THE MOST COMMON VULNERABILITIES AND OPEN SOURCE PREVENTION SOFTWARE

[1]MEHMET MEHDI KARAKOC, [2]MUSTAFA ULAS, [3]ASAF VAROL

[1]Dept. of Computer Education and Ins. Tech., Agri Ibrahim Cecen University, Agri, Turkey
[2,3]Dept. of Software Engineering, Firat University, Elazig, Turkey
E-mail: mkarakoc04@gmail.com, mustafaulas@firat.edu.tr, varol.asaf@gmail.com

Abstract- To store data and information on digital media and be accessible via the internet, makes the life easy. Lots of user save or share personal information on digital platforms which are defined as local or public source. Banks, government agencies, commercial organizations also store and operate institutional customers' information on digital systems. An account information of a customer, a secret project that a company working on, an official secret which saved on a digital platform also attract the attention of attacker. Therefore the concept of information security has been put forward. In the first part of this study; information, information security, privacy, availability, data integrity, the importance of information security; data theft and the reasons of it have been described. Other parts, describe the most common types of malwares, web application security risks and the open source intrusion detection and prevention software that can be used against risks and malwares.

Keywords- Component; Information Security, Malwares, Vulnerabilities, Intrusion Detection Systems, Open Source

I. INTRODUCTION

Many definitions are made for the information concept. In Computer Sciences, it is defined as the form of the data, which is meaningful, or the edited data and the source decreasing the uncertainty, which is about a subject [1-3]. Information security includes all works, which are about forming a secure information editing platform in order to protect the information against unauthorized access or other environmental factors during the processes of saving or transferring it in the sense of electronic environments [3]. From this perspective it can be said that information security includes 3 sub-elements. These are accessibility, privacy, and data integrity. Accessibility is the act of obtaining the information at the time and location, when and where it is needed by the authorized users. Privacy means that not allowing access to the information for unauthorized users. Data integrity is the act of protecting accuracy of the information and information editing methods [4]. Information security is a vital factor because users, companies or even governments want to protect their information / data.

As the necessity of the accessibility principle, which is a part of information security, storing systems are generally connected with some networks. The related networks can be a local one or a global one, which is known also as the Internet. A company can form a local network structure in order to ensure access to the same devices and the Internet connection. It is also possible for banks to make their database accessible to the all over the world, in order to enable their customers to make their online banking operations over the Internet platform. However, all of these networks should ensure the privacy principle, which is another feature of the information security. All data, which are related to personal information, secret information about the government, commercial secrets about companies, or even customer information of banks, should be protected against any invasions that may be encountered over the network. Information theft can be made for different purposes and via different methods. As the technology improves, complexity of invasions also increases; but technical knowledge belonging to invaders decreases [3, 4].

In this study, the most remarkable vulnerabilities that are known by Internet pirates, the foremost invasion methods and also programs, which are used for detecting all these invasions, are explained briefly.

II. CURRENT WORKS

In recent years, there are some works related to web applications and research works, which are developed and done by some scientists in the sense of information security. Goseva-Popstojanova et al. have examined activities of invaders within a classical multi-layer web server, which obtain information via a high interactive, honey cube system. It has seen that the related invaders have scanned especially web 2.0 applications for any vulnerability, tried to crack user accounts encrypted with SSH, and run programs related to scanning vulnerabilities and detecting invasion methods [5]. In 2011, Shar and Tan have done a work on removing XSS vulnerabilities automatically over web applications. They have used a two-stage system for detecting XSS injections and removing them. At the first stage, they have done an analysis for tracking flow of user inputs and defining potential sensitivities expressions within HTML output queries. After that, in the second stage, pattern matching and data addictions analysis have been employed in order to prevent from XSS injection at

the user log in points [6]. Kim et al. has detected that invaders use fake antivirus programs for spreading harmful program codes. Because of this, they have developed a program, which detects web sites providing fake antivirus programs for users. With this program, the harmful web sites have been detected with a 90,4% accuracy and the false positive rate has been found as 0,2% [7]. Montanari et al. have examined the compatibility of using security policies over different devices. They have shown that flexible and big-size policy verification can be possible if only an algorithm and architecture for using security policy and verifications over different devices can be developed [8]. Hubballi and Suryanarayanan have done a work on minimizing false alarms within signature based invasion detection systems. In the work, the related techniques have been explained by classifying them, and also programs using these techniques and some essential scanning features of these programs have been introduced [9]. Stamp et al. have developed a system for detecting http invasions, by using the n-gram analysis method. By using 3 different n-gram techniques, a work aiming to ensure first protection by filtering positive web traffic has been done [10]. Weber has examined new security and privacy problems over the Internet and shown that security and privacy concepts are important, ensuring identity verification, access control and privacy is vital, and expressed that the regulations should keep pace with the technology and revisions in this sense should be done via collaborative works that can be done with the private sector [11]. Sicari et al. have examined IoT (Internet of Things), privacy over Internet, security and trust concepts, determined problems related to these concepts, provided solutions for the problems, and also expressed a road map in the sense of expressed solutions-oriented works. It has been stated in the work that IoT requires special privacy and security levels but it has no vision in this sense yet and security at IoT increases in especially mobile devices. It has been also detected that even there are scientist working on the related issues, some subjects belonging to IoT have not been discussed yet [12]. Zineddine has developed a framework for cloud security and introduced a cost optimization approach by using Cuckoo Search algorithm with Levy flights. It has been shown that cloud providers can balance the costs by using the related optimization approach [13]. Lee and Kim have developed a framework for SQL injection invasions in the database layer by using the SVM (Support Vector Machine) supported via different kernel functions. As a result, the system, which was designed by using SQL intra-query trees, has detected SQL injection invasions with 99.6% rate. The system has also compared with the singular statistical model and it has seen that the developed system has provided better performance at detecting the invasions [14]. Choi and Jang have developed a technique, which is based on analyzing SQL query dimensions in order to make web sites more secure against SQL injection invasions. The system has the ability to detect invasions by using dimension of a query with the query dimension predicted by the program user. It has been shown in the work that by using a temporary query variable and making a cleaning according to the query dimension, SQL queries having infiltration vulnerabilities can be detected and the related methods can be applied successfully [15]. Johns has seen disadvantages of Content Security Precautions against XSS invasions and suggested use of PreparedJS script package as an additional precaution for the Content Security Precautions. It has been seen in the experiments that use of this package eliminates the need for untrusted server-side JavaScript integration, ensures a good control and in this way prevents almost all XSS invasions [16]. Salas and Martins have compared the security tests, which are used for detecting XSS invasions against web services and used two different techniques for testing robustness of web services. These techniques are called as Penetration Test (SoapUI) and Fault Injection (WSInjection). It has been seen that positive and negative rates were high at SoapUI and WSInjection, which is a new approach and were suggested in for robustness test within the study, was able to emulate invasion techniques, which cannot be imitated via other tools [17]. Awoleye et al. have examined official web sites of 64 companies in Nigeria for 2 years in order to evaluate potential application errors within e-government applications. 5 known vulnerabilities have examined in the sense of the research work. As a result, it has been figured out that 67% of web sites were including broken link vulnerabilities whereas 43,8% of them were including non-encrypted password vulnerabilities, 35% of these web sites were having XSS vulnerabilities and one of each 4 web sites were including SQL injection vulnerabilities [18].

III. MOST KNOWN HARMFULS AND SECURITY VULNERABILITIES

While designing a program or system, it is not always possible to check the whole program and remove any errors occurred. It is possible for a program to have some security vulnerabilities even it has been checked very well [19, 20]. Generally, such vulnerabilities are used for reaching to data by eliminating the privacy principle. There are many programs, which have been developed in order to detect such vulnerabilities, perform invasions, make data theft, break down systems, and use it for malicious purposes. Such programs are called as harmful programs, malicious programs or malware briefly [21]. The most known harmful programs and their general features are explained briefly as follows [22, 23]:

Viruses: These are programs, which can integrate themselves to other executable programs, codes or documents and also clone and spread themselves

along other files. Trojans: Trojans are harmful programs that act like useful programs. In this way, they hide themselves from the user. Back Doors: These are programs, which have infiltrated to the computer in order to make unauthorized remote access and can hide themselves and create vulnerabilities in order to make the computer accessible remotely. Browser Hijacking: These are harmful programs that can change options of browser programs. Exploit: Exploits are programs that can perform invasions by using some known security vulnerabilities. Worms: These are harmful programs, which do not need any additional program to be copied / cloned. Spyware: These are harmful programs, which can collect user information and important data to send them to information / data thieves. Keyloggers: Keyloggers are spy programs, which can track / save each typed keyboard characters by user and send the collected data to information / data thieves. Rootkits: Rootkits are programs, which take control of a system and erase all data regarding to them and perform some camouflage related operations in order not to be detected by security programs. Spam: Spam is defined as unwanted e-mails. Users generally do not want to be disturbed via spam mails, which are related to advertisements, commercial or any harmful purposes. Except from the mentioned ones above there also many other harmful programs like Dialer, adware, thiefware, Remote Administration Tool, botnet, flooder, hostile ActiveXhostile Java, hostile script, nuker, packer, binder, password capture - password hijacker, password cracker, key generator, mass mailer, E-mail harvester, web bugs, hoax, phishing, web scam, fraud, phreaking, port scanner, search hijacker, sniffer, spoofer, spyware cookie, tracking cookie, and PIE…etc.

By using the mentioned programs and vulnerabilities, many different invasion methods via different tools have been developed in time. The most encountered web applications security risks, which are related to 2013, are explained briefly as follows by listing them according to their frequencies [24-29]:

Injection: Injection invasions occur when data received from users are used within commands or database queries, without controlling. There are different types of injection like SQL injection, Code injection, UDF injection, and XML injection. By using such vulnerabilities, an invader can have authorization to execute commands at the target system and also gain access to the database. The most used SQL injection invasions belong to executing SQL queries within the database of the target web site. On the other hand, code injection is related to injection operations in the sense of employing codes, which can be executed directly within the target server operating system. Broken Authentication and Session Management: This one belongs to employing functions and methods, which are used for opening sessions but not developed properly. There are two types: session prediction and session fixation. In the sense of this approach, random and predicted cookie information is send to the user and when the user wants to open a session via this cookie and real authentication information the invader obtains an authorized cookie. Cross Site Scripting (XSS):

This is the activity of executing desired client based codes in the user's browser by integrating client based codes into HTML codes. Reflected XSS occurs when query-response data is stolen. In this sense, a link should be clicked or a page should be loaded in order to start the scripting approach. On the other hand, Stored XSS is a stored XSS invasion. It is done by integrating codes into the data, which is a typical database associated with user information. Finally, DOM XSS is the activity of changing objects within web pages and executing / running the page according to invader's purposes. Insecure Direct Object Reference: It belongs to changing information within web address and damaging the application because of insufficient controls for data transferred within address areas. By changing the parameters send within web addresses, the invader can see some hidden information. Security Misconfiguration: In this approach, the invader infiltrate to the system by using default accounts, unused pages, unpatched vulnerabilities, and unprotected files or folders in order to obtain information. Sensitive Data Exposure: It defines the situation in which mandatory information like credit card data, tax number, identification data are obtained because such information are not encrypted or hidden. Missing Function Level Access Control: This is the situation in which request come from Internet is not controlled in the sense of authority / invasion. This situation occurs when invaders can see the pages, which belong to only authenticated users. Cross-Site Request Forgery: That means forgery, which belongs to requests between web sites.

In this approach, the invader makes some operations over the application by using user session information without any permission. This vulnerability generally appears when the developers do not use any approach in order to control requests if they belong to the related user or not. Using Known Vulnerable Components: This is the act of performing invasions by using vulnerabilities belong to libraries or modules, which are used as in full authorization. Unvalidated Redirects and Forwards: That means making redirections without making any first and last data control.
This can also be done by directing the user to an undesired page. The related web security risks explained above are the most seen ones in the sense of 2013. At this point, there can be differences according to years [30]. The most seen web invasioning methods in 2014 are presented in Figure 1.

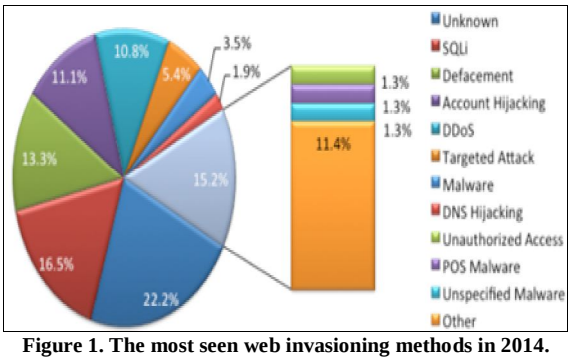

Figure 1. The most seen web invasioning methods in 2014.

IV. EXAMPLES OF INVASIONS HAPPENED RECENTLY

At 12.11.2014, the HSBC bank has announced that some data regarding to customers' bank and credit card information were stolen [31]. In 2014 October, it was claimed that passwords belonging to 7 million members of DropBox, which is both cloud server and social sharing platform, were stolen. At first, the company has not accepted the claims regarding to hacking but after some time, they have explained that it was possible that some user data and also important project document files were stolen [32]. In 2014 September, it has appeared that user name and password data of 4.93 million Gmail users were stolen and shared over the Internet. The Google Company has explained that such data could be obtained from different sources and obtained via some methods belonging to use of some harmful programs and invasion methods [33]. In 2014 September, iCloud, which is a cloud service by Apple Company, has encountered with some invasions. According to the claims, the invaders have used some vulnerabilities of the Finds My iPhone application and reach to photo files. But Apple has explained that there was not any vulnerability like that and after a 40-hour analysis, the company has explained that such data could be obtained by using problems belong to user name and password. At this point, Apple has suggested their users to use more powerful passwords and two-step confirmation system [34]. In 2014 October, the researcher Mohamed Baset has recognized in Samsung cell phones that the invaders can control the phone remotely, lock the device, find the phone over map, view messages over the phone screen and erase data by using some vulnerability. This vulnerability is like the feature, which is available in the Find My Mobile application, which is used for finding stolen phones. This feature allow invaders to act like the real user because there is no control on sources of received requests [35]. In 2014 November, a hacking group: Guardians of Peace has hacked the Sony computer network and shared four new films over the Internet. Additionally, they have also shared some data regarding to some hidden files of the company and passport or e-mail messages of some company staff [36].

V. PROGRAMS FOR INVASION DETECTION AND PREVENTION

Although web applications and network systems are tried to be made more secure, there may be some disadvantages / vulnerabilities, which are not detected yet. But unless these vulnerabilities are discovered by invaders, system managers can also detect preparations done for invasions. Because before an invasion is performed, the invaders should have some information about the target system and detect the related vulnerabilities that can be used. By using some detection tools and programs, such preparations for the invasions can be detected before [37].

Invasion detection programs are generally works in two different approaches: signature based and according to behaviors at runtime (heuristic detection). Some systems can use also both two approaches (hybrid system) [38, 39]. In this section, some open source invasion detection programs are introduced.

SNORT: Snort is an open source, invasion detection and prevention system (IDS), which is developed by the Sourcefire company and distributed with GNU license. Snort includes many package preprocessors that make it possible to prepare the packages, which are obtained via strem5, http, ftp telnet, smtp, to be compared with the rules. It can perform real time traffic analysis and package logging over IP networks. Generally, it works as signature based and can perform protocol and anomaly analysis. Users can write their own rules or just download paid or free rule sets from different web sources. The program can detect invasions and harmful program types like protocol analysis, CGI invasion, content scanning / matching, port scanning, buffer overloading and trying operating system finger print. Two network interface cards are used for the Snort generally. One of these cards is for listening to the network while the other is used for configuring the Snort via remote connection. Snort is generally used for detecting infiltration attempts. In this mode, Snort performs the traffic analysis and makes the related activities according to the rules defined by the user. General architecture of the Snort is shown in Figure 2.

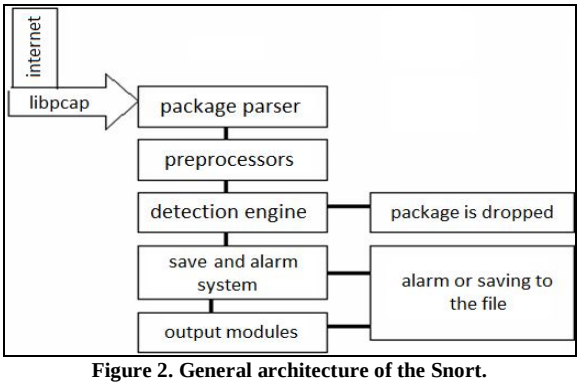

Figure 2. General architecture of the Snort.

```
alert tcp any any -> 156.154.70.1 80 (msg:"Test Rule";
sid:5000853; content:"GET"; content: "cgi-bin/phf";)
```

Figure 3. An example rule written for the Snort.

Libpcap: Libpcap is the package catching / obtaining library, which is used by Linux and UNIX. In Windows, WinPcap is used instead of Libpcap. An example rule written for the Snort is given under Figure 3. According to the rule in Figure 3, a message showing "Test Rule" will be viewed when a package containing "cgi-bin/phf" is send to the port 80 of the address: 156.154.70.1 from any address and port by using the TCP [40-44]. SURICATA: Suricata is an open source, invasion detection and prevention program, which is distributed via GNU/GPL 2 license. It is developed and supported by the OISF (Open Information Security Foundation). Like Snort, Suricata can work as signature/rule based. It also supports the rule sets used by Snort.

Suricata has some remarkable features and functions like including HTTP normalization tool, which is called as HTP library, and working as multithread. It decomposes HTP and HTTP traffics. By using the "security-aware" feature, it can catch / detect some techniques, which can be used by invaders in order to eliminate invasion detection mechanisms. It also has different decomposers for operations in the sense of request row, request title, URI, user component, response row, server response row, cookie, "basic" and "digest" identification approve, which are related to library HTTP protocol. Supporting the multi-thread working allows making the package editing operations as distributed via different process particles, in systems with multi-processors.

In this way, performance is improved by sharing the process load among processors. Invasion detection system (IDS) can be used in working modes like invasion prevention system (IPS). It allows saving the network traffic and analyzing the saved files offline. The configuration is in YAML format and supported by many programming languages. It can track all TCP sessions and order the flow. It is also possible to merge separated packages. It supports second layer protocols like Ethernet, PPP, VLAN, QINQ. Also, it can analyze application layer protocols like HTTP, SSL, TLS, SMB, SMB2, DCERPC, SMTP, FTP, SSH, and DNS. PCRE (Perl Compatible Regular Expressions) can be used in the written rules and matching regarding to file type, size, and MD5 brief value. Without needing any restart, adding, updating, and erasing operations regarding to rules can be done while the system is running. It can save HTTP requests, TLS jamming, SSH connections, and DNS requests/responses. All data are saved in JSON format, which is compatible with many programming languages. Produced alarms can be saved in text or unified2 binary format and stored in a specific database [44-51].

BRO: Bro is an invasion detection system and network analysis and watching program, which is distributed via BSD license and based on UNIX. As different from IDS, which are classical rule based, Bro is a network analysis tool. It includes not only security related approaches, but also solutions for performance analyze and network problems. Bro create save data for many operations within the network. In addition to the mechanism of saving all traffic, it also allows resolving protocols in the application layer. Bro has its own script language and this feature allows it to be flexible and improvable. Users can improve the system by writing special scripts. As general, it is possible to detect harmful activities or anomalies in the network or perform behavioral analyze, thanks to the related scripts. Bro can perform real-time or offline resolving operations. It uses the "libpcap" library for obtaining / catching the packages. In the clustered systems, which have intensive and distributed traffic, it uses Bro Clusters service for working on different servers and ensuring communication among these servers. It can analyze and save all HTTP traffic, DNS queries, SSL certificates, SMTP sessions, FTP traffic, and network flow. The data is saved in text files as ASCII format. It can analyze DNS, FTP, HTTP, IRC, SMTP, SSH, and SSL, which are protocols of the application layer. All of these analyze results then can be saved as text format with MD5/SHA1 brief values. It can also detect some harmful programs by using external sources. It can detect vulnerabilities of applications like Java or Flash and invasions regarding to SSH brute force. With Bro, it is also possible to detect and analyze tunnel protocols. It supports the pattern matching method, which is supported by other IDS programs, and external sources that can used for analyze processes can be integrated to the system in real-time. At this point, different external operations like sending e-mail in case of alarm, ending instant connection and preventing forwarding can be done, thanks to the script language [44, 52]. OSSEC: Ossec means Open Source Intrusion Detection System. This programs watches installed programs, data packages, system calls and any other system activities and warns users in case of unusual situations. Ossec always watches the performed operations and activates different behaviors in case of predefined situations.

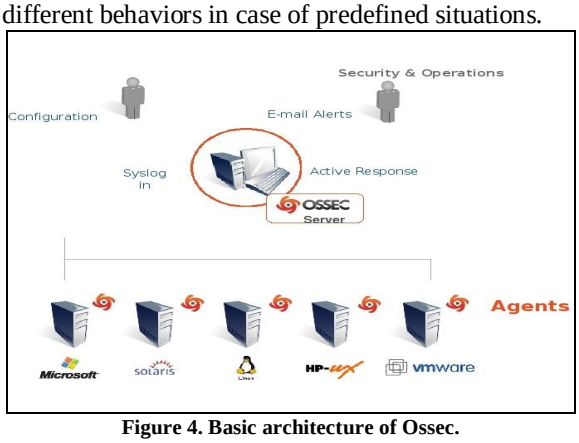

Figure 4. Basic architecture of Ossec.

Ossec can detect anomalies by analyzing log files, control system files and store these file information by using some encryption algorithms and control data integrity by performing analysis on current and previously stored file information, run rootkit scan over systems periodically, and prevent the invaders by sending messages to the firewall and ending their session by cutting the connection. There is more than one component in the Ossec in order to install it on different systems. Ossec architecture is shown in Figure 4. The essential component of Ossec is Manager. Task of this component is to track all structure and analyze information received from remote computers. Agent is the component installed on the system, which will be watched from Ossec. It obtains information from the system, which it is installed over and sends the related information to the Manager. Because it uses a little system sources it does not provide too much additional load for the system. On the other hand, in order to watches systems like firewall, switch, router (on which agent cannot be installed), the Agentless mode can be used. In order to watch virtualization infrastructure, the Virtualization / Vmware mode has been employed. Ossec can also obtain logs from remote systems and analyze them [53-55]. Except from the related programs, some other examples are open source programs like Security Onion, Openwips-Ng, Dig Deeper, Firestorm, and Pokemon.

CONCLUSIONS

In this work, the most recent and remarkable harmful program types, security risks related to web applications and the most popular 4 open source detection and prevention programs, which can be used against the related invasions / problems, have been explained. Eventually, different methods can be used alone or together for performing invasions. It has been seen that it is mandatory for the invaders to obtain information from the system, on which they will perform invasions. Because these invaders use some programs while obtaining information from the system, it is possible to detect the related invasions with some programs and methods, before they happen. It is also seen that some preventions can also be done by using some programs against incoming invasions. It can also be expressed that security can be ensured easily within even professional networks, thanks to some free and open source invasion detection programs. Even there is some prevention done already, it is also important to analyze the network management via IDS and control the whole system against any anomalies. It is clear that some open source IDS programs can be run over all current operating systems. IDS can perform analyze operations in both online and offline modes. Such programs have ability to perform analyze on almost all protocols regarding to OSI and TCP/IP layers. IDS can perform analyze on also encrypted content and perform brief analyze like MD5 or SHA1 in order to ensure data security. Although IDS programs have their own script and rule languages, they also support rule expressions associated with other IDS programs. It is seen that IDS programs can use both internal and external sources for analyze and alert activities. In order to make Snort more effective, double network interface cards should be used. Suricata provides more effective performance by running multithread. As different from classical rule-based IDS, Bro can be used for not only for security, but also for performance analyze and solutions for network problems. As general, Ossec runs computer based and ensure security mostly in operating system level by controlling operating system activities, data packages, system calls, and any other system activities.

REFERENCES

[1] M. E. Balcı, A Concept between Pathos and Ethos: Sociology of Science,» [Online]. Available: http://www.flsfdergisi.com/ sayi12/79-89.pdf. [Accessed: 10 November 2014].

[2] TDK, Turkish Language Association, [Online]. Available: http://tdk.gov.tr/index.php? option=com_bts&view=bts.

[3] Ş. S. Gürol Canberk, A Study on Information, Information Security and Process, Journal of Polytechnic, cilt 9, vol. 3, pp. 165-174, 2006.

[4] Turkey Information Security Association, Information Security and Management, [Online]. Available: http://www.tbd.org.tr/ userfiles/4/zeynep/egitim_bgys_sunum.pdf. [Accessed: 10 November 2014].

[5] K. Goseva-Popstojanovaa ve A. D. R. P. B. M. Goce Anastasovskie, Characterization and classification of malicious Web traffic, Computers & Security, vol. 42, pp. 92-115, 2014.

[6] H. B. K. T. Lwin Khin Shar, Automated removal of cross site scripting vulnerabilities in web applications, Information and Software Technology , vol. 54, pp. 467–478, 2012.

[7] Y. P. Z. J. Kim DW, Detecting Fake Anti-Virus Software Distribution Webpages, Computers & Security, 2014.

[8] E. C. K. L. W. Y. R. H. C. Mirko Montanari, Distributed security policy conformance, Computers&Security, vol. 33, pp. 28-40, 2013.

[9] V. S. Neminath Hubballi, False alarm minimization techniques in signature-based intrusion detection systems: A survey, Computer Communications, vol. 49, pp. 1–17, 2014.

[10] K. R. R. M. L. M. S. Aditya Oza, HTTP attack detection using n-gram analysis, Computers&Security, vol. 45, pp. 242-254, 2014.

[11] R. H. Weber, Internet of Things – New security and privacy challenges, Computer Law&Security Review, vol. 26, pp. 23–30, 2010.

[12] L. G. A. C.-P. S. Sicari A. Rizzardi, Security, privacy and trust in Internet of Things: The road ahead, Computer Networks, vol. 76, pp. 146–164, 2015.

[13] M. Zineddine, Vulnerabilities and mitigation techniques toning in the cloud A cost and vulnerabilities coverage optimization approach using Cuckoo search algorithm with Levy flights, Computers&Security, vol. 48, pp. 2015, 1-18.

[14] D. H. L. Mi-Yeon Kim, Data-mining based SQL injection attack detection using internal query trees, Expert Systems with Applications, vol. 41, pp. 5416–5430, 2014.

[15] K. W. M. Z. T. L. Hossain Shahriar, Effective detection of vulnerable and malicious browser extensions, Computers&Security, vol. 47, pp. 66-84, 2014.

[16] M. Johns, Script-templates for the Content Security Policy, Journal of Information Security and Applications , vol. 19, pp. 209-223, 2014.

[17] E. M. M.I.P. Salas, Security Testing Methodology for Vulnerabilities Detection of XSS in Web Services and WS-Security, Electronic Notes in Theoretical Computer Science, vol. 302, pp. 133–154, 2014.

[18] B. O. M. O. I. Olusesan M. Awoleye, Web application vulnerability assessment and policy direction towards a secure smart government, Government Information Quarterly, vol. 31, pp. S118–S125, 2014.

[19] E. Sarıdoğan, Software Engineering, 2. Edition., Ankara: Papatya Yayıncılık, 2008.

[20] A. Gürbüz, Software Test Engineering, Ankara: Papatya Yayıncılık, 2010.

[21] T. I. S. Agency, Anti-Malware Guide, [Online]. Available: http://www.bilgiguvenligi.gov.tr/dokuman-yukle/microsoft/uekae-bgt-1004-zararli-kod-yazilim-korunma-kilavuzu/download.html. [Accessed: 11 November 2014].

[22] B. K. E. G. I. KyoungSoo Han, Malware categorization using dynamic mnemonic frequency analysis with redundancy filtering, Digital Investigation, 2014.

[23] G. Canbek ve Ş. Sağıroğlu, Malwares and Spywares: A Comprehensive Research, Journal of Gazi Univ. Eng. and Arc. Faculty, cilt 22, vol. 1, pp. 121-136, 2007.

[24] G. A. A. D. R. P. B. M. Katerina Goseva-Popstojanova, Characterization and classification of malicious Web traffic, Computers&Security, vol. 42, pp. 92-115, 2014.

[25] OWASP Top Ten 2013 Project, OWASP Top Ten 2013 Project, [Online]. Available: https://www.owasp.org/index.php/Top_10_2013-Top_10. [Accessed: 10 November 2014].

[26] Information Security Academy, Information Security Academy 2012 (in Turkish), [Online]. Available: http://www.bga.com.tr/calismalar/websec/. [Accessed: 29 November 2014].

[27] ULAKBIM, Most Critical 10 Web Application Vulnerabilities, [Online]. Available: http://csirt.ulakbim.gov.tr/dokumanlar/Ceviri_OWASP.pdf. [Accessed: 16 November 2014].

[28] H. Eken, Security in mobile and web applications, [Online]. Available: www.ab.org.tr/ab13/bildiri/167.pdf. [Accessed: 10 November 2014].

[29] Z. Yeşilyurt, Linux Summer Camp 2014, Ankara.

[30] Hackmageddon, November 2014 Cyber Attacks Statistics, [Online]. Available: http://hackmageddon.com. [Accessed: 3 Aralık 2014].

[31] HSBC, Annoumcement to customers, [Online]. Available: http://www.hsbc.com.tr/tr/haberler/haber_detay.asp?NewsId=984&WT.ac=HBTR_main_tr_dyr_12112014. [Accessed: 20 November 2014].

[32] Milliyet, Dropbox hacked!, [Online]. Available: http://www.milliyet.com.tr/dropbox-hacklendi-7-milyon-sifre-internet-1954247. [Accessed: 20 November 2014].

[33] Hurriyet, 5 milions Gmail passwords on the internet, [Online]. Available: http://www.hurriyet.com.tr/ekonomi/27181091.asp. [Accessed: 20 November 2014].

[34] Cumhuriyet, Apple accounts hacked, [Online]. Available: http://www.cumhuriyet.com.tr/haber/bilim-teknik/136318/Apple_hesaplari_hacklendi.html. [Accessed: 20 November 2014].

[35] Sabah, Vulnerabiliy on Samsung mobile phones, [Online]. Available: http://www.sabah.com.tr/teknoloji/2014/10/30/samsung-telefonlarda-guvenlik-acigi. [Accessed: 21 November 2014].

[36] BBC, Whodunnit? The Mystery of the Sony Pictures Hack, [Online]. Available: http://www.bbc.com/news/technology-30530361. [Accessed: 21 November 2014].

[37] University of Berkeley, Intrusion Detection Guideline, [Online]. Available: https://security.berkeley.edu/content/intrusion-detection-guideline. [Accessed: 16 November 2014].

[38] G. Kesici, Bypassing Antivirus Softwares-1, [Online]. Available: https://www.bilgiguvenligi.gov.tr/yazilim-guvenligi/ antivirusleri-atlatma-yontemleri-1.html. [Accessed: 16 November 2014].

[39] V. Hataş, Compressing and controlling applications with EMET 4.0, [Online]. Available: https://www.bilgiguvenligi .gov.tr/yazilim-guvenligi/emet-4.0-ile-uygulamalarin-sikilastirilmasi-ve-denetimi.html. [Accessed: 16 November 2014].

[40] M. Pascucci, Top five free enterprise network intrusion-detection tools, [Online]. Available: http://searchsecurity. techtarget.com/tip/Top-five-free-enterprise-network-intrusion-detection-tools. [Accessed: 16 November 2014].

[41] B. Dayıoğlu, Intrusion detection with Snort, [Online]. Available: http://seminer.linux.org.tr/wp-content/uploads/snort.pdf. [Accessed: 16 November 2014].

[42] Ö. Erdem, Application Awareness with Snort OpenAppID, [Online]. Available: https://www.bilgiguvenligi.gov.tr/ saldiri-tespit-sistemleri/snort-openappid-ile-uygulama-farkindaligi.html. [Accessed: 16 November 2014].

[43] darkreading.com, Cisco Banks On Sourcefire And Snort For Its Security Future, [Online]. Available: http://www.darkreading.com/attacks-breaches/cisco-banks-on-sourcefire-and-snort-for-its-security-future/d/d-id/1140164. [Accessed: 16 November 2014].

[44] Ö. E. Ulaş Kaya, Saldırı Tespit Sistemleri (Snort, Suricata, Bro), [Online]. Available: https://www.bilgiguvenligi.gov.tr/ saldiri-tespit-sistemleri/saldiri-tespit-sistemleri-snort-suricata-bro.html. [Accessed: 15 November 2014].

[45] Comparison of Open Source Network Intrusion Detection Systems, [Online]. Available: https://www.duo.uio.no/bitstream/ handle/10852/8951/Rodfoss.pdf?sequence=1. [Accessed: 16 November 2014].

[46] R. McRee, Suricata: An Introduction, [Online]. Available: holisticinfosec.org/toolsmith/pdf/august2010.pdf. [Accessed: 15 November 2014].

[47] openinfosecfoundation.org, Suricata User Guide, [Online]. Available: redmine.openinfosecfoundation.org /projects/suricata/wiki/Suricata_User_Guide#Suricata-User-Guide. [Accessed: 16 November 2014].

[48] suricata-ids.org, Complete list of Suricata Features, [Online]. Available: http://suricata-ids.org/features/all-features/. [Accessed: 16 November 2014].

[49] IEEE, Combating Cyber Attacks: Using a 288-Core Server, [Online]. Available: http://sites.ieee.org/scv-cs/files/2013/02/TileraApp-Suricata_IEEE-FINAL-2-11-13.pdf. [Accessed: 14 November 2014].

[50] netfilter.org, Suricata, [Online]. Available: http://workshop.netfilter.org/2013/wiki/images /1/1f/Eric_Leblond_IDS-suricata.pdf. [Accessed: 15 November 2014].

[51] openinfosecfoundation.org, Suricata, [Online]. Available: https://redmine.openinfosecfoundation.org/ projects/suricata/wiki/Suricatayaml. [Accessed: 16 November 2014].

[52] bro.org, Bro 2.3.1 documentation, [Online]. Available: https://www.bro.org/ sphinx/intro/index.html. [Accessed: 14 November 2014].

[53] linuxakademi.com.tr, Installation and Configuration of OSSEC (HIDS), [Online]. Available: http://www.linuxakademi.com.tr/ossec-hids-kurulumu-ve-yapilandirmasi/. [Accessed: 17 November 2014].

[54] A. Tasım, OSSEC - Open Source Intrusion Detection Systems and Installation, [Online]. Available: https://www.bilgiguvenligi.gov.tr/saldiri-tespit-sistemleri/ossec-acik-kaynak-kodlu-saldiri-tespit-sistemi-ve-kurulumu.html. [Accessed: 17 November 2014].

[55] OSSEC, Open Source SECurity, [Online]. Available: http://www.ossec.net/. [Accessed: 16 November 2014].

★★★