International Journal of Security and Its Applications
Vol. 9, No. 5 (2015), pp. 163-174
http://dx.doi.org/10.14257/ijsia.2015.9.5.17

Security Assessment by Google Hacking Automation Tools for the Web Sites of Korea and the USA Universities

Mi Young Bae and Hankyu Lim[1]

Department of Multimedia Engineering, Andong National University
mybae73@naver.com, hklim@anu.ac.kr

Abstract

As software exchanges data in the internet environment, it is always susceptible to the malicious attacks of hackers. Google Search enables individuals to randomly search servers with their preferred vulnerabilities using several search words. Using a sample of university homepages in Korea and the US, this paper investigates the security weakness of homepages by using SiteDigger that automatically searches the Googling, which is the most convenient way of collecting data, and examines the security weakness of homepages in Korea and the US. Based on the analyzed weakness, the researcher attempts to conduct future study that develops security diagnosis tool for webpage

Keywords: *Secure coding, Hacking tools, University Website, Personal information*

1. Introduction

With today's development of Internet search engines, one's preferred data can be quickly found by searching through massive amounts of data for the purpose of information search. In addition, search engines can be used maliciously for hacking.

As software exchanges data in the internet environment, it is always susceptible to the malicious attacks of hackers.

Target attacking activities that occurred in one year of 2013 increased by 91% compared to the previous year and the number of spill accidents increased by 62%. Through the spill accidents, more than 552 million IDs were exposed [1].

Recent cyber-attacks have mostly targeted websites. For example, Sony, PayPal, and the Hong Kong Stock Exchange experienced attacks that leaked personal information. In addition, the homepages of major government agencies such as the US CIA and State Department were attacked. Therefore, the protection of information on websites (homepages) is becoming even more important [2].

Through an international study conducted in 2014 in seven countries by a US based company, it was revealed that the average cyber-crime cost of US companies increased by 9% in one year from 11.6 million dollars in 2013 to 12.7 million dollars in 2014. It was also shown that the average time taken to solve cyber-crimes also increased from 32 days in 2013 to 45 days in 2014 [3]. This is part of facts revealed through the 5th annual cyber-crime cost study conducted by Ponemon Institute.

As the cases of searching and attacking vulnerable servers via Google Search are on the increase and individuals can randomly search servers with their preferred vulnerabilities using only several keywords, all searched servers can become the easy targets of attacks.

Hence, this paper investigates the security weakness of homepages of universities in Korea and the US by using SiteDigger that automatically searches the Googling which is the most convenient way of collecting data and examines the security weakness of homepages in Korea and the US

[1] (corresponding author)

ISSN: 1738-9984 IJSIA

2. Checking Software Security Vulnerabilities

All printed material, including text, illustrations, and charts, must be kept within the parameters of the 8 15/16-inch (53.75 picas) column length and 5 15/16-inch (36 picas) column width. Please do not write or print outside of the column parameters. Margins are 1 5/16 of an inch on the sides (8 picas), 7/8 of an inch on the top (5.5 picas), and 1 3/16 of an inch on the bottom (7 picas).

Secure coding means coding with source code such that security weakness such as insertion of SQL or cross website script (XSS) does not exist in SW when implementing SW based on programming languages such as JAVA and C/C++.

Recognizing the importance of secure coding, the Ministry of Security and Public Administration and KISA have continued to promote policies regarding security in software development. 'Information System Development and Operation Guideline' was revised and officially announced in July 2012, which made it mandatory for the electronic government software under the information system surveillance to apply development security to safely develop.

Even in case a developer develops software referring to secure coding guide, software that contains security weakness can be developed, either by mistake or intentionally. Hence, it is necessary to autonomously detect security weakness and remove it in the developmental phase.

When diagnosing security weakness, it is hard for human beings to directly look at the source codes to detect diverse security weaknesses. Hence, using an automatized tool that can detect security weakness is essential for effectively diagnosing and removing software security weakness.

Diagnosis on the software security weakness can be divided into static analysis, which verifies input data and detects diverse security weaknesses such as weak API use by analyzing source code without running the software, and dynamic analysis, which conducts analysis from a functional operational aspect by running the software.

Static and dynamic automatized analytic tools depend on diagnosis rule and false positive can possibly exist in the diagnosis result. Hence, securing the reliability of the tool is critical [4].

The static analytic tools for secure software development that can analyze the security weakness are widely used these days. In case the surveillance corporation uses automatized tool based on static analytic tool of 'Source Code Security Weakness Analytic Tool' for diagnosing security weakness when inspecting domestic national information-oriented business, using assessed and certified products (CC-certified product) became mandatory according to the 'Guideline for Information Protection System Assessment and Certification'. Two types of certified analytic tools launched in May 2014 [5].

In international cases, NIST SAMATE project provides a variety of tools that can be used in each stage of software development aiming at improvement of quality and security of software, which include security weakness analytic tool based on static analysis (commercial and public). Table 1 summarizes the analytic tools for source code security weakness [6].

Table 1. Security Weakness Analysis Tool of Foreign Software

Tool	Language(s)	Avail.
ABASH	Bash	free

ApexSec Security Console	PL/SQL(Oracle Apex)	Recx
Astrée	C	AbsInt
BOON	C	free
bugScout	Java, C#, Visual Basic, ASP, php	buguroo
C/C++test®	C, C++	Parasoft
dotTEST™	C#, VB.NET, MC++	
Jtest®	Java	
HP Code Advisor (cadvise)	C, C++	HP
Checkmarx	Java, C#/.NET, PHP, C, C++, Visual Basic 6.0, VB.NET, Flash, APEX, Ruby, JavaScript, ASP, Android, Objective C, Perl	Checkmarx
Clang Static Analyzer	C, Objective-C	free
Closure Compiler	JavaScript	free
CodeCenter	C	ICS
CodePeer	Ada	AdaCore
CodeSecure	ASP.NET, C#, PHP, Java, JSP, VB.NET, others	Armorize Technologies
CodeSonar	C and C++	GrammaTech
Coverity SAVE™	C, C++, Java, C#	Coverity
Cppcheck	C, C++	free
CQual	C	free
Csur	C	free
DoubleCheck	C, C++	Green Hills Software
FindBugs	Java, Groovy, Scala	free
FindSecurityBugs	Java, Groovy, Scala	free
Flawfinder	C/C++	free
Fluid	Java	call
Goanna Studio and Goanna Central	C, C++	Red Lizard Software
HP QAInspect	C#, Visual Basic, JavaScript, VB Script	Fortify
Insight	C, C++, Java, and C#	Klocwork
Jlint	Java	free
LAPSE	Java	free
ObjectCenter	C/C++	ICS
Parfait	C/C++	Oracle proprietary
PLSQLScanner 2008	PLSQL	Red-Database-Security
PHP-Sat	PHP	free
Pixy	PHP	free
PMD	Java	free

PolySpace	Ada, C, C++	MathWorks
PREfix and PREfast	C, C++	Microsoft proprietary
pylint	Python	free
QA-C, QA-C++, QA-J	C, C++, Java	Programming Research
Qualitychecker	VB6, Java, C#	Qualitychecker
Rational AppScan Source Edition	C, C++, Java, JSP, ASP.NET, VB.NET, C#	IBM (formerly Ounce Labs)
RATS (Rough Auditing Tool for Security)	C, C++, Perl, PHP, Python	free
Resource Standard Metrics (RSM)	C, C++, C#, and Java	M Squared Technologies
Smatch	C	free
SCA	ASP.NET, C, C++, C# and other .NET languages, COBOL, Java, JavaScript/AJAX, JSP, PHP, PL/SQL, Python, T-SQL, XML, and others	Fortify Software
SPARK tool set	SPARK (Ada subset)	Altran
Splint	C	free
TBmisra®, TBsecure®	C, C++, Java, Ada, Assembler	LDRA
UNO	C	free
PVS-Studio	C++	Program Verification Systems
xg++	C	unk
Yasca	Java, C/C++, JavaScript, ASP, ColdFusion, PHP, COBOL, .NET, etc.	free

3. Google Hacking

Google collect information through many major media. The types of collected information include those pieces of information that are directly provided when major tools of Google are used, those pieces of information that are collected by Google robots web crawlers, those pieces of information that are provided by others when they use Google's tools, and those pieces of information that are obtained from third party databases and business partners [7].

3.1. Personal Information Disclosure Vulnerability

Even simple search words such as "member list" and "member list.xls" produced approximately 450,000 search results and quite some of which were files containing students' birth days, phone numbers, and addresses. The contents could be seen through downloading and file opening without any restriction.

This security vulnerability corresponds to the exposure of important information among OWAP security vulnerability items and the file download vulnerability among the security vulnerability checking items of the National Intelligence Service [8].

3.2. SQL Injection Vulnerability

This is a vulnerability item that enables attackers to insert SQL sentences into the input form and URL input section in web applications interlocked with databases to read and manipulate information in the database.

To find administrator pages in order to inject SQLs, administrator pages were searched in Google using the keyword inurl:admin site:ac.kr. Through the searches, quite a few of approximately 26,900 websites exposed administrator log-in screens as they were

3.3. Directory Listing Vulnerability

Since there was vulnerability that all directories or directories that contain important information are listed outside due to the failure of setting index security in public servers, Googling with intitle:index.of inurl:ac.kr produced approximately 1,610,000 search results and quite a few of them listed directories as they were

3.4. Error Messages Vulnerability

Since AP installation information, ID/PW information, and SQL injection attack information are provided when error messages are searched at Google, detailed information on server invasion pathways is provided.

4. Checking Website Security Vulnerabilities

Since 2012, stepwise mandatory application of security by software development has been institutionalized for public web services of domestic public institutions as a countermeasure against security threats [9].

Web security weakness can be divided into source code-related security weakness that is caused by development without considering security in developmental stage of web application program and web server-related security weakness that is caused by poor web server security setting.

For the purpose of decreasing security weakness of homepages, the Ministry of Security and Public Administration published ‘Homepage SW (web) Development Security Guide for Information System Developer and Administrator’. Table 2 shows the web security weakness types and risks presented in this guide’

Table 2. The Type of Web Security Vulnerabilities

Vulnerability type	Attack Damage	Risk
SQL Injection	DB information outflow	high
Execution of operating system commands	System seized	high
XQuery Injection	DB information outflow	high
Xpath Injection	User authentication bypass	high
Cross-site Scripting	Session hijacking	medium
File Upload	System seized	high
File download	Exposure of web server information	high
Buffer overflow	System resource depletion	low
LDAP Injection	System seized	high
HTTP response splitting	Session hijacking	medium
URL/ Parameter modification	User rights deodorant	medium
Create Account vulnerable permit	User Account deodorant	high
Insufficient Session Management	User rights deodorant	medium
Plaintext data transfer	Important Information Disclosure	medium

Cookie corruption	User rights deodorant	medium
Use of weak encryption algorithm	Important Information Disclosure	medium
Weak passwords recovery	User Account deodorant	high
Information disclosure through comment	Important Information Disclosure	medium
Directory indexing	System File exposure	medium
SSI Injection	System seized	high
Error page impression	Sever Information Disclosure	low
Administrator page impression	Website Information Disclosure	medium
Exposure of backup and temporary files	Website Information Disclosure	medium
Web service methods set attack	System seized	high
Exposure search engine information	Important Information Disclosure	low

In this guide, chapter 2 and chapter 3 propose secure coding for each security weakness and chapter 4 introduces diagnosis method for web security weakness.

As for the diagnosis method of web security weakness, we designed such that web application source code and web firewall can be automatically revised when security weakness is detected, referring to the results created by the web approach, instead of individual input of diagnosis code and approval regarding the security weakness by the developer.

When diagnosing security weakness, it is hard for human beings to directly look at the source codes to detect diverse security weaknesses. Hence, using an automatized tool that can detect security weakness is essential for effectively diagnosing and removing software security weakness. However, there are very few automatic analytic tools for detecting web security weakness as of now.

This paper diagnosed and analyzed the security weakness of webpage of 50 Korean and 50 American universities using Googledork Tool for easier detection of web security weakness.

5. Edork Tool

Google collect information through many major media. The types of collected information include those pieces of information that are directly provided when major tools of Google are used, those pieces of information that are collected by Google robots web crawlers, those pieces of information that are provided by others when they use Google's tools, and those pieces of information that are obtained from third party databases and business partners.

Googling is using Google searches to obtain information from the Web. However, Googling has been abused and established as an easy way to extract personal information.

Googling is used not only in extracting personal information but also in attacks that find company computing system administrator account information and push malignant codes onto the accounts because by searching under certain options, even important personal information existing in the relevant sites can be identified.

Even simple search words such as "member list" and "member list.xls" produced approximately 450,000 search results and quite some of which were files containing students' birth days, phone numbers, and addresses. The contents could be seen through downloading and file opening without any restriction.

When searching for administration page for SQL injection in Google using key word 'inurl:admin site:ac.kr', a number of administrators', a number of login

webpages were completely exposed. GoogleDork is a tool that enables automatic and easier Googling of such Google hackings.

There are many types of GoogleDork tools. Among them, Sqli Hunter is an automatic tool that automatically detects the weakness in SQL injection of websites. Dork Searcher is a small utility-type tool that also automatically detects the weakness in SQL injection of websites. GooDork is a simple python script designed such that Google dorking can be directly used in command line. Pentest-tools.com is a website that provides Google search results after searching for nine types of Google hacking when a user enters desired URL.

SiteDigger searches for Google cache in order to find the security weaknesses of the website including weakness error, configuration trouble, proprietary information, etc.

The list of weakness that can be automatically extracted from SiteDigger is summarized in Table 3.

Table 3. Detecting Vulnerabilities List

FSDB (175)	
Backup Files	12
Configuration Management	35
Error Message	39
Privacy Related	30
Remote Administration	8
Reported Vulnerabilities	8
Technology Profile	43
GHDB (1467)	
Advisories and Vulnerabilities	215
Error Message	68
Files containing juicy info	230
Files containing passwords	135
Files containing usernames	15
Footholds	21
Misc.	45
Pages containing login portals	232
Pages containing network or Vulnerabilities data	59
Sensitive Directories	61
Sensitive Online Shopping info	9
Various Online Devices	201
Vulnerable Files	56
Vulnerable Servers	48
Web Server Detection	72

Key words regarding Google hacking was created according to each category and Googling was iterated for a total of 1642 times on the entered homepage address using an operator regarding Google hacking.

6. Vulnerability Assessment of Homepage Security in Sitedigger

This paper diagnosed the security of the homepage of 50 Korean and 50 American universities using SiteDigger. Table 4 is the list of the universities in alphabetic order.

Table 4. Korea and the USA University

	Korea	United States of America
1	KangWon National University	University of California, Berkeley
2	KonKuk University	University of California, Los Angeles
3	Kyonggi University	Northeastern University
4	KyungPook National University	Northwestern University
5	Gachon University	The University of North Carolina at Chapel Hill
6	Kyunghee University	University of Notre Dame
7	Korea University	New York University
8	Kwangwoon University	Dartmouth College
9	Korea National University of Education	Duke University
10	Kookmin University	Rice University
11	Dankook University	Rensselaer Polytechnic Institute
12	Duksung Women's University	University of Rochester
13	Gongguk University	Lehigh University
14	Dongduk Women's University	University of Miami
15	Myongji University	Massachusetts Institute of Technology
16	Pusan National University	University of Michigan
17	Sahmyook University	Vanderbilt University
18	Sangnyung University	University of Virginia
19	Sogang University	Boston University
20	Seoul National university of Science and Technology	Brown University
21	Seoul National University	Brandeis University
22	University of Seoul	University of Southern California
23	Seoul Women's University	Stanford University
24	Sungkyunkwan University	University of Chicago
25	Sungshin Women's University	Emory University
26	Sejong University	Yeshiva University
27	Sookmyung Women's University	Yale University
28	Soongsil University	Washington University in St Louis
29	Ajou University	Wake Forest University
30	Yonsei University	University of Wisconsin—Madison
31	Ewha Women's University	College of William and Mary
32	Incheon National University	University of Illinois—Urbana-Champaign
33	Inha University	Georgia Institute of Technology
34	Chonnam National University	George Washington University
35	Chonbuk National University	Georgetown University
36	Chung-ang University	Johns Hopkins University
37	Chungnam National University	Carnegie Mellon University
38	Chungbuk National University	University of California—Irvine
39	Korea Advanced Institute of Science	California Institute of Technology

	and Technology	
40	The Catholic University of korea	University of California—Davis
41	Pohang University of Science and Technology	University of California— Santa Barbara
42	Korea University of Technology and Education	University of California—San Diego
43	Korea Polytechnic University	Case Western Reserve University
44	Hankuk University of Foreign Studies	Cornell University
45	Information and Communications University	Columbia University
46	Handing Global University	Pennsylvania State University—University Park
47	Hansung University	University of Pennsylvania
48	Hanyang University	Princeton University
49	Korea aerospace University	University of Florida
50	Hongik University	Harvard University

After running SiteDigge, 37 Korean universities homepages and 12 American universities homepages showed no security weakness at all.

A homepage with largest number of security weakness among Korean universities had 33 security weaknesses, while the US university homepage with largest number of security weakness had that of 163. In case of the US, there were two homepages with more than a hundred security weaknesses.

The average security weakness was 1.9 in case of Korea and it was 18.9 in case of the US.

Table 5 shows the number of security weaknesses by each category.

Table 5. Security Vulnerabilities of Korea and the USA University

Security Vulnerability	Korea	United States of America
Backup Files	9	192
Configuration Management	0	106
Error Message	38	200
Privacy Related	2	82
Remote Administration	0	3
Reported Vulnerabilities	0	0
Technology Profile	8	45
Advisories and Vulnerabilities	1	19
Files containing juicy info	20	125
Files containing passwords	1	27
Files containing usernames	0	16
Footholds	0	9
Misc.	0	0
Pages containing login portals	8	25
Pages containing network or Vulnerabilities data	0	2
Sensitive Directories	8	53
Sensitive Online Shopping info	0	8

Various Online Devices	0	20
Vulnerable Files	0	0
Vulnerable Servers	0	5
Web Server Detection	0	8
Total	95	945

The security weakness was mainly observed in case of completely exposed Error Message, easy downloading of files with sensitive contents, and directory listing

7. Conclusion

This paper diagnosed the security of the homepage of 50 Korean and 50 American universities using SiteDigger which is a Google automatic searching tool that makes security diagnosis of web page easier. In overall, homepage in the US turned out to have more security weaknesses.

However, this is mere a numerical result and SiteDigger does not contain Korean key words when searching for sensitive data in Google as it was developed in the US. Hence, it is not desirable to compare the security status between countries only by referring to numbers without considering the difference of the system configuration of each country.

Moreover, even though using SiteDigger has an advantage of making the search of security weakness of homepages easier, it has shortcoming in that sensitive materials or files that are written in Korean cannot be searched since the produced key word is not in Korean.

Development of analytic tool for homepage security weakness is still in fledgling stage, compared to the analytic tool for software security weakness.

Programmers want to completely remove the weakness in their program so that it can operate as a secure program. However, it is not easy to obtain professional knowledge about the weakness items and to recognize how to correct them.

Hence, it is necessary to develop a tool that can analyze the homepage security weakness suitable for Korean cases.

In the future, we would like to develop a web security weakness tool that takes account for the characteristics of the Korean system.

ACKNOWLEDGEMENTS

This work was supported by a grant from 2014 Research Fund of Andong National University.

References

[1] "Symantec", Internet Security Threat Report, 2013 Trends, vol. 19, **(2014)** April.
[2] "Ministry of Security and Public Administration", Homepage SW (web) development security guide for Information System Developer and Operator **(2012)**.
[3] http://www8.hp.com/kr/ko/software-solutions/ponemon-cyber-security-report/.
[4] J. Ban, Editor, "Development trend for analysis tool of open source code security weakness", INTERNET & SECURITY FOCUS, **(2014)** May.
[5] "Ministry of Security and Public Administration", Software development security guide for electronic government SW development operator, May **(2012)**
[6] http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html.
[7] Greg Conti, Google knows you, Bpanbooks Publishers, **(2009)**.
[8] M. Y. Bae and H. Lim, "Security Assessment through Google Tools - Focusing on the Korea University Website", Advanced Science and Technology Letters, vol. 93, **(2015)**, pp. 9-13.
[9] "Ministry of Security and Public Administration", Secure Coding Inspection Guide for e-gov SW, **(2014)**.
[10] J. Lee and H.-S. Kang, Editors, "Study and Implementation for Google Hacking auto protection system", Inje Journal, vol. 20, no. 1, **(2005)** February**.**

Authors

Hankyu Lim, He received the B.S. degree in Electronics Engineering from the Kyungpook National University in 1981. He received the M.S. degree in Computer Engineering from the Yonsei University in 1984. He received the Ph.D. degree in Computer Engineering from the Sung Kyun Kwan University in 1997. He is a professor of Andong National University, Korea. His areas of interest include web application, multimedia and Natural Language Processing.

Mi-Young Bae, She received the B.S. degree in computer engineering from Andong National University, Korea, 1996, and M.S. degree in computer engineering from Andong National University, 2000. She is studying Ph.D. course in Information communication engineering from Andong National University, 2012. Her areas of interest include mobile programming and secure coding.