

Enterprise Wide Web Application Security: An Introduction

Enis Karaaslan

Ege University International Computer Institute, Turkey

About The Authors

Enis Karaaslan is a research assistant at Ege University having his PhD. He is also working as the campus network manager of Ege University. On his academic research, he is working on intrusion detection systems focusing on the application level and multi-layer security. Other research interests include network analysis and security.

Mailing address: Information and Communication Technologies Centre, Network Management Group, UBE building 1st Floor, 35100 Bornova, İzmir / TURKEY; Phone: +90(232) 342-32-32/222; Fax: +90(232) 343-55-42; E-mail: enis@bornova.ege.edu.tr

Tugkan Tuglular

İzmir Institute of Technology, Department of Computer Engineering, İzmir, Turkey

About The Author

Tugkan Tuglular received his Bs and Ms degrees in Computer Engineering at Ege University, in 1993 and 1995, respectively. He has received scholarship and worked for COAST Lab., Purdue University, under Prof. Dr. Spafford between 1996 and 1998. He received his Ph.D. degree in Computer Engineering at Ege University in 1999. Currently, he works at Izmir Institute of Technology as Assistant Professor. His research interests presently include intrusion detection and security policies.

Mailing Address:

Department of Computer Engineering, Izmir Institute of Technology, Gulbahce Koyu, Urla, Izmir, TURKEY; (90) 232 750-6505, tugkantuglular@iyte.edu.tr

Halil Sengonca

Ege University Computer Engineering Department, Turkey

About The Author

Prof. Sengonca is a supervisor and lecturer at Ege University Computer Engineering Department. He has supervised various PhD, MSc and BSc thesis. He has been lecturing the following courses: Software Engineering, Advanced Computer Programming, Database Management Systems, Systems Analysis, Client Server Systems and Security, Management of Software Development Projects. Prof. Şengonca has published 40 papers in various journals and conferences and 2 books on Computer Science and Databases.

Mailing address: Ege University, Department of Computer Engineering, 35100 Bornova, İzmir / TURKEY; Phone: +90(232) 388-72-21; Fax: +90 (232) 339-94-05; E-mail: sengonca@staff.ege.edu.tr

Descriptors: *web security, application level firewall, web application firewall, web application proxy, web application gateway, web crypto gateway, web antivirus gateway, intrusion prevention system, web application security tools.*

Enterprise Wide Web Application Security: An Introduction

Abstract

Nowadays, enterprises should ensure that their Internet security foundation is solid not only through conventional network level solutions but also through application level protection. More and more attacks are likely at the application level due to the fact that web services emerging rapidly without security considerations and network level solutions allow their connections tunnel through.

There is a need for new technologies which are deployed between the users and web applications. These web security technologies should inspect the inputs, not only data but also code, from users and take prevention measures before the attack happens. They are called “Web Application Security Tools”. In this paper, such tools are presented and their deployment strategies are discussed along with some inspection scenarios.

Introduction

There are increasing number of attacks, which occur at the application level. The traditional firewalls can not stop these types of attacks because they can only permit or deny a given protocol and these kinds of attacks run on a permitted protocol like http.

Sometimes confusion can happen with different usages of the word “application”. For a system administrator, a web server can be called as an application, but for an application developer a web server is an infrastructure. As a definition, an application is a piece of code that is designed to perform a specific function for the user or for another application (SANCTUM, 2003). When we speak about applications, we mean client/server systems running on the network; Intranet, Extranet and mostly Internet. These applications are running on application level protocols such as ftp, smtp, http.

The World Wide Web is used more by networked businesses day by day. Applications mostly run on web servers so protection against attacks on web servers is critical. Especially as ecommerce is based on web technologies and money transactions occur, security risks must be taken into consideration seriously. There are increasing number of intrusion attempts that target http (default port 80) which is left open by traditional firewalls for web browsing. These intrusions are not always attacks initiated by a hacker, also include viruses and worms which can cause high damage.

Web applications enable user’s interaction with the web site, transacting and interfacing with all back-end data systems. The code enables the user to access internal data and perform transactions. The web application components are shown in Figure 1 (SANCTUM, 2003).

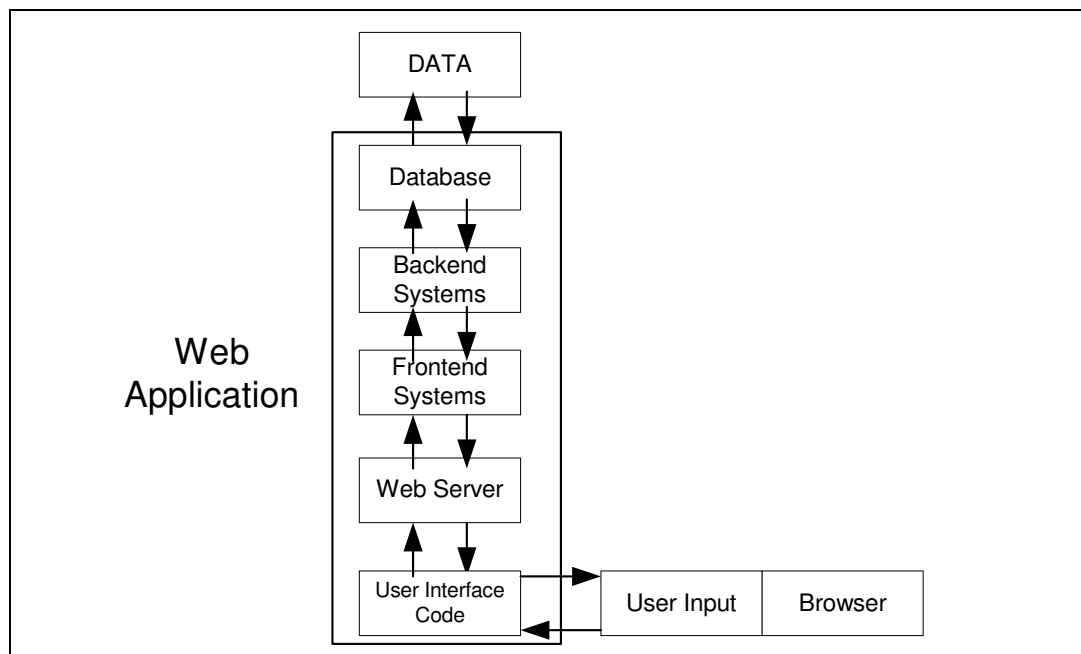


Figure 1. Web Application Components

For web applications to be secure; first of all, the web server's operating system and the web server software must be well configured and well patched. Administrators should be aware of new patches about used software. For details about securing web servers, following references can be used (CERT, 2000a; CERT, 2001; Stein & Stewart, 2002; Nomad, 2002). A scanner like Nessus can be used to scan web servers for known weaknesses (NESSUS, 2003).

When building applications for use on web, technologies like ActiveX, JavaScript, Visual Basic Script, Java Applets and other active content components are used. The executables built with these technologies can carry malicious mobile code viruses or hack scripts (BLUECOAT, 2003). Actually no web application technology can be stated as invulnerable to the inevitable discovery of vulnerabilities that affect its users' security and privacy (OWASP, 2002). Secure programming principles should be applied when programming. For details about secure programming, following references can be used (OWASP, 2002; Burke, 2003; Wheeler, 2003; Grossman, 2001; Panko, 2003). Security testing should not be omitted. For details, see (Hoglund, 2003).

Application level security attacks are possible, as security is not taken into consideration while coding. The main reasons can be given as:

- Programmers are not aware of the threats. They often have no secure programming practice (Burke, 2003).
- Performance considerations,
- In order not to increase coding time,
- Development tools do not prevent simple security bugs such as buffer overflows (Hoglund, 2003).
- Customers buying the software do not demand security measures and technical credible security audits.
- The software which are used in the systems have many components, many developers.

The security threat is increasing day by day because:

- Systems are now having millions line of programming code.
- There exists large number of interactions between different components of a dynamic web site.
- Diversity of the platforms and diversity of the technologies (Burke, 2003).
- New technologies are emerging and not sufficiently tested.
- Mobile devices with mobile applications are becoming more popular.
- More connections and devices are used.

Many of the web attacks can be performed simply by using a web browser. There are also some attacks which require intimate knowledge of underlying system but they are rarely seen. Such attacks aim potential vulnerabilities in the web application layers which are presented in Figure 1. For details, see (Kolodgy, Christiansen & Burke, 2002; SANCTUM, 2003). The web based threat summary can be summarized in a Table 1 (SANCTUM, 2003).

Table 1

Web Based Threat Summary

Threat Category	Description	Consequence
Code Scanning Server/Client	Browsing source code	Learn Vulnerabilities
Cookie Poisoning	Changing cookie content	User impersonation
Hidden Manipulation	Changing hidden HTML fields value	eShoptlifting
Forceful Site Browsing	Use URL address line	Access sensitive data
Third Party Misconfigurations	Default or improper software configuration	Access OS or data
Identified (Known) Vulnerabilities	Published vendor bugs	Access OS, crash server/ application/ database, access sensitive data
Buffer Overflow	Overflow field input	Access sensitive data, crash site/application
Debug Options & Backdoors	Change code setting	Access code/application as developer or admin
Parameter Tampering Server/Client	Removal or alteration of expected parameter fields	Access OS, control application at OS level, site defacement
Cross Site Scripting	Use URL Meta Code to insert Trojan code	Server-side exploitation, access sensitive data
Application DOS	Invalid data input	Crash server/application

These attacks are mostly implemented by using meta-code or invalid data inputs. With these attacks, the attacker can gain info about the system and can gain access also to data and resources that is outside of the application scope.

Enterprise networks have many different web applications running inside. In many of these systems, nobody knows how many different web applications are present and how they interact with each other. Security assessments of all these source codes are mostly not implemented.

There is still a need for other security measures, which will prevent attacks before intrusion or attack occurs. The reasons can be classified as:

- There exists large number of interactions between different components of a dynamic web site,
- Applications running on a company are often bought from another company and no source code is present.
- Traditional security tools are not designed to address the web application as a whole, including how different pieces of application interact with each other (SANCTUM, 2003). Firewalls are effective at blocking network-level attacks, but not at blocking application level attacks.
- Intrusion Detection Systems (IDS) do not sufficiently reduce the risk of attacks against applications. Detection and alerting is not enough, as rapid prevention mechanisms are needed.

Existing network level security tools are not sufficient to secure web applications. There is a requirement for new solutions which will be called as “Enterprise Wide Web Application Security Tools” in this paper.

Enterprise Wide Web Application Security Tools

These prevention mechanisms are deployed between the users and the web application servers in enterprises. They check the validity of user inputs before passing them to the servers. These new intrusion/attack prevention measures to be used in enterprises can be classified as follows:

- Application Level Firewall
- Web Application Firewall (Web Application Proxy)
- Web Application Gateway
- Web Crypto Gateway
- Web Antivirus Gateway
- Intrusion Prevention Systems

These solutions will be efficient, if a security policy of web applications are present. Most of the time there is no policy available. Default configurations or automated tools are used to form a policy. These tools monitor the traffic and automatically generate a policy. These policies though are not always the best ones, a security manager has to work on logs, generated policy and company’s network characteristics to generate a better policy of his own.

Application Level Firewall

Application level firewalls are proxy systems, which examine network traffic at the application-protocol level and can enforce protocol syntax and filter specific protocol commands and content. These systems support multiple applications and web is just one of them (Fratto, 2003). Although this is not a new idea, more efficient and more powerful products are now available as commercial products.

Application level firewalls are candidates for substitution to the traditional firewalls. Traditional firewalls available today are using stateful inspection, for that reason they are also called stateful packet-filtering firewall. This inspection technique monitors TCP session state and drop/reject packets that are not part of the current session or are out of state with a current session. Application proxies can look deeper into sessions and can make drop/pass decisions based on application-protocol headers or in the application payload (Fratto, 2003).

These firewalls are most frequently architected as reverse proxies and also do network layer firewall protection as well. IDC calls these type tools as “Application Shield”. (Kolodgy, Christiansen & Burke, 2002)

Although application level analysis is a very important power, working as a proxy and performing analysis reduces overall performance significantly. **Fratto (Fratto, 2003) run performance tests on hardware supplied by the manufacturer that is supposed to handle up to 1 Gbps of traffic.** According to the performance tests applied in Fratto’s work (Fratto, 2003), none of the application level firewalls could come near 1 Gbps of traffic. As an example,

Checkpoint Firewall NG FP3 ran at 766 Mbps with stateful packet filtering but dropped to 122 Mbps when running as an application proxy.

Application level firewalls rarely look into the protocol payload during examination. However in web applications, HTTP data and form/fields have to be analyzed for security reasons. Therefore, application level firewalls provide partial protection for web applications.

For performance reasons, it may not be suitable to deploy an application level firewall at the perimeter. If such a deployment is to be performed, careful configuration is critical.

Web Application Firewall

Web Application Firewalls (Web Application Proxies) are proxy systems that are placed in front of the web server(s). These systems inspect all web traffic and try to prevent attacks before reaching the web server and applications. These systems are available as software and need to be installed on server machines. These products are adaptable in Web environments and also have support for proprietary products like OWA (Outlook Web Access), Microsoft Frontpage.

A security policy for each application is based on the information extracted from that specific application. There is no defined set of known attack patterns, there is set of known application behaviors. Any different input or any different behavior of that application may mean a security threat (SANCTUM, 2003).

Proxy products come with “adaptive learning” or some common rules. As tough this properties are extremely helpful, these can lead to some incomplete protection. As stated in Forristal’s work (Forristal, 2003), web application firewalls are much complex than firewalls and need to be configured carefully and be tuned perfectly for the deployed site’s characteristics. According to the test in which products’ automated learning/policy generation tools used, none of the products could stop all of the attacks (Forristal, 2003).

Considerations when choosing one include ease and flexibility of configuration. The proxy should give the administrator the power and flexibility to control whatever is wanted. Other important aspect is logging power. The alert logs should be easy to monitor and should give enough amount of information. Some proxies have the option to log entire packet as well.

In a recent work (Scott & Sharp, 2003), a scalable structuring mechanism is presented in which security policies can be stated explicitly using Security Policy Description Language (SPDL-2). These policies are implemented on a security gateway, which is located between the Web server and client systems. This system filters web traffic using the policies installed. It should be noted that this system can also run as a host based solution on the web server machine.

Web Application Gateway

Web Application Gateways (WAG) are web proxy solutions implemented on high-performance devices, which examine outbound (outgoing) web application traffic. These devices have better performance by the use of Application Specific Integrated Circuits (ASIC) and special hardware. They are deployed inline on the application data path and ensure users conform to established security policies and prevent attacks coming from ports 80 (HTTP) or 443 (SSL).

These devices are shown as an alternative to Host Intrusion Prevention (HIP) products which will be explained in “Intrusion Prevention Systems” part of the paper. According to the analyst report (Ogren, 2003a), it is stated that these devices have a lower false positive rate than many of the popular IDS products. As tough it is an emerging technology, it is stated that IDS market is moving to WAG products (Ogren, 2003a).

This technology has some usage benefits of being a hardware box solution. There is no need to bother installing and configuring an operating system (OS). It should not be forgotten that security systems may have some inconsistencies with the underlying OS. WAGs are easy to deploy as only a basic security policy is have to be set and then the system is up and running.

In an another recent analyst report from Yankee Group (Ogren, 2003b), it is predicted that web application security will be one of the hottest segments of the security industry over the next 5 years and HIP products will become essential components in all Web application deployments, WAGs will become common for medium and large-scale enterprises. The market acceptance of WAGs will cause HIP products focus on operating system platform and custom application protection. WAGs will be incorporate with firewalls and security service switches.

It’s predicted that WAGs will incorporate operational performance features such as load balancing, content caching, and content filtering as optional product line extensions. WAGs will also have to evolve to handle new web technologies like XML/SOAP (Ogren, 2003b).

Web Crypto Gateway

Web servers can be configured to use encryption and can form encrypted channels between web clients. The technologies used today for encrypted web connections are SSL (Secure Socket Layer), IETF’s TLS (Transport Level Security), S/HTTP (Secure Hypertext Transport Protocol), and SET (Secure Electronic Transaction) (CERT, 2000b;OWASP, 2002). If the network to be protected has no web server using these protocols, these protocols can be blocked for inbound (incoming) traffic at firewalls. However, it should be noted that there is still trouble for outbound (outgoing) traffic.

Most of the attack prevention measures mentioned in this paper do not or can not look into encrypted web traffic. Some of the new attack types (CryptoHack) take advantage of encrypted Internet traffic to bypass these prevention measures. Malicious code, viruses and attacks can enter the sites using encrypted protocols. There are new products like Microdasys

SCIP Proxy that aim to inspect this encrypted traffic. A dedicated device (Web Crypto Gateway) can be employed to inspect this encrypted traffic and apply the security policy (MICRODASYS, 2003).

Web Antivirus Gateway

Traditional enterprise web virus scanning and active-content security is implemented by two different approaches (BLUECOAT, 2003):

- **On-box Approach:** Virus protection solutions are installed on application level firewalls. This kind of systems can work fine in light traffic environments. Scaling is possible but the performance/price ratio is not satisfactory.
- **Off-box Approach:** Virus protection is implemented on a separate server. Firewall sends all web content to the separate server and selecting which contents to be scanned is left to the virus scanning server. In this architecture, virus scanning server implements the CPU intensive virus scanning. Checkpoint's CVP (Content Vectoring Protocol) which can route email and web content, can be given as an example.

Web Antivirus Gateways are separate devices, which combine web caching and CVP-style architecture. There are examples of these kinds of devices implementing ICAP (Internet Content Adaptation Protocol) Cooperation Architecture. ICAP protocol that is used is conceptually similar to CVP, but has been optimized for web protocols. For details, see (Elson & Cerpa, 2001). However ICAP is not a standard yet and seems that it still needs working on.

Intrusion Prevention Systems

Since protection is necessary in today's networks and detection is not satisfactory, Intrusion Protection Systems (IPS) are becoming more popular as they can block attacks in real time. IPS systems are infact inline systems with intrusion detection capability which help stopping certain types of attacks. It should be noted that IDS' main job is to monitor and audit. IPS' main job is access control and policy enforcement. IDS can show if your network is secure or not (Conry-Murray, 2003).

We can classify IPS by Installation type, as Host Intrusion Prevention (HIP) and Network Intrusion Prevention (NIP). When talking about protecting web, we must not forget that intrusions can occur in a very short time and the defense system must respond rapidly. For this reason prevention is needed more than detection. HIP products installed on web servers, can be more effective in web security. HIP ensures that applications access only required host resources and services; lets applications perform only authorized actions. HIP products are well suited to block attacks at the operating system (Conry-Murray, 2003).

The accuracy of IPS products is still a concern as blocking legitimate traffic is possible. In order to decrease false positives, IPS must be configured to understand the network environment thoroughly. IPS use multiple detection methods in order to improve their ability to detect attacks. IPS will succeed if underlying intrusion detection increases in accuracy.

The latency formed by analysis of incoming data and applying policies must not be long, process must be near wire speed. Most IDS vendors are now moving to IPS hardware appliances. Hardware appliances which have security specific ASICs are used to speed up the process.

Deployment

Integration of tools is necessary for a successful enterprise-wide web application security. Therefore, network design for security is developed by the following methodology. First, web servers are classified by whom they are used; “Internal Use”, “External Use” and “Hybrid Use”. This classification leads us to threat perception which then enables us to determine the deployment strategy for securing web applications. Deployment strategies for all types are given in this section.

In addition to web application security tools, deployment will include the following systems:

- **Firewall:** Today’s networks require a firewall at the perimeter. Traditionally it is a stateful packet-filtering firewall, implementing security on network and transport level (OSI layers 3 and 4). As we mentioned earlier, application level firewalls are also possible with a performance cost.
- **Intrusion Detection Systems:** As a second line of defense, Intrusion Detection Systems should be deployed not only for detection attacks missed by firewalls but also for checking all security applications whether they are hacked or not.
- **Hardware Appliances:** Depending on the number of web servers and throughput of data communication, web application security devices that can handle gigabit speeds must be preferred. For the analysis to be faster, hardware appliances can be preferred.

External Use

Public Web Servers, which are open to all audience is to be protected. These servers can be collected at the DMZ (Demilitarized zone) leg of the firewall. If the firewall to be used is an Application Level Firewall (ALF), most of the application level attacks will probably be prevented. As we mentioned before, most of the ALF do not look into protocol payload and only partial web security can be accomplished. Also the performance will be decreased while working as an application proxy. The network (NW) diagram is shown in Figure 2.

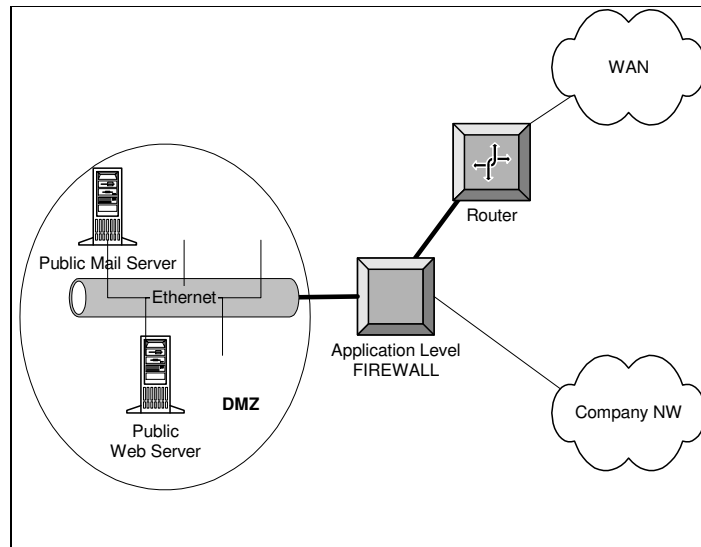


Figure 2. External Use With Application Firewall

In the system suggested, a stateful packet filtering (SPF) firewall is installed at the perimeter. This firewall will deal with all the packets entering and leaving the company's network. It will make the analysis at network and transport OSI layers. It will deal with most of the bulk data and reduce the bulk data as much as it can. This firewall will not look into protocol payloads, therefore will work faster than ALF.

In the system suggested, a Web Application Firewall (WAF) is installed in front of the web servers in the DMZ. This firewall (proxy) will look into web protocol payload and analyze it thoroughly. As the bulk data is reduced by the SPF firewall, WAF will work more efficiently.

If the throughput of web servers does not satisfy the performance requirements, a hardware solution – Web Application Gateway (WAG) can be preferred. An Antivirus Gateway can be installed for content and antivirus checking. IDS can be deployed for auditing purposes. The NW diagram is shown in Figure 3.

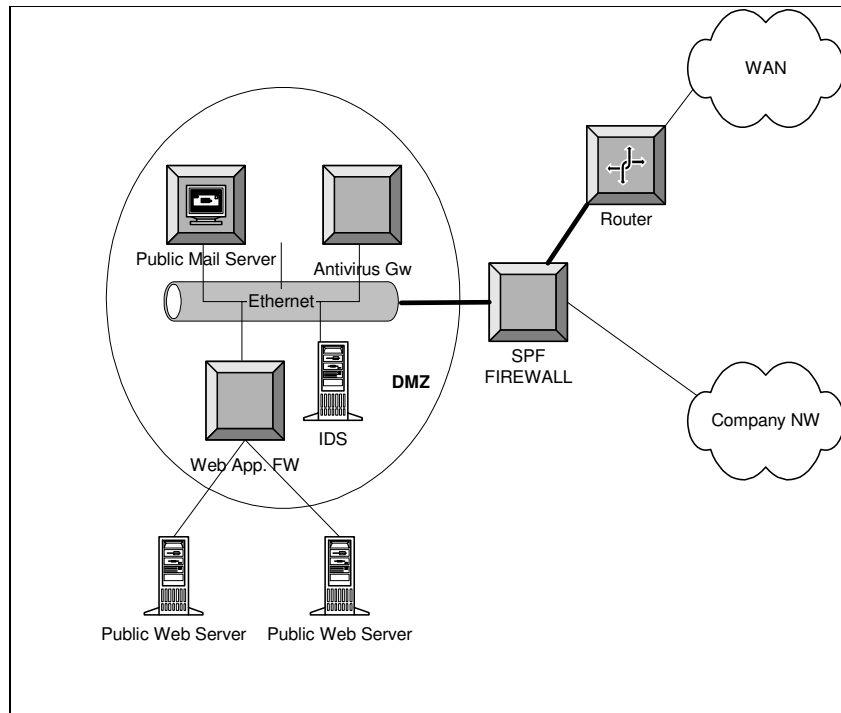


Figure 3. External Use With Stateful Firewall

Internal Use

When talking about internal use; Intranet Web Servers, which are only open to specific group of company users or computers is considered. The main consideration will be protecting services from inside attacks.

If the users of the web applications are specific such as Account Office or Research Department, the basic form of protection should be use of Virtual Local Area Network (VLAN) structure. The web server should be only accessible to specific VLAN(s). By the use of IP level access-lists, access can be controlled and Intranet service can be satisfied. These access-list prevention mechanisms can be installed on OSI layer Layer 3 (L3) type switches. It should be added that this type of security only checks IP addresses and does not make application layer controls.

Being paranoid to live and work long, proxy firewalls can be installed to make the service more secure. These web application (proxy) firewalls will be installed in front of the web servers. Depending on the needs of different web applications, different policies must be installed for each application service. Web servers, which need the same prevention measures can be installed behind a common proxy and can form a Intranet Server Farm. For different needs, application specific separate proxies can be deployed. Correct configuration depending on the needs is critical on performance and security. The system that is suggested is shown in Figure 4.

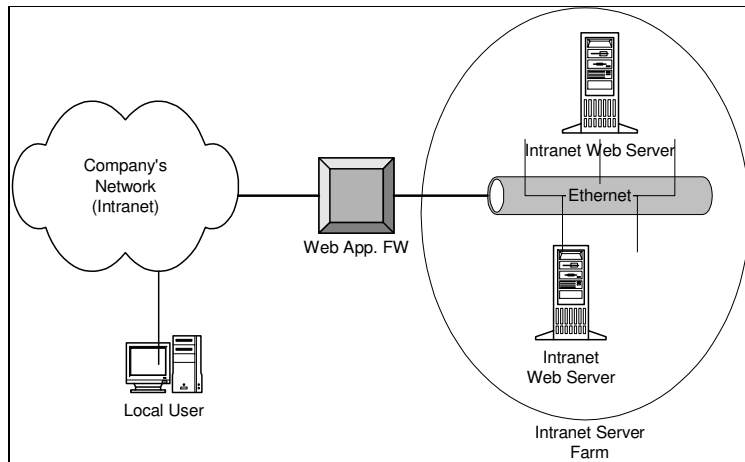


Figure 4. Internal Use

Hybrid Use

As tough optimal is to separate the external use from internal use, sometimes servers for both use have to be installed. These servers can be installed on DMZ or Server Farms depending on the internal or external use frequency. If the servers are used by external users more frequently, it can be installed in DMZ. Also VLAN based access-lists can be used to limit which parts of the local NW are authorized to reach these web servers. The system that is suggested is shown in Figure 5.

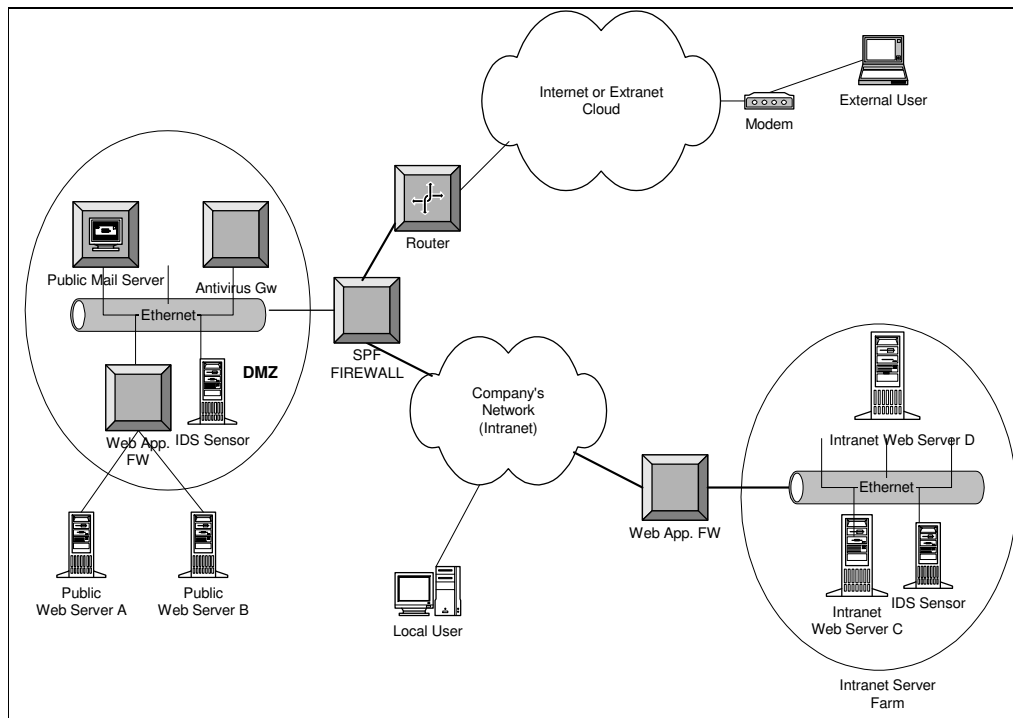


Figure 5. Enterprise Wide Web Security Structure

Inspection

Two inspection scenarios will be given when intrusion is attempted to public and intranet web servers.

Scenario 1

In Scenario 1, an external user aims to access sensitive data by implementing a “Buffer Overflow Attack” on Public Web Server A. For the network under consideration, see Figure 5.

Case1: The system is protected by a WAF.

1. The packet including http request is sent to Public Web Server A from the external user.
2. The http packet is received by the company’s router and routed to the internal network.
3. The packet is inspected by the SPF Firewall and L3 and L4 analysis is done. As http is allowed for Public Web Server A, the request is sent to the WAF.
4. WAF knows the normal behavior of the application that runs on Web Server A, so it discards the request and does not forward it to the web server. The intrusion attempt is logged and alert is produced.

Case2: There is no WAF and no IDS in the system.

Steps 1 and 2 is the same. In the Step3, the packet is sent to the web server. In Step 4, the packet is received by the web server software. The intrusion is unsuccessful if the application designer has programmed the code with inspecting the input carefully. If not, the Buffer Overflow Attack is successful. There is no alert in the system and external user gains access to sensitive data.

Case3: There is no WAF, but an IDS is present in the system.

The steps are the same with Case2. But as an IDS is installed in the system, intrusion is detected, logged and alerted. There are cases an IDS can be fooled by using different encodings so IDS must be configured carefully. Prevention mechanisms will have to be taken manually by the administrators in Case3.

Scenario 2

In Scenario 2, an external user aims to access sensitive data by implementing a “Buffer Overflow Attack” on Intranet Web Server C. For the network under consideration, see Figure 5.

Case 1: The system is protected by access lists and WAF.

1. The packet including http request is sent to Intranet Web Server C from the external user.
2. The http packet is received by the company’s router and routed to the internal network.
3. As the intranet servers are controlled by L3 addresses, the firewall or Intranet Switch blocks the request as it is from an external user. The intrusion attempt is logged and alerted.

Case 2: External user gains access to a company’s machine and makes the attack from there. The system is protected by access lists and WAF.

1. The packet including http request is sent to Intranet Web Server C from the local user which is compromised.

2. The http packet is received by the company's L3 switch. As the request is from a local user, the packet is forwarded to the WAF.
3. WAF knows the normal behavior of the application that runs on Web Server C, so it discards the request and does not forward it to the web server. The intrusion attempt is logged and alert is given.

Case 3: External user gains access to a company's machine and makes the attack from there. The system is protected by access lists only.

1. The packet including http request is sent to Intranet Web Server C from the local user which is compromised.
2. The packet is sent to the web server.
3. The packet is received by the web server software. The intrusion is unsuccessful if the application designer has programmed the code with inspecting the input carefully. If not, the Buffer Overflow Attack is successful. There is no alert in the system and external user gains access to sensitive data.

Case 4: There is no WAF, but an IDS is present in the system.

The steps are the same with Case3. But as an IDS is installed in the system, intrusion is detected, logged and alerted. There are cases an IDS can be fooled by using different encodings so IDS must be configured carefully. Prevention mechanisms will have to be taken manually by the administrators in Case4.

Future Predictions

In the future, we will be seeing hybrid devices implementing web and network security. Devices will probably be hardware appliances with multiple CPU's and ASICs for speeding analyze of network data (intrusion detection). These devices will probably be built on firewalls; working inline, they will be able to sense intrusions and prevent them. There won't be many device types, only one device with different modules. Devices will be designed to enable or disable software modules which will be installed by need. Perhaps there will be hardware modules that can be attached to the device for the implementation specific speedup.

We will also be seeing distributed and cooperative devices and software. The standards will evolve which will enable router, switch, firewall and intrusion prevention mechanisms working together. Two technical committees have been formed under the Organization for the Advancement of Structured Information Standards (OASIS) to develop a common language for describing Web-based attacks and vulnerabilities, and for sharing that data between security devices. These are the Application Vulnerability Description Language (AVDL) committee and the Web Application Security-Extensible Markup Language (WAS-XML) committee. OASIS can be reached by web www.oasis-open.org .

Conclusion

Web based applications are increasing with the evolution of e-commerce. As the applications are designed without giving necessary importance to security, the businesses are vulnerable to web based attacks. Although most of the application attacks can be avoided by using secure programming practices, the fact that the systems have many components of which have

different developers makes securing applications almost impossible. In this work, new measures of prevention for enterprises are presented.

When choosing such a prevention mechanism, the following features have to be satisfied:

- The system must work with high accuracy, not disturbing the normal traffic.
- The system should understand and know each application's behaviors.
- The system should have automatic policy generation feature.
- The system should react and adapt itself when the applications are updated by the developers.
- The system must be scalable. It should be able to scale to heavy usage requirements.
- As web applications are real time systems, the delay should be as short as possible.
- The system should evolve to support for new web based technologies like XML/SOAP.

The systems should be deployed in such a way that overall network performance should not be affected negatively. It should be better if these systems are placed in front of critical application servers only.

Application level prevention mechanisms are the new hot topic of the IT world. We can state the reasons as:

- Application level attacks are becoming more critical.
- Vendors are trying to sell new products, which they classify with different names.

Whatever is the reason, it is obvious that application level security should be considered seriously. It should also be noted that products which prevent application attacks, can only address a portion of web based attacks and users must not be in a "false sense of security" mode. Secure programming practices and security testing should be implemented seriously in all conditions.

References

- BLUECOAT (2003). Web Security for the Enterprise: How to increase protection against port 80 threats. Blue Coat System's White Paper [On-line]. Available: http://www.bluecoat.com/downloads/whitepapers/download_websecurity.html
- Burke R. (2003). Web Application Security. Lecture Notes [On-line]. Available: <http://josquin.cti.depaul.edu/~rburke/courses/f03/ect582/notes/w8/lec1106.ppt>
- CERT (2000a). Securing Network Servers. CERT Security Improvement Module [On-line]. Available: <http://www.cert.org/security-improvement/modules/m10.html>
- CERT (2000b). "Configure the Web server to use authentication and encryption technologies, where required". A practice from the CERT Security Improvement Modules [On-line]. Available: <http://www.cert.org/security-improvement/practices/p080.html>
- CERT (2001). Securing Public Web Servers. CERT Security Improvement Module [On-line]. Available: <http://www.cert.org/security-improvement/modules/m11.html>
- Conry-Murray A. (2003). Emerging Technology: Detection vs. Prevention - Evolution or Revolution? [On-line]. Available:

- <http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId=9400017>
- Elson J., Cerpa A. (2001). The Internet Content Adaptation Protocol, The ICAP Protocol Group, Internet-Draft [On-line]. Available: <http://www.ietf.org/shadow.html>
- Fratto M. (2003). "Application Level Firewalls: Smaller Net, Tighter Filter". Network Computing, pages 57-68
- Forristal J. (2003). Proxies Add a Protective Shield. Network Computing. pages 50-58
- Grossman J., Arquette L. (2001). "Web Application Security - In theory & practice". DefCon9. [On-line]. Available: http://www.whitehatsec.com/presentations/Defcon9_2001/defcon9_presentation2001.ppt
- Hoglund G. (2002). Bad Software. CTO, Cenzic Inc. [On-line]. Available: <http://opensores.thebunker.net/pub/mirrors/blackhat/presentations/bh-usa-02/bh-us-02-hoglund-software.ppt>
- Kolodgy C.J., Christiansen C., Burke B. (2002). Web Intrusion Protection: Defending Web Servers and Applications. IDC [On-line]. Available: <http://www.idc.com>
- MICRODASYS (2003). "Digital Certificates and Encryption – Security Implications for Enterprises" Microdasys White Paper [On-line]. Available: <http://www.microdasys.com>
- NESSUS (2003). Nessus Remote Security Scanner, The Nessus Project, [On-line]. Available: <http://www.nessus.org/>
- Nomad S. (2002) The Unofficial Web Hack FAQ [On-line]. Available: <http://www.nmrc.org/faqs/www/index.html>
- Ogren E. (2003a). Web Application Gateways Bolster Security, CSO Analyst Reports, Yankee Group [On-line]. Available: <http://www.csoonline.com/analyst/report883.html>
- Ogren E. (2003b). Assessing What's Hot in Web Application Security, Yankee Group [On-line]. Available: <http://www.csoonline.com/analyst/report1586.html>
- OWASP (2002). The Open Web Application Security Project (OWASP) Guide to Building Secure Web Applications and Web Service. [On-line]. Available: <http://www.owasp.org/documentation/guide>
- Panko R. (2003) Corporate computer and network security. ISBN: 0-13-038471-2. Prentice Hall.
- SANCTUM (2003). Anatomy of a Web Application [On-line]. Available: <http://www.sanctuminc.com/solutions/whitepapers/>

- Scott D., Sharp D. (2003). Specifying and Enforcing Application-Level Web Security Policies. IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No. 4
- Stein L.D., Stewart J.N. (2002). The World Wide Web Security FAQ, Version 3.1.2 [On-line]. Available: <http://www.w3.org/Security/Faq/>
- Wheeler D.A. (2003). Secure Programming for Linux and Unix HOWTO. [On-line]. Available: <http://www.tldp.org/HOWTO/Secure-Programs-HOWTO/>