A Study on Security Diagnosis Using Automated Google Hacking Tools-Focusing on the US Government Website

Mi Young Bae, Han Kyu Lim, and Dae Jea Cho
Department of Multimedia Engineering, Andong National University, South KOREA
Email: mybae73@naver.com, {hklim, djcho}@andong.ac.kr

Abstract—Because various software use internet to transmit data in recent internet environment, there is always a possibility of malicious attacks by hackers. With Google search, with a few search words servers with desired vulnerabilities can be randomly searched. The study used SiteDigger that automatically searches Google which can most easily collect data to explore the security vulnerability status of US State Department web sites and analyzed the detected security vulnerabilities. In the future, based on the analyzed security vulnerabilities, the study plans to develop a webpage security diagnostics tool.

Index Terms—secure coding, Google hacking, secure web site

I. INTRODUCTION

With the development of internet search engines, it has become possible to search desired data from vast amounts of data for the purpose of information search. But also it has become possible to maliciously use search engines for hacking.

Within the year of 2013 targeted attack activity has increased by 91% compared to the previous year, data leakage has also increased by 62%, and through data leakages, over 552 million IDs have been exposed [1].

Cyber-crime average costs for US-based companies conducted in seven countries in 2014 increased 9% yearly and from the $11.6 million in 2013, it rose to $12.7 million. The average time consumed solving cyber-crime has also rose from 32 days in 2013 to 45 days, as reported by the cyber-crime costs research conducted by Ponemon Institute [2].

There are increased cases of finding and attacking vulnerable servers through Google search and because with a few search words, desired servers with vulnerabilities can be randomly searched, all servers searched can easily be a target of attack.

For diagnostic method of web security vulnerability, individual developers input individual diagnostic codes for individual security vulnerability or based on the results of web access in an unauthorized method find security vulnerabilities then edit web application source code or web firewall according to each element.

During diagnosing security vulnerabilities, because it is difficult for a person to find various security vulnerabilities by looking at the source code, by using an automated tool that inspect security vulnerabilities in short time, software security vulnerabilities can be effectively diagnosed and removed but currently there is lack of automatic analysis tools for inspecting web security vulnerabilities [3].

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 KOREA 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 I summarizes the analytic tools for source code security weakness [6].

The study used Googledork (Google search hack) tool SiteDigger is a method to more easily inspect web security vulnerabilities to explore the security vulnerability status of US State Department homepages and diagnosed and analyzed the security vulnerabilities of web pages.
II. GOOGLEDORK TOOL

Google collects data from various major media. Types of collected data include data directly provided when using main Google tools, data collected by Google bot web crawler, data provided by people when they use Google tools, and data acquired from third-party databases and business partners [7].

Googling refers to using Google search to acquire data from the web. However, googling is being exploited to become an easy way to steal personal information. Googling is not only used in simple personal data leakage, but finding administrator’s account information from IT systems to be used in attacks inserting malicious codes. This is because by searching using specific options, one can search even major personal information existing within a specific site.

There are various types of Googledork tools. Among them, SqlHunter is an automated tool that automatically detects SQL injection vulnerabilities of a website. Dork Searcher is a small utility type tool that automatically detects SQL injection vulnerabilities and GoogleDork is a simple Python script designed so that google dorking can detect SQL injection vulnerabilities of a website. Dork Hunter is an automated tool that automatically searches even major personal information existing within a specific site.

The list of vulnerabilities that can be automatically detected by SiteDigger is represented in Table II. FSDB is Found Stone database and SiteDigger is developed by Found Stone [8]. GHDB is Google hacking database [9].

| Table I. SECURITY WEAKNESS ANALYSIS TOOL OF SOFTWARE |
|-----------------------------------------------|
| Tool                                           | Language(s) | Avail. |
| ABASH Security Console                         | Bash        | free   |
| ApexSec                                        | PL/SQL(Oracle Apex) | Recx |
| Astree                                         | C           | AbsInt |
| BOON                                           | C++         | free   |
| bugScout                                      | Java, C#    | Visual Basic, ASP, php | bugruru |
| C/C++test®                                     | C, C++      | Parasoft |
| dotTEST™                                       | C#, VB.NET, MC++ |      |
| Jrest®                                         | Java        |        |
| HP Code Advisor (cadvice)                      | 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 |
| DoubleCheck                                    | C, C++      | Green Hills Software |
| FindBugs                                       | Java, Groovy, Scala | free |
| FindSecurityBug                                 | Java, Groovy, Scala | free |
| Flawfinder                                      | C/C++       | Free   |
| Flurl                                          | 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 |
| Jint                                           | Java        | free   |
| LAPSE                                          | Java        | free   |
| ObjectCenter                                   | C/C++       | ICS    |
| Parfait                                        | C/C++       | Oracle proprietary |
| PLSQL Scanner 2008                             | PLSQL       | Red Database-Security |
| PHP-Sat                                        | PHP         | free   |
| Pixy                                           | PHP         | free   |
| PIMD                                           | Java        | free   |
| PolySpace                                      | Ada, C, C++ | MathWorks |
| PREfix and PREfast                             | C, C++      | Microsoft proprietary |
| QA-Q C QA-C++, QA-J                            | C, C++, Java | Programming Research |
| Qualitychecker                                 | VB6, Java, C# | Qualitychecker |
| Rational AppScan Source Edition                | C, C++, Java, JavaScript, ASP.NET, VB.NET, C# | IBM (formerly Ounce Labs) |
| RATS (Rough Auditing Tool for Security)        | C, C++, Perl, PHP, Python | free |
| Resource Standard Metrics                      | C, C++, C#, and Java | M Squared Technologies |
| SCA                                            | ASP.NET, C, C++, C# and other .NET languages, COBOL, | Fortify Software |

| Table II. LIST OF VULNERABILITIES DETECTED BY SITEDIGGER |
|-----------------------------------------------------------|
| FSDB(175)                                                 |
| Backup Files                                             | 12 |
| Configuration Management                                 | 35 |
| Error Message                                             | 39 |
| Privacy Related                                           | 30 |
| Remote Administration                                     | 8 |
According to each item, there are Google hacking related search words and for the inputted homepage address, a total of 1642 Googling is conducted using Google hacking related operator.

III. SECURITY VULNERABILITY DIAGNOSIS USING SITEDIGGER

The study used SiteDigger to diagnose security vulnerabilities of 50 US State Department homepages. The list of the 50 US State Department homepages and the detected security vulnerabilities are represented in Table III.

| No. | State         | Detected number | No. | State         | Detected number |
|-----|---------------|-----------------|-----|---------------|-----------------|
| 1   | Delaware      | 8               | 26  | Michigan      | 0               |
| 2   | Pennsylvania  | 2               | 27  | Florida       | 9               |
| 3   | New Jersey    | 0               | 28  | Texas         | 5               |
| 4   | Georgia       | 0               | 29  | Iowa          | 4               |
| 5   | Connecticut   | 8               | 30  | Wisconsin     | 0               |
| 6   | Massachusetts | 16              | 31  | California    | 0               |
| 7   | Maryland      | 17              | 32  | Minnesota     | 19              |
| 8   | South Carolina| -               | 33  | Oregon        | 0               |
| 9   | New Hampshire | 11              | 34  | Kansas        | 0               |
| 10  | Virginia      | 0               | 35  | West Virginia | 0               |
| 11  | New York      | 0               | 36  | Nevada        | 8               |
| 12  | North Carolina| 0               | 37  | Nebraska      | 0               |
| 13  | Rhode Island  | 0               | 38  | Colorado      | 8               |
|     |               |                 |     |               |                 |

After running SiteDigger, there were 26 sites among State Department homepages where no security vulnerabilities were found and for South Carolina and Alabama there existed web addresses for the State Department homepages http://www.sc.gov/, http://www.alabama.gov/, but it was not possible to access that website.

Table IV represents the number of security vulnerabilities found by each security vulnerability category.

| Security vulnerability category          | Result |
|-----------------------------------------|--------|
| Backup Files                           | 21     |
| Configuration Management                | 83     |
| Error Message                          | 2      |
| Privacy Related                        | 5      |
| Remote Administration                   | 0      |
| Reported Vulnerabilities                | 0      |
| Technology Profile                      | 8      |
| Advisories and Vulnerabilities          | 10     |
| Files containing juicy info             | 8      |
| Files containing passwords              | 35     |
| Files containing usernames              | 0      |
| Footholds                               | 0      |
| Misc.                                   | 0      |
| Pages containing login portals          | 17     |
| Pages containing network or Vulnerabilities data | 0 |
| Sensitive Directories                   | 60     |
| Sensitive Online Shopping info          | 0      |
| Various Online Devices                  | 0      |
| Vulnerable Files                        | 0      |
| Vulnerable Servers                      | 0      |
| Web Server Detection                    | 16     |
| Total                                   | 265    |

TABLE IV. FOUNDATION NUMBER OF SECURITY VULNERABILITIES
The area where many security vulnerabilities were found was ‘Configuration Management’ item and there was exposure of files that could represent handling methods of components.

The next area where much vulnerability was found was ‘Sensitive Directories’ item and this is an item where directories that could contain web security sensitive data are searched and there was ‘Files containing passwords’ item which showed there were many security vulnerabilities.

IV. CONCLUSION

The study used SiteDigger, a Google automation search tool that can conveniently conduct webpage security diagnosis, to diagnose security of US State Department homepages. Overall, there were not too many security vulnerabilities.

However this is only numerical figure and it is difficult to compare security status with numbers.

While there are many tools that analyze software security vulnerabilities, there are lacking development of homepage security vulnerability diagnosis tools.

Programmers want their programs to operate securely with vulnerabilities completely removed. However it is difficult to acquire professional knowledge about vulnerability items and there are difficulties in recognizing how the vulnerabilities must be edited.

Therefore, it is necessary for development of homepage security vulnerability analysis tools. In the future, the study plans to develop a tool diagnosing web security vulnerabilities appropriate for global standard system characteristics.

ACKNOWLEDGMENT

This work was supported by a grant from 2015 Research Funds of Andong National University.

REFERENCES

[1] Symantec, Internet Security Threat Report, 2013 Trends, vol. 19, April 2014.
[2] Larry Ponemon, HP CISO. 2014 Global Report on the Cost of Cyber Crime. [Online]. Available: http://www8.hp.com/kz/ko/software-solutions/ponemon-cyber-security-report/index.html
[3] M. Y. Bae and H. K. Lim, “Security assessment by Google hacking automation tools for the web sites of Korea and USA universities,” IJSIA, vol. 9, no. 5, pp. 163-174, 2015.
[4] J. Ban, “Development trend for analysis tool of open source code security weakness,” Internet & Security Focus, May 2014.
[5] Ministry of Security and Public Administration, Software Development Security Guide for Electronic Government SW Development Operator, May 2012.
[6] Source Code Security Analyzers. [Online]. Available: http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html
[7] G. Conti, Google Knows You, Bpanbooks Publishers, 2009.
[8] Google Hacking Database (GHDDB). [Online]. Available: https://www.exploit-db.com/google-hacking-database/
[9] Software & Application Security Service. FSDB. [Online]. Available: http://www.mcafee.com/us/services/technology-consulting/software-and-application-security-services/index.aspx

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.

Han Kyu 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.

Dae Jea Cho He received the Ph.D. degree in computer engineering from Kyungpook National University, Korea, 2001. He is professor at Andong National University, Korea. His areas of interest include digital watermarking and multimedia contents authentication.