情報保護學會論文誌
第18卷 第1號, 2008. 2

Windows 기반의 PC 보안 정책 관리 및 취약성 점검을 위한 시스템 설계 및 구현*

박 병 연[1,2†], 양 종 원[1], 서 창 호[1‡]
[1]공주대학교, [2]한국과학기술정보연구원

System Design and Implementation for Security Policy Management of Windows Based PC and Weakness Inspection*

Park, Byung-yeon[1,2†], Jongwon Yang[1], Changho Seo[1‡]
[1]Kongju National University, [2]KISTI

요 약

현재 많은 컴퓨터들은 해킹 및 바이러스, 웜, 트로이 목마 등으로부터 개인 컴퓨터를 보호하기 위한 다양한 시도가 진행되고 있다. 하지만 Windows 기반의 개인 PC의 보안 안정성을 높이기 위하여 설정하여야 하는 정보는 일반 이용자들이 이해하기 어려운 측면이 많으며, 정보 접근성에 대한 인식 부족과 각종 설정에 대한 필요성 및 이에 대한 효과를 인지하지 못함으로 인하여 많은 문제점을 야기하고 있다.

이에 따라 정보통신망 및 개인 PC를 자동화된 방법으로 보호하기 위한 효율적인 시스템 개발이 요구된다. 본 논문은 Windows 시스템의 각종 보안 정책 및 취약성 분석을 통해서 개인용 컴퓨터의 문제점을 도출하고, 이를 통해 Windows 시스템의 다양한 문제점을 쉽고 편리하게 해결할 수 있는 시스템을 설계 및 구현한다.

ABSTRACT

Attempt to protect personal computer from hacking, virus, worm, and the troy wooden horse is progressed variously. Nevertheless, it is very difficult for public users to understand configurations to enhance security stability in windows based personal computer, and many security problem is due to there lack of recognize about information accessabilty, various kind of configuration, these necessity, and efficiency. Accordingly, it is demandded to develop an efficient system to protect networks and personal computer with automated method. In this paper, we derive problems of personal computer by analyzing various vulnerableness and policy on security, through which we design and implement the system to solve various windows system problem conveniently.

Keywords : Security Policy, 개인 컴퓨터 보안, XML

Ⅰ. 서 론

정보화 산업 발달로 네트워크 인프라의 증대와 개인용 컴퓨터의 네트워크 접속률이 증가됨에 따라 개인용 컴퓨터가 해커들의 주요 공격 대상으로 전환되었고, 특히 Windows 운영체제를 사용하는 개인용 컴퓨터는 다양한 취약성이 노출되면서 이를 이용한 다양한 공격 패

접수일: 2007년 7월 31일; 채택일: 2007년 10월 16일
* 이 논문은 2007년도 한국과학재단 특정기초사업의 지원에 의하여 연구되었음(R01-2005-000-10200-0)
† 주저자, bypark@kongju.ac.kr
‡ 교신저자, chseo@kongju.ac.kr

턴을 보이고 있다.

마이크로소프트사는 이러한 문제에 대처하기 위하여 다양한 보안기술을 운영체제에 적용하며 개인용 운영체제에 대한 보안성을 높이기 위해 노력하고 있으나, 여전히 다양한 취약성이 보고되고 있고 이로 인한 보안 침해사고 역시 증가하고 있는 추세이다. 또한 대부분의 보안 문제는 사용자가 프로그램의 취약점을 통해 자원의 접근 권한을 획득하면서 일어나므로 강력한 "사용자 인증(authentication) 및 승인(authorization)", "접근제어(access control)", "감사(audit)" 기능을 제공하여야 한다. 접근제어는 운영체제의 핵심 기능 중 하나로써, 임의적 접근제어(Discretionary Acces Control, DAC)[1]를 사용하여 각 소유자가 자신의 소유객체에 대해 임의로 접근제어 정책을 수립하며, 다른 접근제어 방법으로는 강제적 접근제어(Mandatory Access Control, MAC)[2], 역할기반 접근제어(Role based Access Control, RBAC)[3]들이 있다.

그러나 이러한 정보 접근제어에 대한 인식 부족으로 인해 정보 접근성이 떨어져 여전히 수많은 개인용 컴퓨터는 공격의 대상이 되고 있다. 특히 악의적인 행위 중 반달(Vandal) 공격을 가하는 컴퓨터 프로그램 형태[4]가 보안의 효율성을 떨어뜨리고 있다. 이에 따라 정보통신망 및 개인 PC를 자동화된 방법으로 보호하기 위한 효율적인 시스템 개발이 요구된다.

이에 본 논문은 Windows 시스템의 각종 보안 정책 및 취약성 분석과 접근제어를 통해서 개인용 컴퓨터의 문제점을 도출하고, 이를 통해 Windows 시스템의 다양한 문제점을 쉽고 편리하게 해결할 수 있는 시스템을 설계 및 구현하여 개인 및 기업 혹은 기관의 중요 정보와 시스템 자원을 보호하고자 한다.

본 논문의 구성은, 2장에서 관련연구를 조사하여 설명하며, 3장에서는 제안하는 Windows 기반의 PC 보안 정책 관리 및 취약성 점검을 위한 시스템 설계 및 구현을 다룬다. 마지막으로 4장에서 결론을 맺는다.

Ⅱ. 관련연구

2.1. Windows 레지스트리분석

Windows는 레지스트리(registry)에 프로그램이나 시스템에 관한 다양한 정보를 저장하고 있으므로 이를 분석할 수 있어야 한다. 레지스트리 Hive 파일들은 [SystemRoot]₩System32₩Config 폴더에 위치하며, regedit와 같은 명령으로 살펴볼 수 있다. 레지스트리 Hive 파일들 중 SAM 파일은 패스워드들의 해시 정보를 가지고 있으며, 운영체제에 의해서 암호화되어 보호되고 있다. 또한 SAM 파일의 패스워드들을 복구할 수 있어야 한다.

2.2. 덤프 메모리 분석

증거 수집에서 얻어진 데이터들로부터 유용한 정보를 얻는 것을 증거 분석이라고 한다. 유용한 정보는 사건에 따라 다르겠지만 일반적으로 다음과 같은 증거 분석 기술들이 사용될 수 있다.

2.2.1. Windows 레지스트리분석

Windows는 레지스트리(registry)에 프로그램이나 시스템에 관한 다양한 정보를 저장하고 있으므로 포렌식 툴은 이를 분석할 수 있어야 한다. 레지스트리 Hive 파일들은 [SystemRoot]₩System32₩Config 폴더에 위치하며, regedit와 같은 명령으로 살펴볼 수 있다. 레지스트리 Hive 파일들 중 SAM 파일은 패스워드들의 해시 정보를 가지고 있으며, 운영체제에 의해서 암호화되어 보호되고 있다. 포렌식 툴은 SAM 파일의 패스워드들을 복구할 수 있어야 한다.

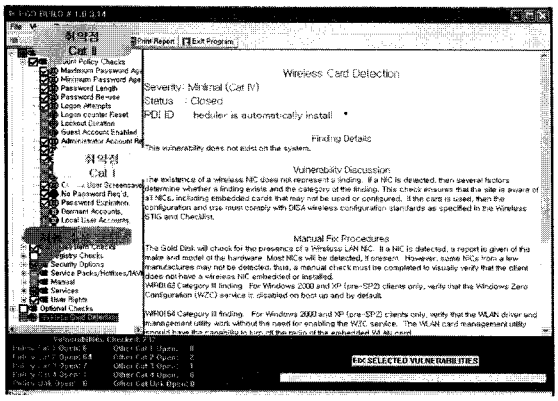
2.2.2. 덤프 메모리 분석

프로세스가 사용중인 가상 메모리의 덤프를 획득했을 경우에 사용자 ID나 패스워드와 같은 유용한 정보가 가상 메모리에 남아 있을 수 있다. 프로세스를 위한 가상 메모리는 보통 코드 영역, 데이터 영역, 스택 영역 등으로 나뉘어지며, 데이터 영역이나 스택 영역이 프로세스에서 필요한 여러 정보를 저장하고 있으므로 프로세스가 가상 메모리를 어떻게 사용하는지를 분석할 수 있어야 한다.

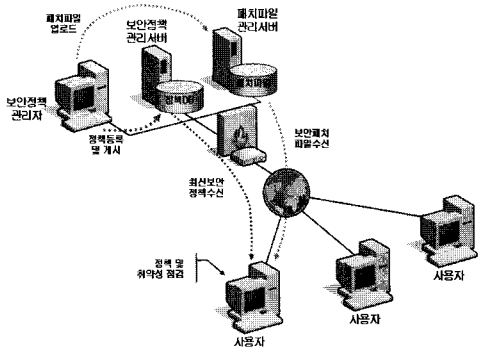
이외에도 Timeline 분석, 삭제된 파일복구, 비정상적인 파일찾기, 이메일분석, 로그분석, 슬랙 공간 분석, 스트링 서치 분석 등이 있다.

2.3. Gold Disk

GGold Disk[5]는 Windows XP를 대상으로 PC에 대



〔그림 1〕. Go;d Disk의 취약점 점검 결과



〔그림 2〕. 전체시스템 구조도

한 사용자 전문지식이 없어도 쉽게 보안진단을 수행하고 자동으로 보안설정을 할 수 있도록 DISA에서 개발한 응용 프로그램 도구이다. 이 프로그램은 Windows XP 운영체제만을 지원하고 있으며, 인터넷 익스플로어 6.0이상의 버전 등이 필요하고, 실행 프로그램은 로컬 하드디스크에서 수행하도록 개발 되었으며, 실행할 계정, 검사할 PC의 관리자 권한, 감사 및 보안 로그를 관리할 수 있는 권한 등을 요구한다.

[그림 1]은 Windows XP에 대하여 보안진단을 수행하여 사용자PC에 대한 취약한 항목을 보여주고 있는 점검 결과 화면이다. 각각의 취약점 점검 결과 항목은 윈도우의 왼쪽 메뉴에 포함되어 있으며, 252개의 취약점 항목으로 점검을 실시하였으며, 취약점 점검결과는 둥근 원모양에 Ⅰ, Ⅱ, Ⅲ을 표시하여 카테고리별 위협 항목을 시각적으로 보여 주고 있다. 또한 PC 및 네트워크의 보안진단을 위한 도구들로 NCSA의 Clumon[6], 버클리 대학(UC,Berkeley)의 Ganglia[7], SIFT의 NVisionCC[8] 등이 있다.

Ⅲ. Windows 기반의 PC 보안 정책 관리 및 취약성 점검 구조도

본 논문에서 제시한 Windows 기반 PC 보안 정책 관리 및 취약성 점검 시스템에 대한 물리적인 구성 및 시스템의 논리적인 구성 그리고 모듈 구성을 보인다.

3.1 구현 시스템의 모듈

개인용 PC의 보안 정책으로 활용될 Windows 시스템의 보안 정책 및 체크 리스트 정립을 통해, Windows 시스템의 Registry 자원과 보안 체크리스트 간의 연관관계를 분석하고, 정책에 따른 취약성 점검 모듈 구성을 보인다.

[그림 2]와 같이 클라이언트는 정책 및 취약성 점검 툴을 이용하여 Windows 운영체제에 대한 점검을 수행하고, 이때 정책 및 취약성 점검 리스트를 항상 최신의 정보를 유지하기 위해 보안정책 게시 서버로부터 수신하고, 점검이 완료된 후 발견된 취약성에 대한 패치가 필요한 경우 필요한 파일을 패치파일 게시 서버로부터 수신하여 시스템에 설치하는 작업을 수행한다.

3.2 보안정책 관리 모듈

보안정책 관리 모듈은 사용자가 Windows 시스템에 대한 보안정책과 취약성을 점검하기 위한 다양한 체크리스트와 패치 방법에 대한 정규화 된 규칙을 저장하고 관리할 수 있는 기능을 제공하는 모듈이다.

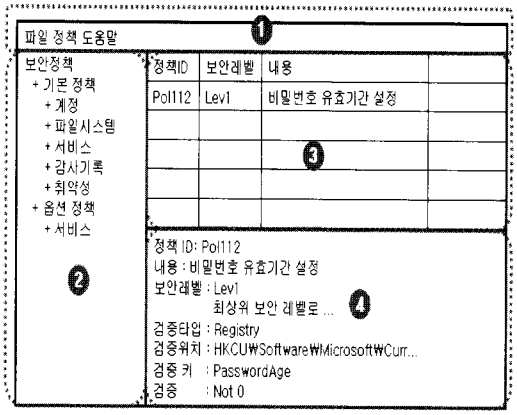
보안정책 관리 모듈을 통해서 생성된 보안정책 파일은 XML 형태로 생성되며, 웹 서버에 게시되거나 클라이언트 검증 모듈에 포함되어 배포되게 된다.

보안정책 관리 모듈의 UI는 [그림 3]과 같다.

화면 구성을 통해서 관리자는 현재 등록된 보안정책에 대한 정보를 쉽게 유지보수 할 수 있게 되며, 다양한 메뉴 제공을 통해서 검증 항목을 추가하거나 삭제할 수 있게 하기위해서 보안정책 관리 모듈에 의해서 생성되는 결과물은 XML 구조로 되어야 한다. 다음은 XML 문서의 대략적인 정보이다.

〔표 1〕. 시스템의 고려사항

정책의 변경 및 추가, 삭제가 가능.
새로운 취약성 발생시 취약성 검증 항목을 추가.
온라인 클라이언트 시스템은 항상 최신의 정책과 취약성 검증 항목을 이용해 검증.
오프라인 클라이언트를 위해서 기본적인 정책 및 취약성 검증 항목을 사용.



〔그림 3〕. 보안정책 관리 모듈 UI구성

3.3 구현 시스템 보안 점검 프로파일 정의

보안 점검 프로파일은 다음과 같은 파일로 정의된다.

- Category 파일
- SubCategory 파일

```
[FILE INFO]
version=1.0
date=2007.01.23
org_name=공통

[CATEGORY]
0=정책 관련,policy.ini
1=패치 관련,patch.ini
2=보안 S/W 관련,extsw.ini
```

상기에서 Category 파일은 sub category 파일들에 대한 정보를 담고 있으며, 보안점검 프로파일의 버전 정보 등을 담고 있다.

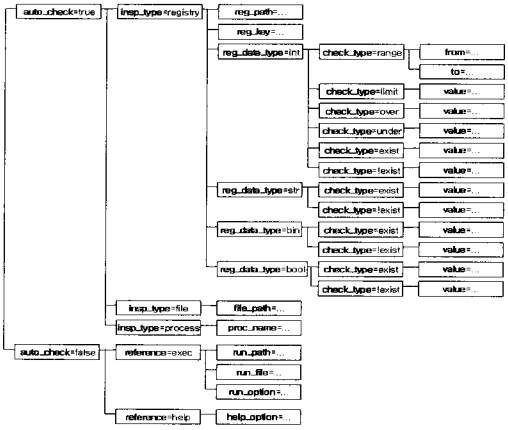
〔표 2〕. Catagory 파일의 상세 정보

```
<?xml version="1.0" encoding="euc-kr"?>
<CHECKLIST>
  <ITEM>
      <POLICY_ID>Pol112</POLICY_ID>
    <LEVEL>Lev1</LEVEL>
<CHECK_DESC>비밀번호 유효기간설정</CHECK_DESC>
<DETAIL>비밀번호 유효기간은 최소 1일에서 최대 90일 이하로 권장합니다. 유효기간이 너무 길면 비밀번호가 유출될 우려가 있으므로 유효기간을 설정하여야 합니다.</DETAIL>
<CHECK_METHOD>Registry</CHECK_METHOD>
<REGISTRY>
<PATH>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters</PATH>
<CHECK_NAME>maximumpasswordage</CHECK_NAME>
    <TYPE>REG_DWORD</TYPE>
    <CHECK>X LT 0 OR X GT 90</CHECK>
  </REGISTRY>
</ITEM>
```

[표 2]의 내용은 Category 파일인 category.ini 파일의 내용을 보이고 있다.

[표 2]에서 CATEGORY 타이틀 하위에 Sub Category로 분류되는 보안점검 프로파일 정보가 들어가게 되며, 해당 정보에는 보안점검을 위한 실제 파일명이 들어가게 되며, 해당 파일에 보안점검을 위한 프로파일 정보가 포함되게 된다.

이 파일들이 Sub Category 파일이 되며, 해당 파일은 다음에 설명되는 것처럼 보안 점검 항목 리스트를 프로파



〔그림 4〕. 자동점검 프로파일에 따른 필드 계층구조

일 정책에 의해 정규화된 형태의 정보를 가지고 있게 된다.

3.2.1. 자동 점검 항목 프로파일

자동 점검 항목 프로파일은 해당 점검 항목이 자동으로 점검할 수 있는 항목인지 아니면 사용자가 수동으로 점검을 수행하여야 하는 항목인지에 대한 정보를 보이게 된다. 자동점검 프로파일은 'true' 혹은 'false' 값을 가질 수 있으며 점검 값에 따른 하위 필드들의 계층 구조를 살펴보면 다음과 같다(그림 4).

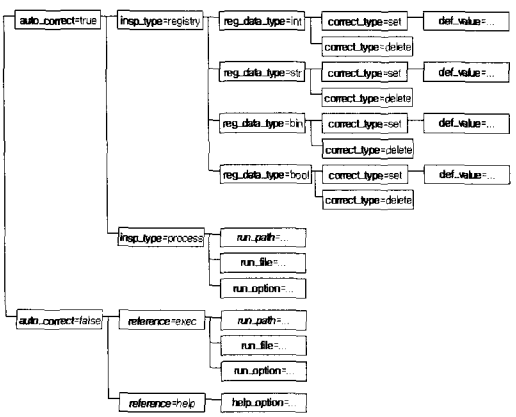
3.2.2. 자동 보정 항목 프로파일

자동 보정 프로파일은 앞의 자동점검 옵션인 auto_check가 true의 값을 가지는 경우에 한하여 보안점검 결과에 문제가 있는 경우 이를 보정하기 위해서 자동으로 보정 가능 여부에 대한 정보를 포함하는 옵션으로 auto_correct 필드명을 가지며 'true' 혹은 'false'의 값을 가진다. 역시 설정 값에 따른 하위 계층구조의 정보를 살펴보면 다음과 같다(그림 5).

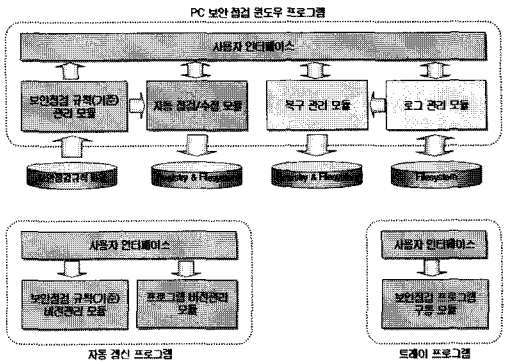
Ⅳ. 실험 및 성능 평가

4.1. 구현 시스템의 환경 및 구성

본 논문에서는 구현된 프로그램은 크게 클라이언트를 담당하는 PC보안 점검 윈도우 프로그램과 관리자용 PC보안 점검 규칙 관리 프로그램으로 나눈다. 클라이언



[그림 5]. 자동점검 프로파일에 따른 필드 계층구조



[그림 6]. PC 보안 점검 윈도우 프로그램 구조

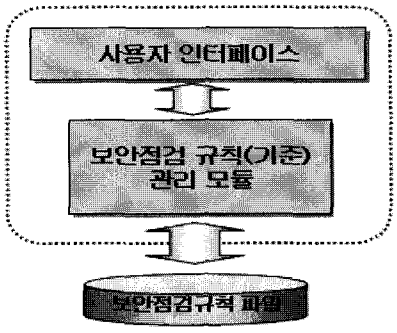
트용 프로그램은 개인용 PC에 설치되어 시스템의 보안 점검을 직접 수행하고, 점검 결과를 사용자에게 알려주어 시스템의 문제점을 파악하고 이를 수정할 수 있도록 기능을 제공한다.

PC 보안 점검 윈도우 프로그램의 시스템 구조는 다음과 같다(그림 6).

[그림 7]과 같이 PC 보안 점검 규칙 관리 프로그램은 PC 보안 점검 윈도우 프로그램이 개인용 PC에 보안 점검을 수행하기 위한 점검 규칙정보를 생성하고 편집, 관리할 수 있는 기능을 제공하는 관리자용 프로그램이다. PC 보안 점검 규칙 관리 프로그램의 구조는 [그림 7]과 같다.

4.2. 구현 시스템의 개발 환경 및 라이브러리

4.2.1. 구현 시스템 개발 환경



[그림 7]. PC 보안 점검 규칙 관리 프로그램 구조

- 보안정책 관리 도구
  - Platform : 윈도우 기반
  - 개발 언어 : C/C++ (MS Visual C++)
- 보안정책 및 취약성 검증 도구
  - Platform : MS Windows XP
  - 개발 언어 : C/C++ (Visual Studio v2003)

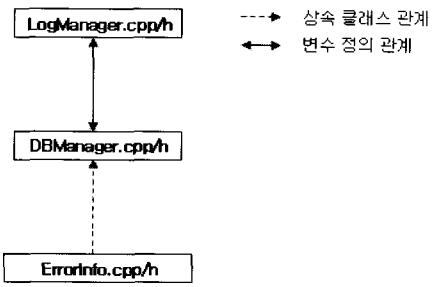
4.2.2. 구현 시스템 라이브러리

크게 라이브러리는 LogManager와 Inspector 라이브러리로 나누어진다.

먼저 LogManager 라이브러리에서는 보안점검 클라이언트 프로그램을 통해서 PC의 취약점 점검 항목을 점검한 후 점검 결과 및 취약점 수정 내용에 대한 감사 기록 정보를 저장한다. 또한 이를 관리하여 사용자가 쉽게 처리 내용을 파악할 수 있도록 기능을 제공한다. [그림 8]은 라이브러리 모듈 내부의 각 소스간 관계도이다.

Inspector 라이브러리는 보안점검 규칙 관리 모듈에 의해서 제공되는 점검항목 리스트를 이용하여 PC의 정보를 확인하고, 점검 결과를 사용자에게 보여주어 사용자가 점검 결과를 볼 수 있도록 기능을 제공한다. 또한 점검결과에 대한 수정이 필요한 경우 사용자에 의해 선택된 항목에 대해서 점검 항목의 기준 값으로 변경을 수행한다. 세부적으로 제공하는 기능은 다음과 같다.

- 보안 점검 규칙에 지정된 정보로 점검
  - 레지스트리의 정보를 점검하여 기준 값에 맞지 않는 경우 사용자에게 보이고, 사용자가 수정할 수 있도록 기능 제공
- 보안 점검 규칙에 지정된 정보를 기준으로 외부 프로그램 점검
  - 외부 프로그램(개인방화벽, 백신)에 대한 특정 파일 혹은 레지스트리 점검을 통해 프로그램 정보를 사용자에게 알리고, 프로그램 정보에 문제가 있는 경우 이를 처리할 수 있도록 기능 제공
  - 외부 프로그램 구동 상태 정보를 확인
- 사용자의 수정 요청에 의한 수정(보정) 처리
  - 보안 점검 규칙에 위배된 항목에 대해서 사용자가 수정(보정)을 선택한 경우 해당 항목에 대해서 기준 값으로 설정하는 기능을 제공
  - 보안 점검 규칙에 위배된 항목 중 프로세스가 구동되어야 하는 경우 프로세스 구동에 대한 도움말을 보임
  - 보안 점검 규칙에 위배된 항목 중 업데이트가 필요한 경우 도움말 혹은 업데이트 프로그램을 자동 구동



[그림 8]. LogManager 라이브러리 관계도

4.3. 구현된 시스템 실행 환경

- Windows XP용
  - Pentium 3 500Mhz 이상의 CPU
  - 128 MByte 이상의 메모리
  - 20 MByte 이상의 하드디스크 여유 공간
  - Windows XP Home/Pro 버전

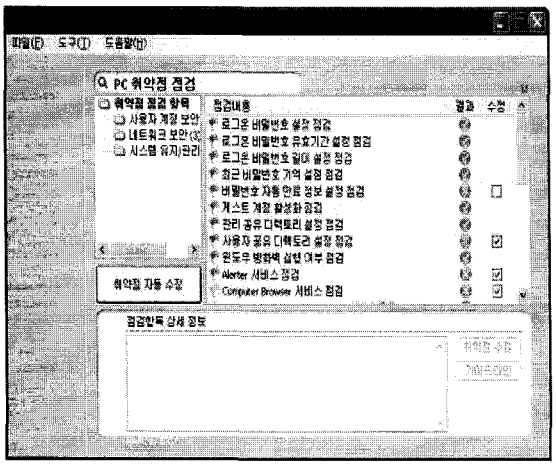
- Windows 98용
  - Pentium 3 500Mhz 이상의 CPU
  - 128 MByte 이상의 메모리
  - 20 MByte 이상의 하드디스크 여유 공간
  - Windows 98 SE 이상 버전

구현된 시스템의 기능이 적절하게 구현되어지는지 확인하기 위해, 먼저 해당 운영체제에서 프로그램의 정상적인 수행 가능성, 취약점 점검 항목의 점검 가능성, 취약점 발견 항목에 대한 취약점 수정 가능성 등을 테스트하였다.

[표 3]은 구현된 시스템에서 클라이언트 프로그램에 대한 항목별 시험 및 시험결과의 예를 보여주고 있다.

[표 3]. 취약점 점검 항목별 시험 및 시험결과 예

시험 내용	로그인 비밀번호 설정 점검
시험 방안	로그인 비밀번호를 제거
예상 결과	로그인 비밀번호 설정되지 않았음이 나옴
시험 결과	정상



〔그림 9〕 취약점 점검 화면

[그림 9]와 같이 취약점 점검은 Windows의 취약점을 점검하기 위한 항목 리스트를 화면에 보이고, 해당 항목에 대한 자세한 정보를 보인다.

또한 점검 항목을 기능별로 분류하여 보이며, 취약점 점검을 수행할 수 있다.

V. 결 론

현재 컴퓨터 해킹 및 바이러스, 웜, 트로이 목마 등으로부터 개인 컴퓨터를 보호하기 위한 다양한 시도가 진행되고 있다. Anti-Virus 프로그램 및 방화벽 설치 등을 통하여 사이버 위협으로부터 개인 PC를 보호하는 것은 당연시 되고 있다. 반면 이들 프로그램의 최신 상태 유지 및 주기적인 점검, 최신 보안 패치의 유지와 같은 설치 후 지속적으로 수행되어야 할 가장 중요한 작업에 대한 검사를 지원하지 않는다. 이는 값 비싼 방범용 대문을 설치하고 이에 대한 지속적인 관리를 수행하지 않음으로 인하여 제대로 된 효과를 거두지 못하고 있는 실정이다.

본 논문에서는 이러한 Windows 시스템의 각종 보안 정책 및 취약성 분석을 통해서 개인용 컴퓨터의 문제점을 도출하고, 이를 통해 Windows 시스템의 다양한 문제점을 쉽고 편리하게 해결할 수 있는 시스템을 설계 및 구현하였다.

추후 다양한 기준에 대한 여러 운영체제에서의 검사 및 미흡한 설정에 대한 수정 기능을 제공하는 시스템을 제공 할 예정이다.

참고문헌

[1] R. SANDHU, P.SAMARATI, "Access control: Principles and practice", IEEE Commun. Mag.32, 9, 40~48 1994.

[2] D.E. DENNING, "A lattice model of secure information flow", Commun, ACM 19, 2, 236~243. 1976

[3] D. F. Ferraiolo, D. Richard Kuhn, "Role-Based Access Controls", Proceedings of the 15th NIST-NSA National Computer Security Conference, Baltimore, Maryland, October, 13~16, 1992.

[4] http://www.symantec.co.kr

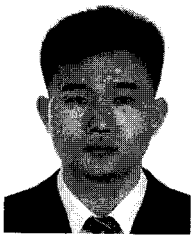
[5] http://iase.disa.mil/stigs/SRR/index.html

[6] T. Roney, A. Bailey, and J, Fullop, "Cluster Monitoring at NCS," 2nd LCI Int'l Conf. on Linux Clusters.2001.

[7] M. Massie, B. Chun, and D. Culler, "The Ganglia Distributed Monitoring System: Design, Impleementation and Experience," Paralled Computing. Vol.30, Issue. 7, 2004.

[8] W. Yurcik, X. Meng, and N. Kiyanclar, " NVisionCC: a Visualization Framework for High Performance Cluster Security," Proc. of VizSEC 2004, ACM Press, New York, NY, USA, Oct. 2004. pp. 133-137.

〈著者紹介〉



박 병 연 (Park Byung-Yeon) 정회원

1997년 : 대전산업대 재료공학과(학사)
2004년 : 공주대 교육정보대학원 교육정보학(이학석사)
2006년 : 공주대학교 일반대학원 바이오정보학과 (정보보호전공) 박사과정
1990년~1996년 : 시스템공학연구소 연구원
1996년~1998년 : 한국전자통신연구원 연구원
1999년~현재 : 한국과학기술정보연구원 선임연구원
<관심분야> 무선 인터넷 보안, 시스템 보안, 생체인식, RFID/USN 등 인식, 암호 알고리즘

양 종 원 (Jongwon Yang) 학생회원

2003년 : 공주대학교 전자계산학과(학사)
2005년 : 공주대학교 일반대학원 컴퓨터공학과 (공학석사)
2005년 : 공주대학교 일반대학원 바이오정보학과(정보보호전공) 박사과정
2006년~현재 : 한국전자통신연구원 위촉연구원
<관심분야> 무선 인터넷 보안,시스템 보안, 생체인식, 암호 알고리즘,

서 창 호 (Changho Seo) 종신회원

1990년 : 고려대학교 수학과(학사)
1992년 : 고려대학교 일반대학원 수학과 (이학석사)
1996년 : 고려대학교 일반대학원 수학과 (이학박사)
1996년~1996년 : 국방과학연구소 선임연구원
1996년~2000년 : 한국전자통신연구원 선임연구원, 팀장
2000년~현재 : 공주대학교 응용수학과(정보보호전공) 부교수
2001년~현재 : 공주대학교 바이오정보학과 부교수
<관심분야> 암호 알고리즘, PKI, 무선 인터넷 보안, 시스템 보안 등