情報保護學會論文誌
第 22 卷 第 5 號, 2012. 10

시스템 취약점 분석을 통한 침투 경로 예측 자동화 기법*

김 지 홍,† 김 휘 강‡
고려대학교 정보보호대학원

Automated Attack Path Enumeration Method based on System Vulnerabilities Analysis*

Ji Hong Kim,† Huy Kang Kim‡
Graduate School of Information Security, Korea University

요 약

조직 내에 정보자산들의 숫자가 증가하고, 관련된 취약점의 종류와 개수가 증가함에 따라 조직 내 네트워크에 어떠한 취약점이 존재하는지 파악하는 것이 점차 어려워지고 있다. 취약점 데이터베이스 및 이를 활용한 취약점 분석 관련 정량적 분석 기준들 역시 존재하지만, 각각의 보안전문가들의 주관적인 기준에 따른 평가방식과, 수동적 측정 방법으로 인해 네트워크 시스템의 위험도 및 공격 침투 경로를 정량적 평가에 기반하여 예측하기에 효율적이지 못한 문제가 있다. 본 논문에서는 자동화된 취약점 평가 및 공격 침투 경로 예측 시스템인 HRMS(Hacking and Response Measurement System)를 제안하고, 네트워크 시스템 내 예상 공격 경로를 도출한 결과를 제시하였다. HRMS는 정보자산에 대한 충분한 정보가 주어지지 않았다 하더라도, 기존에 알려진 시스템 또는 어플리케이션의 보안관련 평판치를 취약점 평가지표를 기반으로 계산하여, exploitability를 산정하고 attack graph 경로를 생성한다는 점에서 효율적이라 할 수 있다. 본 논문에서 제안한 HRMS를 이용하여 적극적인 취약점 분석을 통한 보안관리를 할 때에 공격경로 예상을 통한 능동적인 대응책을 마련할 수 있을 것이다.

ABSTRACT

As the number of information asset and their vulnerabilities are increasing, it becomes more difficult for network security administrators to assess security vulnerability of their system and network. There are several researches for vulnerability analysis based on quantitative approach. However, most of them are based on experts' subjective evaluation or they require a lot of manual input for deriving quantitative assessment results. In this paper, we propose HRMS(Hacking and Response Measurement System) for enumerating attack path using automated vulnerability measurement automatically. HRMS can estimate exploitability of systems or applications based on their known vulnerability assessment metric, and enumerate attack path even though system, network and application's information are not fully given for vulnerability assessment. With this proposed method, system administrators can do proactive security vulnerability assessment.

Keywords: Network Security, Attack Graph, System Vulnerability Evaluation

접수일(2012년 6월 18일), 수정일(2012년 9월 27일), 게재확정일(2012년 9월 27일)
* 본 연구는 국방과학연구소의 지원을 받아 수행하였습니다.(UD110051ED)
† 주저자, inside15@korea.ac.kr
‡ 교신저자, cenda@korea.ac.kr

I. 서 론

네트워크에 연결된 모든 시스템에 대한 보안은 방화벽이나 침입탐지시스템과 같은 장비만으로는 해결하기 어렵다. 중요 데이터를 지켜내기 위해서 네트워크 경계, 내부 네트워크, 운영체제와 어플리케이션 등 각 계층마다 적절한 보안대책을 실시하여야 한다. 이를 해결하기 위해 다계층 방어(The defense in depth)와 다계층 보안(multi layered security)이라는 개념이 생겨났다. 이 개념이 시사하는 바는 '특정 장비를 통해 해결할 수 없는 문제점을 여러 단계를 이용한 보안만이 정보보호를 할 수 있다.'다는 점이다. 한편, 계층 방어를 우회하거나 보안 상태를 점검하기 위해서 attack graph와 같은 공격 시나리오와 확률 기반 평가 모델들이 제안되고 있다.

Attack graph는 계층방어와 같은 네트워크 내에 있는 타겟 시스템에 공격자가 쉽게 접근할 수 있는 특정 경로를 알려준다[1]. 네트워크 보안을 측정하기 위한 확률적 매트릭을 제시함으로써 현재 네트워크 환경이 얼마나 취약한지를 확인할 수 있다. Attack graph를 발전시켜 최적화 값을 구할 수 있도록 Attack graph와 Genetic Algorithm을 결합한 hybrid 기법도 연구되었다[2].

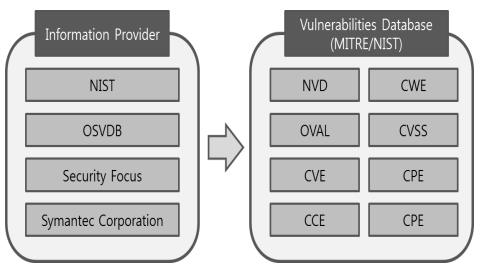
한편, 대표적인 취약점 정보와 분석 지표로는 CVE(Common Vulnerabilities and Exposures), CVSS(Common Vulnerability Scoring System), OSVDB(The Open Source Vulnerability Database)가 있다. 이들은 취약점 정보에 대한 데이터베이스를 제공한다[3][4]. Nikto, snort, nessus 등과 같은 네트워크 스캐닝 도구에 호환되어 스캐닝 도구를 이용했을 때 많은 취약점과 상세 정보를 알려준다. 그리고 많은 하드웨어와 소프트웨어들의 보안 등급을 평가하기 위해서 CVE를 이용하여 CVSS로 취약점에 대한 점수를 일반화하고 위험 우선순위를 통해 다른 취약점들과의 연관성을 결정하여 보안 수준을 점검할 수 있게 해준다[5].하지만 기존에 이러한 연구들에도 한계점이 존재한다. Attack graph 생성에 필요한 취약점에 관련 정보가 제시되지 않았다.

본 논문에서는 이러한 문제점들을 개선하여 자동화 취약점 측정 모델을 제안하였다. HRMS는 계층 방어 전략과 같은 네트워크에 존재하는 취약점을 통해 그래프 경로의 노드를 순차적으로 분석하고, 해당 시스템에 접근할 수 있는 최적의 경로와 확률적 접근 성공 가능성을 자동으로 계산해준다. 2장에서는 기존에 연구되어 있던 attack graph와 CVE, CVSS, OSVDB에 대해서 소개하고, 3장에서는 앞에서 소개된 제안방법을 개선하여 향상된 취약점 분석 모델을 소개하였다. 4장에서는 본 논문의 실험 네트워크 구조에서 HRMS를 이용한 실험 및 결과를 설명하였다. 마지막으로 5장에서는 결론 및 향후 연구방향을 제시하였다.

II. 관련 연구

보안 취약점에 따른 큰 피해의 우려로 인해서 미국, 일본, 중국과 같은 나라에서 국가 차원의 취약점 데이터베이스를 구축하고 있다[6]. 대표적으로 미국의 경우에는 취약점 데이터베이스인 NVD (National Vulnerability Database)를 구축하여 정보를 제공하고 있다. NVD의 정보가 많아지고, 같은 취약점에 대한 정보가 상이할 경우를 대비하여 취약점 식별자 체계인 CVE를 구축하였다. CWE(Common Weakness Enumeration)는 소프트웨어 취약점 목록을 작성한 것으로, 소프트웨어 취약점의 용어가 기업 및 보안업체마다 달라 용어 통일에 도움을 주기 위해 만들어 졌다. OVAL (Open Vulnerability and Assessment Language)은 컴퓨터 시스템의 보안설정 상태를 검사하기 위한 언어, CPE(Common Platform Enumeration)는 하드웨어·소프트웨어·운영체제·응용프로그램 등과 같은 취약점 정보를 식별하기 위한 데이터베이스이다. 이처럼 미국뿐만 아니라 일본, 중국 등의 국가에서 보안취약점에 대한 연구가 진행되고 있다. 최근 미국에서는 CVE, CWE, OVAL, CPE 등을 이용한 자동화 기술을 만들기 위해 추진 중에 있다[7].

이 장에서는 본 논문에서 제안한 자동화 기술에 필요한 취약점 데이터베이스인 NVD, CVE, CVSS, OSVDB의 체계들을 간략하게 설명하고, 자동화된 attack graph를 만들기 위해 기존에 제안된 attack



[그림 1] 미국의 취약점 관리 시스템

graph를 설명하였다.

2.1 NVD(National Vulnerability Database)

미국 정부는 대응책이 마련된 취약점에 대해서는 이를 공개하고, 보안 업체 및 일반 사용자, 해외 협력 기관과 공유해야겠다는 필요성을 느끼고 국토안보부 책임 하에 국가 취약점 데이터베이스인 NVD를 운영하여 효율적으로 통합 관리하고 있다. NVD의 정보가 방대해지고 NVD 이외의 취약점 데이터베이스들이 생겨나자, 미국 정보는 한 가지 취약점에 대해 각 취약점 데이터베이스의 정보가 상이할 경우에 가져올 혼란을 방지하기 위해 MITRE를 주축으로 하여 단일 취약점 식별자 체계인 CVE를 구축하였다.

2.2 CVE(Common Vulnerabilities and Exposures)

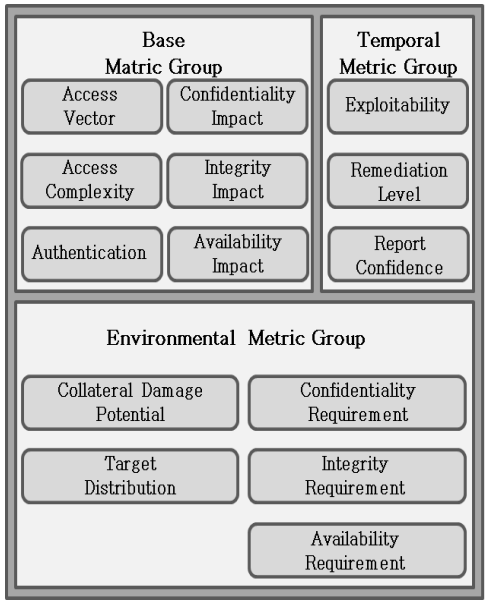
NVD의 정보가 많아지고, 여러 취약점 분석 기관 및 벤더들에 의해 중복된 취약점 정보들이 생성되면서 취약점 정보의 중복으로 인한 혼선을 최소화하기 위해 CVE를 구축하였다. MITRE(미연방정부의 정보보안 연구 기관)에서 CVE를 개발하여 관리하고 있으며, 아래와 같은 정보를 담아 취약점 식별정보를 고유하게 제공하고 있다.

- CVE 식별자 번호 (즉, "CVE-1999-0067")
- 보안 취약점에 대한 간략한 설명
- 관련된 모든 레퍼런스(즉, 취약성 보고서, 대응 방안)

2.3 CVSS(Common Vulnerability Scoring System)

다양한 취약점 분석 기관 및 벤더들에서 각각의 보안 취약점 등급을 이용하여 취약점 등급을 분류하고 있어, 효율성을 높이기 위해 취약점에 대한 정량적인 등급을 CVSS로 표준화하는 방안이 추진되고 있다. CVSS는 시스템, 네트워크 취약점의 영향과 특징을 전달하기 위한 오픈 프레임워크를 제공하고 있으며 [4][5], base metric, temporal metric, environmental metric 3가지 그룹을 각각 0부터 10까지 수치화된 점수를 만들어 취약성 수준을 판단할 수 있도록 해준다.

Metric 그룹은 [그림 2]와 같은 metric이 포함되며, 각 metric 그룹은 다음과 같은 특징을 가진다.



[그림 2] CVSS 매트릭 그룹

Base metric은 시간과 환경에 영향을 받지 않으면서 취약점의 고유 특성을 나타낸다. Temporal metric은 시간의 흐름에 따라 변경되는 취약점의 특징을 나타내고, environmental metric은 사용자 환경에 유일한 취약점의 특징을 말하고 있다. 또한, CVSS는 취약점 평가할 때 취약점 분석에 도움이 되는 대응방안을 제시하고, 각 metric에 대한 알고리즘을 제공한다.

CVSS와 관련된 연구로, 표준화된 CVSS 결과를 향상시키고자 Ruyi Wang은 base equation에 공격 목표 대상의 서버와 운영체제의 종류에 따른 취약점 기준을 나누어 제안하였다[8]. 그리고 Christian Fruhwirth는 Temporal equation 계산 방법에 CVSS가 구체적인 특정 계산 방법을 제공하지 않아 신뢰성 있는 결과를 나타내기 위해서 취약점의 exploit code는 Pareto 분포에 수렴한다는 것과 많은 수의 취약점이 발견될 때 Weibull 분포를 따른다는 것을 통해서 결과 값의 신뢰도를 높이고자 제안하였다[9].

2.4 OSVDB(The Open Source Vulnerability Database)

OSVDB는 취약점 정보를 누구나 제공하고 수정할 수 있는 공개형 데이터베이스로 NVD와 대등한 양의

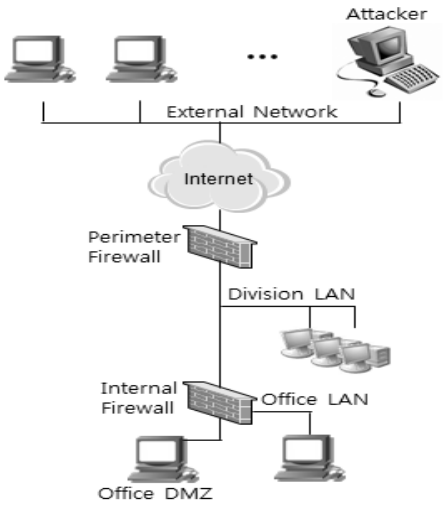
〔표 1〕 OSVDB에서 제공하는 데이터베이스 정보

구 분	설 명
Timeline	취약점 정보 게시 및 업데이트에 관련된 시간 및 날짜 등에 대한 정보로, 취약점 공개 날짜, 취약점 발견 날짜, 공격 코드가 공개된 날짜, 대응책이 제공된 날짜, 벤더에게 취약점을 통보한 날짜, 벤더로부터 응답을 받은 날짜, 벤더가 아닌 제 3자로부터 대응책을 받은 날짜를 제공
Keywords	취약점들의 검색 등을 용이하게 하기 위한 항목으로, 해당 취약점의 유형, 소프트웨어 및 버전, 벤더 등의 정보
Description	취약점 전반에 대한 사항을 요약한 항목으로 OSVDB를 사용하는 사용자들이 이 항목을 읽음으로써 취약점의 내용을 이해할 수 있도록 하기 위한 정보
Classification	해당 취약점을 위치, 공격형태, 영향, 취약점 발견 형태와 같은 항목을 기준으로 분류한 정보
Solution	해당 취약점에 대한 해결 및 대응 방법을 기술한 정보
Products	다른 DB의 affected products와 동일한 정보로, 취약점이 존재하는 제품을 기술
References	CVE, Secunia, 벤더 등의 외부 참고 링크
Credit	취약점 정보 작성자
CVSSv2 Score	CVSS에 의한 취약점의 위험도 평가 점수
Blogs	해당 취약점이 소개된 블로그
Comments	추가 의견

취약점 정보를 보유하고 있다. 공격 방법이나 형식 등을 상세하게 기술하고 있으며, 이에 대한 대응 방법도 제공하고 있다. OSVDB는 소프트웨어 벤더, 보안 전문가 및 연구자, 고객, 보안 기관, 전문 사용자, 일반 사용자 등 취약점 정보를 제공하려는 사람은 누구든지 제작에 참여할 수 있다. 이러한 공개 형태로 운영되는 OSVDB와 같은 데이터베이스의 가장 큰 문제는 제공 되어지는 정보를 모두 신뢰하기는 힘들다는 것이다. 그래서 OSVDB에서는 이러한 문제를 중재자(Moderator), 자료 분석가(Data Mangler), 개발자(Developers), 사용자(User)의 4가지 역할을 통해 체계적이고 효율적으로 데이터베이스를 운영/관리하여 제공되는 취약점 정보의 신뢰성을 보장하고 있다. OSVDB에서 제공하는 데이터베이스에 대한 정보는 〔표 1〕과 같다.

2.5 Attack Graph

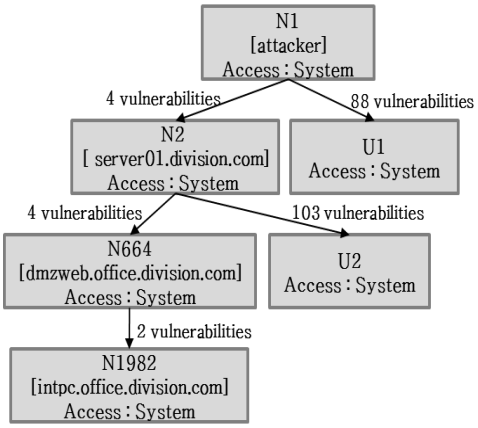
현재 가장 보편적으로 쓰이는 Attack graph는 Sushil Jajodia, Steven Noel에 의해 제안되었다〔10〕. Attack graph는 네트워크 시스템의 보안 취약성을 분석하는데 유용한 도구로 제안되고 있다. 이는 가능한 모든 공격 경로에 대한 최적의 공격 경로를 제시하기 위해 사용되는 데이터 구조를 말한다. Attack graph를 사용하여 최종 목적 시스템에 도달하기 위한 최적의 경로를 확인할 수 있다. 〔그림 3〕과 〔그림 4〕는 attack graph의 예시이다. 먼저 〔그림 3〕에서는 간단한 네트워크 구조를 나타낸다. 이 네트워크를 이용해 attack graph를 나타내면 〔그림 4〕



〔그림 3〕 실제 네트워크의 구조

와 같이 간략하게 표현된다. 〔그림 4〕의 그래프에서 루트 지점은 공격자이다. 선은 해당 네트워크에 접근하기 위해 공격자가 실행한 취약점의 수를 나타낸다. 공격자는 취약점이 있는 시스템에 접근해 또 다른 시스템의 취약점을 이용해서 최종 목적지까지 접근할 수 있다.

Attack graph의 확률을 계산하는 방식에 있어서 다양한 기존 연구들이 있는데, Steven Noel가 제안한 attack graph는 각 시스템에서 공격 대상 시스템의 취약점을 이용하여 다른 시스템으로 접근할 수 있는 확률적인 값을 계산한다〔11〕. 확률 값은 취약점의 상대적인 난이도를 나타내고 있으며, 공격 성공 확률



〔그림 4〕 실제 네트워크의 attack graph

의 산정법은 Bayesian Network, Markov Modeling을 이용하여 계산한다. 그리고 Lingyu Wang, Tania Islam은 attack graph에 존재할 수 있는 cycle의 처리방법에 관한 매트릭과 알고리즘을 제시하고 있다〔12〕.

Bin Wu와 Andy Ju An Wang은 CVSS Score을 기반으로 EVMAT(Enterprise Vulnerability Modeling and Assessment Tool)를 이용해 기업 자산의 취약점 관리를 제공한다〔13〕. 이는 네트워크 내의 취약한 소프트웨어로 네트워크 전체의 취약성 평가를 알려준다. 이처럼 CVSS Score 계산이나 EVMAT 같은 측정 방식은 관리자가 네트워크 시스템들의 어플리케이션들을 알고 있다는 가정아래에 계산이 가능하다. 다음 장은 이러한 과정 없이 공격자가 확인할 수 있는 최대한의 정보를 이용하여 자동화된 측정방법을 제공한다.

기존에 제안된 연구에서 몇 가지 단점과 개선이 필요하다. 첫째, 보안전문가들의 주관적인 의견에 대한 점수 부여 방법을 개선해야할 필요가 있다. 둘째, 어느 한 곳에서 제공되지 않는 취약점을 개선하기 위해 여러 데이터베이스를 중복되지 않게 수집하여 다양한 취약점이 존재하도록 해야 한다. 셋째, 자동화된 모델링을 구축할 필요가 있다. 지금까지 연구되어온 attack graph와 취약점 관련 정보를 활용하여 자동화된 취약점 측정 모델을 제안하고자 한다.

III. 제안 기법

HRMS는 알려진 취약점 정보를 이용하여 최적의 침입 경로를 제시한다. 자동화된 모델로 취약한 시스템의 접근 성공 확률을 자동으로 계산하여 우선순위에 맞게 상황을 제시하고, 공격 대응 방안으로 취약점에 대한 해결책을 제공한다.

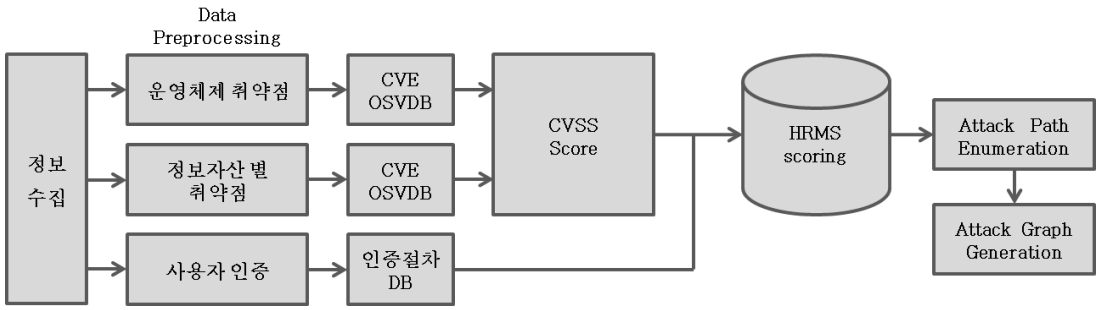
시스템 취약성을 확인하기 위하여 〔표 2〕와 같이 많은 정보들을 활용할 수 있지만, 자동화된 방법을 위해서 앞장에서 제시한 NVD, CVE, CVSS, OSVDB를 이용하여 scoring하는 방법과 attack graph를 활용하여 최적의 attack path가 나오도록 하였다. 더불어, 정보자산에 대한 상세 취약점 정보가 없을 경우를 고려하여, 그간 누적된 소프트웨어 및 벤더들의 과거 취약점 데이터에 근거하여 점수를 산출하여 계산하도록 하였다. 이는 현실적으로 보안 관리자 역시 내부의 모든 정보자산들에 존재하는 취약점에 대해 파악하기 어려운 점과, 해커가 외부에서 침입을 할 때 해커가 외부와 접점이 없는 내부 정보자산들에 대해서는 1차적인 정보가 없는 상황인 것을 반영할 수 있다는 점에서 현실적인 방안이라 할 수 있다.

〔그림 5〕와 같이 HRMS는 nmap, nessus 등의 네트워크 스캐닝 툴을 활용하여 시스템 정보를 알 수 있다. 네트워크 시스템 정보수집을 통해 운영체제, 시스템 등의 정보를 알 수 있으며, CVSS에서 사용자 권한 부분에 대한 정보가 있으므로 이들을 활용하여 시스템이 얼마나 취약한지 확인할 수 있다. 운영체제, 시스템에 대한 취약점 점수는 NVD와 OSVDB에 저장되어 있는 CVSS Score을 활용한다. 〔그림 5〕는 다음의 점수부여 방법을 통해서 네트워크 시스템들의

〔표 2〕 시스템 취약성을 확인할 수 있는 주요 지표와 도구

Attack Path Enumeration	Attack Tree, Attack Graph
Vulnerability Enumeration	CVE, CWE, CAPEC*, CPE
Vulnerability Scoring System	CVSS, CWSS**
Vulnerability Database	OSVDB, NVD
Scan Tool	nmap, nessus

*CAPEC(Common Attack Pattern Enumeration and Classification)
**CWSS(Common Weakness Scoring System)



〔그림 5〕 HRMS Overall Process Diagram

취약점 점수를 계산하고, 그 점수로 attack path enumeration을 생성하여 attack graph를 생성하는 diagram을 보여준다.

3.1 사용자 운영체제에 대한 점수부여 방법

취약점들의 대부분은 운영체제가 어떤 것인지와 관계가 깊다. 운영체제는 Windows 계열, UNIX 및 Linux 계열로 나눠볼 수가 있다. 참고로, Ruyi Wang은 운영체제 취약성에 대한 가중치를 다른 운영체제와 비교했을 때 상대적으로 취약한 운영체제를 기준으로 가중치를 부여하고 있다[8]. 이 연구에서 부여된 가중치는 주관적인 기준에서 점수를 분류하고 있는 단점이 있다. 운영체제에 존재하는 취약점은 운영체제 벤더의 보안개발능력 성숙도(예: 시큐어코딩 및 보안개발과 관련된 프로세스가 어떤 형태로 정립되어 있느냐)에 의존성이 높아 취약점이 많이 발견되고 패치대응이 느린 운영체제의 경우 지속적으로 동일한 문제를 보여 왔음을 관찰할 수 있다. 이러한 과거 데이터 추세를 근거로 운영체제 별 보안성의 평판도(reputation value)를 산정할 수 있다. 우리가 제안하는 방법은 이러한 관점에서 각 운영체제 별로 알려진 취약점의 CVSS 점수의 평균을 구하고, 취약점의 수를 이용하여 가중치를 부여하는 것이다.

OSVDB에서는 운영체제의 취약점 정보를 제공하고 있다. 현재 OSVDB에는 모든 운영체제에 대하여 4798개의 취약점에 대한 정보가 있다. 일반적으로 취약점이 많이 발견된 운영체제는 전반적으로 공격에 취약하다고 볼 수 있으며, 공격 시도 횟수도 많아지기 때문에 가중치를 부여한다. HRMS에서 가중치 부여 방식은 모든 운영체제의 취약점 수에 대하여 타겟 시스템의 운영체제에 대한 비율로 가중치를 부여하며, 취약점 데이터베이스가 업데이트 될 경우 가중치가 변화될 수 있다. 운영체제 별 가중치는 현재까지 각 벤더별로 1990년대 후반부터 2012년 중반까지 발견된 취약점의 수를 근거로 부여하였다.

3.2 사용자 인증 절차에 따른 점수부여 방법

HRMS에서는 시스템 및 네트워크의 취약성에 대

〔표 3〕 대표적인 OS에 대한 가중치 점수 분류

운영체제 종류	알려진 취약점의 개수 (2012년5월28일까지)	가중치	운영체제 종류	알려진 취약점의 개수 (2012년5월28일까지)	가중치
Windows XP Service Pack 2	250	0.5	HP-UX 11.x	29	0.1
Windows XP Service Pack 3	125	0.3	Solaris 2.x(2.6이하)	100	0.6
Windows 7	132	0.3	Solaris 2.8	82	0.1
Windows 7 Service Pack 1	55	0.1	Solaris 2.9	68	0.1
Windows Server 2008 SP2	186	0.4	Mac OS X 10.5.x	144	0.3
HP-UX 10.x	50	0.1	Mac OS X 10.6.x	45	0.1

해 산정을 할 때 영향을 미치는 factor로 운영체제 자체에 대한 취약점과 어플리케이션 등 정보자산에 대한 취약점 및 접근제어 시 사용되는 인증방식을 선정하였는데, 이는 해커가 외부로부터 침입을 할 때에 운영체제 및 어플리케이션의 취약점을 이용하여 구현된 인증방식과 무관하게 exploit을 하는 방법도 있고, 운영체제 및 어플리케이션의 취약점이 없는 경우에는 접근제어(access control)에 구현된 인증의 강도에 따라 공격의 성공유무가 좌우될 수 있다는 점을 고려한 것이다. 인증 기법은 크게 Password 인증, Biometric 인증, Token 인증, Ticket 인증, SSO 인증 방식으로 나눌 수 있다[14].

Password 인증은 비밀번호를 변경하기 전에는 static한 비밀번호가 유지된다는 점과 비밀번호의 길이와 복잡도를 높여 비밀번호가 강력하게 만든다 하더라도, 다양한 암호 해독기술을 이용하여 무력화 될 수 있음을 고려하여 인증기법 중에서도 가장 취약한 것으로 분류하였다.

Biometrics 인증 기법은 생체 인식 패턴과 저장된 패턴 데이터베이스와의 one-to-one 인증기법이다. Biometrics은 지문 스캔, 홍채 스캔, 심장 박동 패턴 인식 등이 있다. 생체 인식 장치가 지나치게 민감한 경우에는 false negative의 오류가 발생하게 되며, 충분히 민감하지 않은 경우에는 false positive의 오류가 발생하게 된다. 이러한 오류를 잘 활용한다면 공격성공도 가능하다.

Token 인증 기법은 비밀번호 인증보다는 한층 더 강력한 보안 수단이다. 이 인증 방법은 사용자 신원을 입증하기 위해서는 두 가지 이상의 인자를 사용해야 한다. 사용자 이름, 비밀번호, PIN 등과 함께 주체는 물리적 Token장치를 소유해야 한다. 하지만 이것은 분실이나 도난 위험이 있으며, 쉽게 노출될 수 있다는 단점이 있다.

Ticket 인증 기법은 신원을 증명하고 인증을 제공하기 위해 제 3의 기관을 이용한 메커니즘이다. 이는 서버가 해킹을 당하면 네트워크에 모든 비밀 키가 노출될 위험이 있다.

SSO 인증 기법은 사용자가 시스템 상에서 한번만 인증 되도록 하는 메커니즘이다. SSO를 이용하여 인증이 되면 사용자는 인증에 대한 재요청을 받지 않는다. 이러한 이유로 시스템에서는 강력한 패스워드만 허용한다.

공격자는 시스템에 접근하게 될 경우 인증요구를 고려해야 한다. 시스템 로그인에 사용할 수 있는 인증

〔표 4〕 사용자 인증 개수에 따른 점수 분류

인증 개수	점 수
Nothing	0.704
Single	0.56
Multiple	0.45

```
If(CVE>0 or OSVDB>0){
    cv=MAX(CVE CVSS score)
    cs=MAX(OSVDB CVSS score)
    (0≤cv, cs≤10)
    SV=MAX(cv, cs)

+{MIN(cv,cs)×{10-MAX(cv,cs)}}/10
  Else
    SV=0
```

〔그림 6〕 정보자산 별 공개된 취약점에 따른 점수 부여 알고리즘

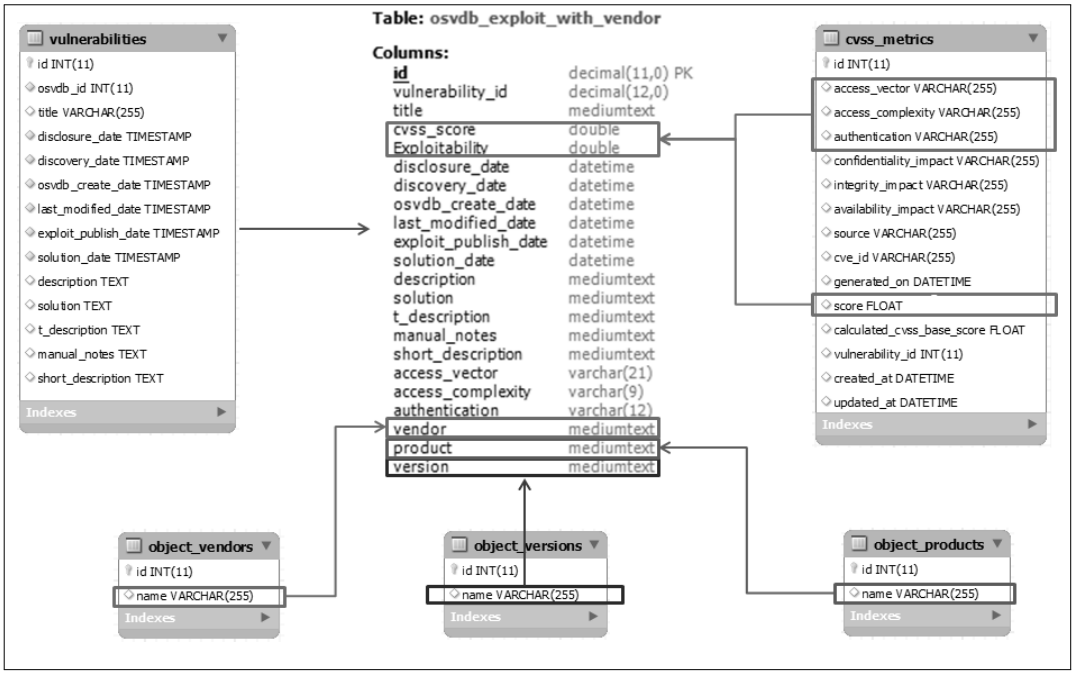
방법은 위에서 나타낸 것 같이 다양한 방법으로 적용할 수 있다. 공격자가 시스템에 접근하기 위해서 거쳐야할 인증 절차는 인증 절차가 적을수록 시스템 권한을 얻을 확률은 높다고 볼 수 있다. CVSS에서는 인증 방법이 없을 경우, 인증 방법이 하나만 있을 경우, 인증 방법이 여러 개일 경우로 나누어서 점수를 부여하고 있으며, 각 시스템 취약점 따른 인증 점수를 데이터베이스에서 제공하고 있다[4]. HRMS에서는 CVSS에서 제공하는 인증절차 개수에 따른 취약점 점수를 활용하여 〔표 4〕와 같이 공격 성공 확률 계산시 활용하였다.

3.3 정보자산 별 공개된 취약점에 따른 점수부여 방법

사용자 운영체제의 점수부여 방식과 마찬가지로 공개된 취약점 데이터베이스에서 정보자산(본 논문에서는 라우터, 스위치 등의 네트워크 장비, Apache, IIS 와 같은 응용프로그램 등을 모두 정보자산으로 표현)과 관련된 CVSS 값을 계산할 수 있다. Nessus와 같은 스캐너를 이용하여 탐지한 정보자산들에 대해 취약점을 스캔 한 후, 공개된 취약점 데이터베이스를 이용하여 발견된 취약점에 대해서 아래와 같이 점수를 계산한다.

3.4 데이터베이스 구축

본 절에서는 CVSS값을 HRMS에 어떻게 적용할지 제시하며 HRMS 자동화를 위한 데이터베이스 구축방법을 설명한다.



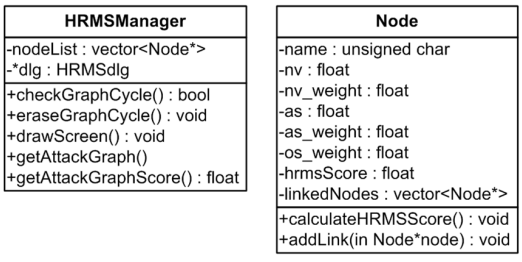
〔그림 7〕 OSVDB 데이터베이스를 활용한 Exploitability 테이블 구축

OSVDB와 CVE에서는 시스템 및 어플리케이션에 대한 CVSS score을 제공한다. 정보수집 과정에서 확인된 정보를 이용하여 〔그림 7〕과 같은 정보로 데이터베이스에 저장된 CVSS score을 HRMS에 적용시킨다.

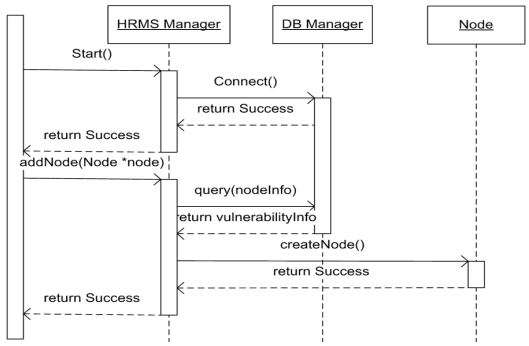
3.5 HRMS Display와 구조

〔그림 12〕의 HRMS Display는 타겟 네트워크 내의 노드를 검색하여 시스템 정보를 확인할 수 있도록 해준다. HRMS Display는 많은 필수 정보를 보존하면서 그래프를 생성하고 간결한 방식으로 attack graph를 생성해 준다.

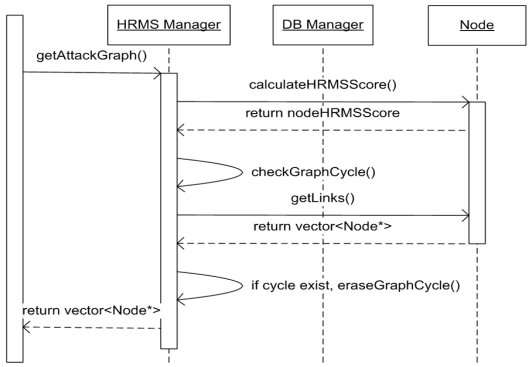
HRMS의 주요 클래스로는 〔그림 8〕과 같이



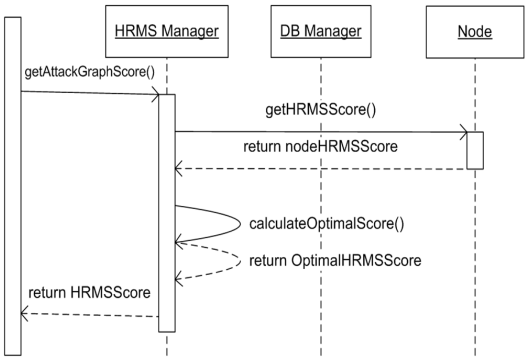
〔그림 8〕 Node Class와 HRMSManager Class Metric



〔그림 9〕 HRMS 시작 및 노드 정보 추가



〔그림 10〕 Attack Graph 정보 추출



〔그림 11〕 HRMS Score 도출

Node Class와 HRMSManager Class가 있다. Node Class는 각 노드들의 정보를 보관하며, HRMSManager Class는 Node Class와 view인 dlg class를 관리하고 제어하는 역할을 한다.

3.6 각 점수별 가중치 및 HRMS 계산

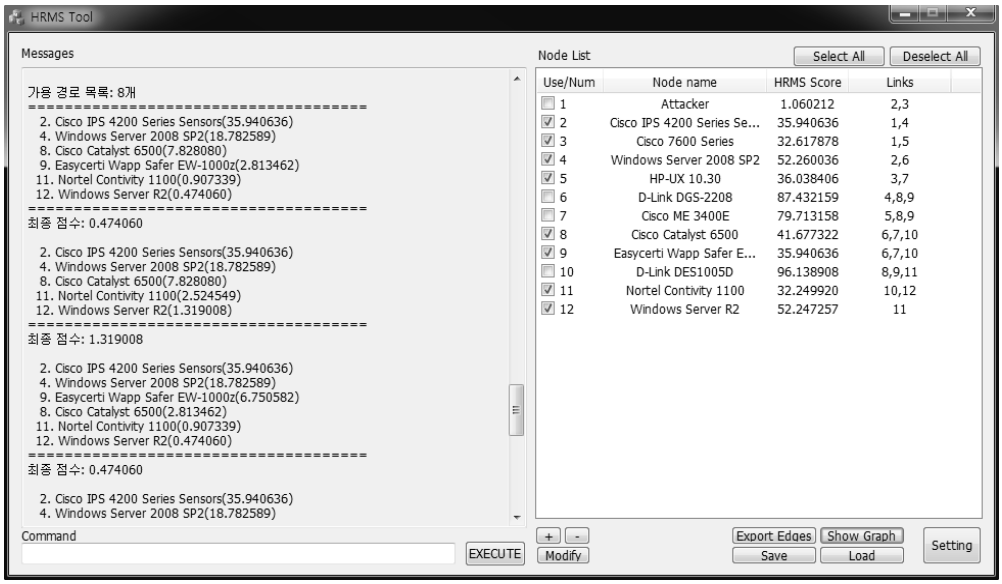
HRMS는 공격 대상의 사용자 OS, 사용자 인증, 정보자산 별 공개된 취약점에 대해서 점수를 부여하고 점수가 자동으로 계산된다. 점수가 높을수록 공격자가 시스템 권한을 얻을 확률이 높다. 각각의 메트릭이 최대 100점이 되게 가중치를 설정하여 HRMS의 최고 점수가 100점이 되게 설정 하였다.

$$HRMS = \frac{TO \bullet TO_w + SV \bullet SV_w + AS \bullet AS_w}{3} \quad (1)$$

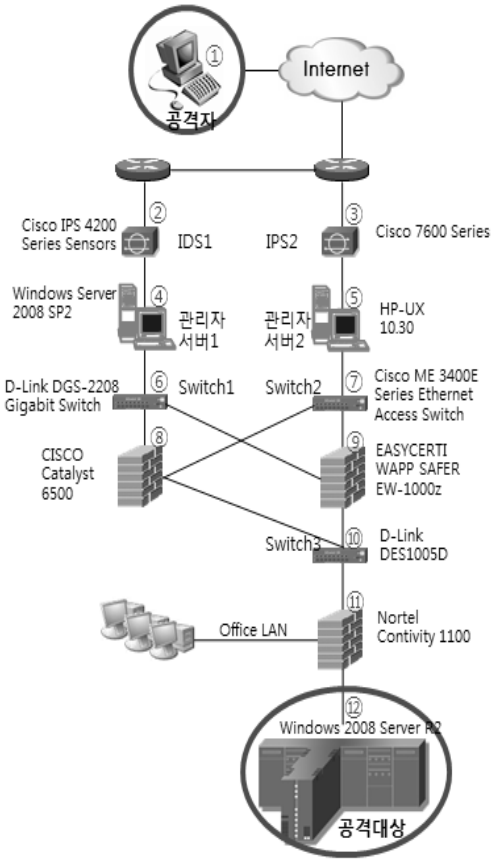
TO는 사용자의 특정 운영체제의 모든 취약점에 대한 CVSS 점수의 평균을 구한 뒤 10배해서 계산하고, TO_w는 모든 운영체제의 취약점의 수에 대한 타겟 시스템의 취약점에 대한 비율로 가중치를 부여한다. SV는 정보자산 별 공개된 취약점에 대해 CVE와 OSVDB를 이용하여 가장 큰 CVSS 값을 구하여 계산해주고 〔그림 6〕에서 보여준 식과 같이 계산한다. SV_w는 정보자산 별 공개된 취약점의 점수가 100점이 되게 설정하기 위해서 10의 가중치를 적용한다. AS는 사용자 인증 절차에 따른 점수이고, AS_w는 사용자 인증 방법에 따른 점수가 100점이 되게 설정하기 위해서 10의 가중치를 적용한다.

IV. 실험 및 결과

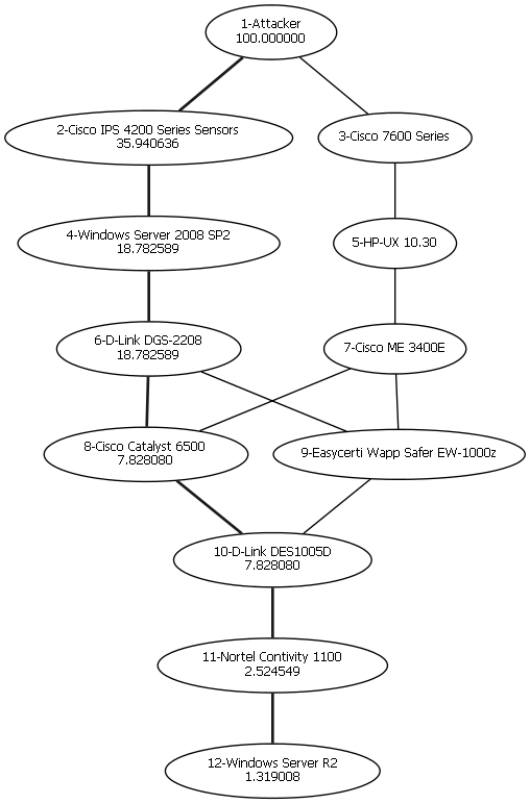
실제로 HRMS를 이용하여 attack path에 대해 enumeration을 해 보았다. 실험에서는 〔그림 13〕과 같은 환경에서 이루어질 수 있는 공격에 대한 HRMS를 계산하여 attack graph를 생성하는 것을 보여준다. 공격자의 위치는 외부 네트워크에 있고, 내부에 타겟이 되는 서버에 접근하기 위해서는 다양한 침투 경로가 존재하게 될 것이다. 공격자는 네트워크



〔그림 12〕 HRMS Display



〔그림 13〕 실험 네트워크 구조



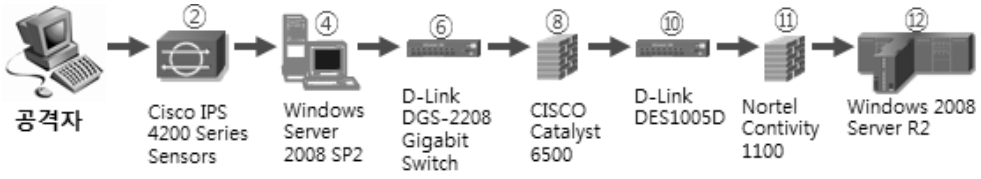
〔그림 14〕 실험 HRMS를 이용한 attack

를 전체적으로 스캐닝한 후에 〔그림 12〕와 같이 각각의 시스템들의 HRMS를 계산하여 공격가용 경로의 수를 확인하고 최적의 attack graph를 생성하게 될 것이다. 기존에 제안된 attack graph와는 다르게 HRMS는 2계층장비인 스위치와 같이 쉽게 우회 할 수 있는 장비를 확인하여 최적의 attack path를 찾기 위한 최종 점수에 영향을 주지 않게 설정할 수 있다〔15〕.

〔그림 13〕과 같은 네트워크에서 공격자는 먼저 네트워크 스캐닝을 이용하여 시스템의 정보를 얻을 수 있다. 다음으로 공격자는 각 시스템에 대한 HRMS 점수를 계산하게 될 것이다. 이 때 모든 공격 가능한 경로에 대한 점수를 계산하는 것이 아니라, 각 시스템에 대하여 HRMS 점수를 계산하여 상대적으로 점수가 낮은 경로는 제거하는 방법으로 시간 복잡도를 크게 줄일 수 있다. 이 방법은 Ingols가 제안한 Predictive attack graph를 이용한다〔1〕. Full attack graph는 공격 가능한 모든 경로를 표현하는

〔표 5〕 실험 네트워크 구조의 HRMS 점수

장비	HRMS 점수	장비	HRMS 점수
Cisco IPS 4200 Series Sensors	35.94	Cisco 7600 Series	32.62
Windows Server 2008 SP2	52.26	HP-UX 10.30	36.04
Cisco Catalyst 6500	41.68	Easycerti Wapp Safer EW-1000z	35.94
Nortel Contivity 1100	32.25	Windows Server 2008 R2	52.25



〔그림 15〕 실험 네트워크에서 최적의 attack path

것이고, Predictive attack graph는 그래프를 바로 생성하지 않고 동일한 노드를 제거한 후에 생성하는 그래프이다. Predictive attack graph를 사용한 이유로는 Full attack graph에 비해 attack graph를 생성하는 시간이 짧기 때문이다. HRMS에서는 기존에 제안된 attack graph와는 다르게 전체 네트워크 구조에 Predictive attack graph 알고리즘을 적용하여 graph로 나타낼 수 있다. HRMS를 이용한 각 시스템의 계산 결과는 〔표 5〕와 같다.

〔표 5〕와 같이 모든 계산 결과로 인한 최적의 attack graph는 〔그림 14〕와 같이 만들 수 있다. HRMS는 네트워크 내의 모든 시스템에 대하여 HRMS 점수를 계산한다. 〔표 5〕에 계산된 HRMS 점수를 이용하여 attack graph를 생성하게 되는데 먼저 방화벽이 있는 지점 전까지의 구간에서 ③, ⑤, ⑦시스템이 있는 구간 보다는 ②, ④, ⑥시스템이 있는 구간이 상대적으로 더 취약한 것을 볼 수 있다. ⑥시스템에서 ⑨시스템보다는 ⑧시스템이 취약하다는 것을 확인하며, ⑧시스템을 통해 또 다른 시스템에 접근가능 하다는 것을 HRMS는 기억하여 다음에 다시 ⑧에 대한 시스템이 나오게 되면 이후의 경로는 생각하지 않은 채로 해당 경로에 대해서는 attack graph를 생성하지 않는다. 이렇게 attack graph를 생성하게 되면 〔그림 15〕의 화살표 방향과 같이 가장 공격에 최적의 경로를 확인할 수 있다.

V. 결론 및 향후 계획

지금까지 시스템 및 네트워크 취약점 경로 분석을 위한 attack path 생성을 위해 다양한 연구들이 있었지만, 정량적인 스코어링을 위해 모든 정보자산들에 수동으로 입력 작업을 해야 하거나, 해당 정보자산에 대해 취약점 정보가 충분하지 않은 경우 연산이 어려운 모델링 기법들이 많았다. 이러한 문제점을 개선하기 위해 본 논문에서는 자동화된 침투경로 예측모델(HRMS)을 제안하였으며, 그간 알려진 취약점 정보들을 최대한 활용하여, 확보하고 있는 정보가 적다하더라도 최적화된 attack path를 생성할 수 있도록 하였다. HRMS를 이용하여 CVE, CVSS, OSVDB를 이용하여 취약점을 분석해주고, 침투 테스트 시 attack path enumeration을 통해 높은 공격성공률 획득 및 해킹시나리오를 제공함을 실험을 통해 확인할 수 있었다. HRMS에서 제공해주는 attack graph와 취약점 정보를 통해 보안 관리자가 시각적으로 쉽게 조직 내 정보자산들에 대하여 취약점 분석 및 대응방안을 도출할 수 있을 것으로 기대한다.

기존 연구에서도 확인할 수 있듯이 자동화 기법에 대한 연구가 많이 진행되고 있다. 하지만 아직까지 자동화된 모델링 기법에 대한 제안만 있을 뿐 구체적인 모델은 제시되어 있지 않다. 향후 연구 계획으로는 제안된 HRMS를 더 효율적으로 개선하기 위해 연구할 것이고, 다른 시스템이나 공격도구들과 연동이 될 수 있게 발전시킬 것이다.

참고문헌

〔1〕 K. Ingols, C. Scoot, K. Oiwowarski, K. Kratkiewicz, M. Artz, and R. Lippmann, "Validating and restoring defense in depth using attack graphs," Military Communications Conference (MILCOM 2006), pp. 1-10, Oct. 2006.

〔2〕 M. Jun-chun, W. Yong-jun, S. Ji-yin, and C. Shan, "A minimum cost of network hardening model based on attack graphs," Procedia Engineering, vol. 15, pp. 3277-3233, Dec. 2011.

〔3〕 B. Martin, C. Sullo, and J. Kouns, "OSVDB: open source vulnerability database," http://www.osvdb.org/

〔4〕 P. Mell, K. Scarfone, and S. Romanosky, "CVSS: a common vulnerability scoring system," http://www.first.org/cvss/cvss-guide.html/

〔5〕 S.H. Houmb and V.N.L. Franqueira, "Estimating toe risk level using CVSS," International Conference on Reliability and Security, pp. 718-725, Mar. 2009.
〔6〕 김동진, 조성제, "국가 DB 기반의 국내외 보안취약점 관리체계 분석," Internet and Information Security, 1(2), pp. 130-147, 2010.
〔7〕 S. Quinn, D. Waltermire, C. Johnson, K. Scarfone, and J. Banghart, "The technical specification for the security content automation protocol(SCAP)," National Institute of Standards and Technology(NIST), sp. 800-126, 2010.
〔8〕 R. Wang, L. Gao, Q. Sun, and D. Sun, "An improved CVSS-based vulnerability scoring mechanism," Third International Conference on Multimedia Information Networking and Security(MINES), pp. 352-355, Nov. 2011.
〔9〕 C. Fruhwirth and T. Mannisto, "Improving CVSS-based vulnerability prioritization and response with context information," Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 535-544, 2009.
〔10〕 S. Jajodia, S. Noel, and B. O`Berry, "Topological analysis of network attack vulnerability," Managing Cyber Threats, vol. 5, pp. 247-266, 2005.
〔11〕 S. Noel and S. Jajodia, "Managing attack graph complexity through visual hierarchical aggregation," Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, pp. 109-118, 2004.
〔12〕 L. wang, T. Islam, A. Singhal, and S. jajodia, "An attack graph-based probabilistic security metric," Lecture Notes in Computer Science, vol. 5094, pp. 283-296, 2008.
〔13〕 B. Wu and A.J.A. Wang, "EVMAT : an OVAL and NVD based enterprise vulnerability modeling and assessment tool," Proceedings of the 49th Annual Southeast Regional Conference, pp. 115-120, 2011.
〔14〕 김윤호, 이승, 강희조, "사용자 인증 방법의 분류방법에 대한 연구," 보안공학연구논문지(Journal of Security Engineering), 4(1), Feb. 2007.
〔15〕 L. Wiliams, R. Lippmann, and K. Ingols, "An interactive attack graph cascade and reachability display," VizSEC 2007 Mathematics and Visualization, pp. 221-236, 2008.

〈著者紹介〉

김 지 홍 (Ji Hong Kim) 학생회원
2011년 8월: 숭실대학교 수학과 학사 졸업
2011년 9월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 네트워크 보안, 시스템 보안, 개인정보보호

김 휘 강 (Huy Kang Kim) 종신회원
1998년 2월: KAIST 산업경영학과 학사 졸업
2000년 2월: KAIST 산업공학과 석사 졸업
2009년 2월: KAIST 산업및시스템공학과 박사 졸업
2004년 5월~2010년 2월: 엔씨소프트 정보보안실장, Technical Director
2010년 3월~현재: 고려대학교 정보보호대학원 조교수
〈관심분야〉 온라인게임 보안, 네트워크 보안, 네트워크 포렌직, 침입탐지시스템, 봇넷탐지