전산망보안점검 도구의 설계 및 구현
Design and Implementation of Network Security Scanner for Analyzing UNIX Vulnerabilities

*변경근, *심영철, **신훈, **임휘성, **임채호
*홍익대학교, **한국정보보호센터

요약

인터넷에서의 보안 취약점을 점검분석하기 위한 방법으로서 네트워크를 통해 외부에서 시스템의 문제점을 점검 보고하는 도구를 WWW 서버의 CGI 를 이용하는 시스템형태로 설계하고 구현하였다.

1. 서론

인터넷에서는 유닉스시스템들이 보안에 가장 취약하며 특히 네트워크에 문제점이 많아 외부 침입자들이 손쉽게 시스템을 공격하고 있다. 시스템관리자는 자신이 관리하는 시스템의 보안 문제점들을 수시로 점검하여 보안문제점을 막음으로서 시스템을 안전하게 관리하고자 한다. 시스템의 보안문제점을 점검하고자할 경우 보통 COPS[22], Tiger[23], Crack[24], Tripwire[25] 등의 보안도구들을 이용하여 예방적인 측면의 결과를 얻을 수 있지만, 전산망 외부에서의 불법적인 접근에 대한 구체적인 점검이 어려울 수 있다. 해커 등 외부 전산망 침입자들의 시각에서 보다 실질적인 공격시도를 통한 점검이 아니기 때문인데, 최근 발표된 SATAN[26], ISS[27] 등은 보안 점검을 외부 전산망에 존재하는 해커의 입장에서 보다 실질적인 점검, 보고를 하는 도구이므로 가치가 있다.

한편 국내에서는 시스템관리자들이 보안에 대한 관리기술력이 매우 낮은편이며, 또한 시스템보안관리에 대한 인식과 시간 투자가 어려우면서도 관리할 시스템 수는 매우 많은 편이다. 그러므로 관리자가 모든 시스템 보안관리가 어려워져 전산망의 보안이 매우 허술해지고 있어 더욱 외부 침입자에 대한 방지 대책을 구현관리하는데 문제가 발생한다.

본 논문에서는 어떤 기관의 관리자가 전산망을 관리하는데 있어 전산망에 연결된 시스템들의 보안문제점들을 일일이 개별 보안점검하는 것이 아니라 하나의 중앙관리시스템에서 여러 시스템들의 보안문제점들을 점검하는 도구를 설계하였으며, 실제 구현은 WWW서버의 CGI 인터페이스로 구현하여 실제 보안점검을 요구하는 사용자 측며의 관리자들이 자신의 시스템을 점검하고 보고받을 수 있는 형태로 구현하였다.

이를 위해 기존에 개발되어 알려진 ISS, SATAN 등 관련 스캐너에 대한 분석을 하였으며, 시스템 요구사항 및 설계, 실제 구현을 진행하였다. 그리고 마지막으로 향후 개선 방향 등에 대해 보이고자 한다.

2. 기존의 보안스캐너 분석

보안 스캐너에 대한 분석을 하는 이유는 현재 소개되고 있는 보안스캐너들이 해커들을 흉내낸 자동 해킹에 의한 시스템 보안 점검을 하므로 이러한 분석을 통해 실제적인 해킹 수법들을 파악할 수 있기 때문이며, 본 연구과제와 연관된 선행연구라고 볼 수 있기 때문이다. 1995년 발표되어 전세계에 큰 반향을 일으킨 SATAN 을 비롯한 공개도구들과 최근 국내에서 개발된 것들 등을 분석하고자 한다.

2.1 SATAN[26]

Dan Farmer와 Wietse Venema가 만든 SATAN(시스템 Administrator Tool for Analyzing Networks)은 원격 시스템을 시험한다는 것을 제외하고는 COPS(Computer Oracle and Password 시스템)와 그 원리가 같다. 따라서 이 프로그램을 작동시키는 모든 사용자는 다른 시스템의 보안상 문제점을 발견할 수 있다. SATAN은 네트워크를 통하여 원격 시스템의 보안정도를 조사하고 그 자료를 데이터베이스에 저장한다. 이 결과를 http 프로토콜을 지원하는 HTML 브라우져

(browser)를 통하여 쉽게 볼 수 있으며, 호스트의 타입, 서비스, 취약요소 등으로 분류, 설명된 보고서를 만들어 낼 수도 있다. 보통 SATAN의 실행은 설정파일(config/satan.cf)에 의하여 제어된다.

SATAN은 문제를 발견하면 검색기 상에 그것이 왜 문제가 되는지와 어떻게하면 문제를 해결할 수 있는지를 알려준다. 하이퍼링크(hyper link)로 연결된 보안대책을 제공하며, 점검할 수 있는 보안취약점은 다음과 같다.

- NFS가 일반사용자 프로그램에 대해 export되는지 점검
- NFS의 Worldwide 외부 개방(Export) 점검
- portmap을 통해 NFS가 외부에 개방되어 있는지 점검
- NIS의 패스워드 파일 접근 여부 점검
- REXD 접근 가능 점검
- Sendmail 취약성(8.6.9이하 버전) 점검
- TFTP의 파일 접근 가능성 점검
- rsh의 접근 가능성 점검
- X 서버로의 접근 가능성 점검
- 쓰기권한이 있는 FTP의 홈 디렉토리 점검

2.2 ISS (공개용)[27]

최초의 multi-level 보안 점검 스캐너이며, 상용제품은 모든 알려진 취약성과 방화벽, 서비스거부(Denail of Service)를 검사하는 반면 공개제품은 명백하고 단순한 취약성만을 검사한다. 점검 항목은 다음과 같다.

- 처음 telnet 포트로 연결을 시도, 연결이되면 그 호스트가 보여주는 모든기록을 로그한다.
- 대부분의 시스템에 있는 'sync' 라는 계정으로 로그인을 시도한다.
- mail 포트로 접근하여 OS type과 호스트이름, sendmail 버전을 알아낸다.
- mail aliases와 wiz 뒷문프로그램을 조사한다.
- ftp 포트에 익명(Anonymous)으로 접근하여 홈 디렉토리의 쓰기 취약성을 조사한다.
- rpcinfo를 통해 NIS, rexd, bootparam, NFS, selection_svc가 수행중인지 본다.
- NIS 서버에 대해 ypserv를 이용하여 추측한후 패스워드를 가져온다.
- selection_svr이 있는지 확인한다, (아무나 외부에서 시스템 파일을 읽을 수 있다)
- Rexd가 있는지 확인한다.
- 모든포트를 조사하여 gopher, mud등과 같은 가능한 접근목록을 찾는다.
- showmount -e 이용하여 NFS의 export 여부를 점검한다.

2.3 Internet Scanner SAFEsuit (상용)[29]

기존의 네트워크 보안 평가 툴인 ISS를 기반으로한 툴이며, 웹서버(web site), 방화벽(Firewal)I, 유닉스, Win95, WindowNT, 워크스테이션에 이르는 140가지의 취약성을 검사한다. ISS상용시스템의 구성은 다음과 같다.

- 웹보안 점검스캐너 : 100가지이상의 알려진 웹서버 취약점을 점검
- 방화벽 점검스캐너 : 100가지 이상의 알려진 방화벽 취약점을 점검
- 인트라넷 점검스캐너 : 인트라넷 환경의 취약요소 점검
- 시스템 점검스캐너 : 시스템에서의 취약요소 점검

이 ISS 제품의 특성은 다음과 같다.

- 많은 보안 취약성을 검사한다. (140가지 이상)
- 자동적으로 운영 및 구성가능한 보안점검 스캐너
- GUI 제공 (Window NT GUI, Motif UNIX GUI, HTML)
- 새로운 취약성이 발견될때마다 버전을 올릴 수 있음
- 여러대의 시스템들을 동시에 점검할 수 있음
- 대상 네트워크를 명시하여 해당 네트워크 시스템만을 점검할 수 있음

실제 검사 항목으로서는 다음과 같은 것들이 있다.

- 웹서버의 문제 여부
- 방화벽 문제 여부
- UNIX TCP/IP 서비스 취약성
- WindowNT, Window95 취약성
- 계정 문제점

- 서비스거부공격(Denial-of-service) 점검
- 라우터 취약성

2.4 PINGware[30]

TCP/IP 네트워크상의 보안 취약성을 감지하는 Bellcore사의 제품이다. 백그라운드로 실행되어 효율적으로 실행하는데, 점검 항목은 다음은 다음과 같다.

- ftp
- rsh (/etc/hosts.equiv파일 조사)
- tftp
- sendmail (wiz, debug옵션 검사)
- NFS
- xhost

2.5 Smash[28]

이 프로그램은 KAIST의 김휘강이란 학생이 만든 취약점 점검도구로서 주로 내부 시스템을 실제로 해킹점검하는 방법으로 시스템의 취약점을 보고하며 검사하는 취약점은 다음과 같다.

- autotreply : 실제로 공격 진단
- bootparam : "rpcinfo -p"로 점검
- loadmodule : 실제로 공격 진단
- portscan : SATAN의 방식을 이용
- rdist : 실제로 공격 진단
- rexd : "rpcinfo -p"로 점검
- rlogin : 실제로 공격 진단
- rstatd : "rpcinfo -p"로 점검
- sendmail : 실제로 공격 진단
- yp-passwd : ypcat 이란 프로그램을 이용한 공격 진단
- X server : SATAN의 방식 이용

2.6 UVchecker 1.0[3]

한국전산원의 95년도 "유닉스 시스템 보안 취약성 분석 및 진단에 관한 연구"의 부속물로서 개발된 UVchecker는 시스템 검색과 네트워크 검색으로 나누어진다. 현재 SUN과 Solaris에서 동작가능하고 tcl/tk를 이용한 GUI를 가지고 있다. 그러나 실제 해킹 시도·점검하는 도구가 아니라 취약점 진단이 목적인 프로그램으로서 검사하는 취약점은 다음과 같다.

(1) 시스템 취약점
- rhost
- hosts.equiv
- autoreply (화일의 존재와 모드 점검)
- rdist (화일의 존재와 mode 검사)
- loadmodule(화일의 존재와 모드 점검)
- psrace(tmp 디렉토리 접근 권한 점검)

(2) 네트워크 취약점
- tftp 검사 : 실제 공격
- rexd : "rpcinfo -p"로 점검.
- rsh : rsh 명령 수행 후 진단
- X server : XopenDisplay()를 호출
- sendmail : 8.6.12이전 버전인지 아닌지 점검
- NIS 검사 : "rpcinfo -p"로 NIS 관련 프로그램이 수행되고 있는지 점검
- NFS : "showmount -e" 의 결과로 점검

2.7 비교 요약

다음 <표 1>은 각 스캐너가 어떤 보안 취약점들을 점검하는지 비교 요약한 것이다. 이 표를 토대로 본 전산망보안점검도구에서 점검해야할 항목들을 고려해 본다.

[표 1] 각 보안 스캐너 비교 요약표

* SAT:SATAN, ISP:ISS(공개), ISC:ISS(상용), PGW:PingWare, SMS:Smash, UVC: UVChecker

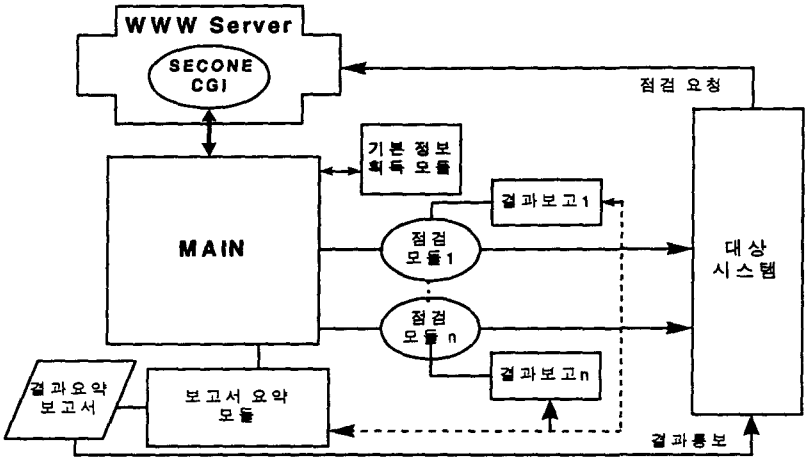
취약점 점검 항목/내용	SAT	ISP	ISC	PGW	SMS	UVC	
.rhost, hosts.equiv	X	X	X		X	X	
autoreply bug					X	X	
ps race						X	
rdist bug					X	X	
loadmodule bug					X		
rstatd			X		X		
weak 패스워드	X	X	X		X		
sendmail 문제	X	X	X	X	X	X	
rxed 문제	X	X	X		X		
tfp	X	X	X	X		X	
X 서버	X		X	X		X	
NFS		X	X	X	X		X
bootparamd	X	X	X		X	X	
NIS	X	X	X			X	
rlogin		X	X	X	X		
sync login	X	X	X				
rusers	X	X	X				
rpc check(rpcinfo -p)	X	X	X		X		
fingerd	X	X	X	X			
ftp	X	X	X	X			
modem 체크	X						
NCSA httpd V1.4.1			X				
rsh		X	X	X	X		X
Selection_svc		X	X				
UDP Bomb			X				
X.25			X	X			
IP Spoofing			X				
소스포팅, 라우팅			X				

3. 설계

3.1 요구사항 분석

3.1.1 전산망보안점검도구의 개략적 개념

다음 <그림 1>은 전산망 보안점검도구의 개략적 개념도를 보여주고 있다. 이 그림에서 보듯 전산망 국내의 보안관리자는 사용자로서 자신의 시스템을 스스로 점검하고자 이 점검 서비스가 제공되는 WWW 서버에 접근하게되고 적당한 정보를 입력함으로서 자신의 시스템 보안 문제를 점검받고 결과를 통보받을 수 있다.



<그림 1> 전산망보안점검 서비스 개념도

3.1.2 사용자 요구사항
사용자는 보다 편리하고 안전한 방법으로 자신의 시스템의 보안을 점검하기를 원한다. 사용의 편이성을 위해 1)기존의 넷스케이프 인터페이스를 이용하며, 2)사용자들이 새로운 프로그램설치는 없으며, 3)그래픽인터페이스로 사용의 편이성을 높힌다. 또한 사용의 안전성을 위해 1)타 시스템 보안점검 행위 방지, 2)시스템 취약성 정보 유출 방지 등을 고려한다.

3.1.3 관리자 요구사항
이 시스템을 동시에 다수의 사용자가 보안점검 서비스를 이용할 수 있으므로 이를 고려하여야 한다. 이때 악의의 사용자에 의한 보안점검 서비스의 남용을 방지하여야 한다. 또한 누적된 시스템 보안점검 결과를 이용하여 관리하에 있는 전산망내 시스템의 보안상태에 대한 통계를 알고자 한다. 관리자들을 위해 고려할 사항은 1)단일의 서버에 의한 보안점검 수행 및 결과의 보관, 2) 다중 사용자 환경에의 적용, 3)접근 제어를 통한 서비스 이용 제한, 4)사용 빈도 제한을 통한 서비스 남용 제한, 5)관리자용 통계처리 기능의 지원 등이다.

3.2 도구 설계

전산망 보안점검 도구는 위와같은 사용자 요구사항 분석을 거쳐 편이성, 확장성, 안전성을 고려해 전체적인 구조를 설계하였다.

3.2.1 기본 구조
전체 보안점검 도구의 인터페이스 및 기능은 월드와이드 웹 기술을 활용하여 구성하도록 한다. 즉, 사용자 인터페이스는 HTML을 이용하며, 전체 프로그램은 CGI를 이용하여 구성한다. 이것은 원격 시스템상에 있는 보안점검 도구를 기동시키고 필요한 정보를 제공하여 실제 보안점검 기능을 실행시키기에 가장 용이한 방법이다. 또한 대다수의 시스템에 월드와이드 웹 브라우저가 설치되어 있으며 이미 그 인터페이스에 익숙해져 있기 때문에 실제 사용자의 입장에서도 가장 적합한 구조이다. 또한 개발자의 입장에서도 사용자 인터페이스의 구성 및 변경이 매우 자유롭고, 잘 정의된 프로그램 인터페이스에 의해 사용자 인터페이스의 구조와는 다소 독립적으로 실행 프로그램을 개발할 수 있으며, 사용자와의 통신에 관련된 문제를 대부분 웹 서버-클라이언트 구조가 해결해 줄 뿐만 아니라 인쇄, 브라우징, 스크롤링등의 모든 부수적인 기능들을 클라이언트측에서 이미 보유하고 있기 때문에 전체적인 프로그램 개발이 용이해 진다.

사용자 인터페이스는 HTML을 이용하여 구성한다. 보안점검도구의 대부분의 기능이 사용자에 의해 초기에 제공된 기초 정보를 이용하여 보안점검프로그램이 스스로 세부 정보를 획득하여 점검을 수행하여 결과를 보고하도록 구성되어 있으므로 안내 및 정보의 제공, 초기 정보의 입수 및 확인 등에 관련된 부분에 제한된다. 보안점검 도구의 전체적인 구성은 다음과 같은 기능별 모듈들로 이루어 진다

- 사용자 정보의 획득
- 보안 취약점 점검
- 결과 보고서 전송
- 입력 정보의 확인 및 사용자 검증
- 결과 보고서 작성

3.2.2 사용자 정보의 획득
사용자가 사용자 시스템에서 웹 브라우저를 이용하여 보안진단 서비스에 접속하면, 대상 시스템에 대해 초기 접근하기 위해 필요한 기초 정보들과 보안점검결과의 배포등에 필요한 사용자 정보를 요청한다. 사용자의 이용 편의성과 사용자의 검증을 위해 필요한 정보간에 타협하여 입력 정보를 제한한다.

3.1.3 입력입력 정보의 확인 및 사용자 검증 기능
보안점검도구의 안전한 사용을 위해, 보안점검도구는 사용자에 의해 입력된 시스템 및 사용자 관련 정보와 네트워크 환경상에서 실제적을 입수된 정보들을 비교 분석하여 서비스를 요구하는 사용자의 적법성을 판단하여 제3자의 시스템의 보안 취약성을 점검하는 등의 부당한 사용을 방지한다. 별도의 사용자 인증 기능을 도입하여 사전에 등록된 사용자 만이 서비스를 이용할 수 있도록 할 수도 있다.

3.2.3 보안 취약점 점검 기능
실제 구조에 있어 보안점검도구의 핵심은 시스템에 대한 실제 공격을 수행하여 취약점을 점검하는 취약점 점검 모듈이다. 이 취약점 점검 모듈은 향후의 취약점 발견시 지속적인 점검 기능 추

가를 위해, 각 취약점별로 공격하여 취약성 여부를 판별하는 공격 모듈들과, 이들을 관리하는 공격관리 모듈로 구성된다.

여기에 추가하여, 공격 모듈에 의해 실제적인 공격을 통한 보안진단 기능의 수행을 위해 필요한 각종 정보를 수집하기 위한 정보수집 모듈이 있다. 정보수집 모듈은 사용자 인터페이스를 통하여 입수된 정보를 이용하여 대상 시스템에 접근하여 이후의 실제적 공격을 통한 보안진단 기능의 수행을 위해 필요한 각종 정보를 수집한다. 이와같은 정보는 공격관리 모듈에 의해 각 공격 모듈에 제공되어 점검 대상 시스템에 직접 접근을 시도하여 시스템의 취약성을 판단하게 된다.

공격관리 모듈은 주어진 정보를 이용하여 각각의 공격 모듈을 수행시켜 이들에 할당된 취약점들을 점검하게 한다. 각 공격 모듈에 의해 시스템의 취약성이 판단되면 공격관리 모듈은 그 결과를 요약된 형태의 점검결과 보고서로 작성한다. 이 점검결과 요약보고는 결과 보고서 작성 모듈에 전달되어 최종 보고설를 작성하는데 사용된다. 또한 이 점검결과 요약보고는 추후에 별도의 툴을 이용하여 완성된 결과보고서를 재작성하거나, 통계의 처리 등에 활용될 수 있다.

3.2.4 보안점검 결과보고서 작성 기능
결과보고서처리 모듈은 공격관리 모듈에 의해 작성된 요약보고를 이용하여 최종 결과보고설를 작성한다. 이 보고서에는 다음과 같은 정보들이 포함된다.

- 점검 대상 시스템 이름
- 점검 대상 시스템 도메인 이름
- 점검 대상 시스템 IP 주소
- 점검 시간
- 취약점 목록 - 취약점 이름(Vulnerability Name)
  - 취약점에 대한 CERT/8lgm 권고서(Advisory Document Number)
  - 취약점 심각도(Severe Level)
  - 취약점 설명(Summarised Description)
  - 취약점 CERT/8lgm 권고문서 본문(Full Description)

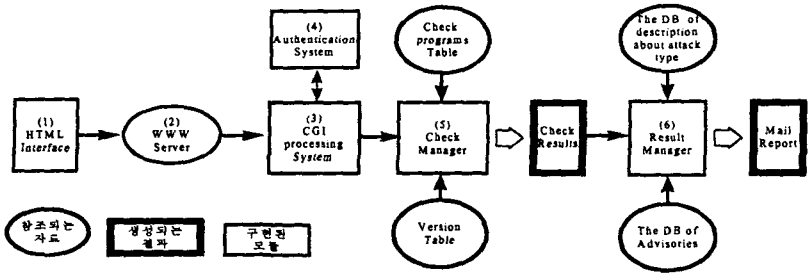
3.2.5 보안점검 결과보고서 전송 기능
작성된 보안점검 결과보고서는 즉시 전자우편을 이용하여 점검을 요청한 사용자에게 전송되며 시스템의 자원을 절약하기 위해 전송이 성공하면 즉시 삭제한다. 만약 결과 보고서를 다시 작성해야할 필요가 있을 경우, 별도의 유틸리티를 이용하여 공격관리 모듈에 의해 작성되어 보관하는 요약보고로부터 다시 작성할 수 있다.

3.2.6 기타
본 보안점검 도구는 다수의 클라이언트의 요청에 따라 하나의 서버에서 모든 작업을 수행하게 되므로 멀티 유저 환경에 적응하도록 구현되어야 한다. 예를들며, 특정 사용자가 중복 또는 단 시간 내에 반복하여 서비스를 요청하여 접속이 가능한 사용자의 수를 초과하게할 경우, 다른 사용자가 서비스를 사용하지 못하게 되므로 동일 사용자가 일정한 시간내에 반복하여 서비스를 요청할 경우, 이를 거부하도록 해야한다. 또한 다수의 사용자가 동시에 서비스를 이용할 경우, 보안점검 도구가 작성하는 임시 파일, 결과 파일등이 중복되지 않도록 해야할 것이다.

4. 구현

4.1 전체 구조

RVCS(원격 Vulnerability Check 시스템)의 전체적인 구조는 <그림 2>과 같다.



<그림 2> RVCS 구조

4.2 HTML 인터페이스

HTML 인터페이스는 RVCS(원격 Vulnerability Check 시스템)에 관한 소개와 점검 대상 기계의 Network hole을 찾아내기 위해 필요한 정보를 얻는데 그 목적이 있다. 필요한 정보의 종류로는 E-mail 어드레스, IP 어드레스, 기계이름, 공격 수준, OS 버전 등이다.

- E-mail 어드레스 : 점검 대상 기계를 점검한 후 결과를 통보할 주소
- IP 어드레스 : 점검 대상 기계의 IP 주소 (예 : 203xxxx..249.xxx)
- 기계 이름 : 점검 대상 기계의 공식적 이름 (예 : example.xxx.co.kr)
- 공격 수준 : 수준이 높을수록 충분히 점검한다. (예 : level1, level2, level3)
- OS 버전 : 점검 대상 기계의 OS version. 문제점을 점검하는 각각의 프로그램들은 OS 버전에 의존적인 점검 방법들을 가지고 있다.

4.3 WWW 서버

WWW 서버는 HTML 인터페이스를 통해서 얻은 정보를 CGI 처리 시스템에 전달한다.

4.4 CGI 처리 시스템

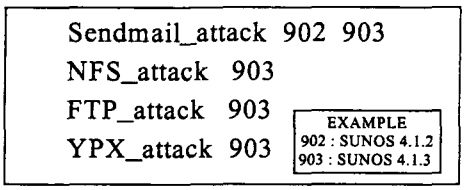
WWW 서버로부터 넘겨받은 정보를 이용하여 인증 시스템을 실행시켜서 그 값이 참이면 취약점 검사 모듈를 실행시킨다. 그렇지 않고 거짓이면 진단 작업의 진행을 중단시킨다. 취약점 검사 모듈를 실행시킬 경우 어느 정도 공격이 이루어져 있는가에 대한 상태를 HTML 인터페이스를 통하여 사용자에게 알려준다. 그리고 동시에 여러 사용자가 점검을할 경우를 대비하여 취약점 검사 모듈에게 자신의 process id와 현재 시간을 파라미터로 넘겨준다. 왜냐하면 취약점 검사 모듈는 각각의 점검 프로그램들이 점검 도중에 생기는 중간파일이나 전체 결과를 담은 파일들의 이름을 만들 때 이들 정보를 이용함으로써 유일성(uniqueness)을 보장할 수 있다.

4.5 인증 시스템

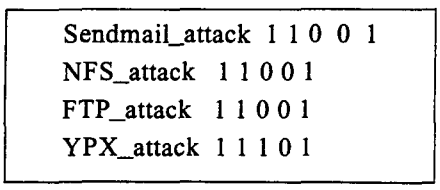
CGI Processing 시스템으로부터 받은 정보 중에서 IP 주소와 기계 이름을 가지고서 생성한 IP 주소와 CGI 환경 변수를 이용하여 구한 현재 접속중인 기계의 원격 IP 주소 등, 이 세 가지 IP 주소를 비교하여 같으면 참을 돌려주고 그렇지 않으면 거짓을 돌려준다. 인증 시스템의 결과가 참이라는 것은 현재 점검을 요청한 사용자가 현재 점검할 대상 기계를 사용하고 있다는 증거이다. 결과적으로 인증 시스템을 사용함으로써 다른 기계를 해킹할 목적으로 다른 기계를 점검 대상으로 삼고, RVCS(Remote Vulnerability Check System)을 악용하는 것을 막아준다.

4.6 취약점 검사 관리자

CGI 처리 시스템으로부터 E-mail 주소, IP 주소, 기계 이름, 공격 레벨, OS 버전 등을 넘겨받는다. 이들중 특히 OS 버전을 보고서 어떠한 공격이 가능한 가를 Version Table(<그림 3>) 에서 참조하여 각각의 점검 프로그램을 골라낸 후 차례대로 실행을 시킨다. 점검 프로그램들을 실행시킬 때에는 그 프로그램이 어떠한 정보가 필요한 지를 Check Programs Table에서 참조하여 필요한 정보들을 파라미터로 넘겨준다. Check Programs Table은 <그림 4>과 같다.



<그림3> Version Table



<그림 4> Check Program Table

<그림 3>에서의 한 항목에 대한 해석은 <그림 4>의 형식으로 만약 1 이면 그 에 해당하는 정보가 필요한 것이고 0 이면 그 에 해당하는 정보가 필요 없는 것이다. 이러한 식으로 하나 하나의 점검 프로그램을 실행시킨 후 그 결과를 Check Results 파일에 담는다. <그림 6>는 Check

Results 파일의 예이다. 1은 그 점검 내용에 문제점이 존재한다는 뜻이고 2 는 문제점이 존재하지 않는다는 뜻이다. 그러므로 <그림 6>와 같은 결과를 보이면 NFS_attack, FTP_attack, YPX_attack은 문제점을 발견한 것이다. 반면에 Sendmail_attack은 문제점을 발견하지 못한 것이다. 그리고 만약 0 이라는 결과가 나타나면 현재 네트워크의 과부화나 고장으로 제대로 문제점 여부를 점검하지 못한 것이다.

senmail_attack	공격 실행 파일 이름
1	File Name(Extension PID & Time)
1	IP Address
0	Machine Name
0	NIS Server
1	OS Version

1 : 필요함 , 0: 불필요함

<그림 5> Check Program Table Format

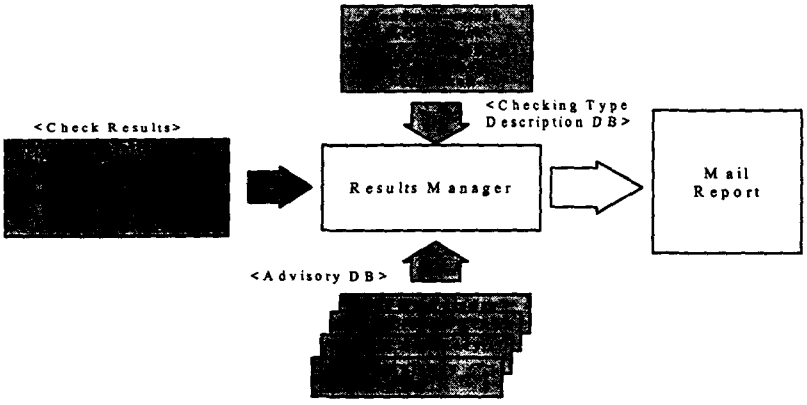
Sendmail_attack 2
NFS_attack 1
FTP_attack 1
YPX_attack 1

0 : 점검 자체의 실패
1 : Hole 이 존재
2 : Hole 이 없음

<그림 6> Check Results 사례

4.7 결과 관리자

결과 관리자는 취약점 검사 모듈가 생성한 Check Results를 어떠한 점검이 행하여 졌는지는 점검 항목에 관련된 DB를 참조하고 문제점을 발견한 점검 항목에 대하여서는 각각의 권고 DB를 참조하여 Mail Report를 작성하여 점검 의뢰인에게 mail로 보내준다. 그리고 Check Results에서 0 으로 표시된, 제대로 실행되지 못한 점검 프로그램들이 만들어 놓은 중간파일들을 지운다.(만약 제대로 실행되면 각각의 점검 프로그램들이 자기 자신의 중간파일들을 지운다. 결과 관리자의 구조는 <그림 7>과 같다.



<그림 7> 결과 관리자

5. 결론 및 향후 발전 방향

이 논문을 통하여 한국정보보호센터가 준비하는 온라인 보안 점검 및 서비스를 위한 전산망보안 점검 도구의 개발을 구현하고자 하였다. 이를 위해 기존에 알려진 유닉스 보안 취약점 및 해킹 수법 분석이 이루어졌으나 이 논문에서는 밝히지 았았으며, 시스템의 보안 안전성 점검을 위한 기존에 개발된 스캐너들을 분석하고 비교하였으며, 이를 기반으로 네트워크를 경유한 보안 문제점 점검방법론을 연구하였다.

다만 이 연구개발에서는 실제 점검 모듈들을 개별적으로 동작하도록 시험한 후 이 시스템에 통합하였으며, WWW서버에 CGI 를 통해 사용자가 직접 온라인으로 자신의 시스템 점검을 신청하도록 환경을 바꾸어 동작하고 이러한 일련의 작업들이 자동적으로 이루어질 수 있도록하면서 신청자에게 요약 및 대책을 전달해주도록 하였다.

이 도구의 개발을 통해 기존에 스캐너들이 점검하는 항목들은 모두 지원하도록 하였으며, 추후 새롭게 발표되고 시도되는 해킹 및 침해 수법들은 점검모듈로 하나씩 추가할 수 있도록 확장성을 꾀하였다.

참고문헌

[1] 한국정보보호센터, 정보시스템 해킹 현황 및 대응, 1996. 11
[2] 한국전산원, 유닉스 보안 취약점 분석 및 진단에 관한 연구, 1995. 12
[3] 한국전산원, 인터넷/유닉스 관리자를 위한 보안기술 지침서, 1995
[4] 한국정보보호센터, 정보보호현황, 1996. 10
[5] 노정석, 김휘강, 조용상, 최재철, "해킹의 최신 형태와 방지 테크닉", 월간 인터넷, 1996.
[6] 이석찬, "Hackers and His Attacks", 개인 자료, 1995.
[7] KUS, PLUS, "인터넷 침입 수법과 대응 방안", NETSEC-KR96 특강 자료집, 1996
[8] 임채호 외, 관리자를 위한 인터네트 보안지침서 Version 1.4, 시스템공학연구소, 1995.
[9] 김진식, 김희상, 장환용, 김영민, 해커들의 해킹기법, 연암출판사, 1993.
[10] 오영희 외, Security+ for UNIX, PLUS, 1995.
[11] 이서로 외, 파워 해킹 테크닉, 파워북, 1995.
[12] 나이트메어, 슈퍼해커의 해킹 비밀, 연암출판사, 1995.
[13] 서삼영 외, 주전산기 I 용 보안유틸리티 개발보고서, 1993.
[14] 이재우 외 정보화 역기능 현황 및 분석, 한국전산원, 1995.
[15] 임채호, "해킹 수법 현황 및 방지 대책", '96 정보보호심포지움 1996.
[16] RAND, "Emerging Challenge: Security and Safety in Cyberspace", IEEE Technology and Society Magazine, Vol.14 No.4., 1996.
[17] Marcus J. Ranum, A Taxonomy of Internet Attacks : What You Can Expect, IWI, 1995.
[18] Simson Garfinkel and Gene Spafford, Practical Internet and UNIX Security, O'Reilly & Association, 1996.
[19] CERT-Advisory Series, CERT.
[20] http://8lgm.org, 8lgm Security Advisories.
[21] http://www.cert.org
[22] D. Farmer and E. Spafford, The COPS Security Checker System, USENIX Conference Proceedings, Anaheim, CA, 1990.
[23] David R. Safford, Douglas Lee Schales, and David K. Hess. The TAMU security packages: An ongoing response to internet intruders in an academic environment. In Edward Dehart, editor, Proceedings of the Security IV Conference, pages 91-118, Berkeley, CA, 1993. USENIX Association.
[24] D. V. Klein, Foiling the Cracker: A Survey of, and Improvements to, Password Security, EKKUUG Summer 90, London, July 1990, 147-154
[25] Gene H. Kim and Eugene H. Spafford. Experiences with tripwire: Using integrity checkers for intrusion detection. In Systems Administration, Networking and Security

Conference III. Usenix, April 1994.
[26] ftp://ftp.win.tue.nl/pub/security/satan-1.0.tar.Z
[27] ftp://ftp.iss.net/pub/iss/
[28] ftp://ftp.cert-kr.or.kr/pub/KUS
[29] http://www.iss.net/prod/
[30] ftp://sunsite.docic.ac.uk/computing/Systems/sun/sunflash/1994/1000-1099/