678

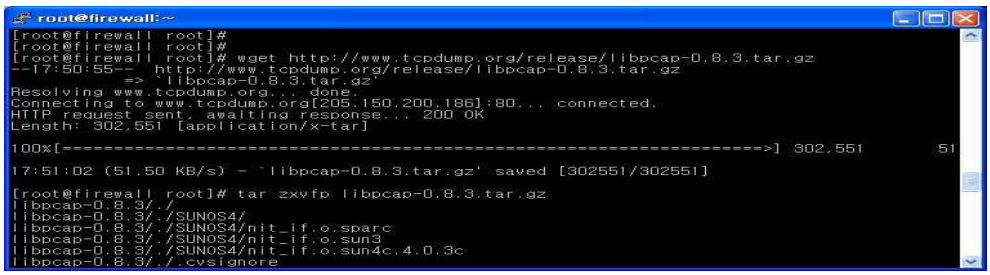
→ http://blog.syszone.co.kr/435 - 각종 설명 및 Preprocessor (플러그인) 설명

3. 시스템 개발

3.1.1 snort 설치 및 설정

snort가 패킷을 스니핑 하려면 패킷캡처 라이브러리인 libpcap을 기반으로 동작하므로 snort를 설치하기 전에 libpcap이 먼저 설치되어 있어야 한다.
설치는 다음과 같은 순서를 따라서 하면 된다.

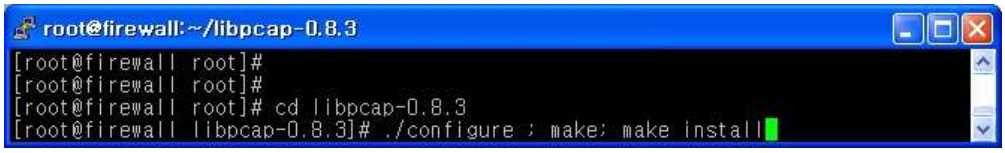
STEP 1. libpcap 다운로드 및 설치



[그림 5] libpcap 다운로드 및 압축해제

wget http://www.tcpdump.org/release/libpcap-0.8.3.tar.gz
(파일 다운로드)tar zxvfp libpcap-0.8.3.tar.gz (압축 풀기)

libpcap은 http://www.tcpdump.org/에서 최신 버전을 다운로드하여 설치하면 된다.
cd libpcap-0.8.3 (압축 푼 파일 디렉토리 이동)
./configure ; make ; make install (컴파일 및 설치)



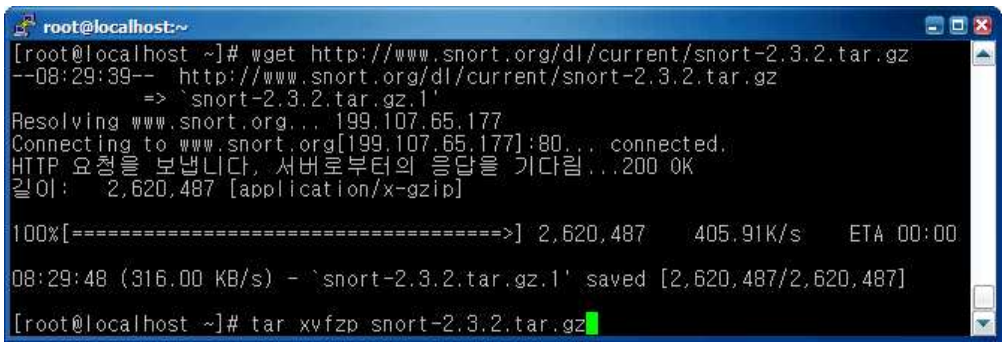
[그림 6] libpcap 컴파일

만약 설치가 잘 안될 경우에는 http://www.rpmfind.net/에서 각자 배포판 버전에 맞는 rpm 파일을 다운로드하여 설치해도 된다.

STEP 2. snort 다운로드 및 설치

wget http://www.snort.org/dl/current/snort-2.3.2.tar.gz (snort 파일 다운로드)

tar xvfzp snort-2.3.2.tar.gz (압축 풀기)



[그림 7] snort 다운로드 및 압축해제

libpcap을 설치한 후snort를 설치하도록 한다.

snort는 http://www.snort.org/dl/에서 최신 버전의 소스파일을 다운로드 후 압축 해제한다.

cd snort-2.3.2 (압축 푼 파일 디렉토리 이동)

./configure ; make ; make install (컴파일 및 설치)

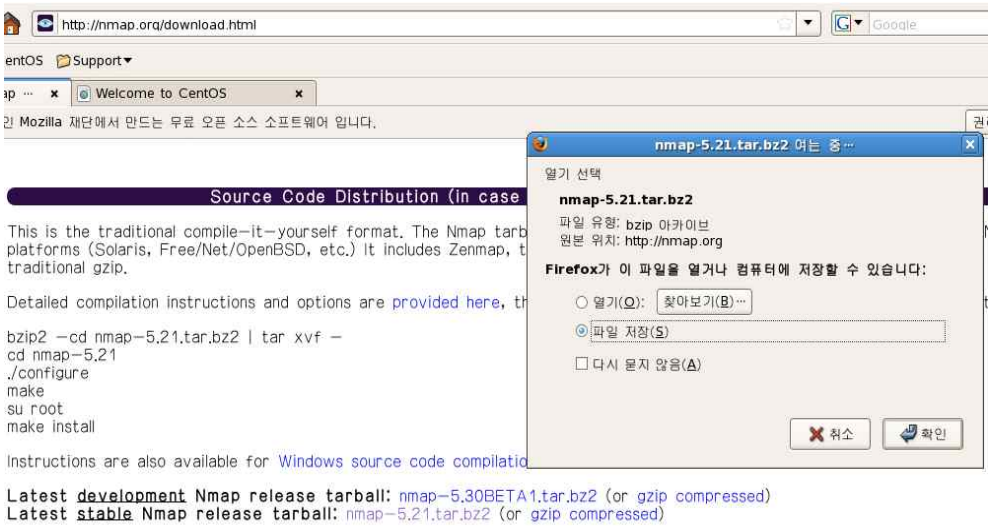
[그림 8] snort 컴파일

같은 방법으로 configure로 Makefile을 만든 후 make; make install로 설치하면 된다. 위와 같이 컴파일하면 snort가 설치된 snort-2.3.x/src 디렉토리와 /usr/local/bin/ 디렉토리에 snort 라는 실행 파일이 생기고, snort-2.3.x/etc 디렉토리에는 snort.conf 등 설정 파일이 생성된다. 그리고 snort-2.2.x/rules 디렉토리에 침입 탐지를 위한 각 종 룰 파일이 위치하게 된다.

3.1.2 nmap 설치 및 설정

Nmap 인터넷 사이트로 이용하여 Nmap을 다운로드 받는다.

http://nmap.org/download.html



[그림 9] nmap 다운방법

나. Configure

다음 명령을 실행하여 인스톨에 필요한 환경을 설정한다.

configure

[그림 10] nmap Configure 환경

Configure 명령이 완료되면 위와 같은 화면을 볼 수 있다.

다. make

make 명령을 수행하여 configure에 의해 만들어진 makefile을 기반으로 컴파일을 수행한다.

make

[그림 11] nmap makefile 기반 컴파일

라. Install

다음 명령을 수행하여 인스톨을 수행한다.

make install

[그림 12] Nmap make install 과정

위와 같은 과정이 정상적으로 완료되어 install이 되면 위와 같은 화면을 볼 수 있다.

만약 make install을 수행하는 과정에서 자신의 계정이 root 계정으로 수행하고 있지 않아 권한의 문제로 설치가 되지 않는다면 다음과 같은 명령으로 통해 계정을 변경하여 수행한다.

3.1.3 Nikto 설치 및 설정

설치방법

1. Nikto 다운로드

wget https://cirt.net/nikto/nikto-2.1.4.tar.gz

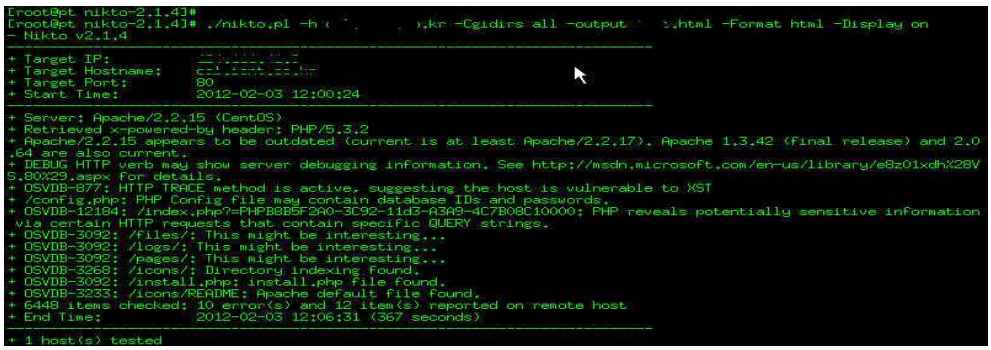
2. 다운로드된 파일을 압축해제 후 실행합니다.

tar xvfz nikto-2.1.4.tar.gz

cd nikto-2.1.4

perl로 실행되므로 perl을 다운로드 합니다.
yum install perl

./nikto.pl -h www.test.com -Cgidirs all -output test_site.html -Format html -Display on
로 실행합니다.



[그림 13] Nikto 다운로드 및 압축해제

 # 설치가이드문서
http://cirt.net/nikto2-docs/installation.html#id2727109

 # 스캔된 웹서버 access_log 화면

[그림 14] 웹서버 스캔화면

2. 과제목표 및 추진체계

2-1-1.개발동기

수동적인 방법 보다 도구를 이용하여 웹 서버의 종류나 버전, 디렉터리 정보나 중요 파일 정보가 존재하는지, 웹 서버 자체의 취약점은 무엇인지 검사하는 것이다. 대상 시스템에 HTTP 요청을 보낸 뒤 이에 대한 응답 코드를 받으면, 그 코드를 분석해서, 요청한 페이지가 존재하는지, 서버의 어느 부분이 취약한 지 등을 알 수 있다. 다만 주의해야 할 점은 현행법상 스캐닝만 하여도 공격으로 간주하여 처벌되기 때문에, 개인 테스트 컴퓨터 또는 친구 또는 주위에서 허가한 컴퓨터를 대상으로 스캐닝 하여야 된다.

2-1-2. 시스템 구축목표

본 연구에서 개발한 리눅스 쉘 프로그램을 이용한 서버 시스템을 사용자가 손쉽게 관리할 수 있으며, 네트워크에 존재하는 서버들의 취약점에 대한 내용을 보고서로 작성하여 그 결과를 손쉽게 확인하고 대비할 수 있는 장점을 가지고 있다.

2-2. 팀 구성원

2-2-1. 구성원

구성원			역할분담
5조	조장	허종만	개발담당, 프로젝트총괄
	조원	장순현	자료 종합 담당, 프로그램 검증 담당
		하상우	자료조사. 자료분석 담당

3. 주간활동보고서

3-1. 조별주간활동보고서

내용 / 주기		주간활동사항	
1주차		· 조원의 역할분담 및 일정계획, 자료수집	
2주차		· 취약점에대한 개인 공부	
3주차		· 각종취약점 분석툴 설치 및 연구	
4주차	허종만	· 프로젝트의 전반적인 틀 구상	· 역할 별로 각자 시스템 부가 프로그램 구현에 대한 회의 및 각각의 계획 수립
	장순현	· 취약점 분석 자료 종합	
	하상우	· 취약점 분석 담당	

<table>
<tr><td colspan="2">5주차</td><td>· 쉘 프로그램 개발
시작하는 메인 화면 구축</td><td> 
</td></tr>
<tr><td rowspan="3">6주차</td><td>허종만</td><td>· 프로그램 메인 화면 구축</td><td> 
</td></tr>
<tr><td>하상우</td><td>· 웹 서버 취약점
· 웹 서버 스캔 조회
· 포트 및 서버버전 스캔
프트 스캔 조회 화면 구축</td><td> 
</td></tr>
<tr><td>장순현</td><td>· IP또는web주소
· type(html,txt,Xml)
· 저장할파일명 화면구축</td><td> 
</td></tr>
<tr><td rowspan="2">7주차</td><td>허종만</td><td colspan="2">· 프로그램 메인 화면 프로그램에 각종 스캔목록 작성</td></tr>
<tr><td>장순현</td><td colspan="2">· 웹 서버 취약점 툴 분석 (nikto,nmap)</td></tr>
</table>

	하상우	· 웹서버 및 포트스캔에관한 취약점공부
8주차	허종만	· 각 프로그램에 전체 흐름 파악
	장순현	· 시스템 오류 검사 및 수정
	하상우	· snort에관한 자료수집
9주차	허종만	· web서버 프로그램 분석
	장순현	· 각종 서버의 대한 자료 조사
	하상우	· Snort 관련 프로그램 구현 
10주차	허종만	중간점검
	장순현	
	하상우	
11주차	허종만	· 쉘 프로그램에 대한 추가디자인 구성
	하상우	
	장순현	
12주차	허종만	· 관리시스템 연동 및 최종 디자인 수정
	하상우	
	장순현	
13주차	허종만	· 각 서버에 디자인 수정 및 오류 체크
	하상우	
	장순현	
14주차		· 발표 준비 및 선 테스트
15주차		· 제 작 발 표

4. 발표자료


5조 'The Creator'
졸업작품 중간발표
조 원 : 허종만(조장), 장순현, 하상우
담당교수 : 양환석 교수님

CONTENTS
I. The Creator 소개
II. The Creator 는 무엇을?
III. The Creator 는 어떻게?
IV. The Creator 진행사항
V. The Creator 결론
VI. Q&A


누가?

누가? - 조원 및 역할 소개 -
조원 역할
허종만 조장, 개발 담당, 프로젝트 총괄
장순현 자료 종합 담당(ppt), 프로그램 검증 담당
하상우 자료조사(snort, 각종서버), 자료 분석 담당


무엇을?

무엇을? - 주제 소개 -
ESM (Easy Security Management) 제작
• 웹 서버 취약점 분석
• Snort Log파일 관리
• 포트스캔을 통한 취약점 분석

왜?

왜? - 주제 선정 이유 -

사용자가 여러 서버를 사용할 때 관리자는 툴을 이용하여 취약점 분석을 한다. 이러한 취약점을 분석할 때 복잡한 명령어 없이 다수의 툴을 한번에 사용할 수 있고 이에 대한 결과값이 복잡하게 나오는데 그 결과값을 보고서 형식으로 보기 좋게 정리할 수 있도록 도와주는 프로그램을 만드는 것이 목적입니다.

어떻게?

어떻게? –개발 환경-

운영체제 : CentOs 5.3

프로그래밍 언어 : Bash Shell

사용도구 : Nmap , Nikto2, Sscan , Snort

어떻게? - 용어설명 -

Snort 란 무엇인가?

Snort는 "sniffer and more" 라는 말에서 유래되었는데, 처음 공개되었을 때는 코드도 얼마되지 않는 단순한 패킷 스니퍼 프로그램이였다. 그러나 이후 현재의 IDS와 같이 rule을 이용한 분석 기능이 추가되고, 커뮤니티를 통해 계속적인 기능 보완과 향상을 이룩해 지금과 같이 다양한 기능과 탁월한 성능을 갖춘 프로그램이 됐습니다.

Nikto2 란 무엇인가?

Nikto 웹 스캐너는 위험한 파일과 오래 된 서버의 소프트웨어 및 기타 문제에 대 한 웹 서버를 테스트해 특정검사를 수행합니다.

Nmap 란 무엇인가?

네트워크 보안을 위한 유틸리티로, 대규모 네트워크를 고속으로 포트 스캔하는 툴. 네트워크 탐색 툴이며 동시에 보안 및 포트 스캐너

어떻게?

어떻게?

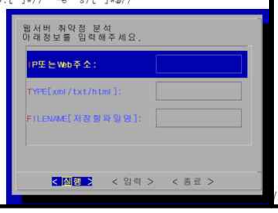
```
$DIALOG --default-item "OS/2" --clear --title "Easy Security Management" \
        --menu "The Creator \n\n
\n\n
\n\n
    ESM(Easy Security Management) \n\n
 \n\n
 \n\n
 \n\n
          Choose the Scanning you like:" 20 51 4 \
        "[1]"  "Server Scan" \
        "[2]"  "Snort  Scan" \
        "[3]"  "종 합 스 캔" 2> $tempfile
retval=$?
choice='cat $tempfile'
case $retval in
  0)
    ./$choice.dia
    ;;
  1)
    echo "Cancel pressed.";;
  255)
    if test -s $tempfile ; then
      cat $tempfile
    else
      echo "ESC pressed."
    fi
    ;;
```

어떻게?

```
exec 3>&1
value='$DIALOG --extra-button --extra-label "이전" \
        --default-item "" \
        --clear \
        --title "Easy Security Management" \
        --menu "The Creator \n\n
\n\n
\n\n
    ESM(Easy Security Management) \n\n
 \n\n
 \n\n
 \n\n
          Choose the Scanning you like:" \
        "Web"  "웹서버 취약점 스캔" \
        "web"  "웹서버스캔조회" \
        "Port"  "포트 및 서버버전 스캔" \
        "port"  "포트스캔조회" 2>&1 1>&3'
retval=$?
exec 3>&-

case $retval in
  0)
    ./$value.dia;;
  1)
    echo "프로그램 종료.";;
  2)
    echo "Help pressed ($value)";;
  255)
    if test -n "$value" ; then
      echo "$value"
```

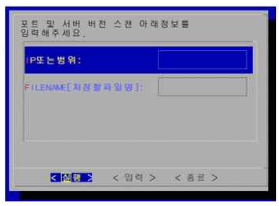
어떻게?

```
perl /nikto-2.1.5/nikto.pl -h $ip -Format $type -output /zolzak/webs/$file.$type >> /de
null

        ;;

        1)
                returncode=99
                ;;
        esac
                ;;
        3)
                tag=`echo "$value" |sed -e 's/"RENAMED //' -e 's/:.*//'`
                item=`echo "$value" |sed -e 's/.*[ ]*//' -e 's/[ ]*$//'`

                case "$tag" in
                IP또는Web주소)
                        ip="$item"
                        ;;
                TYPE₩(xml/txt/html₩))
                        type="$item"
                        ;;
                FILENAME₩(저장할파일명₩))
                        file="$item"
                        ;;
                esac
                ;;
]       255)
                echo "프로그램 종료."
                break
                ;;
```



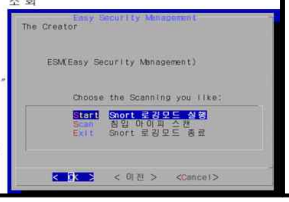
어떻게?

```
"$DIALOG" ₩
--clear ₩
--backtitle "$backtitle" ₩
--yesno "Port & Sever Version Scan
-아이피:        $ip ₩n₩
-파일명:        $file ₩n₩
               " 10 40
case $? in
0)
  left=5
 unit="seconds"
 while test $left != 0
 do
sleep 1
$DIALOG --title "$backtitle" ₩
 --infobox "  현재 스캔중입니다 잠시만 기다려주세요
You have $left $unit ..." 10 52
left=`expr $left - 1`
test $left = 1 && unit="second"
done

nmap -sV $ip |egrep "^[1-9]|interesting|Service" | grep -v "filtered" >> /zolzak/
$file
```

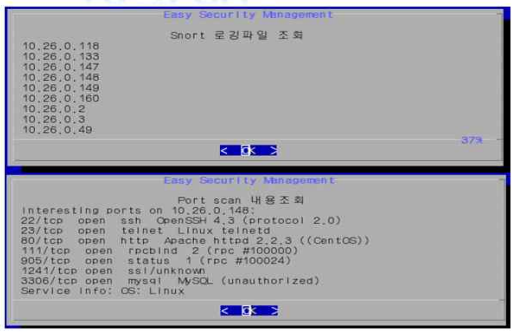


어떻게?

```
case $retval in
  0)
        if [ $value = "Start" ]
        then
            snort -vde -K ascii -D >> /dev/null
             echo "실행되었습니다"
             sleep 1
            ./[2].dia
        elif [ $value = "Scan" ]
        then
                $DIALOG --title "Easy Security Management" --clear ₩
--msgbox "
                Snort 로깅파일 조회
`ls /var/log/snort`  " 15 70
case $? in
        0)
                ./[2].dia ;;

        255)
                echo "프로그램 종료"
        esac

        elif [ $value = "Exit" ]
        then
                killall snort
                echo "종료되었습니다."
                sleep 1
                ./[2].dia
        fi
```



어떻게?



결 론!

결 론! - 특징 -

➢장 점

1. 각종 서버를 손쉽게 관리할 수 있다
2. 보고서 작성시 쉽고 보기 좋게 취약점 결과를 볼 수 있다.
3. GUI형식으로 명령어 없이 쉽게 사용할 수 있다.

➢단 점

1. 관리해야 할 서버의 취약점을 수동으로 관리해야 한다.
2. 사전의 설치해야 할 툴이 있다.

결 론! - 향후방향 -

➢향후 방향

- 관리해야 할 서버 종류의 폭을 확대한다.
- 디자인을 개선하여 보기 좋고 사용자가 쉽게 이용할 수 있도록 한다.
- 사전 설치 없이도 사용이 가능한 프로그램 만들기

Thank you !

Any Questions?

5. 참고문헌

1. 뇌를 자극하는 RedhatFedora 리눅스서버&네트워크 우남재저 한빛미디어(2007.8.5.)

2. 리눅스 쉘 스크립트 프로그래밍 송인우저 도서출판 대림 (2003.10.1.)

3. Centos 기반의 리눅스 서버구축 및 활용하기 DIM연구소저 (2011.8.10.)

4. BackTrack4 샤킬알리 . 테디 헤리얀토저 (2011.7.29.)

6. 결　　론

- 각종 서버를 손쉽게 관리할 수 있다.
- 보고서 작성시 쉽고 보기 좋게 취약점 결과를 볼 수 있다.
- GUI형식으로 명령어 없이 쉽게 사용할 수 있다.