Enterprise Wide Web Security Infrastructure

Enis Karaarslan[1], Tugkan Tuglular[2], Halil Sengonca[3]

[1] Ulak-CSIRT, TUBITAK – ULAKBIM, Turkey,
enis.karaarslan@ege.edu.tr

[2] Izmir Institute of Technology,
Department of Computer Engineering, Izmir, Turkey
tugkantuglular@iyte.edu.tr

[3] Ege University, Computer Engineering Department,
Izmir, Turkey
halil.sengonca@ege.edu.tr

Abstract. There is increasing number of intrusion attempts to the web infrastructure, so web and web application security are becoming more vital everyday. There is need for security measures, which will detect and prevent vulnerabilities before intrusion and attack occur. In this work, an Enterprise-wide Web Security Infrastructure model where different techniques work cooperatively to achieve better security is defined. Network awareness and training are focused on. The solution and statistical results of the implementation of the open source tools and the management platform are given.

Keywords: web security, web application security, network awareness, web system awareness, intrusion detection, vulnerability analysis

Introduction

Web has become the most efficient way to supply information to public and accomplish e-commerce. Also web servers have become the easiest tool to manage devices. There is increasing number of intrusion attempts to web infrastructure, so web and web application security is becoming more vital everyday.

Most of the time, the source of vulnerability is programming errors. Secure coding and vulnerability testing must be implemented in the Software Development Life Cycle (SDLC). In spite of the obvious requirements, they're not implemented as they are supposed to be in real life. This is usually caused by the nearby deadline of the project and programmer's lack of security-awareness. There are deployed systems which are vulnerable so security tests and network based measures must be taken anyway. These systems should preferably be proactive, rather than reactive. Different technologies are present to detect and prevent web attacks [1] [2]. Each technique has its own strength and weaknesses. These systems also produce too many logs as they don't have enough information about the network they are installed on. In this work, an Enterprise-wide Web Security Infrastructure model is defined where all techniques work cooperatively to achieve better security. Open source tools and a management platform are being implemented in Ege University Campus Network. Implementation details and statistics of the current project will be presented in this study.

Web attacks are now mostly implemented at the application layer. Open Web Application Security Project (OWASP) lists the most critical web application and database security vulnerabilities and provides basic methods to protect against these vulnerabilities [3]. User supplied data are mostly not validated in the web applications and invalidated input is the root of many security problems. Attacks like Sql injection, Cross Site Scripting (XSS), buffer overflow use this weakness. Applications can leak information about the system because of improper error handling. Attackers use this weakness to learn about the system and use this information in further attacks [3].

In enterprise networks, several servers which host hundreds of domains and different web applications can be present. Detailed information about the web servers and applications running on them may not be present. Intrusion detection systems (IDS) can be used to detect web attacks but mostly those systems are not network aware, that is they don't have enough information about the network they are installed on [2]. The difference when sufficient information is supplied to IDS will be shown in this study.

The next section describes Ulak-CSIRT and ULAKNET. In Section 2, network and web system awareness concept and analysis methods (active, passive and hybrid) are described. In Section 3, Enterprise Web Security Model is given. Implementation in Ege University and Turkish Academic Network (ULAKNET) are given in section 4. In Section 5, conclusion and future work are given.

Ulak-CSIRT and ULAKNET

Ulak-CSIRT (Computer Security Incident Response Team) is a security unit that has been established within the context of Turkish Academic Network (ULAKNET). Ulak-CSIRT is responsible for preventing the potential security violation of external networks to ULAKNET, ascertaining the attacks and the people in charge and in the same way, preventing the attacks of ULAKNET to the outside world and if there was an attack, ascertaining the people in charge of the attack and sharing the information with the administrators of this network.

Turkish Academic Network (ULAKNET) is an interactive system which uses new technologies and connects the innovation centres to each other in the national scale. There are 110 distinct units connected to ULAKNET. The total number of the nodes reached to 650 with the connection of Higher Institutes of Technology, Faculties, Graduate Schools, Schools of Higher Education, Conservatories, Vocational Schools and Research Centres of the universities. Over 100.000 lecturers and 2.000.000 students are using ULAKNET in these nodes.

Network and Web System Awareness

Network awareness is the concept that describes the ability of knowing what is happening on the network [4]. If the attacker has more information about the targets on a network than the security systems, that knowledge can be used to evade the security systems. The first aim of the defence should be having detailed and current information of the network assets. In this work, web security is focused on and a specialized form of network awareness which is called "web system awareness" is introduced. Web system awareness is essential to reduce the Intrusion Detection System's false positive alarms which can occur in huge amounts. For example, Code Red attack is the one that can exploit vulnerabilities of Microsoft IIS Server. If such an attack is implemented against a UNIX server that runs Apache web server software, this attack can not be successful. A network aware system should understand those types of unsuccessful attempts and should not log this attempt or reduce the priority of this attempt in the logs.

Web server characteristics can be described to security systems by configuring manually. Especially in enterprise networks, the network administrators may not know all the servers and their all characteristics. In addition the systems can change in time; so a dynamic process (network analysis) must be involved. This process can be done in three ways which have different positive/negative sides [2]:

- Active Analysis: The enterprise's web server characteristics can be learned by advanced port scanning. Active analysis can be optimized to scan only specific ports and routinely check changes on IP addresses and software that are found in previous scans. HTTP server's identity can be determined by "HTTP fingerprinting" which is implemented by sending requests to the HTTP server and then analyzing the characteristics of the HTTP response messages [5] [6]. Some of the http fingerprinting methods may be evaded by specific configuration on the web servers. Although such configurations make information gathering harder, it's always possible to gather information about the web servers by http fingerprinting [6] [7].
- Passive Analysis: Passive techniques use sensors to monitor network traffic and investigate packets. Passive techniques have some advantages like; no bandwidth usage, stealth run and vision of the real time web activity. It should be noted that passive techniques usually take more time to learn assets than active techniques. Also idle web servers will not be visible to the system [8]. The locations of the sensors are very important as it will only see the traffic of that segment. Many sensors may be needed to get the whole picture of the enterprise network web server structure.
- Hybrid: Hybrid analysis is the combination of active and passive techniques for optimum performance. Active analysis can be optimized to run efficiently like scanning specific ports and running in specific intervals. As active analysis will not use through investigation, passive analysis can be used to find information like unlisted web servers and unlisted web ports. The proposed architecture is given in Section 3.

The data that is to be collected by analysis methods is as follows [2]:

- Web server IP addresses
- Protocols used (https, http)
- Site domain names (ex. socrates.ege.edu.tr)
- Web server ports (80, 8080, etc)
- Operating system (Linux, Windows, etc)
- Web server software types and versions (Apache 2.0, IIS 6.0, etc)
- Content Management Systems (CMS), Portals, Wikis, Bulletin Boards, discussion forums

- Web frameworks (PHP, .NET, J2EE, Ruby on Rails, ColdFusion, Perl, etc) and all types of web applications
- Application file names
- Path to the applications, the directory structures
- Parameters that are being passed to the application and their types

Enterprise Web Security Model

A security model which consists of different security measures that work together is introduced. The system consists of the following modules:

- Standardization
- Awareness
- Training/Testing
- Detection
- Prevention
- Coordination Centre

Standardization

Standards and rules must be defined and used for a more stable and more secure web system. The system should consist of the following:

- Web Server's platform (OS, web server software, programming environments, etc) limitations, vulnerability analysis methods, patch management and backup procedures must be defined.
- The number of web servers should be reduced for better management and security. It can be a good practice to use a central web server and encourage the departments to use this server only.
- Secure coding principles should be used when developing web applications [9] [10] [11].
- Enterprises should develop their own database and program libraries and classes which are built with secure coding principles. It must be assured that these codes are used in all software projects. These codes will supply more secure environments to the developers.
- Enterprises can also choose development frameworks like Struts that force the developers to write programs with secure coding principles.
- Vulnerability testing should be implemented in the SDLC.
- Web applications should be documented by appropriate tools like javadoc, phpdoc, etc.
- Enterprise web pages should be in a specific structure and template.
- Enterprises should build their own Content Management Systems, portals if possible. The resulting systems will be more flexible and secure. The enterprise can also choose a proprietary CMS and can monitor these systems constantly. The system administrators should make sure that the systems are timely patched and stable.
- Enterprises should use a template error page that will be shown when a software error occurs on the webpage. This will prevent sensitive information leakage.
- Network Management Group should have right to close the web service if standards not met.

Awareness

The network and security administrators should know what's happening on their network. This includes knowing the assets to be protected, their current status, the threats and the vulnerabilities. The system should consist of the following:

- **Web System Awareness:** The aim should be detecting and identifying web server and web application specifications. As described in detail in Section 1, active and passive analysis techniques can be used cooperatively to have an up-to-date view of the web server infrastructure. Also it can be a good idea to use a custom search engine to index the web content in the enterprise for collecting information about the web system.
- **Vulnerability Analysis:** Vulnerability Analysis Systems can be used to find the vulnerabilities on the web servers and web applications. If it is possible, source code analysis is recommended as it will help finding most of the potential vulnerabilities. Analysis programs like OWASP's WebScarab, Firefox's Web Developer Toolbar, Greasemonkey and the XSS Assistant can be used [12]. For details of adding vulnerability analysis to SDLC, see [13]. Black box (automated) testing is chosen if testing is implemented from outside and the source code of the web application is not reachable. Automated web application vulnerability testing reveals only some of the vulnerabilities and there are challenges that need to be solved

[14] [15]. Programs like Nikto, Wapiti, Paros proxy, Burpsuite can be used for black box testing. For a detailed list of security test tools, see [16]. Web systems should periodically be scanned whose generated reports will help the staff to fix the holes before an attack occurs. Vulnerability Analysis Systems should work cooperatively with the Web Awareness System and the IDS.

- **System Monitoring:** Corporate web servers should constantly be monitored for abnormal activities. SNMP agents can be used to collect network traffic, CPU usage, memory usage and process statistics for further analysis.

Training/Testing

Enterprise networks need to have some guidelines and standards for their web security infrastructure. Web application developers and web server administrators need to be informed about the security threats and the countermeasures. The system consists of the following:

- **Workshops:** Meetings can be arranged with the web application programmers. The aim is to show the importance of secure coding principles. Secure coding practices should be made. Input validation and output filtering must be focused on. It's recommended to form software libraries for input/output control for each language used. Also some attack/penetration tests should be implemented to show what an attacker can do by using web application vulnerabilities.
- **Training Portal:** An Intranet portal can be established for related secure coding best practices of the enterprise. Some do and don'ts can be given with examples. Also web server configuration guidelines and standards can be presented here.
- **Test Servers:** Web applications can be put on a security enhanced server and security teams can try to find the weaknesses in the applications. Source code analysis and black box testing can be implemented on these test servers. For implementing penetration testing against web servers and databases, see [17].

Detection

There should be mechanisms to detect abnormal activity on the network. The system should consist of the following:

- **Intrusion Detection:** One or more Intrusion Detection System (IDS) can be deployed on the critical segments of the network. The IDS should have network awareness capability which is obtained from the network awareness component. The IDS configuration and rules should be optimized for web security. Also host based IDS can be deployed. The integrity of the critical system files should be monitored by programs like Tripwire [18].
- **Log Control:** Server logs should be sent to the management (coordination) centre for analysis. Server logs contain web server (such as apache) logs or web application firewall (such as mod_security) logs. Both access and error logs should be analyzed since some of the legitimate actions on web servers can give information about the attacks.
- **Honeypot:** Honeypot systems can be installed to learn possible new vulnerabilities and collect information about attacker characteristics. Enterprise specific web applications with fake data can be deployed on this honeypot. Honeypot systems should be monitored constantly for attacker activity. Implementation details and results of such a system are given in [18].

Prevention

Prevention and risk reduction systems should be used where appropriate. The system should consist of the following:

- **Access Control:** Access to the web servers should be controlled and restricted by firewalls and/or access-lists. For example, access from outside must be permitted to the specific public servers. The number of public servers should be kept in minimum and those servers should preferably be located in the DMZ (Demilitarized Zone) or in a separate VLAN (Virtual LAN). Network access to the ports associated with the database services should be restricted [12]. The databases should preferably be kept in a separate server. Access to intranet and extranet web servers should be given to the selected network segments. Enhanced authentication mechanisms should be used to reach those systems.
- **Server Local Security:** Guidelines for securing web servers and databases should be applied. It must be ensured that all systems are patched up to date [12]. Web application firewalls (such as mod_security) can be installed on each web server.

- **Reverse Proxy/Web Application Firewall (WAF):** Some systems may not be secured because of hardware/software restrictions or there can be problems in reaching their system administrators. Those web servers should be put behind a reverse proxy or a web application firewall (WAF). The reverse proxy should be configured to have filtering capabilities which will provide security for the web servers.

Coordination Centre

Each system/technique described above, has specific strengths and also weaknesses. These systems should work together to achieve the best solution. The coordination of the systems can be implemented in a central system which is called the Management Centre. The aim of the management centre is:

- Optimizing the system's performance
- Reducing false alerts and focusing on real alerts
- Knowing the system's vulnerabilities and focusing on threats to those vulnerabilities

Implementation

In Ege University Campus, several web security enhancements are on process. This section aims to tell what has been implemented and the results. The main focus is on network/web server awareness and training. When this project had started, the situation was as follows:

- All of the faculties and most of the departments had separate web servers and those servers were administered by the local staff. Most of the administrators were not informed about web server security.
- Most of the sites preferred to use free software portals. These software portals had several security holes and most of the administrators hadn't fixed them.
- Web programmers who prepare campus wide information systems had not enough knowledge about secure coding.
- Some of the used web applications didn't have enough documentation and some of the programmers were not reachable.
- There were some network devices (camera systems, wireless devices, printers ...etc) which had their web services open without password authentication or had default passwords.
- The network management group hadn't got a detailed current view of the enterprise wide web system.

The following have been implemented which will be explained widely in the following subsections:

- Standardization and Reorganization
- Web System Awareness
- Training and Vulnerability Testing

Standardization and Reorganization

Standardization and reorganization process is as follows:

- The security policy is changed and security needs of the web environment are focused on.
- The number of web servers is reduced for better management. A web server with backup facilities is located and the departments are encouraged to use this server. Open source web application firewall (Mod security) is installed on the web server and rules from Gotroot (http://www.gotroot.com) with exclusions are used. Configuration is optimized and logs are monitored with the Modsec Security Console.
- The servers are collected in a separate vlan and access to this network is controlled and logged. This vlan will be behind a web application firewall (Mod security) which runs as a reverse proxy. The test implementation of this system is still going on.
- The campus programmers are preparing a software web portal which will be used as a template in all departments. This portal will satisfy the needs of the departments and also be more secure.
- Logs and alerts of the web servers and IDS are collected in a central database. The data is analyzed and statistics are collected. Alerts are summarized and false alerts are reduced. The configurations of the systems are modified using the information found by the awareness systems.

Web System Awareness

Hybrid approach is preferred for web system awareness in the Ege University Campus Network. NMAP and AMAP are used for active analysis and Snort IDS is used for passive analysis. Snort is optimized for web attacks

in order to operate efficiently. Snort only analyzes the traffic destined to the web servers and uses different rule sets for different types of web servers. For details of the implementation see [2]. The Network Awareness System's model is shown in Figure 1. The phases are as follows [2]:

- Periodic Scan: Active analysis is implemented periodically on the network. The network is scanned to find the web servers and detailed information is collected. The information is stored in the Server Info database.
- Update Config: The IDS configuration is updated by using the information from the Server Info database.
- Passive Analysis: Passive analysis process runs all the time on the network. The network data is analyzed to find the unlisted servers and the obtained info (unlisted IP addresses, port info and programming environments used) are put into the Suspect Info database.
- Triggered Scan: The system starts a new scan by using the IP address and ports that are stored in the Suspect Info database. If the information obtained in passive analysis phase is validated, the IDS configuration is updated. This phase is actually improved version of the periodic scan phase.

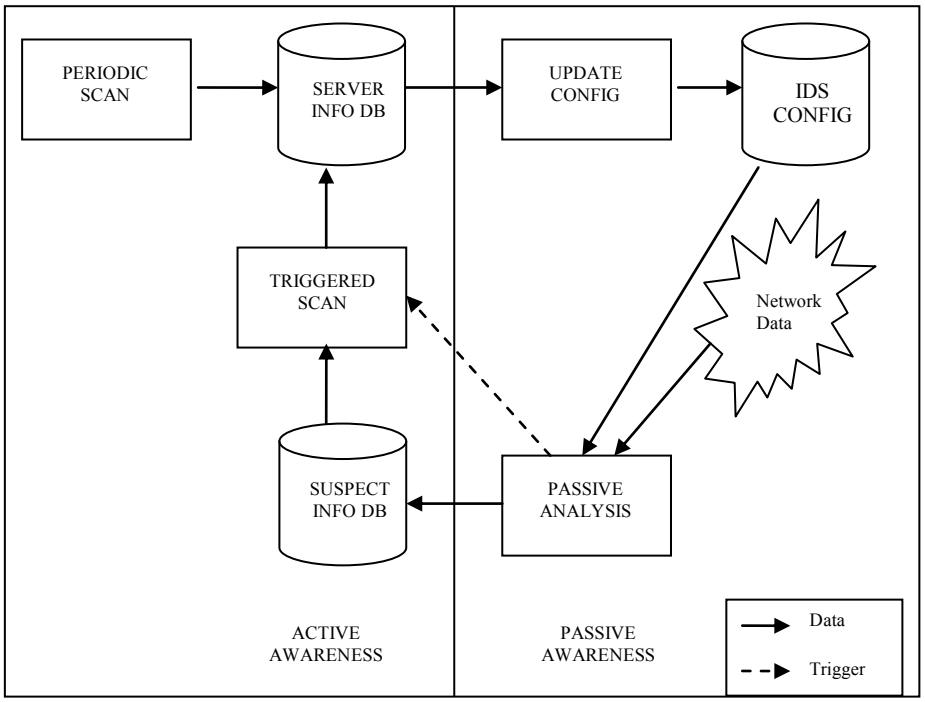

Fig. 1. Active & Passive Network Awareness System Model

It's not always possible to collect enough information about the applications because of the non-standard implementations. The logic of web applications are hard to detect, so human participation and analysis are needed for better results. A module which uses Artificial Intelligence methods is planned to be implemented as a future work.

The web servers are periodically scanned for vulnerabilities. Web application vulnerability analysis is implemented for all domains located on the servers. The recommended system is shown in Figure 2. Nessus vulnerability analysis tool is used for detecting device vulnerabilities. Signature based (Nikto) and anomaly based (Wapiti) tools are used for web application vulnerability testing. Also manual testing is implemented. Reports are generated and system administrators are warned before a compromise occurs. IDS configuration will be changed when a vulnerable system found, that is priority and detailed logging will be activated for the vulnerable systems. Access to the page that has vulnerable content will be blocked when an IPS or Web Application Firewall is used.

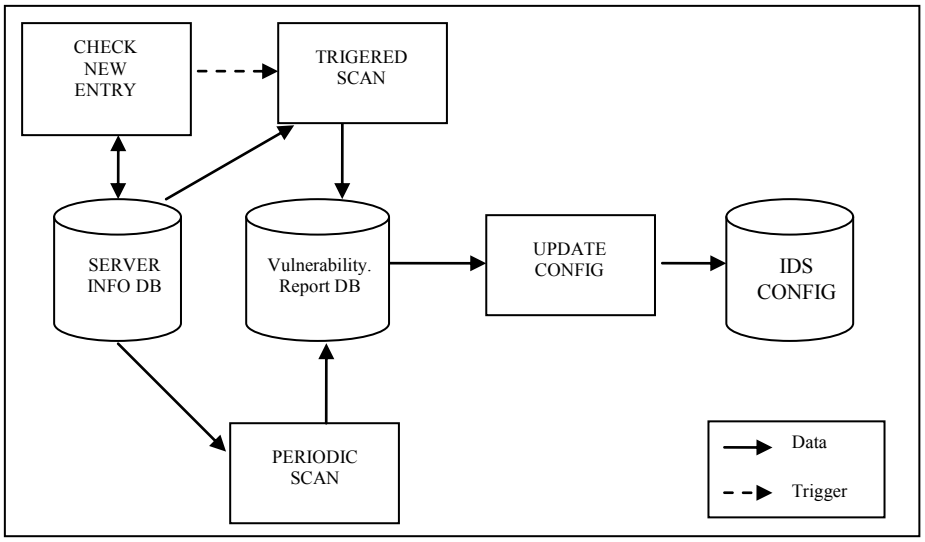

Fig. 2. Vulnerability Analysis System

A portal where web server administrators can see detailed reports about their web systems is being developed in Ege University and there is a plan to expand this implementation to control the critical web servers of the universities in the Turkish Acedemic Network ULAKNET. The portal will give summarized information about the vulnerabilities and recommend actions to solve the problems. The system will also track the changes on the systems. The system's main database schema is shown in Figure 3 and 4. Meanwhile, Inprotect software (http://inprotect.sourceforge.net) is used in the ULAKNET, which automates vulnerability scan with Nessus and Nmap.

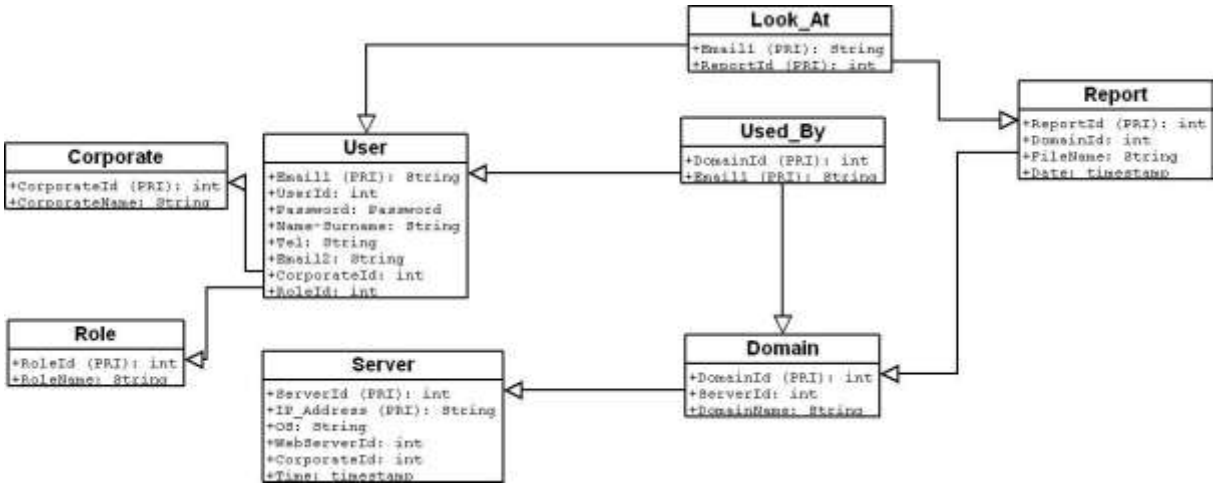

Fig. 3. System Database Schema

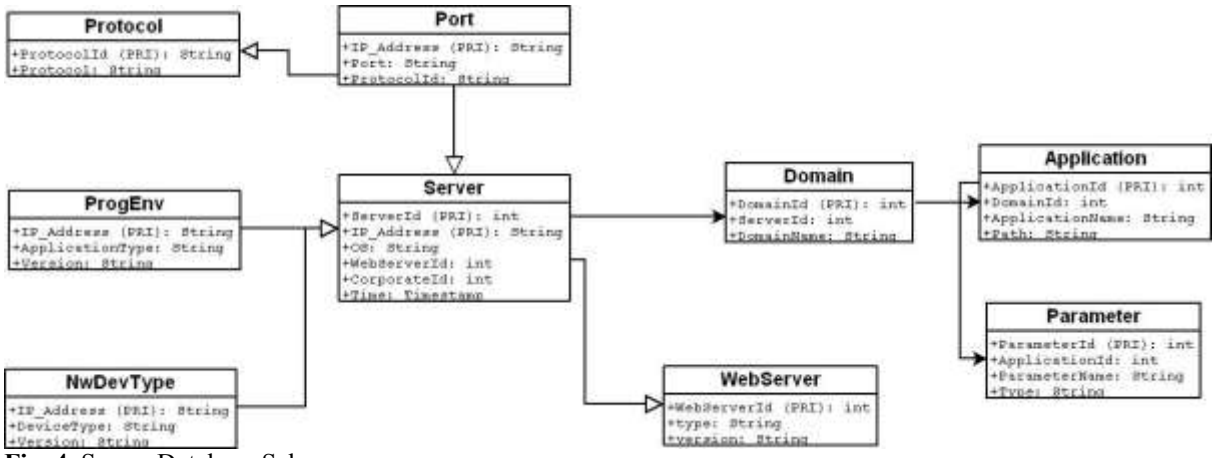

Fig. 4. Server Database Schema

Training and Vulnerability Testing

- A security portal about web security (http://websecurity.ege.edu.tr) is put online. Documents about web security best practices are given in this portal.
- Sites are scanned periodically with vulnerability scanners. Also manual testing is implemented to find the vulnerabilities. Reports are shared with the site administrators and precaution methods are given.
- Some workshops and meetings are made with web application developers and web site administrators of the university. Vulnerabilities in their applications and ways to avoid those vulnerabilities are explained. Security tools from OWASP (http://www.owasp.org) are being used in the workshops for training.
- A session about Web security and Vulnerability Testing is given in ULAKBIM Workshop 2007 where several universities from Turkish Acedemic Network ULAKNET are attended.
- The e-learning portal of Istanbul Technical University will be used to provide free education about web security to the public. The content is available at http://www.ninova.itu.edu.tr/?e=129.
- Web security for the client machines is becoming more critical everyday. Several privacy issues and also phishing attacks must be considered. University members are being informed about these threats and prevention methods are described on the web.
- Honeynets are set in ULAKNET. Some honeypots are being used for detecting web application attacks.

Experiment

In this experiment, the effect of network awareness is shown. Two identical IDS are deployed at the perimeter to detect the web attacks from WAN to Ege University Campus Network. Incoming traffic is mirrored to the IDS by the core switch. Scanner machine is used to scan the network and send information to the TWEBIDS. The deployment is shown in Figure 5. TWEBIDS is the IDS which knows the campus web servers and its rules are optimized. The information comes from the network awareness system.

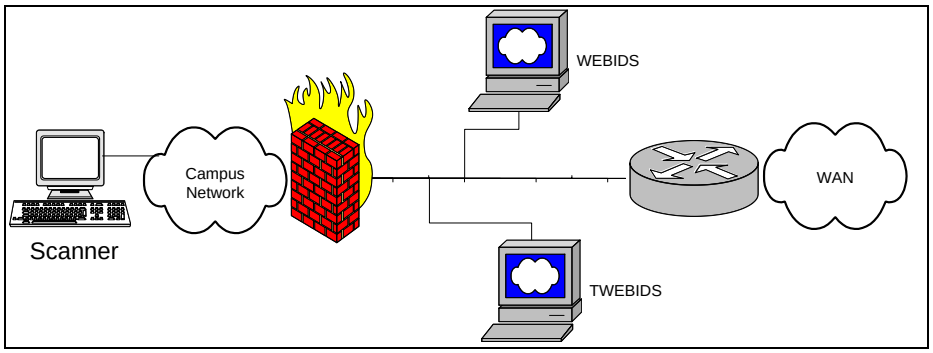

Fig. 5. Test Deployment Schema

The Scanner machine scanned the entire B class IP address range in the first run and 2193 active machines are scanned, analyzed and web server characteristics are discovered. All these discovery process takes about 25 minutes to run. According to the analysis, 199 web servers were detected. 45 (22.6%) web servers use Windows OS, 30 (15%) web servers use Linux/Unix OS. 39 (19.5%) web servers use IIS, 48 (24%) web server use Apache software. Also printers, UPS, wireless devices and also camera systems were discovered. Detecting the wireless network devices that are used without the permission of the Network Management Group is an additional aim of this project. Some systems are undefined and detailed investigation needs to be implemented for these devices. For details see [2]. Also passive analysis is implemented by using Snort IDS and systems that run web server software in ports other than 80 are discovered. These include proxy systems and some other special purpose systems. The IDS configuration is changed according to the information found.

The network is analyzed with Snort having only web rules and “http inspect” engine. The alerts are collected in the snort database and analyzed with the Basic Analysis and Security Engine (BASE). The experiment lasted for a month and logs were collected. The experiment statistics is given in Table 1 and 2. As it is seen from the statistics in Table 1, TWEBIDS produced less and more specific alerts. It’s clearer for a security administrator to analyze the logs of TWEBIDS.

It’s obvious that most of the logs are false alarms. The noisiest rules are coming from http_inspect and web application activity. Detailed alert classification of TWEBIDS is given in Table 3. Detailed analysis of the IDS logs show that most of the logs are because of non standard implementations in the enterprise. The IDS mostly

warns that there may be a misuse and it will warn each time that pages with non standard implementations is accessed. It should be an aim to decrease the non standard implementations in the enterprise. Some noisy signatures which are raised from several source addresses can be filtered or can be given less priorities. Such configurations should be done carefully. It can be a good practice to investigate new IDS alerts which hadn't occurred before. It is always a good practice to search for source IP addresses which caused the most alarms and also with different signatures. In a previous test that lasted for a week, totally 37,677 alerts were collected in TWEBIDS. Most of the logs (5,897) were from a unique IP to our student information system. The logs were between 16:01 and 16:48; 136 different signatures were raised. That was surely a vulnerability scan.

	WEBIDS	**TWEBIDS**
Total Number of Alerts	902,151	92,046
Source IP Address	79,419	17,010
Destination IP Address	106	106
Unique IP Links	87,062	10,657
Unique Alerts	112	99

Table 1. Alert Classification

	WEBIDS	**TWEBIDS**
http_inspect	700,162	23,214
Web application attack	8,251	7,523
Web application activity	185,078	52,748
Attempted Recon	8,398	8,304
Trojan Activity	173	173
Attempted DOS	54	50
default-login-attempt	7	7
attempted-user	28	27

Table 2. Alert Classification

	Total Alert	**Signature**	**Source Address**	**Destination Address**
Http_inspect	23,214	6	5,389	25
Web application attack	7,523	37	8,937	46
Web application activity	52,748	19	3,248	19
Attempted Recon	8,304	28	858	76
Trojan Activity	173	1	20	11
Attempted DOS	50	1	14	2
Default-login-attempt	7	2	3	3
Attempted-user	27	1	2	5

Table 3. Detailed Alert Classification of TWEBIDS

Conclusions and Future Work

Sometimes security measures can make all the system look complex and difficult to manage. As Dan Sullivan says, no "best practice" will turn IT management into a mechanical process. This study aims to reduce the attack surface area as much as possible and aims to make the security administrators work easier. In this work, network/web server security awareness and staff training is focused on. Manageable and easily modifiable open source system is used for this purpose. The system is modular and each module will enhance web security. Some modules may be omitted according to the network characteristics.

Web systems and web services are becoming more complex than ever with the use of the new technologies. It's obvious that detection and prevention mechanisms will not be able to detect the new threats. The importance of the secure coding must be emphasized. Programmers should be trained and secure coding must be included in the software development life cycle (SDLC). Implementing input validation and output filtering will prevent most of the attacks. It is also very important to know the web system assets. Web system awareness enables us to have detailed and up-to-date information of the enterprise web system. Web information gathering is implemented by using a hybrid analysis approach. The current work extends the related work with an efficient hybrid method which is specific to the http protocol. The proposed solution enables to have an up-to-date view

of the web server infrastructure. A network aware IDS prototype is implemented where the configuration is customized by using the web server information which results in few and more specific logs.

As a future work, Artificial Intelligence and log prioritization are planned to be added to this system. Web application vulnerability analysis will be focused on. Prioritization can be set by using the information found by the vulnerability scanners. Different IDS instances with different configurations and priorities can be run for the vulnerable systems until the system is patched. Also audit and forensics modules are planned to be developed. As intrusions often consist of more than one step, it's also important to actively monitor consecutive intrusion attempts.

References

[1] Karaarslan E., Tuglular T., Sengonca, H.: Enterprise Wide Web Application Security: An Introduction, EICAR 2004 (2004)
[2] Karaarslan E., Tuglular T., Sengonca, H.: Does Network Awareness Make Difference In Intrusion Detection of Web Attacks, ICHIT 2006 (2006)
[3] OWASP: OWASP Top Ten Most Critical Web Application Security Vulnerabilities, http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#OWASP_Top_10_2007 (2007)
[4] Hughes E., Somayaji A.: Towards Network Awareness, Lisa 2005 (2005)
[5] Lee D.W.: HMAP: A Technique and Tool for Remote Identification of HTTP Servers, http://seclab.cs.ucdavis.edu/papers/hmap-thesis.pdf (2001).
[6] Web server/application Fingerprinting, http://www.webappsec.org/projects/threat/classes/fingerprinting.shtml
[7] Lee D., Rowe J., Ko C., Levitt K.: Detecting and Defending against Web-Server Fingerprinting, 18th Annual Computer Security Applications Conference (ACSAC '02) p. 321 (2002)
[8] Montigny-Leboeuf A.D., Massicotte F.: Passive Network Discovery for Real Time Situation Awareness (2004)
[9] OWASP: Secure Coding Principles, http://www.owasp.org/index.php/Secure_Coding_Principles
[10] OWASP: OWASP Guide to Building Secure Web Applications, http://www.owasp.org/index.php/Category:OWASP_Guide_Project
[11] Lipner S., Howard M.: The Trustworthy Computing Security Development Lifecycle, Microsoft Corporation, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/sdl.asp (2005)
[12] Web Applications, SANS Top-20 Internet Security Attack Targets (2006 Annual Update), http://www.sans.org/top20/#c1 (2006)
[13] Curphey M., Araujo R.: Web Application Security Assessment Tools, IEEE Security & Privacy (2006)
[14] Grossman J.: Challenges of Automated Web Application Scanning – "Why Automated scanning only solves half the problem", Blackhat Windows 2004, http://www.whitehatsec.com/presentations/challenges_of_scanning.pdf (2004)
[15] Grossman J.: 5 challenges of web application scanning, http://jeremiahgrossman.blogspot.com/2006/07/5-challenges-of-web-application.html (2006)
[16] Peine H.: Security Test Tools for Web Applications, IESE Report-Nr. 048.06/D, A Fraunhofer IESE Publication, (2006)
[17] Whitaker A., Newman D.: Penetration Testing and Network Defense, Cisco Press, ISBN:1-58705-208-3, 2005
[18] Riden J., McGeehan R., Engert B., Mueter M.: Know your Enemy: Web Application Threats, Using Honeypots to learn about HTTP-based attacks, http://honeynet.org/papers/webapp/ (2007)