OVAL을 이용한 분산 네트워크 환경에서의 통합 보안취약성 진단도구 설계

백승엽*, 김영섭*, 이극*
*한남대학교 컴퓨터공학과
e-mail: psy9511@is.hannam.ac.kr

A Design of Enterprise Vulnerability Assessment Tool on a Distributed Network Environment using OVAL

Seung-Yub Baek*, Young-Sub Kim*, Geuk Lee*
*Dept of Computer Engineering, Han-Nam University

요 약

취약성 점검 및 진단을 위해서는 표준화된 취약성 분석 자료와 취약성 진단의 단계가 구축되어야 한다. 본 논문에서는 취약성 점검 및 진단의 기능이 분산화 된 네트워크 환경에서 효율적으로 수행할 수 있도록 분산 취약성 점검도구를 설계한다. OVAL을 기반으로 하여 설계하였고, 취약성 데이터베이스는 표준화되고 정확한 취약점을 진단 할 수 있도록 항상 최신의 정보를 유지하여 분산된 네트워크 환경에 적합하도록 구성하였으며, Nessus와 OVAL을 기반으로 취약성 진단도구를 구현하였다. 본 논문에서 제안하는 통합 취약성 진단도구를 사용하면 분산 네트워크 환경에서의 취약점을 빠르고 정확하게 진단하며 그에 따른 해결방안을 제시할 수 있다.

1. 서론

취약성 진단도구는 분산 네트워크 환경에서 정보 및 각 시스템을 보호하기 위하여 시스템에 존재할 수 있는 취약점을 사전에 발견하고 그에 따른 보안 대책을 제시하여 시스템을 보호하는데 목적이 있다.

취약점 분석도구는 크게 네트워크 취약성 진단도구와 시스템 취약성 진단도구로 나눌 수 있다. 현재 상용 및 공개용 도구들이 존재하지만 거의 대부분이 네트워크 기반 진단도구와 호스트 기반 진단도구로 분리되어 있으며, 취약성을 진단하기 위한 취약성 데이터베이스도 진단도구마다 상이하게 이루어져 있어 진단 결과를 별도로 관리해야 하는 문제점이 있다. 이런 문제를 해결하기위해 표준화되고 정확한 취약성 진단 데이터베이스를 구축하고 연동할 수 있는 통합된 취약성 진단도구가 필요하다.[1]

본 논문에서 제시하는 통합된 취약성 진단도구는 다음과 같다. 네트워크와 시스템의 통합된 취약점 데이터베이스를 취약성 표준 명명인 CVE(Common Vulnerabilities and Exposures)와 해당 취약점에 대한 해결방안으로 설계하여 취약점 진단 도구와 연동한다. 취약점 진단 결과를 통합 관리하고 네트워크와 시스템의 취약점 진단을 동시에 함으로써 시간적 비용을 절감할 수 있는 중앙 집중식 시스템이다. 또한 취약점 진단과 함께 해당 취약점의 해결방안을 제시하여 시스템과 네트워크의 유지보수 효율을 증대할 수 있다.

본 연구는 산업자원부 지역협력연구사업 (R12-2003-004-02003-0) 지원으로 수행되었음.

2. 관련연구

2.1 취약성 데이터베이스

취약성 데이터베이스(vulnerability database)란 컴퓨터 시스템의 취약성 정보를 분류하여 저장한 데이터베이스로 취약점 점검도구에서 가장 중요한 요소 중 하나이다.

취약성 데이터베이스는 보통 취약점 진단도구의

진단 결과를 레포팅 하는데 사용한다.[2] 취약점 진단도구는 진단한 시스템의 취약점 정보를 취약점 데이터베이스에 질의하여 취약점의 식별번호, 이름, 내용, 해결방안 등을 알아낸 후 보고서를 작성한다.

현재 국내 및 국외에서 취약성에 대한 데이터베이스화가 활발히 진행 중에 있으며 대표적인 곳은 퍼듀 대학 보안연구팀인 COAST와 NIST에서 운영중인 ICAT Meta Vulnerability Database가 있다.

2.2 취약성 진단도구

취약점 진단도구는 일반적으로 취약점 스캐너로 불리며 시스템과 네트워크에 산재되어 있는 컴퓨터 시스템의 보안 취약점을 진단하고 분석하여 취약점 정보를 제공해주는 도구이다. 취약성 진단도구는 크게 네트워크스캐너와 시스템스캐너로 분류할 수 있다.

2.2.1 시스템 스캐너

시스템 스캐너는 시스템 내부적으로 가지고 있는 취약점을 진단하는 도구로 시스템 레벨의 보안취약점 진단하여 상세하고 정확하게 진단할 수 있는 장점이 있다. 하지만 플렛폼에 의존적이고 모든 점검대상에 시스템 스캐너가 설치되어 있어야 하며 실행 시 관리자의 권한이 필요하다.[4]

대표적인 시스템 취약성 진단도구로는 COPS, Tiger, STAT, CyberCop 등이 있지만 이러한 대부분의 시스템은 분산 환경에서의 취약성 점검에 한계를 가지고 있다.

2.2.2 네트워크 스캐너

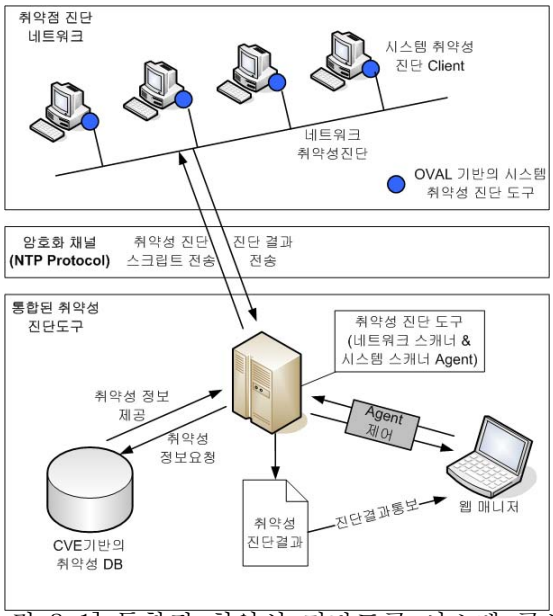
네트워크 스캐너는 탐지 대상 시스템에 대하여 운영중인 서비스 점검 후 각 운영 서비스에 대한 데몬 정보를 통하여 취약성 유무를 점검하는 도구로써 시스템 레벨에서 취약점을 진단한다. 점검대상으로는 스캐너가 설치된 네트워크에 대하여 불필요하게 열려진 포트, 서비스 거부공격(DoS), RPC, HTTP, SMTP, FTP, FINGER 등이 있다. 현재 네트워크 스캐너는 Nessus, SAINT, ISS Network Scanner 등이 있으며 호스트 점검 기능을 위해 호스트 점검 도구의 기능을 추가하는 하이브리드 방식을 채택하고 있다.

2.3 CVE(Commom Vulnerabilities and Exposures)

CVE는 컴퓨터의 취약점에 대해서 표준화된 이름을 제공하기 위한 Naming-schema로서 미국의 Mitre에서 처음 제안하였고, 현재 수많은 보안 업체들이 참여하여 취약점 이름의 표준화 작업을 진행하고 있다.[3] CVE를 사용하지 않은 취약성 진단도구들은 각 벤더들의 보안 정책에 의해 취약성이 평가되어 타제품들과는 호환이 이루어지지 않는 문제점이 지적된다. 하지만 CVE를 기반으로 취약성 데이터베이스를 설계하여 취약성 진단도구와 연동한다면 이런 문제점은 모두 해결될 것이다.

3. 통합된 취약성 진단도구 설계 및 구현

3.1 취약성 진단 시스템 구성



[그림 3-1] 통합된 취약성 진단도구 시스템 구성도

본 논문에서 설계한 통합된 취약성 진단도구는 [그림 3-1]과 같으며 취약성 DB는 CVE를 기반으로 구축하였고 네트워크 스캐너는 Nessus를 사용하며 시스템스캐너는 OVAL을 기반으로 한 취약성 진단 스크립트 인터프리터를 설계하였다. 취약성 진단도구는 취약성 진단 엔진인 Agent를 중심으로 웹 매니저를 통하여 Agent를 제어하며 진단 엔진을 통하여 취약성을 탐색한다. 취약성 진단 시스템은 서로 NTP 프로토콜을 이용하여 통신을 하며 데이터들은 SSL로 암호화되어 전송되므로 패킷이 스니핑 되더라도 취약성 관련 정보를 안전하게 보호 할 수 있다.

3.2 취약성 데이터베이스

통합된 취약성 진단도구에 구성되는 취약성 데이터베이스는 [표 3-1]과 같은 스키마를 구성하고 있으며 웹 인터페이스를 통하여 외부로 공개할 수 있게 설계하였다.

취약점의 식별자로 취약성 표준 명명인 CVE ID

를 사용하며 CVE를 사용함으로써 다른 취약점 데이터베이스들과 100% 참조할 수 있는 호환성(CVE-Compatible)을 가진다. 취약성 데이터베이스는 I-CAT 데이터베이스를 파싱하여 최적화시키며 해결방안과 자세한 정보를 추가할 수 있는 Web Client를 제공한다. 이 Web Client를 통하여 취약성 정보를 실시간으로 업데이트 할 수 있는 모듈을 포함하고 있다.

[표 3-1] E-R모델을 이용한 취약성 DB의 엔티티

개체	설명
Vulnerability	취약점에 대한 전반적인 내용으로 취약점 식별자(CVE ID), 유형, 위험도, 손실유형, 도메인 유형, 공격 요구사항 등이 있다.
Related Reference	취약점에 대한 참조 가능한 레퍼런스 정보를 나타낸다.
vulnerable Software	취약한 S/W에 대한 정보로 Reference 개체와 마찬가지로 하나의 취약점에 하나 이상의 취약한 소프트웨어를 가질 수 있다.
Vulnerability Name	취약점에 대한 이름으로 영문과 한글로 나누어진다.
Vulnerability Summary	취약점에 대한 요약
Vulnerability Description	취약점에 대한 설명
Vulnerability Solution	취약점에 대한 해결책
Vulnerability Author	취약점 정보를 수정한 사용자
User	사용자에 대한 정보

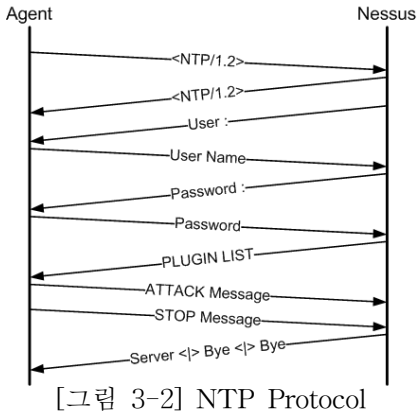
취약성 진단 Agent가 진단한 결과와 취약성 데이터베이스를 연동하여 취약성 진단 결과 및 해결책을 상세하게 제시하여 취약성에 대한 빠른 조치가 가능하도록 설계하였다.

3.3 통합 취약성 진단도구

통합 취약성 진단도구는 네트워크 스캐너와 시스템 스캐너로 이루어져 있고 취약성 데이터베이스와 연동하여 취약성을 진단하고 진단결과 및 해결책을 출력한다.

3.3.1 네트워크 취약성 진단 스캐너

네트워크 취약성 진단 스캐너는 Nessus를 사용한다.[7] Nessus는 Client/Server의 구조로 되어 있으며 Client는 Server에게 명령을 전달하여 취약성을 진단한다. Server에 로그인 과정을 거쳐 접속하며, 단일 호스트 혹은 다수의 호스트에 대하여 네트워크 취약성 진단을 수행하고 결과를 반환한다. (Nessus의 취약성 스크립트는 사용자 그룹이 주기적이고 신속한 업데이트를 하여 항상 최신의 진단 플러그인(스크립트)를 보유하고 있으며 대규모의 네트워크에 대한 진단이 가능하다.[7]

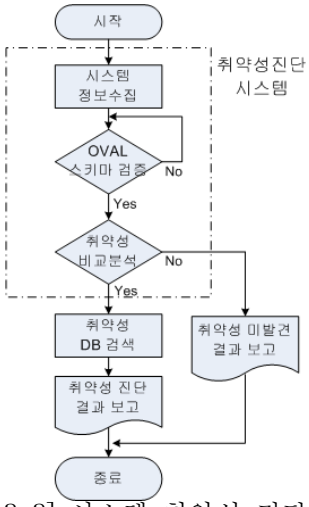


[그림 3-2] NTP Protocol

Web-Manager와의 통신은 NTP프로토콜을 이용하며 SSL로 암호화 통신을 한다. [그림 3-2] Web-Manager 는 시스템(Agent)을 통하여 Nessus를 제어하며 환경설정 및 네트워크의 취약성 진단을 위한 플러그인 설정이 가능하다. 네트워크의 취약성 진단을 실행하고 취약성 데이터베이스와 연동하여 CVE기반의 표준화된 진단 결과를 출력한다.

3.3.2 시스템 취약성 진단 스캐너

시스템 스캐너는 OVAL(Open Vulnerability Assessment Language)을 기반으로 취약성을 탐지하고 탐지 결과는 취약성 데이터베이스와 연동하여 결과를 출력한다. OVAL을 이용한 스캐너는 CVE에 등록된 모든 취약점을 탐지할 수 있도록 설계하였으며, XML 및 SQL로 취약성 진단 스크립트가 이루어져 있어 명료하고 논리적으로 취약성을 기술할 수 있다.[2, 5]

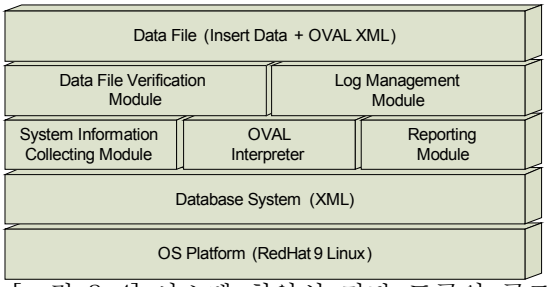


[그림 3-3] 시스템 취약성 진단 흐름도

취약성 정보수집 도구인 Client를 이용하여 시스템의 정보를 수집하고 수집된 정보를 Agent인 OVAL 인터프리터를 이용하여 비교분석하여 취약성

을 진단한다.[7] 시스템의 정보는 플랫폼에 따라 상이하며 본 논문에서는 RedHat9 시스템에서 설계 및 구현하였으며, 취약성 진단 흐름도는 다음 [그림 3-3]과 같다.

시스템 취약성 진단도구는 OVAL정의, 시스템 정보수집 모듈, OVAL정의 인터프리터로 나눌 수 있다. OVAL 정의는 Mitre에서 표준으로 정의된 자료에 의거하여 정의하며, XML로 저장한 파일들로서 "OVAL+식별번호"의 이름을 가지고 있다.[3] 시스템 정보수집 모듈은 시스템, 파일, 네트워크 데몬, 프로세스, RPM 패키지, 쉐도우 패스워드, 운영 플랫폼 정보 등을 수집하는 모듈로 시스템의 중요 정보를 수집하여 OVAL정의 XML과 비교하여 취약성을 점검한다.[그림 3-4]



[그림 3-4] 시스템 취약성 진단 도구의 구조

OVAL 정의 인터프리터는 XML 포맷의 OVAL 정의를 해석하여 시스템으로부터 어떠한 정보를 수집할 것인지를 파악한 후 시스템 정보 수집 모듈을 호출하여 수집된 정보와 정의에 기술되어 있는 취약점 존재 조건을 비교하여 결과를 리포팅 모듈에 전달하는 역할을 한다.

3.4 Web-Manager System

통합 취약성 진단도구에 의해 네트워크 및 시스템에서 발견된 취약성 진단 결과는 취약성 데이터베이스를 참조한 후 Web-Manager에게 전송하고 실시간으로 Manager가 설치된 컴퓨터에서는 로그인절차를 거쳐 언제 어디서나 취약성 진단이 가능하도록 설계하였다.

모든 데이터를 SSL로 암호화하여 정보유출을 미연에 방지하도록 설계하였다. Web-Manager는 시스템 취약성 진단 파트와 네트워크 취약성 진단 파트로 나뉘어 있으며 각각의 취약성 진단 결과는 취약성 데이터베이스의 상세한 진단 Report로 보여주고 소프트웨어의 업데이트나 응용프로그램 및 시스템의 설정 변경 등에 해결방안을 관리자에게 제공한다.

4. 결 론

본 논문에서는 분산 네트워크 환경에서 취약성을 진단할 수 있는 취약성 진단도구를 설계 및 구현하였다. 통합 취약성 진단도구는 Web을 통하여 원격에서 네트워크와 시스템에 잠재하고 있는 취약성을 탐지하고 해결책 제공을 목적으로 하고 있다.

취약점 진단도구인 Agent에서 중앙 집중적인 관리를 통해 취약점을 관리하고, 그에 따른 로그를 관리하여 기존 도구들이 분산된 네트워크 환경에서 취약성 탐지 시에 발생하는 문제점들을 해결할 수 있도록 하였다.

본 논문에서 구현된 분산 네트워크환경에서의 통합된 취약성 진단 도구는 Nessus와 OVAL을 기반으로 한다. 네트워크 진단도구와 시스템 진단도구는 각각 'Nessus사용자 그룹'과 'MITRE기관'에서 취약성 스크립트에 대한 업데이트가 주기적으로 이루어지며, 많은 플러그인(스크립트)을 보유할 수 있어 최신의 취약성 진단 스크립트로 취약성을 진단을 할 수 있다. 또한 취약성 데이터베이스는 CVE기반의 표준화된 진단 결과에 해결방안을 추가하여 취약성 진단과 동시에 해결방안을 얻을 수 있다.

참고문헌

[1] 김정희, "컴퓨터시스템 취약성평가 국제표준화동향", 한국정보보호진흥원 해외정보보호동향, 6월, 2003년, p.31 ~ p.41

[2] Ragi Guirguis, "Network and Host-based Vulnerability Assessments",http://www.sans.org, February 2004.

[3] Introduction to CVE, The Key to Information Sharing, http://cve.mitre.org/docs/docs2000/key_to_info_shar.pdf

[4] UNIX Security Checklist v2.0, http://www.cert.org/tech_tips/usc20_full.html

[5] M. Wojcik, T. Bergeron, T. Wittbold and R.Roberge "Introdution to OVAL," http://oval.mitre.org/documents/, MITRE Corporation, November 2003.

[6] Young Mi Gwon, Hui Jae Lee, Geuk Lee, "A Vulnerability Assessment Tool Based on OVAL in Linux System", IFIP International Conference, NPC2004, Network and Parallel Computing, October 2004

[7] Introduction to Nessus, http://www.securityfocus.com/infocus/1741