情報保護學會論文誌
第17卷 第5號, 2007. 10

정보시스템 취약도 계산 방법 개발

박 중 길
한국전자통신연구원 부설 연구소

A development of weakness calculation method for information system

Joong-Gil Park
ETRI Network & Communication Security Division

요 약

취약점 분석은 운영중인 정보시스템을 구성하는 다양한 장비에 존재하는 취약점을 식별하여 제거할 수 있도록 도와준다. 취약점 분석의 결과로 각 장비의 취약점에 대한 설명과 제거방법을 기술하는 것은 가능하나, 해당 정보시스템의 전체적인 취약도 및 안전도를 나타내기는 어렵다. 이를 위해, 취약점 분석을 수행한 인력이 수행 과정에서 획득한 여러 정보를 종합하는데, 이는 수행 인력의 자의적 판단이 비중이 높아 객관적이지 못 하다. 본 논문에서는 식별된 취약점을 종합하여 각 장비별, 시스템별, 정보시스템 전체의 취약도를 수치로 표현하는 방법을 제안한다.

ABSTRACT

Vulnerability analysis helps to remove discriminating vulnerabilities that exist in various equipment that composes operating information system. It is possible to explain representation and exclusion method about vulnerabilities of each equipment by vulnerability analysis. But it is difficult to display the weakness of whole information system. To do this, analyst synthesizes several information that achieved by vulnerability analysis. But the existing method does not provide fair evaluation because operators' personal opinion. In this paper, we explain about method that unites discriminatively vulnerable point and expresses whole weakness degree in numerical value by equipment, by system class, or by overall system.

Ⅰ. 서 론

조직의 주요업무 수행을 지원하는 정보시스템에 존재하는 취약점과 관련한 사항을 정확하게 파악하고 있는 것은 중요하다.

취약점은 각종 침해사고에 악용될 수 있기 때문에 이를 공격자보다 미리 파악하여 제거해야 하기 때문이다[1].

이러한 이유로 각 조직은 자신의 정보시스템에 존재하는 취약점을 식별하기 위해 취약점 분석을 시행한다[2]. 취약점 분석은 정보시스템을 구성하는 장비에 존재하는 취약점을 식별하여 제거할 수 있도록 도와준다. 취약점 분석의 결과로 각 장비의 취약점에 대한 설명과 이를 제거하는 방법을 기술하는 것은 가능하나, 해당 정보시스템의 전체적인 안전도를 객관적으로 설명하기는 어렵다[3]. 안전도 설명을 위해 취약점 분석을 수행한 인력이 수행 과정에서 획득한 여러 정보를 종합하여 표현하는데, 수행 인력의 주관이 많이 첨가되어 객관적 결과를 생산하지 못 하는 것이 일반적이다.

접수일: 2007년 7월 6일; 채택일: 2007년 8월 14일
† 주저자, jgpark@etri.re.kr
‡ 교신저자, jgpark@etri.re.kr

그러나 한 조직의 정보시스템에 존재하는 취약점의 상황을 상징적으로 표현하는 방법이 필요한데, 이는 조직 경영진에게 현황을 단순한 형태로 보고하거나 다른 조직 간 또는 다른 정보시스템과의 비교를 위해 필요하다[3]. 본 논문에서는 정보시스템의 취약점 상황을 표현하기 위해 사용할 수 있는 방법을 제안하다. 2장에서 현재 사용하고 있는 방법의 문제점을 알아 보고, 3장에서는 이와 관련된 연구내용을 알아 본다. 4장에서는 제안하는 방법을 설명하고 기존 방법과 비교하고, 마지막으로 5장에서 향후에 진행되어야 할 연구 내용을 살펴 본다.

Ⅱ. 기존의 취약점 현황 표현 방법

2.1. 종합점수 계산

특정 정보시스템의 취약점 현황을 상징적으로 표현하기 위해서 사용한 방법은 취약점 분석 결과로 획득한 내용을 수치화하는 것이다. 내용을 수치화하는 방법은 결과를 직관적으로 전달하는 데에 유리하기 때문이다. 취약점 현황을 수치로 표현하는 데에는 위험도가 적용된다. 즉, 해당 분야에서 기대할 수 있는 최대의 위험도 중에서 어느 정도의 위치에 있는 지를 표현한다. 예를 들어, [표 1]과 같은 현황수준과 기준을 활용하여 취약점 분석 수행자는 해당 분야가 어느 수준에 해당하는 지를 결정한다[2][3].

[표 1]과 같은 기준을 취약점 분석을 수행한 모든 분야에 적용하여 분야별 점수와 시스템 전체 점수를 계산한 후에 이를 백분율로 계산한다. 이러한 과정을 모두 수행하면 [표 2]와 같은 결과가 계산된다.

[표 2]의 내용을 보면, 4개 분야의 점수가 계산된 후에 이를 평균한 종합점수를 점수와 등급으로 표현하고 있다.

2.2. 기존 방법의 문제점

기존에 사용하고 있는 방법이 위험도를 계산하여 취약점 현황이 어느 정도 수준에 있는 지를 수치로 표현하는 데, 이 방법에 대한 문제점은 다음과 같은 것이 있다.

- 기준의 주관적 판단 가능

해당 분야에 대한 점수는 표 1의 기준에 근거한다.

〔표 1〕 위험도 계산에 활용하는 현황수준과 기준

현황수준	점수	기준
1등급	1	전체적으로 통제가 체계적으로 수립되어 있고, 정확하게 이행 및 관리되고 있음
2등급	2	전체적으로 통제가 구현되어 있고, 이행 및 관리가 양호함
3등급	3	전체적으로 통제가 구현되어 이행 및 관리되고 있으나, 주기적인 개선 및 점검이 필요함
4등급	4	기본적인 통제만 수립되어 이행 및 관리되고 있음
5등급	5	기본적인 통제조차 수립되어 있지 않고, 이행 및 관리가 거의 이루어지지 않고 있음

〔표 2〕 정보시스템 취약점 분석 종합점수(등급)

분야	분야별 점수(등급)	종합점수(등급)
A 분야	83점(우)	79점(미)
B 분야	78점(미)	
C 분야	89점(우)	
D 분야	66점(양)	

해당 분야에 대한 통제 구현 여부와 이행 및 관리 실태에 근거하는데, 분석자마다 이를 다르게 해석할 수 있다. 예를 들어, 2등급과 3등급의 차이는 통제는 구현되어 있고 이행 및 관리로 이루어지나 주기적 개선 및 점검이 필요한 지 여부로 구분한다. 그러나 이 두 등급의 차이를 명확하게 객관적으로 설명하기 어렵다. 그러므로 취약점 분석을 수행한 인력의 경험 즉, 주관이 크게 작용한다. 해당 분야에 대한 점수 배정은 궁극적으로 정보시스템의 종합점수에까지 활용되므로 첫 단계에서 반영된 주관적 판단이 종합점수까지 영향을 주게 된다[4].

- 존재 취약점 특성 파악의 어려움

분야별로 점수를 계산했으나 특정 분야가 어떤 취약점 때문에 점수가 낮게 계산되었는지를 확인하기 어렵다. 예를 들어 표 2의 A 분야와 D 분야의 경우, 83점과 66점으로 큰 차이를 보이고 있지만 D 분야에 존재하는 어떤 취약점 때문에 A 분야보다 낮은 점수가 계산되었는 지를 확인하기 어렵다.

- 만점에 대한 정의 부재

점수는 100점을 만점으로 어느 정도의 수준인지를 설명하는 것인데, 만점의 의미를 정의하기가 어렵다. 특정 정보시스템이 만점이 나오기 위해서는 모든 분

야가 표 1의 1등급에 해당되어야 한다. 그러나 1등급의 기준에 대한 설명도 명확하지 않다. "전체적으로 통제가 체계적으로 수립되어 있고, 정확하게 이행 및 관리되고 있음"이라고 해서 해당 분야에 취약점이 전혀 존재하지 않는다고 정의할 수 없다.

- 점수 차이의 의미 부재

분야별로 계산된 점수의 차이를 설명할 수 없다. 정보시스템을 관리하는 조직의 관점에서는 90점으로 계산된 분야와 80점으로 계산된 분야의 차이가 명확하게 설명되어야 한다. 그러나 기존의 방법은 취약점 분석 수행자가 표 1에 근거하여 각 분야에 대한 점수를 지정한 것이기 때문에 주관적 판단에 의해 차이가 발생하므로 명확한 설명이 어렵다.

Ⅲ. 취약점 점수 계산 방법 : CVSS

CVSS(Common Vulnerability Scoring System)는 취약점의 심각도를 계산하여 제거의 시급성과 우선순위 결정을 지원하는 점수 계산 방법이다[5].

DHS(Department of Homeland Security)의 NIAC (National Infrastructure Advisory Council)에서 시작하였고 현재는 FIRST(Forum of Incident Response and Security Teams)에서 관련 내용을 연구중이다[6].

CVSS의 가장 큰 역할은 취약점의 심각도를 발표하는 주체에 따라 상이한 표현을 하나의 수치로 단일화한다는 것이다. 예를 들어, 동일한 취약점에 대해 마이크로소프트에서는 "Important"라 하고, CERT에서는 "47.31", Secunia에서는 "Less Critical"로 정의하여 혼돈을 발생시킨다. 이러한 혼돈을 방지하고 먼저 제거해야 할 취약점의 순위를 정하는 데에 CVSS를 사용할 수 있다[5].

3.1. Metric과 Formula

CVSS는 세 개의 Metric과 Formula로 구성되며, 세 개의 Metric에는 총 12개의 항목이 포함된다[7][8][9].

3.1.1. Base Metric과 Formula

Base Metric은 취약점의 기본적 특징 즉, 취약점을 실현할 수 있는 위치와 복잡도 등을 활용하여 계산되며 시간이 경과해도 변경되지 않는 값이다. [표 3]은 Base Metric의 7개 항목에 대한 설명이며, 각 항목에 대한 선택값 지정은 해당 소프트웨어를 개발한 주체에 의해 지정되는 것이 일반적이다.

〔표 3〕 Base metric 설명

문 항	설 명	선택값	점수
Access Vector	취약점을 실현(Exploit)할 수 있는 위치	Local	0.7
		Remote	1.0
Access Compl -exity	취약점 실현을 위해 필요한 동작의 복잡도	High	0.8
		Low	1.0
Authenti -cation	취약점 실현을 위해 인증과정을 통과해야 하는 지 여부	Required	0.6
		Not Required	1.0
Confide n -tiality Impact	취약점 실현이 목표시스템의 기밀성에 주는 영향의 정도	None	0.0
		Partial	0.7
		Complete	1.0
Integrity Impact	취약점 실현이 목표시스템의 무결성에 주는 영향의 정도	None	0.0
		Partial	0.7
		Complete	1.0
Avail -ability Impact	취약점 실현이 목표시스템의 가용성에 주는 영향의 정도	None	0.0
		Partial	0.7
		Complete	1.0
Impact Bias	특정 보호기능에 더 큰 비중을 주기 위한 값		
	- Normal인 경우 · Confidentiality : 0.333 · Integrity : 0.333 · Availability : 0.333	- Confidentiality를 강조하는 경우 · Confidentiality : 0.5 · Integrity : 0.25 · Availability : 0.25	
	- Integrity를 강조하는 경우 · Confidentiality : 0.25 · Integrity : 0.5 · Availability : 0.25	- Availability를 강조하는 경우 · Confidentiality : 0.25 · Integrity : 0.25 · Availability : 0.5	

Base Metric에 의해 지정된 값은 [표 4]의 계산식에 의해 계산된다. 계산결과는 취약점의 심각도를 의미하며, 최대값 10점에서 최소값 0점 사이의 점수로 계산된다. 이 결과는 다음 단계인 Temporal Formula에 입력된다.

〔표 4〕 Base Score 계산식

```
Base Score = (10 * Access Vector * Access Complexity
              * Authentication
              * ((Confidentiality Impact * Impact Bias)
                + (Integrity Impact * Impact Bias)
                + (Availability Impact * Impact Bias))
```

3.1.2. Temporal Metric과 Formula

Temporal Metric은 취약점의 실현과 치료방법에 대한 설명이며, 시간의 경과에 따라 변경될 수 있는 값이다. [표 5]는 Temporal Metric의 3개 항목에 대한 설명이며 Base Metric과 같이 해당 소프트웨어를 개발한 주체에 의해 지정된다.

〔표 5〕 Temporal metric 설명

문항	설명	선택값	점수
Exploit-ability	취약점 실현 동작의 복잡도 (Exploit Code 존재 여부)	Unproven	0.85
		PoC	0.9
		Functional	0.95
		High	1.0
Remedi-ation Level	취약점 치료방법의 수준	Official Fix	0.87
		Temporary Fix	0.9
		Workaround	0.95
		Unavailable	1.0
Report Confi-dence	취약점을 발표한 주체에 대한 신뢰 수준	Unconfirmed	0.9
		Uncorroborated	0.95
		Confirmed	1.0

Temporal Metric에 의해 지정된 값은 표 8의 계산식에 의해 계산된다. 계산결과는 취약점의 심각도를 의미하며, Base Formula의 결과에 따라 0~10점 사이의 점수를 계산한다. 이 결과는 다음 단계인 Environmental Formula에 입력된다.

〔표 6〕 Temporal Score 계산식

```
Temporal Score = Base Score * Exploitability
                             * Remediation Level
                             * Report Confidence
```

3.1.3. Environmental Metric과 Formula

Environmental Metric은 취약점이 실현됨에 따라 예상할 수 있는 피해의 규모를 나타내는 값이며, 해당 소프트웨어를 사용하는 정보시스템의 환경에 따라 가변적이므로 정보시스템을 운영하는 주체가 지정한다. [표 7]은 Environmental Metric의 2개 항목에 대한 설명이다.

Environmental Metric에 의해 지정된 값은 [표 8]의 계산식에 의해 계산된다. 계산결과는 최종적으로 계산된 취약점의 종합점수이며, 제거해야 할 취약점의 우선순위로 사용할 수 있다.

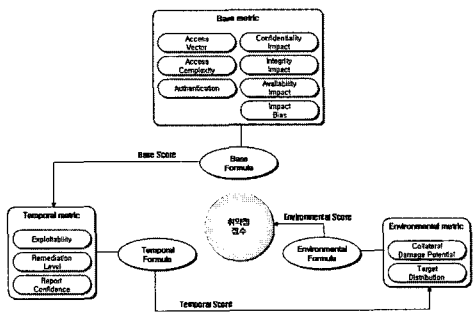
〔표 7〕 Environmental Metric 설명

문항	설명	선택값	점수
Collateral Damage Potential	취약점 실현으로 인해 발생 가능한 실제 피해의 수준	None	0.0
		Low	0.1
		Medium	0.3
		High	0.5
Target Distribution	해당 취약점이 실현될 수 있는 목표시스템의 규모	None	0.0
		Low	0.25
		Medium	0.75
		High	1.0

〔표 8〕 Environmental Score 계산식

```
Environmental Score = ((Temporal Score
                        + ((10 - Temporal Score)
                        * Collateral Damage Potential))
                        * Target Distribution)
```

CVSS의 Metric과 Formula를 활용한 취약점 점수 계산 방법을 종합하면 [그림 1]과 같다



〔그림 1〕 CVSS 계산 과정

3.2. CVSS 계산 예제

CVSS를 활용하여 취약점의 점수를 계산한 예의 결과는 [표 9]와 같다. [표 9]에서 취약점 #1은 Microsoft Outlook Express Scripting Vulnerability (CAN-2004-0380)이며, 취약점 #2는 Microsoft LSASS Vulnerability(CAN-2004-0533), 취약점 #3은 BGP potential DoS(CAN-2004-0589)를 의미한다.

[표 9] CVSS 계산 예제

		취약점 #1	취약점 #2	취약점 #3
Base Metric	Access Vector	Remote	Remote	Remote
	Access Complexity	High	Low	High
	Authentication	Not -required	Not -required	Not -required
	Confidentiality Impact	Complete	Complete	Complete
	Integrity Impact	Complete	Complete	Complete
	Availability Impact	Complete	Complete	Complete
	Impact Bias	Normal	Normal	Availability
Base Score		8.0	10.0	4.0
Temporal Metric	Exploitability	Functional	Functional	Unproven
	Remediation Level	Official-fix	Official-fix	Unavailable
	Report Confidence	Confirmed	Confirmed	Confirmed
Temporal Score		6.6	8.3	3.4
Environ-mental Metric	Collateral Damage Potential	None	None	None
	Target Distribution	High	High	High
Environmental Score		6.6	8.3	4.0

[표 9]의 결과를 보면, Base Score, Temporal Score, Environmental Score 모두 취약점 #2가 가장 높게 계산되었으며, 이는 취약점 #2를 가장 먼저 제거해야 한다는 의미로 해석할 수 있다.

Ⅳ. 정보시스템 취약도 계산 방법 제안

2장에서 설명한 바와 같이 기존의 종합점수는 정보시스템의 안전도를 표현한 것으로서, 100% 안전한 상태를 기준으로 해당 정보시스템은 어느 수준인가를 표현했다. 그러나 제안하는 방법에서는 종합점수를 취약도로 표현하며, 취약점이 발견되지 않은 상태를 0점으로 하고 취약점 발견시마다 일정 점수를 가산하는 형태이다. 즉 취약도는 점수가 클 수록 취약한 상태를 의미한다.

4.1. CVSS 활용 및 수정

CVSS를 정보시스템 취약도 계산에 활용하기 위해 다음과 같은 수정이 필요하다.

4.1.1. Metric과 Formula 수정

- Environmental Metric과 Formula 제외
  Environmental Metric과 Formula는 취약점 자체에 대한 내용이 아니고 해당 취약점이 존재하는 장비와 시스템에 관한 질문이므로 해당 조직의 특성에 따라 결과가 달라질 수 있기 때문에 제안하는 방법에서 활용하지 않는다. 취약점만의 특징을 활용함으로서 조직 특성 반영시에 발생할 수 있는 오류를 방지하고자 하는 것이다.
- Temporal Metric의 Report Confidence 제외
  Report Confidence는 취약점을 발표한 주체에 대한 신뢰수준인데, 취약점 분석 결과로 정리되는 취약점은 표 9에서 사용된 취약점과는 설명하는 수준이 다르기 때문에 제외한다. 표 9에서 사용된 취약점은 하나의 H/W나 S/W에서 발견된 취약점이지만, 취약점 분석에서 획득하는 취약점은 이와는 다른 수준의 취약점들이며, 대표적인 예는 다음과 같다.
  - 추측 가능한 패스워드를 사용하는 계정이 존재
  - Guest 계정이 존재
  - Anonymous FTP가 활성화

이러한 형태로 설명되는 취약점에 발표 주체를 정의할 수 없기 때문에 Report Confidence 항목은 제외한다.

4.1.2. 용어의 한글화

제안하는 방법에서 사용하는 Metric과 항목에 대해 표 10과 같이 한글화한다.

〔표 10〕 제안 방법에서 활용할 용어

CVSS 사용 용어		한글화 용어	
Base Metric	Access Vector	기본 측정	접근 위치
	Access Complexity		접근 복잡도
	Authentication		인증 필요 여부
	Confidentiality Impact		기밀성 영향
	Integrity Impact		무결성 영향
	Availability Impact		가용성 영향
	Impact Bias		보호기능 비중
Base Score		기본 점수	
Temporal Metric	Exploitability	가변 측정	실현가능성
	Remediation Level		보호대책 수준
Temporal Score		가변 점수	

4.2. 제안하는 취약도 계산 방법

4.2.1. 취약점 점수 계산 방법

취약점 분석 결과로 생산된 취약점에 대해 점수를 계산하는 식은 [표 11과 같이 CVSS의 계산식에서 제외하기로 한 항목 부분만을 뺀 형태이다.

〔표 11〕 제안한 취약점 가변 점수 계산식

```
기본 점수 = (10 * 접근 위치 * 접근 복잡도
             * ((기밀성 영향 * 보호기능 비중)
             + (무결성 영향 * 보호기능 비중)
             + (가용성 영향 * 보호기능 비중))

가변 점수 = 기본 점수 * 실현 가능성 * 보호대책 수준
```

서버에 Guest 계정이 존재하는 취약점에 대해 [표 11]을 활용하여 취약점 점수를 계산하면 [표 12]와 같다.

Guest 계정이 존재하는 취약점의 기본 점수는 6.9점으로 계산되었고, 가변 점수는 6.0점으로 계산되었으며,

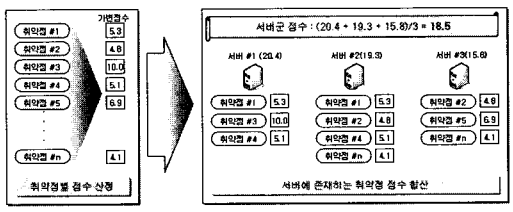
〔표 12〕 Guest 계정 존재 취약점에 대한 가변 점수 계산 예제

측정분야	항목	선택값	선택값 점수
기본측정	접근 위치	원격	1.0
	접근 복잡도	단순	1.0
	인증 필요 여부	불필요	1.0
	기밀성 영향	부분적 영향	0.7
	무결성 영향	부분적 영향	0.7
	가용성 영향	부분적 영향	0.7
	보호기능 비중	Normal	각 항목에 0.333
기본점수	10*1.0*1.0*((0.7*0.333)+(0.7*0.333) +(0.7*0.333)) = 6.9		
가변측정	실현 가능성	가능	1.0
	보호대책 수준	대책 존재	0.87
가변점수	6.9 * 1.0 * 0.87 = 6.00		

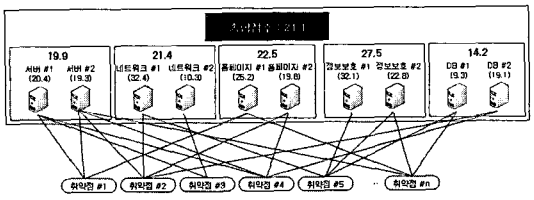
취약점을 대표할 수 있는 점수는 가변 점수인 6.0점이다. 해당 취약점을 제거할 수 있는 보호대책이 존재하는 이유로 가변 점수가 기본 점수보다 낮게 계산되었다.

4.2.2. 정보시스템 취약도 계산 방법

계산된 취약점 점수를 기반으로 정보시스템의 취약도를 계산하는데 [그림 2]와 같다. [그림 2는 서버군의 점수를 계산하는 방법으로 모든 서버에 존재하는 취약점을 정리하고 각 취약점별 점수를 계산하여, 각 서버별로 존재하는 취약점의 점수 합계를 계산한다. 이 결과가 서버별 점수가 되고, 모든 서버의 점수를 평균한 것이 서버군의 점수가 된다. 여기에서 계산된 모든 점수는 취약도의 의미를 갖는다. 각 서버의 점수를 평균하여 서버군 점수를 구한 방법으로 취약점 분석의 모든 분야에 대해 동일하게 적용하면 [그림 3]과 같다.



〔그림 2〕 정보시스템 취약도 계산방법



〔그림 3〕 정보시스템 종합점수 계산 방법

서버, 네트워크, 홈페이지, 정보보호시스템, DB 군의 점수를 계산한 후, 이의 평균을 구한 21.1점이 해당 정보시스템의 종합점수가 된다. 종합점수는 취약도를 의미한다.

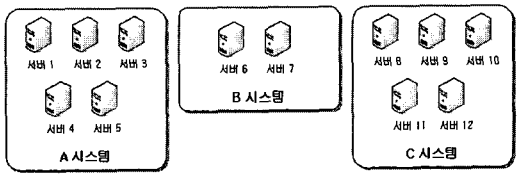
10점의 취약도가 계산되기 위해 필요한 조건을 기본 측정 항목과 가변 측정 항목을 활용하여 설명하면 다음과 같다.

- 공격자가 원격에서 별도의 인증 절차 없이 매우 단순한 방법으로 실현할 수 있음
- 기밀성, 무결성, 가용성 중에서 적어도 하나의 기능을 완전하게 방해할 수 있음
- 취약점을 실현할 수 있는 Exploit Code를 수정 없이 사용할 수 있거나 또는 Exploit Code 없이 취약점을 실현할 수 있으며, 현재 이에 대한 보호대책은 없음

종합하면 공격자에 의한 실현이 언제나 가능하며 보호대책이 없는 취약점이 한 개 존재할 때에 취약도는 10점을 갖게 된다. [그림 3]에서 종합점수로 21.1점이 계산된 것은 해당 정보시스템을 구성하는 모든 장비에 평균적으로 2개 이상의 치명적인 취약점이 존재한다는 것을 의미한다.

4.3. 실험 결과

제안한 방법을 근거로 실제 데이터를 이용하여 취약도를 계산한다. [그림 4]와 같이 12대의 서버가 3개 시스템을 구성하고 있는 상황에서 취약점 분석 결과, [표 13]과 같이 13개의 취약점이 발견된 상황이다. [표 13]은 13종의 취약점에 대한 기본 점수를 계산한 결과이다. 최소 2.52점에서 최대 10점까지의 결과가 계산되었다. [표 14]는 각 취약점의 가변 점수를 계산한 결과이다.



〔그림 4〕 실험 환경 서버 구성

〔표 13〕 취약점별 기본 점수 계산 결과

번호	취약점명	접근 위치	접근 복잡도	인증 필요 여부	기밀성	기밀성 비중	무결성	무결성 비중	가용성	가용성 비중	기본 점수
#1	시스템 정보 노출	원격	단순	불필요	큰 영향	0.50	부분적 영향	0.25	영향 없음	0.25	6.75
#2	추측가능한 패스워드 사용	원격	단순	불필요	큰 영향	0.33	큰 영향	0.33	큰 영향	0.33	10.00
#3	패스워드가 없는 계정 존재	원격	단순	불필요	큰 영향	0.33	큰 영향	0.33	큰 영향	0.33	10.00
#4	Guest 계정 존재	원격	단순	불필요	부분적 영향	0.33	부분적 영향	0.33	부분적 영향	0.33	7.00
#5	불필요한 서비스 활성화	원격	복잡	필요	부분적 영향	0.50	영향 없음	0.25	부분적 영향	0.25	2.52
#6	Anonymous FTP 활성화	원격	단순	불필요	부분적 영향	0.33	영향 없음	0.33	영향 없음	0.33	2.33
#7	쓰기 가능한 Anonymous FTP 활성화	원격	단순	불필요	부분적 영향	0.25	부분적 영향	0.50	영향 없음	0.25	5.25
#8	SMTP expn/vrfy 명령을 통한 시스템 정보 획득 가능	원격	복잡	불필요	부분적 영향	0.50	영향 없음	0.25	영향 없음	0.25	2.80
#9	구 버전의 SNMP 사용	원격	단순	불필요	부분적 영향	0.50	영향 없음	0.25	부분적 영향	0.25	5.25
#10	쓰기 가능한 공유 폴더 존재	원격	단순	필요	큰 영향	0.25	큰 영향	0.50	영향 없음	0.25	4.50
#11	보안패치 미적용	원격	단순	불필요	부분적 영향	0.33	부분적 영향	0.33	부분적 영향	0.33	7.00
#12	NetBIOS 널 세션 취약점	원격	단순	불필요	부분적 영향	0.33	부분적 영향	0.33	부분적 영향	0.33	7.00
#13	숨김 공유 존재	원격	복잡	필요	부분적 영향	0.50	부분적 영향	0.25	영향 없음	0.25	2.52

〔표 14〕 취약점별 가변 점수 계산 결과

번호	취약점명	기본 점수	실현 가능성	보호 대책 수준	가변 점수
#1	시스템 정보 노출	6.75	가능	대책 존재	5.87
#2	추측가능한 패스워드 사용	10.00	가능	대책 존재	8.69
#3	패스워드가 없는 계정 존재	10.00	가능	대책 존재	8.69
#4	Guest 계정 존재	7.00	가능	대책 존재	6.08
#5	불필요한 서비스 활성화	2.52	PoC 수준	대책 존재	1.97
#6	Anonymous FTP 활성화	2.33	가능	대책 존재	2.03
#7	쓰기 가능한 Anonymous FTP 활성화	5.25	가능	대책 존재	4.57
#8	SMTP expn/vrfy 명령을 통한 시스템 정보 획득 가능	2.80	가능	대책 존재	2.44
#9	구 버전의 SNMP 사용	5.25	가능	대책 존재	4.57
#10	쓰기 가능한 공유 폴더 존재	4.50	가능	대책 존재	3.92
#11	보안패치 미적용	7.00	PoC 수준	대책 존재	5.48
#12	NetBIOS 널 세션 취약점	7.00	가능	대책 존재	6.08
#13	숨김 공유 존재	2.52	가능	대책 존재	2.19

이러한 상황에서 서버별, 시스템별, 전체 시스템의 취약도를 계산한 결과는 표 15와 같다.

〔표 15〕 가상 취약도 계산 결과

시스템명	서버 종류	취약점 종류	서버별 취약도	시스템별 취약도	전체 시스템 취약도
A 시스템	서버 1	#1 #2 #7 #6	21.57	15.08	16.11
	서버 2	#1 #2 #8	17.00		
	서버 3	#2 #8	11.13		
	서버 4	#2 #8	11.13		
	서버 5	#1 #2	14.56		
B 시스템	서버 6	#3 #4 #10 #11 #12 #13	32.44	24.00	
	서버 7	#5 #6 #11 #12	15.56		
C 시스템	서버 8	#7 #12	10.65	13.99	
	서버 9	#2 #8 #9	15.70		
	서버 10	#2 #8 #9	15.70		
	서버 11	#5 #7 #12 #13	14.81		
	서버 12	#8 #9 #12	13.09		

가장 취약도가 높은 서버는 32.44점이 계산된 서버 6이며, B 시스템이 24점으로 계산된다. 서버 6에는 치명적 취약점이 세 개 이상 존재한다는 의미이다. 가장 낮은 취약도를 갖는 서버 3, 서버 4에도 치명적 취약점이 하나 이상은 존재한다는 것을 나타낸다.

4.4. 제안한 방법의 장점

제안한 방법으로 종합점수를 계산하면 다음과 같은 장점을 갖는다.

- 결과의 객관성 확보
  기존의 방법은 [표 2]에서 나타낸 것처럼 점수가 계산된 근거를 확보하기가 어렵다. 이는 점수들이 취약점 분석 수행자의 주관에 의존하는 바가 크기 때문이다. 그러나 제안된 방법은 취약점 분석 과정에서 획득한 취약점 정보만을 근거로 계산되기 때문에 모든 것이 수치로 계산되며 수행자의 주관이 반영될 여지가 적다[10].
- 종합점수 차이 설명 가능
  이미 설명한 바와 같이 종합점수 10점의 의미를 정의할 수 있기 때문에 서버별 또는 시스템별 종합점수 차이를 비교할 수 있다.]표 15]에서 서버 6은 서버 7에 비해 약 2배 정도 더 취약하다고 할 수 있다[11].
- 취약점 제거 후의 상태 확인 가능

취약점 제거 계획 수립시에 특정 취약점을 제거한 후의 취약도를 미리 계산할 수 있어 취약점 제거의 우선순위를 정할 수 있다. [표 15]의 결과에서 #2, #12 취약점을 제거하는 경우의 취약도 계산 결과는 표 16과 같다. 전체 시스템의 취약도가 16.11점에서 8.51점으로 감소된 것을 확인할 수 있다. 이와 같이, 특정 취약점을 제거할 경우에 취약도가 변화하는 모습을 보여줄 수 있어서 효율적인 취약점 제거에 활용할 수 있다.

V. 결론 및 향후 연구방향

취약점 분석의 결과인 장비별 취약점을 수치화하여 정보시스템 전체의 종합점수를 계산하는 방법을 제안하였다. 제안한 방법은 수행자의 주관적 판단이 반영될 수 있는 부분을 최소화하여 객관성을 확보하였고, 장비별 또는 시스템별 차이를 설명할 수 있다. 또한 특정 취약

〔표 16〕 특정 취약점 제거 후의 취약도 결과

시스템명	서버 종류	취약점 종류	서버별 취약도	시스템별 취약도	전체 시스템 취약도
A 시스템	서버 1	#1 #7 #8	12.88	6.39	8.51
	서버 2	#1 #8	8.31		
	서버 3	#8	2.44		
	서버 4	#8	2.44		
	서버 5	#1	5.87		
B 시스템	서버 6	#3 #4 #10 #11 #13	26.36	17.92	
	서버 7	#5 #6 #11	9.48		
C 시스템	서버 8	#7	4.57	6.87	
	서버 9	#8 #9	7.01		
	서버 10	#8 #9	7.01		
	서버 11	#5 #7 #13	8.73		
	서버 12	#8 #9	7.01		

점 제거 후의 상태를 미리 계산할 수 있어서 제거해야 할 취약점의 우선순위를 결정할 수 있는 장점을 가지고 있다. 이 장점들은 앞에서 제시한 기존의 방법이 갖는 문제점을 모두 해결한 것이다.

향후에는 다양한 형태의 정보시스템에 적용함으로서 계산에 사용되는 수치를 조정하는 연구가 이어져 H/W 및 S/W의 발전을 반영해야 한다. 그리고 장비의 중요도를 계산하는 방법 개발이 필요하다. 장비의 중요도에 따라 동일한 취약점이라도 정보시스템에 끼치는 영향에 차이가 발생할 것이며 정보시스템의 취약도는 이를 반영해야 하기 때문이다.

참고문헌

[1] 정보통신부, "정보통신기반보호법 가이드", 2004.

[2] TTA, "공공정보시스템 보안을 위한 위험분석 표준-개념과 모델", TTAS.KO-12.007, 1998.

[3] 박중길, "정량적 방법을 이용한 위험분석 방법론 연구", 정보처리학회 논문지 VOL. 13-C, pp. 851-858, 2006.

[4] GAO, "Homeland Security : A risk management approach can guide preparedness efforts", GAO-02-208T, 2001.

[5] Mike Schiffman, "A Complete Guide To Common Vulnerability Scoring System", FIRST, www.first.org/cvss/cvss-guide.html, 2005.

[6] John T. Chambers외 1명, "The Common Vulnerability Scoring System", National Infrastructure Advisory Council, 2004

[7] Ramaswamy Chandramouli외 3명, "Common Vulnerability Scoring System", IEEE Computer Society, pp. 85-89, 2006.

[8] Mike Schiffman 외 4명, "CVSS : A Common Vulnerability Scoring System", National Infrastructure Advisory Council, 2004.

[9] Peter Mell외 1명, "CVSS : A Complete Guide To Common Vulnerability Scoring System Version 2.0", NIST, 2007

[10] British Standards Institution(BSI), BS7799, 1999.

[11] ISO/IEC TR 13335, 2000.

〈著者紹介〉

박 중 길(Joong-Gil Park) 정회원

1986년 2월: 동국대학교 전자계산학과(학사) 졸업
1988년 2월: 서강대학교 전자계산학과(석사) 졸업
2002년 2월: 충남대학교 컴퓨터과학과(박사) 졸업
1988년~1999년 국방과학연구소 선임연구원
2000년~현재 한국전자통신연구원 부설 연구소 책임연구원
<관심분야>정보보호(컴퓨터 보안, 네트워크 보안)