시큐어코딩 프로그램 웹쉘 연동 시스템 설계

김민진*, 송석화*, 김만식*, 강정호*, 전문석*
*숭실대학교 일반대학원 컴퓨터학과
e-mail : minjini57@ssu.ac.kr
shsong@ssu.ac.kr
mansik@ssu.ac.kr
kjh7848@naver.com
mjun@ssu.ac.kr

A Design of Secure Coding Program and Web Shell Linkage System

Minjin Kim*, Seokhwa Song*, Mansik Kim*, Jungho Kang*, Moon-Soeg Jun*
*Dept. of Computer Science & Engineering, Soongsil University

요 약

시큐어 코딩은 2014년부터 행정자치부에서 법제화되어 의무사용이 이뤄지고 있다. 기존 소프트웨어 시장의 성장과 함께 여러 해킹방법도 고도화됨에 따라 근본적인 설계 및 코딩단계에서의 취약점 보완 필요성이 제시 되었다. 특히 웹쉘 공격은 해킹당하는 웹 페이지의 대부분이 해당 공격으로 피해를 받고 있으며, 위장하여 침투하기 때문에 백신으로 인한 검출도 어렵다. 따라서 본 논문에서는 시큐어코딩 프로그램을 웹쉘과 연동하여 취약점 분석하는 시스템을 제안하고 동작 과정에서 웹쉘 분석 후 생성되는 파일리스트를 확인해 보았다. 이것은 각 파일을 동기화하고 이후 운영과정에서도 변경되는 소스코드들을 반영하기 때문에 웹쉘로부터 웹 페이지를 효과적으로 방어할 수 있을 것으로 기대된다.

1. 서론

시큐어코딩은 소프트웨어 개발과정 중 코딩단계(소스코드 구현단계)에서 보안을 적용하는 코딩 기법을 의미한다 [1]. 현재, 다양한 해킹 및 침해와 시간이 지날수록 지능형으로 발전하는 해킹공격에 대응하기 위해 소프트웨어의 취약점을 근본적으로 보완할 수 있는 시큐어코딩이 필수적으로 요구되고 있다. 이러한 시큐어코딩 기술은 구현 단계에서 원천적인 보안 취약점을 차단할 방법으로 근본적인 취약점 제거를 통해 소프트웨어 구현 이후의 취약점 수정 비용을 줄일 수 있어 매우 경제적이다. 실제, 설계 및 코딩 단계에서 취약점 수정 시에는 개발완료 단계 이후보다 최소 10배에서 최대 30배까지 적은 비용으로 수정할 수 있다. 더 나아가 개발단계에서 시큐어코딩이 적용되었다 하더라도 운영단계에서 변경되는 소스코드로 취약점이 발생할 수 있으므로 시큐어코딩은 개발 단계뿐 만 아니라 소스코드가 변경될 수 있는 운영 단계까지도 포함해 적용해야 한다 [2].

이렇듯, 시큐어코딩에 대한 중요성이 증가함에 따라 다양한 기업에서는 시큐어코딩 시스템을 '서버-클라이언트'의 형태로 서비스를 제공하고 있다. 이때, 시큐어코딩 웹 서버의 경우 외부와 커뮤니케이션이 빈번한 환경적인 특성으로 인해 다양한 취약점에 노출될 수 있으며, 이로 인한 피해 예측이 어려우므로 실시간 모니터링을 통한 시큐어 코딩 적용이 필요하다. 또한, 웹쉘을 이용한 공격은 기존 백신으로 검출되지 않는 경우가 많으며 탐지 우회 기법이 고도화되어 웹 방어를 위한 웹쉘 방어 시스템 구축이 필요하다. 실제로 현재 해킹 공격의 89-90%는 웹을 겨냥하고 있으며 KISA의 사고신고 기준으로 웹 해킹 피해 서버 중 91%에서 악용된 웹쉘의 흔적을 발견했다고 발표했다. 한 예로 2014년 K사 서버는 웹쉘 업로드를 통해 침투당하여 홈페이지 변조 및 악성코드 유포지로 악용되어 접속 PC 6,541대를 감염시킨 경우가 있었다 [3]. 기존 Firewall, IDS/IPS, Web Firewall, Anti-Virus 등의 방법으로는 웹쉘 공격을 막기 어려우며 실제 virustotal에서 웹쉘 백신탐지에서 미탐지율이 약 73% 달하였고, 일반적인 서버 관리자들이 해킹 여부를 확인하기 어려우므로 더 큰 피해를 일으킨다 [4].

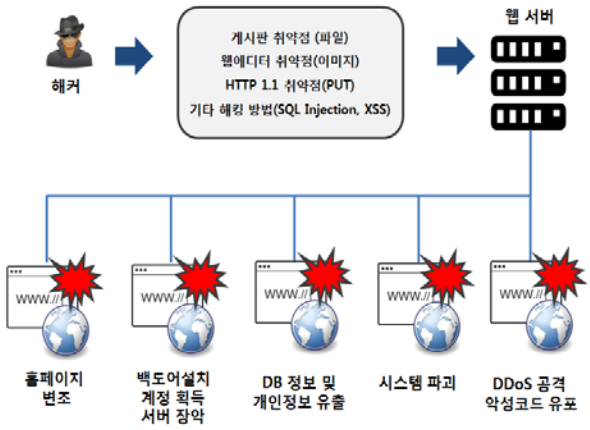
따라서 본 논문에서는 '서버-클라이언트' 형태의 시큐어코딩 시스템 환경에서 네트워크상의 취약점 및 웹쉘을 이용한 공격에 능동적으로 대응하기 위해 악성코드 탐지 솔루션(웹쉘 탐지 모니터 시스템)과 상호 기능 연동을 통해 소스코드 보안약점 진단 및 웹쉘 파일을 탐지하는 시큐어코딩 웹쉘 연동 시스템 설계 방안을 제안한다.

※ 이 논문은 2016년도 중소기업청의 산학연구마을 지원사업의 지원을 받아 수행된 연구임.

2. 관련연구

2.1 웹쉘(Web Shell)

웹쉘이란 웹 페이지의 웹(Web)과 기능과 서비스를 구현하는 인터페이스인 쉘(Shell)의 합성어 이다. 웹쉘공격은 악성코드가 담긴 웹 스크립트 파일(asp, jsp, php, cgi)을 정상적인 파일 위장해 업로드 한 이후에 원격에서 웹 서버와 접속자들을 공격한다 [5].



(그림 1) 웹쉘 공격

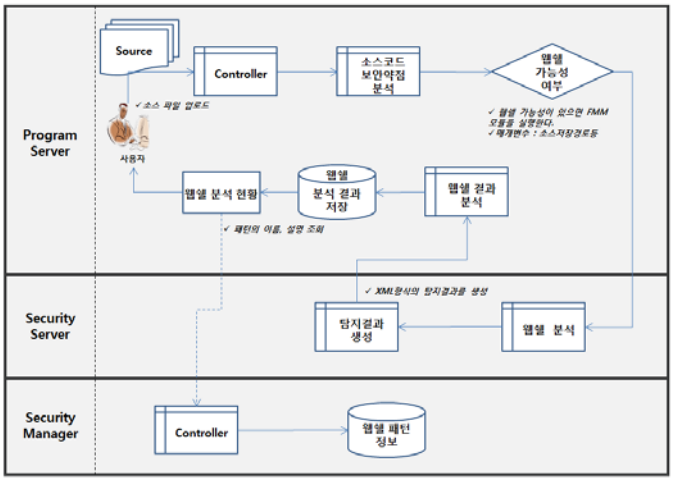
(그림 1)과 같이 해커들은 게시판, 웹 에디터, HTTP 1.1, SQL Injection, XSS 등과 같은 웹 취약점을 통해 웹쉘 파일을 웹 서버에 업로드 한다. 이후 해당 웹 서버는 해커의 원격 조종으로 홈페이지 변조, 백도어 설치와 관리자 계정 획득을 통한 서버 장악, DB정보 및 개인정보 유출, 시스템 파괴, DDoS 공격 및 악성코드 유포로 이어질 수 있다. 해커에 의해 점령된 웹사이트는 해킹 경유지로 활용되어 접속된 사용자들은 동시에 악성코드에 감염된다. 또한, 잠복하여 웹 서버 스캐닝을 통해 내부 정보 유출과 함께 웹 서버에 등록되어 있는 개인정보 유출을 시도한다.

하지만 이러한 웹쉘은 백신을 회피하기 위해 정상적인 파일스크립트에 짧은 코드를 삽입, Script Encoder를 활용, Signature로 사용되는 문자열 분산 삽입, 정상적인 스크립트 메소드 사용, HTTP 서비스의 80/tcp 포트를 이용하는 등의 지능적인 수법으로 이용되고 있다. 또한 난독화된 웹쉘이 90%이상을 차지하므로 더욱 백신 프로그램을 이용한 탐지를 어렵게 하고 있다 [6].

이렇듯, DB정보 및 개인정보 유출, 악성코드 유포지 등과 같은 문제를 일으킬 수 있는 웹쉘 공격에 웹 서버가 노출된다면, 소스코드 유출을 통해 기업의 핵심 원천 기술이 노출되거나, 소스코드를 변조하여 소프트웨어상에 다양한 버그를 일으킬 수 있다 [7].

3. 시큐어코딩 프로그램 웹쉘 연동 시스템 설계

제안하는 시큐어코딩 프로그램 웹쉘 연동 시스템은 (그림 2)와 같다. 먼저, 시큐어코딩을 실질적으로 수행하는 SecureCoding Program Server와 웹쉘 보안 기능을 제공하는 Security Server 웹쉘 서버의 분석 결과 및 현황을 관리하는 Security Manage로 구분한다. Program Server에서는 사용자가 소스 파일을 업로드 했을 경우 우선 시큐어코딩 프로그램 실행을 통해 소스코드의 보안 약점을 분석



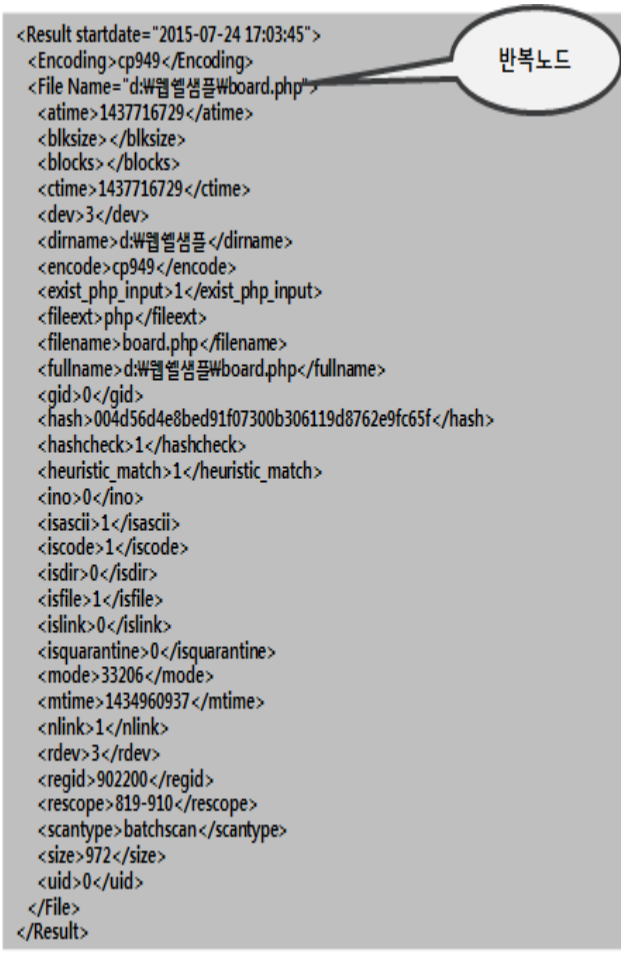
(그림 2) 웹쉘 동기화 운영 단계

한다. 분석이 끝나게 되면 웹쉘의 가능성 여부를 판단(파일 업로드 취약점, 파일 다운로드 취약점, 운영체제 명령어 삽입 취약점 중 하나 이상 발견된 파일을 웹쉘 가능성이 존재한다고 판단)하게 되며, 이때 웹쉘 가능성이 있다고 판단될 경우 명령 줄 인터페이스를 통해 매개변수를 소스저장경로로 가지는 FMM 모듈을 실행한다. 이후 Security Server를 통해 정밀 분석을 실행하며, 웹쉘 여부 탐지 결과를 XML형식으로 생성하여 결과 웹쉘 분석 결과(프로젝트 명, 웹 쉘 파일명, 탐지된 웹 쉘 패턴 명칭, 설명이 포함)를 도출한다. 분석 결과의 내용은 실시간으로 DB에 저장되며, 사용자와 프로그램 관리자는 웹쉘 분석 현황을 실시간으로 확인 가능하다. 또한, 발견된 웹쉘 패턴정보를 Security Manager에 의해 관리하여, 이후 새롭게 발생할 수 있는 새로운 형태의 취약점에 능동 대응이 가능하도록 제어 한다.

(그림 3)은 실제 SecureCoding Program Server와 웹쉘 보안을 위한 Security Server 연동 후, 웹쉘 분석을 실행했을 때의 예시로서 '시작시간|현재시간(기록시간)|검사 대상 수(2depth디렉터리기준)|완료 대상 수(2depth디렉터리기준)|검사한 전체파일 수|검사대상 파일 수(예외 등 제외된 개수)|탐지된 파일 수|패턴형식 탐지 파일 수|URL형식 탐지 파일 수|휴리스틱 형식 탐지파일 수|백업파일 탐지 수| 현재 검사 하는 파일명' 등을 출력하도록 구현한 결과의 예시이다.

4. 결론

정보통신기술의 발달로 소프트웨어의 활용 용도가 증대됨에 따라, 소프트웨어 취약점을 이용한 공격이 증가하였다. 이에 따라, 행정자치부에서는 2011년부터 시큐어코딩 의무화를 단계적으로 진행하였으며, 2015년 정보화 사업 전 분야에 걸쳐 시큐어코딩을 의무적으로 적용했다. 이처



(그림 3) 웹쉘 실행 결과 리스트

럼 시큐어코딩의 중요성이 증대되고 있는 상황에서 네트워크상의 취약점, 서버에 대한 웹쉘 공격 등에 대한 대응방안 조치는 미비한 상황이다. 웹쉘 공격은 서버의 DB에 접근하여 중요한 정보를 유출하고 웹사이트를 위/변조, 악성코드 유포지로 악용하기 때문에 조기 발견이 중요하다. 또한, 공격자가 피해서버를 장악 후 지속적인 관리가 가능하기 때문에, 시간이 지날수록 그 피해가 기하급수적으로 증가하게 된다. 시큐어코딩 환경에서는 이러한 취약점 공격을 통해 소스코드의 핵심 원천 기술을 노출하거나 소스코드 악의적 변조를 통해 프로그램 상 버그를 일으킬 수 있는 등 다양한 문제를 발생시킬 수 있다.

제안하는 시큐어코딩 프로그램은 악성코드 탐지 솔루션(웹쉘 탐지 모니터 시스템)과 상호 기능 연동을 통해 다양한 취약점에 능동적 대응과 실시간 취약점 탐지 및 선제 대응이 가능하다. 따라서 제안하는 설계 기법을 적용한다면, 안전한 시큐어코딩 시스템 서비스 제공이 가능할 것으로 기대된다.

참고문헌

[1] JONES, Russell L.; RASTOGI, Abhinav. Secure coding: building security into the software development life cycle. Information Systems Security, 2004, 13.5: 29-39.

[2] GRAFF, Mark; VAN WYK, Kenneth R. Secure coding: principles and practices. " O'Reilly Media, Inc.", 2003.

[3] HU, Jiankang, et al. Research of Webshell Detection Based on Decision Tree [J]. Journal of Network New Media, 2012, 6: 005.

[4] TU, Truong Dinh, et al. Webshell detection techniques in web applications. In: Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on. IEEE, 2014. p. 1-7.

[5] YANG, Chung-Huang; SHEN, Chung-Hsiang. Implement web attack detection engine with snort by using modsecurity core rules. Graduate Institute of Information and Computer Education, National Kaohsiung Normal University Kaohsiung, TAIWAN, 2009.

[6] KIM, Bumryong, et al. A Design of Inter-Working System between Secure Coding Tools and Web Shell Detection Tools for Secure Web Server Environments. Journal of the Korea Society of Digital Industry and Information Management, 2015, 11.4: 81-87.

[7] DENG, Lawrence Y., et al. Lexical Analysis for the Webshell Attacks. In: 2016 International Symposium on Computer, Consumer and Control (IS3C). IEEE, 2016. p. 579-582.