Journal of The Korea Society of Computer and Information
Vol. 20, No. 3, March 2015
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.3.075

분석단계에서 취약점 관리의 보안 요건 정의에 관한 연구

신 성 윤*, 이 현 창**

A Study on the Definition of Security Requirements of Vulnerability Management in Analysis Step

Seong-Yoon Shin*, Hyun-Chang Lee**

요 약

취약점 관리란 보안 정책을 준수하여 사업의 연속성과 가용성을 보장하는 것이다. 본 논문에서는 시스템의 어플리케이션 및 IT 인프라에 대한 취약점 관리는 식별되어야 한다는 것이다. 그리고 개발 단계에서 실행 가능한 취약점 관리 방안이 도출되어야 한다는 것이다. 취약점의 식별 및 분류에서 식별 및 인증, 암호화, 접근제어의 영역에서 정의되지 않은 취약점들이 많다. 이들은 기술적, 관리적, 운영적 관점에서 해당 영역별로 누락 없이 정의하도록 한다. 식별된 취약점의 대응여부를 판단하고, 해당 취약점을 제거하기 위한 대응방안을 선택하도록 한다.

▸ Keywords : 취약점 관리, 식별 및 인증, 암호화, 접근제어

Abstract

Vulnerability management is in compliance with security policies, and then, this is to ensure the continuity and availability of the business . In this paper, the application vulnerability management and IT infrastructure of the system is that it must be identified. And a viable vulnerability management plan should be drawn from the development phase. There are many that are not defined vulnerability in the area of identification and authentication, encryption, access control in identification and classification of vulnerabilities. They define the area without missing much in technical, managerial, and operational point of view. Determining whether the response of the identified vulnerability, and to select a countermeasure for eliminating the vulnerability.

▸ Keywords : Vulnerability Management, Identification and Authentication, Encryption,

•제1저자 : 신성윤, 교신저자 : 이현창
•투고일 : 2015. 1. 6, 심사일 : 2015. 1. 12, 게재확정일 : 2015. 2. 11.
* 군산대학교 컴퓨터정보통신공학부(School of Computer & Information Communication Engineering, Kunsan National University)
** 원광대학교 정보전자상거래학부(융복합창의연구소)(School of Information and Electronic Commerce(Institute of Convergence and Creativity), Wonkwang University)

Access Control,

I. 서 론

취약점 관리란 IT 환경을 보다 더 안정적으로 운영하고 기업체들이 개별 보안 정책을 지키는지를 계속해서 감시하고 대응하여 위협을 완화시켜 사업의 연속성과 가용성을 가능하도록 하는 것이다. 여기에서 취약점이란 보유하고 있는 정보 시스템(하드웨어(PC 포함), 네트워크 장비, 운영체제, 소프트웨어, 등)의 탑재하고 있는 소프트웨어의 오류와 결함이나 설치하는데 있어서의 오류 등을 말한다.

효율적이고 효과적인 취약점 관리는 전문적인 인적자원, 관리의 프로세스, 관리의 기술의 조화를 통한 취약점 관리를 말한다. 보안에 관한 전문 지식과 비즈니스에 능통하며, 소통능력이 뛰어난 전문지식을 겸비한 고급 인적 자원, 비즈니스의 관점이 보안에 접목되어 반영된 취약점 관리 체계인 프로세스, 그리고 프로세스가 통합된 취약점 관리 시스템인 기술의 세 가지 조화를 말한다.

웹 사이트에 관한 취약점 관리는 [1]에서는 웹 취약점 분석, [2]에서는 웹 사이트 구축 시 취약점 분석, [3]에서는 웹 어플리케이션의 취약점 분석 등으로 웹에 관하여 전반적으로 취약점 분석을 수행한 논문들이다.

시스템들의 취약점 관리를 보면, [4]에서는 클라우드 컴퓨팅 시스템 보안의 취약점 해결을 위해, 가상머신의 취약점에 대한 유형 및 영향분석 타입(impact type)을 정하고, 가상머신의 취약점에 대한 위험도 평가에 따른 우선순위를 정하였다. [5]에서는 검색 사이트인 구글을 이용하여 디렉토리 리스팅 취약점이 있는 사이트를 찾는 법, 구글 검색에 내가 운영하는 사이트가 검색되지 않는 방법과 웹 서버의 운영자가 할 수 있는 취약점 제거 방법을 제안하였다. [6]에서는 산업제어 시스템에서 통상적으로 발견되는 취약점은 우선순위, 발생빈도 및 영향의 심각성들과는 무관하게 정책 및 절차, 플랫폼 및 네트워크 등으로 분류하여 그들의 특성에 맞도록 취약점을 줄이거나 제거하는 방범을 모색하였다. 또한 [7]에서는 미국, 일본, 중국, 유럽 등의 보안취약점 관리체계를 조사 및 분석하여 국내 환경에 적합한 보안취약점 관리체계 구축방안을 제시하였고, [8]에서는 가상머신의 취약성 및 보안위협들을 조사하여, 안전한 클라우드 컴퓨팅을 보호하기 위한 가상머신 보안 취약점 탐지 도구를 제시하였다. 그 밖에도 [9-12]에서는 다양한 취약점 관리 방법을 제시하고 있다.

본 논문에서는 어플리케이션 구현 단계 중 분석 단계에서 취약점 관리에 대한 요건을 정의하도록 한다. 이에 따라 본 논문의 구성은 2장에서는 취약점 관리의 원칙, 3장에서는 취약점 식별 및 분류, 4장에서는 취약점 대응방안 선택, 그리고 5장에서는 취약점 관리의 사례를 들어 설명하도록 하며, 6장에서 결론을 맺도록 한다.

II. 취약점 관리의 원칙

최근의 변화하는 IT 환경을 보면 기술의 획기적이고 매우 빠른 변화에 따른 위협의 증가로 경계 네트워크가 사라짐에 따라 취약점을 이용한 침해 사고의 증가와 정부의 규제 강화로 인한 산업적 측면에서 제약이 따른다. 이러한 보이지 않는 위협이 산업 전선에 파급효과를 가져오고 취약점 관리 절차가 없으므로 제한된 자원의 사용과 비효율성을 가져오게 된다.

여기에서 취약점에 관하여 다시 한 번 논하고자 한다. 취약점이란 흔히 정보시스템에 불법적인 사용자의 접근을 허용할 수 있는 위협 요소들을 말하는데, 정보시스템의 정상적인 서비스를 방해하는 위협 요소들과 정보시스템에서 관리하는 중요한 데이터의 유출, 변조, 삭제에 대한 위합 요소들을 통틀어서 취약점이라고 한다.

위에서 언급한 취약점을 분석하는 것이 정보시스템 보안의 시작이라고 한다. 그럼 그 이유는 다음과 같이 3가지로 말할 수 있다.

첫째, 취약한 부분을 전혀 모르는 상황에서의 완벽한 방어태세는 없는 것처럼 우리의 몸을 지켜주기 위한 건강검진과 같은 역할을 수행하기 때문이다.

둘째, 정보시스템의 보안취약점에 대하여 정기적 또는 주기적인 점검조치를 수행한다면 방화벽(Firewall) 이나 침입차단 시스템(Intrusion Prevention Systems)이 없더라고 보다 안전한 정보시스템 유지가 가능하기 때문이다.

셋째, 효율적이고 효과적인 취약점 관리 시스템을 구축한 조직의 경우에 성공한 외부의 공격에 대하여 60%의 감소 효과가 발생하여 그만큼 취약점 관리 시스템을 구축하려 한다.

취약점 관리로 인하여 발생하기 쉬운 오류들로는 취약점 점검 보고서는 있지만 취약점 결과에 대한 어떠한 조치도 없

다는 것이다. 또한 취약점 관리는 기술적인 문제뿐 아니라 다양한 프로세스와 자원에 대한 문제이며, 산업적인 측면을 전혀 알지 못하는 경우가 많다는 것이다.

따라서 다음과 같은 원칙을 가지고 취약점 관리를 수행한다.

(원칙 1) 시스템의 어플리케이션 및 IT 인프라에 대한 모든 중요한 기술적 취약점 관리는 식별되어야 한다.

(원칙 2) 개발 단계에서 실행 가능한 취약점 관리 방안이 도출되어야 한다.

III. 취약점 식별 및 분류

식별 및 인증, 암호화, 접근제어의 영역에서 정의되지 않은 취약점들은 기술적, 관리적, 운영적 관점에서 해당 영역(어플리케이션, IT 인프라)별로 누락 없이 정의한다. 이러한 취약점을 관리하기 위하여 솔루션이 제시되었는데 솔루션을 취약점들을 관리하고 식별 및 분류하는 것이다.

취약점 관리의 솔루션은 특성에 따라서 다음의 세 가지로 분류되며, 각 분류된 특징에 따라 점검 가능한 범위 및 한계가 분명히 확실하다. 표 1은 취약점 관리 솔루션의 분류 및 특징[13]을 나타내고 있다.

표 1. 취약점 관리 솔루션의 분류 및 특징
Table 1. Classification and Characteristics of Vulnerability Management Solution

	시스템	DB	애플리케이션
점검대상	PC, 서버, 네트워크 장비, 기타 장비	공개용 DB 상용 DB	웹 기반 애플리케이션
점검내역	패치여부, 취약한 서비스, 취약한 패스워드, 기타 설정 오류,	DB 패치, 환경설정	웹 기반 애플리케이션 취약점
한계점들	DB, 애플리케이션 취약점	공격이 애플리케이션을 경유한 경우에 점검의 한계	개발 단계에서 발견하지 못할시 취약점 제거에 장시간 소요

다음 표 2는 취약점 식별의 예시를 나타내고 있다.

표 2. 취약점 식별 예시
Table 2. Vulnerability Identification Example

영역	어플리케이션	IT인프라
기술적 취약점	- SQL 인젝션 - Cross Site Scripting - 파일 다운로드 - 파일 업로드 - 클라이언트 메모리에 대한 해킹 - 웹페이지 내 악성코드 감영 -	- 시각 동기화 미설정 - 중요 보안패치 미설정 - 불필요 서비스 구동 - 취약한 네트워크 서비스 구동 - 원격 신뢰관계(rlogin) 접근 허용 - .....
운영적 취약점	- 테스트용 소스코드 방치 - 관리자 페이지 노출 - 사용하지 않는 계정 존재 - 웹 해킹 시도에 대한 감사 부재 - 어플리케이션 사용자에 대한 부적절한 권한 부여 -	- 사용하지 않는 계정 미 제거 - 중요한 테이블 권한 및 세션 권한이 일반 사용자에게 부여 - DB 데이터 파일의 부적절한 권한 부여 - .....

IV. 취약점 대응 방안

식별된 취약점의 대응여부를 판단하고, 해당 취약점을 제거하기 위한 대응방안을 기본적인 대응방안, 강화된 대응방안, 엄격한 대응방안 등의 단계적 대응 관점에서 정리하여 최종적인 대응방안을 선택한다. 표 3은 취약점 대응 방안 선택의 예시이다.

표 3. 취약점 대응방안 선택 예시
Table. 3 Vulnerability Countermeasure Selection Example

취약점 (예시)	취약점 대응 단계			단계 선택
	1단계	2단계	3단계	
SQL Injection	시큐어 코딩 실시	자동화 된 소스코드 점검	네트워크에서 공격 사전 필터링	2단계
DB 권한설정 미흡	자체 권한설정 및 검토	자동화 된 권한설정 분석	n/a	1단계
불필요한 서비스 구동	정기적 자체 점검	자동화 된 불필요 서비스 탐지 및 경고	네트워크에서 서비스 모니터링	3단계

이와 같은 대응 방안들은 다음과 같은 법적 근거에 따른 주요 SW 개발보안 요건들에서도 자세히 나타나 있다. 본 논문에서는 법에 대한 세부 항목은 생략하였고 아래와 같이 제

목만 적어서 살펴보았다.

1. 개인정보보호법 안정성 확보 조치기준 제6조 참조
2. 금융감독원: 전자금융감독규정(제17조) 및 시행 세칙 참조
3. 금융감독원: 홈페이지 보안 취약점 및 점검 체크리스트 참조
4. 행정안전부: 소프트웨어 개발 보안 가이드(2011.9월) 참조

V. 취약점 관리의 사례

취약점 관리의 사례를 R사의 개발 단계 보안 요건 중 취약점 관리의 사례를 들도록 하였다. 먼저, 우선 취약점 관리의 사례로서 입력값 검증의 사례를 들어 보도록 한다. 요건 ID와 요건 명은 설계단계의 보안에서 취약점 관리의 대상 ID와 이름을 나타내고, NUM은 취약점의 상세내용에 대한 번호이다.

표 4. 입력값 검증의 사례
Table. 4 Examples of Input Validation

요건 ID	요건명	NUM	상세요건
00-00-01	입력값 검증	1	입력되는 데이터의 필터링 (Black-listing) - 명령어 행에 대한 입력값 필터링 - DB 쿼리문에 대한 입력값 필터링 ※ 필터링 세부목록은 '어플리케이션 보안 아키텍처'의 '입력값 검증' 참조
		2	허용 가능한 입력값 정의 (White-listing) - 허용 가능한 최대 또는 최소값 점검 - 허용 가능한 문자 형태 정의 - 기 정의된 값들로 부터의 선택
		3	입력값을 재확인 한다. - 예: 비밀번호 생성 시 두 번 반복 입력 등
		4	사용자가 입력한 값에 대해 서버측에서 검증을 수행하도록 한다.

다음으로는 취약점 관리의 사례로서 안전한 코딩의 사례를 들어 보도록 한다.

표 5. 안전한 코딩의 사례
Table. 5 Examples of Secure Coding

요건 ID	요건명	NUM	상세요건
00-02	코딩		- 인증 후에도 사용자의 세션 및 권한 체크 유지
		2	안전한 암호 알고리즘 사용 - 암호 알고리즘은 어플리케이션 보안 아키텍처 1.4.2 암호화 알고리즘 적용 기준 참조
		3	비인가 파일 다운로드 금지 - 다운로드 직접 링크 금지 - 다운로드 기능 필요시 사용자 권한 및 세션을 체크한 후 허용
		4	비인가 파일 업로드 로직 구현 - 사용자의 파일 업로드 구현 시 허용되는 형태 외의 파일이 업로드 금지 * 업로딩 파일 형태(확장자) 필터링 로직 구현
		5	취약한 함수 사용 금지 - 길이 제한 없는 동적 할당, 길이 제한 없는 버퍼 사용 등

다음으로는 취약점 관리의 사례로서 보안 프로그램 환경 유지의 사례를 들어 보도록 한다.

표 6. 보안 프로그램 환경 유지의 사례
Table. 6 Examples of Sustainable Security Program

요건 ID	요건명	NUM	상세요건
00-00-03	보안 프로그램 환경 유지	1	프로그램 엔진을 최신 버전으로 보안 패치 프로그램 엔진의 보안 설정 적용 - 예: Cookie Protection, Password Protection, Request Validation 설정 등

다음으로는 취약점 관리의 사례로서 프로그램 정보 노출 방지의 사례를 들어 보도록 한다.

표 7. 프로그램 정보 노출 방지의 사례
Table. 7 Examples of Program Information Exposure Prevent

요건 ID	요건명	NUM	상세요건
00-00-04	프로그램 정보 노출 방지	1	불필요 파일 삭제 - .bak 등 임시 파일 제거 - 개발 시 임시 사용된 로직 및 코드 제거 등
		2	에러 처리 - 에러 페이지로 인한 시스템 정보 노출 금지 -에러 발생 시 지정된 에러 페이지 호출

다음으로는 취약점 관리의 사례로서 보안 취약점 관리의 사례를 들어 보도록 한다.

표 8. 보안 취약점 관리의 사례
Table. 8 Examples of Security Vulnerability Management

요건 ID	요건명	NUM	상세요건
00-00-05	보안 취약점 관리	1	프로그램 소스 코드의 보관 및 변경 등에 대한 이력 관리가 되어야 한다. - 프로그램 소스 코드에 대해 허가 받지 않은 사용자의 접근을 금해야 한다. - 프로그램 소스 코드의 변경에 대한 이력 관리가 되어야 한다.
		2	운영 이관 전 보안성 검토 절차를 적용한다.
		3	운영 이관 전 모의해킹 및 어플리케이션 취약점 진단을 통한 취약점 제거를 수행한다.
		4	감사지원 - 테스트 데이터 신청서 - 소스 코드 품질 검사 - 보안성 검토 체크 리스트 점검 기록 (어플리케이션부문) - 모의 해킹 및 보안 취약점 점검 결과

본 논문에서는 실험환경으로 R사의 시스템, DB, 어플리케이션을 대상으로 현재 상태의 보안 등급과 취약점 관리 후의 보안 등급을 최고 100%에 맞추어 실험하였다. 그리하여 분석 단계의 보안에서 보안 요건의 정의의 마지막 단계인 보안 취약점 관리에 대하여 알아보았다. 취약점 관리의 사례를 물론 R사에 국한된 것이지만 전반적으로 다음 그림 1과 같이 보안의 등급이 취약점 관리 전에 비하여 취약점 관리 후의 상태가 전반적으로 향상된 것으로 나타났다.

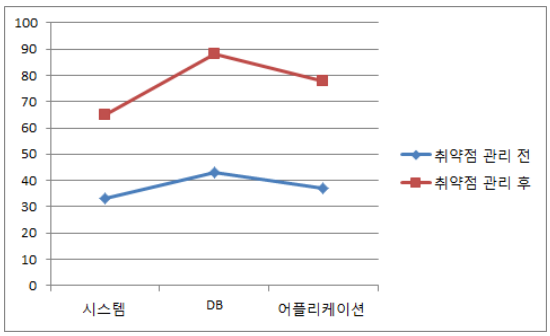

그림 1 . R사의 보안 등급 향상도
Fig. 1 Enhance the Security Level of R's

그림 1에서 전체를 100%로 보았을 때 여러 가지 요건에 의하여 취약점 관리 전에는 대략 40% 정보의 보안성을 가지지만 취약점 관리 후에는 대략 70% 정도 까지는 보안성을 올릴 수 있다. 하지만 취약점 관리만 한다고 해서 보안성이 높아지는 것은 아니다. 다른 요인들은 이전에 작성한 논문에서 확인할 수 있다.

V. 결 론

본 논문에서는 제시하는 점은 시스템의 어플리케이션 및 IT 인프라에 대한 모든 중요한 기술적 취약점 관리는 식별되어야하고 개발 단계에서 실행 가능한 취약점 관리 방안이 도출되어야 한다는 것이다. 취약점의 식별 및 분류에서 개발 단계의 식별 및 인증, 암호화, 접근제어의 영역에서 정의되거나 다루지 않은 취약점들은 기술적, 관리적, 운영적 관점에서 해당 어플리케이션 및 IT 인프라 별로 누락이나 예외 없이 정의하도록 한다. 이렇게 식별된 취약점의 대응 여부를 식별 및 판단하고, 해당 취약점을 제거하기 위한 대응방안을 기본적인 대응방안, 강화된 대응방안, 엄격한 대응방안 등의 단계적 대응 관점에서 정리하여 최종적인 대응방안을 선택하도록 한다.

REFERENCES

[1] Gwang-Hyun Kim, "Implementation and Design of Proxy System for Web vulnerability Analysis", JKIECS, Vol. 9, No. 9, pp. 1011-1018, 2014

[2] Kim ChinGo, "(A)study on the verification of improved security vulnerability when building website", Master Thesis, Graduate School of Namseoul University, 2014

[3] Jae-Chan Moon, Seong-Je Cho, "Vulnerability Analysis and Threat Mitigation for Secure Web Application Development," JKSCI, Vol.17, No. 2, pp. 127-137, 2012

[4] Mi-Young Park, Hyen-Woo Seung, Yang-mi Lim, "The Vulnerability Analysis for Virtualization Environment Risk Model Management Systematization," JKSII, Vol. 14, No. 3, pp. 23-33, 2013

[5] Sunghyuck Hong, "Vulnerability of Directory

List and Countermeasures," Journal of Digital Convergence, Vol. 12, No. 10, pp. 259-264, 2014
〔6〕 Do-Yeon Kim, "Vulnerability Analysis for Industrial Control System Cyber Security," JKIECS, Vol. 9, No. 1, pp. 137-142, 2013
〔7〕 Dong-Jin Kim, Sung-Je Cho, "An Analysis of Domestic and Foreign Security Vulnerability Management Systems based on a National Vulnerability Database," Internet and Inormation Security, Vol. 1, No. 2, pp. 130-147, 2011
〔8〕 Young-Gi Min, Kab-Seung Ko, "A Designed of Virtual Machine Security Vulnerability Detection Tool in a Cloud Computing Environment," Journal of Security Engineering, Vol. 9, No. 6, pp. 519-530, 2012
〔9〕 Jin-Seok Yang, Tai-Myoung Chung, "An Efficient Agent Framework for Host-based Vulnerability Assessment System in Virtualization Environment," KIPS Tr. Comp. and Comm. Sys., Vol. 3, No. 1, pp. 23-30, 2014
〔10〕 Woo-Sung Chun, Dea-Woo Park, "A Study of Security Measures and Vulnerability Analysis on the Application using WiBro Service" JKIICE, Vol. 16, No. 6, pp. 1217-1222, 2012
〔11〕 Jang Seung-Ju, "Implementation of User Account Vulnerability Checking Function System using MS-SQL Database," JKIICE, Vol. 18, No. 10, pp. 2482-2488, 2014
〔12〕 Ji Hong Kim, Huy Kang Kim, "Automated Attack Path Enumeration Method based on System Vulnerabilities Analysis," JKIISC, Vol. 22, No. 5, pp.1079-1090, 2012
〔13〕 http://www.ahnlab.com/kr/site/securityinfo/secunews /secuNewsView.do?seq=18181

저 자 소 개

신 성 윤
2003년 2월 : 군산대학교 컴퓨터과학과 이학박사
2006년~현재 : 군산대학교 컴퓨터정보통신공학부 교수
관심분야 : 영상처리, 컴퓨터비전, 가상현실, 멀티미디어
Email : s3397220@kunsan.ac.kr

이 현 창
2001년 : 홍익대학교 컴퓨터과학과 (공학박사)
2008년~현재 : 원광대학교 전자상거래학부 교수
관심분야: Semantic Web, Image Processing, Ubiquitous Computing
Email : hclglory@wku.ac.kr