



IIS MIGRATION AND SECURITY THE APACHE SOLUTION

INTRODUCTION

Several highly publicized worms, such as Code Red, Code Red II, and Nimda target specific Microsoft Internet Information Server (IIS) security shortcomings. Organizations using have experienced a number of security and reliability issues related to these recent worm attacks. The operational resource drain of these issues have led them to consider other alternatives to IIS, particularly for Web servers exposed to the public Internet. The fall out of those worms is not limited to just the IIS servers; but the aggressive scanning and propagation attempts of infected machines, inside and outside, causes a significant drain on human and network resources.

Executives and IS management of the affected organizations are faced with the risks and escalating costs of ownership of IIS. New viruses and worms are constantly targeting IIS, and each new worm or virus typically means another IIS security update from Microsoft. Each new attack is likely to be responsible for additional server and network downtime during the period while no fix is available and subsequently while the update is obtained and applied. The loss of revenue and customer confidence from public-facing Websites when new attacks strike, and the labor and opportunity costs from repairing damages and installing updates on dozens or hundreds of Web server can easily dwarf other operational costs.

This white paper provides architectural, execution, and development alternatives for current IIS deployments. It focuses on a three-step phased process:

- Immediate: Protection for current IIS servers and applications
- Short Term: Migration of current IIS applications to Apache with minimal application changes
- Longer term: Alternative application development and execution environments

For many organizations, the solution to this ever-increasing Total Cost of Ownership is to migrate away from IIS. For this to be a viable solution, the new Web server platform must offer compatibility with the current applications in use on IIS, offer much higher security and reliability, and be widely supported by Independent Software Vendors (ISVs) to ensure a wide range of development and application alternatives. The Apache Web server, combined with value added products and enterprise class support from Covalent, meets these requirements.

The Apache Web server is an open source product of the Apache Software Foundation, and is the most popular server on the Web. Over 16 million sites currently run Apache, and over 1 million sites per month are being added. Even with this wide exposure, Apache remains among the most secure and reliable Web servers available, with no history of the security issues and vulnerability to worms and viruses exhibited by other servers. Covalent Technologies, Inc. specializes in the distribution, enhancement, and support of the Apache Web server with a complete suite of products intended for use in enterprise-class applications.

REMOVING IIS FROM PUBLICLY-ACCESSIBLE NETWORKS

IIS has become a frequent target for hackers because of its well-known security issues and wide distribution, combined with poor administrative habits of some Web site owners that facilitate the spread of attacks. IIS Web sites accessible from the public Internet are at greatest risk because of the ready availability of automated

scanning tools, and because of the actions of some of the worms themselves. For organizations concerned about these issues, one of the first steps is to protect systems running IIS from direct contact with the Internet.

Objectives

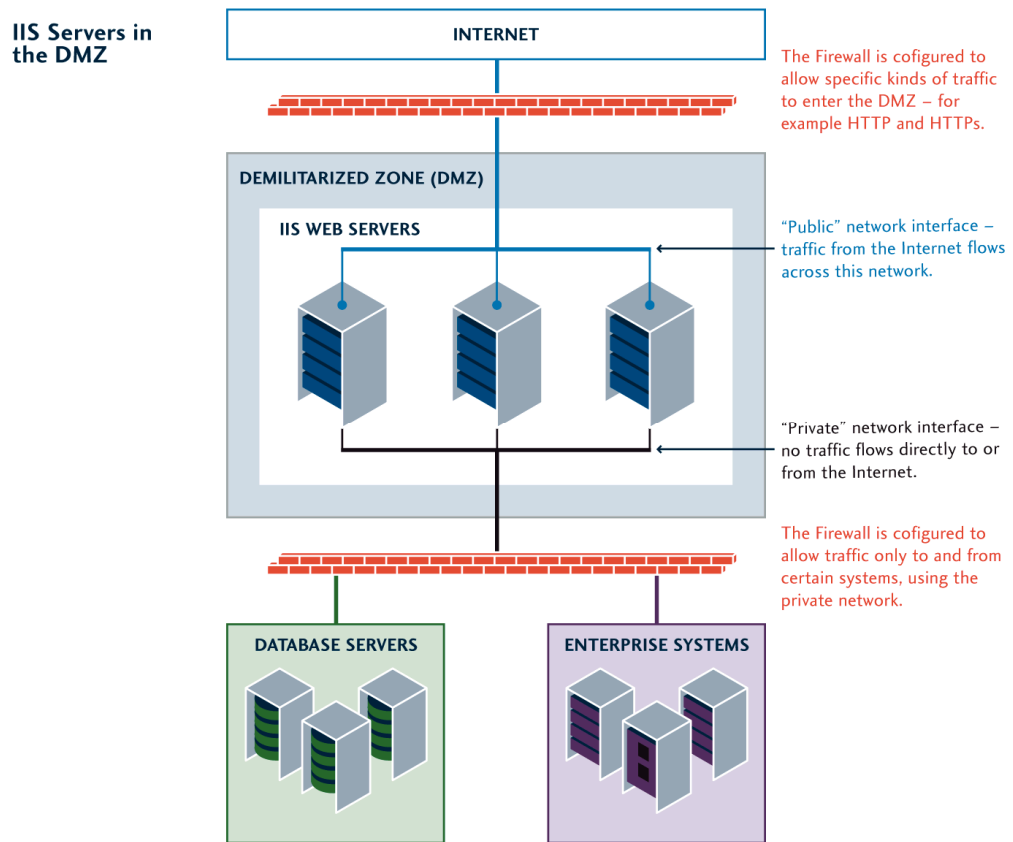
The objectives of this effort are as follows:

- Remove systems running Microsoft IIS from the publicly accessible portion of a network, typically outside the firewall protected area or in the DMZ network segment
- Conceal the internal network structure from view
- Provide a screening mechanism to help prevent known attacks against IIS
- Provide a filtering mechanism to protect IIS against a common group of attacks: buffer overruns in the URL, the headers or the post block.

Network architecture

Typical networks today are implemented using multiple tiers, with each tier being protected by an increasing more restrictive set of firewall configurations. As shown in Figure 1, Web servers are usually placed in a Demilitarized Zone (DMZ), which is the segment of the network that is accessible to the public Internet. Other resources, such as connections to enterprise systems and databases are often connected to an internal network; connections between the DMZ and internal network are tightly controlled.

Figure 1

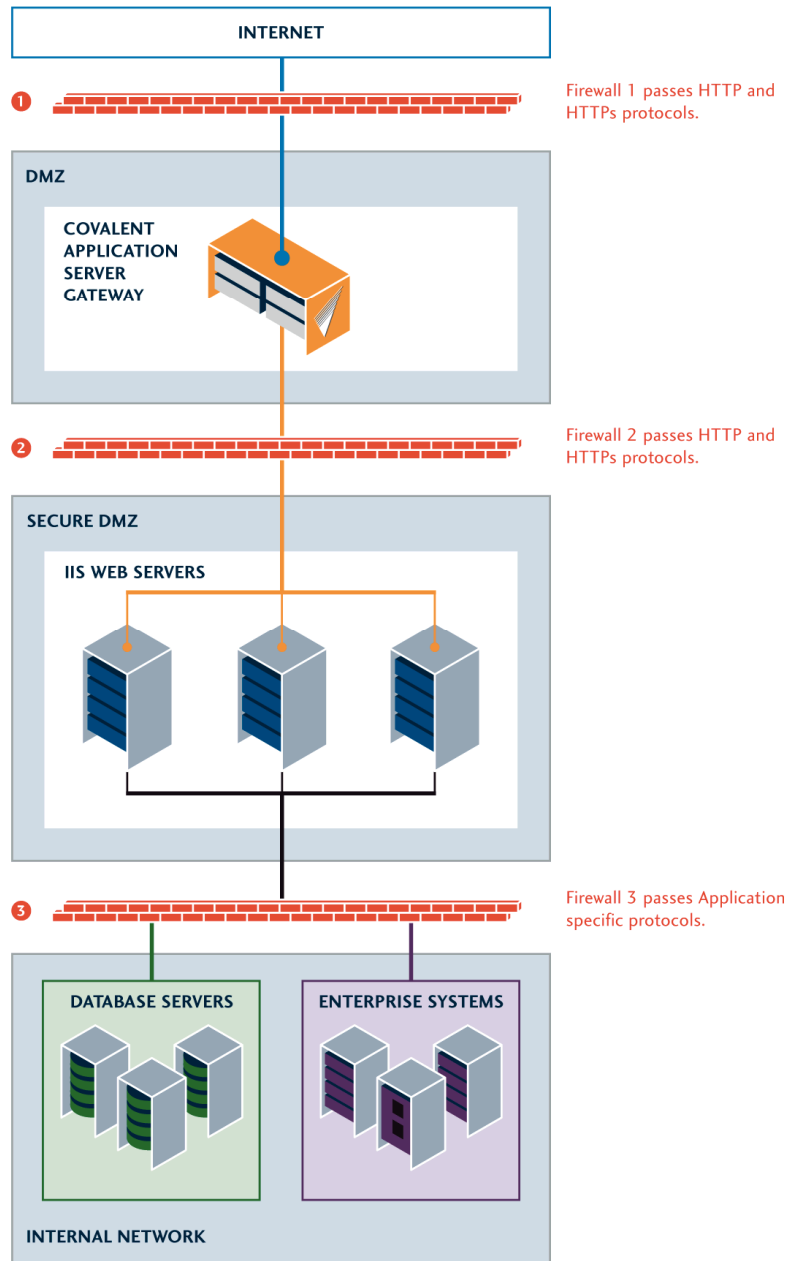


One step to improving the security of IIS is to remove those systems from the DMZ, placing them in a more protected network environment. As shown in Figure 2, this involves placing a 'gateway' system or systems in the DMZ, with all requests and responses to IIS being routed through that gateway. The gateway requires software such as Covalent's Application Server Gateway (ASG) product. ASG provides a secure, high

performance solution that not only forwards requests to the IIS servers, but also provides substantial protection against the most common IIS attacks.

Figure 2

Securing IIS Servers



This architecture allows the internal network (Firewall 3 in the diagram) to be configured in a way that significantly limits the type of network traffic that is allowed to flow from the DMZ into the internal network and vice-versa. In this example, the gateway system uses two network interfaces: one that connects to the public Internet, and one that connects only to the secure section of the DMZ. Because the IIS systems are no longer exposed to the public Internet, they are screened from some of the more damaging attacks. Although the network diagrammed looks fairly complex, many networks today use routers and firewall software from Cisco and other vendors that can be reconfigured to support this architecture relatively easily with minimal changes to the physical network. Of course, other network architectures can also be effective.

Implementing ASG

ASG is an Apache module that implements a gateway solution based on the Apache Web server. In this application, ASG receives all requests ultimately bound for the IIS systems. Based on the URL of the request, it routes the request to the appropriate IIS system. This routing is determined by the ASG configuration files, and can be changed and updated as necessary. Responses from the IIS systems are routed back through the ASG, preserving the isolation of those systems from network clients.

Individuals who have targeted specific Web sites and networks spend significant effort in trying to determine the structure of the network, host names and IP addresses, the presence of specific routers and gateways, and many other pieces of information. Information returned in URLs and HTTP headers can often reveal important pieces of the puzzle to a skilled hacker. Therefore, part of any secure solution should be to conceal this information.

ASG accomplishes this by rewriting a portion of the HTTP header to return the name of the ASG server rather than the name of the actual IIS server, concealing details of the internal network. Other HTTP headers can also be manipulated, both to further conceal the internal network architecture and to maintain compatibility with various custom applications running on IIS.

Typical attacks on IIS are based on a so called “buffer overrun”. This type of attack – whose principles are well understood, as it is one of the oldest type of compromise, essentially consists of sending very long requests or requests with series of unusual characters to the server. The long request ensures that a buffer is filled beyond capacity, causing the unusual characters to be stored in a place where they might be executed. These characters typically represent a series of processor and OS specific machine code instructions which trigger further internal errors or security shortcomings of IIS, its configuration, or the underlying OS. These errors can allow attackers to take control over aspects of the system on which IIS is executing, or to crash IIS itself.

ASG resolves this issue by limiting the length of the URL and header passed to the IIS server and by ‘normalizing’ the URL passed on. Known attacks on IIS require that the ‘payload’, which is the executable code that actually exploits a specific IIS vulnerability, be included either as part of the URL itself, or as large block in the HTTP header area. A standard configuration parameter within ASG which limits this length to safe values, and prevents a malicious code block from being passed to IIS. A further safeguard – is the normalization or ‘%’ escaping (named after the %2e characters commonly seen in URL’s) places further constraints which make passing arbitrary binary/byte code with enough control harder.

Issues Resolved

- Potentially vulnerable IIS systems are removed from the publicly available network, placing them instead in a more protected network
- The structure of the protected network is concealed from the outside world
- Firewalls protecting the internal network may be able to be configured in more restrictive, secure manner
- Attacks against known vulnerabilities in IIS can be prevented from reaching the IIS system.

TRANSITIONING IIS APPLICATIONS TO COVALENT APACHE

Objectives

The ASG solution protects against some of the most severe IIS issues, particularly those that depend on depositing and executing malicious code on the IIS server. However, other attacks that are intended to crash or impair the target Web server, rather than exploiting it for other attacks, may not be stopped with this strategy. Because IIS is continually targeted with new variations of this type of attack, it is very difficult for any solution to remain dependably current. ASPs executing in the IIS environment have been demonstrated to be especially vulnerable, along with standard functions in the ISAPI. Even ASG can’t completely protect against these kinds of attacks.

Many organizations have decided to move away from IIS entirely because of these ongoing issues. This section describes alternatives for moving applications off of the IIS platform on to Covalent's Apache platform. For organizations that have made a decision to move some or all of their current applications from IIS to Apache, the objectives are:

- Assure a high degree of compatibility with the current IIS execution environment
- Provide a development environment and tools that are similar as possible to those currently in use, to maintain developer productivity during the transition
- Maintain a migration path for future development that may not depend on proprietary Microsoft technologies.

IIS Application Compatibility

Web applications written to the IIS server can use a number of technologies for creating and executing the Web application. While there are literally dozens of alternatives, the most popular approaches are:

- Application Server Pages: A programming and execution environment that allows applications written in Visual Basic or JScript (Microsoft's version of the standard ECMA Script) to create dynamic Web pages, accessing resources both on the server system and on the network.
- Internet Server Application Programming Interface (ISAPI): An execution environment that allows program modules written in C or other compiled languages to work with the IIS's API.
- Application Servers: Vendor specific environments, such as BEA WebLogic and IBM WebSphere, that provide a complete Java programming and execution environment for complex applications
- Packaged applications: Applications such as SAP and PeopleSoft that use IIS as a front end Web server.
- In-house applications, often written using the standard Common Gateway Interface (CGI) and languages such as PERL, PHP, or C.

Apache offers direct compatibility for many application servers and packaged applications, including all of the major vendor environments. For these situations, the transition to Apache should be straightforward, since the only step required will be the configuration of the Apache server to work with the vendor specific plug-in, and the transfer of any static content currently served by the IIS server.

ISAPI Applications

Applications using ISAPI as their programming interface can take advantage of Apache's built-in compatibility with the ISAPI. Apache includes as part of the standard definition an ISAPI compatibility module that provides application developers with a compatible API for Windows NT/2000 based systems. For applications written using ISAPI, the application will need to be re-compiled to execute in the Apache environment.

No source code changes should be required, although it should be noted that the Apache ISAPI module does not support ISAPI filters or Microsoft specific additions to ISAPI for asynchronous file operations. Other tasks would include configuring the Apache server to recognize the ISAPI module, and transferring any static content.

ASP Applications

ASP applications are perhaps the most popular programming and execution environment for IIS servers today, primarily because of the use of Visual Basic as one of the underlying programming languages, and because of the extensive development environments (such as Visual Studio) available from Microsoft and other vendors. ASP includes not only the ability to generate dynamic Web pages using Visual Basic Script (VBScript) or JScript, but also includes ActiveX Data Objects (ADO) that allow the Visual Basic or JScript program to access any ODBC-compliant data source, such as relational databases.

To execute ASP applications on Apache, the entire programming environment, including VBScript, JScript, and ADO must be available and compatible with the Microsoft distribution. ChiliSoft (a division of Sun Microsystems) provides this capability today. The ChiliSoft ASP product (CASP) provides a compatible execution environment across a wide variety of UNIX platforms, as well as Windows NT/2000. CASP includes

VBScript (licensed from Microsoft), JScript, and ADO in its distribution. CASP is also compatible with the major development environments, including Visual Studio.

Moving current ASP applications from IIS to Apache by using CASP should require no changes in either the application or in developer behavior. Once the CASP environment is installed and configured, application code and Web page assets can be transferred to the Apache server without changes.

Other Development and Execution Environments

Many current production applications running on IIS use the Common Gateway Interface (CGI) to communicate between an executing program and the IIS Web server. The CGI standard predates IIS, and is widely used across UNIX and other platforms. CGI itself defines that standard of how programs and the Web server interact; it doesn't specify what language the program is written in.

While there are a large number of choices for writing CGI programs, the most common in use today are PERL and PHP. PERL is a full-featured scripting language that provides extensive text and data manipulation capabilities that have made it extremely popular with Web application programmers for a number of years. PERL itself is not specific to Web applications, but has had a number of extensions added that make Web application development easier and more reliable. Current applications written in PERL can be easily moved to Apache, as Apache includes `mod_perl`, a full implementation of PERL with Web application extensions. Because `mod_perl` integrates PERL with the Apache server itself, applications often execute more quickly because the overhead associated with the standard CGI method of creating new processes for each page served is eliminated. Minimal source code changes would be required to effect this transition.

PHP, with its built-in database access capabilities and its C language-like syntax, is rapidly growing in popularity, and is often considered as a direct alternative to ASP when selecting a Web application environment. PHP, like PERL, is inherently cross-platform, and minimal if any changes should be required to move an application from IIS to Apache using PHP. PHP includes a method to mimic ASP tags in HTML pages, making it compatible with visual user interface tools. `Mod_PHP` uses the same method as `mod_perl` to eliminate the overhead associated with standard CGI processes.

Issues Resolved

Because of Apache's popularity and deep support from a wide variety of vendors, moving the vast majority of applications from IIS to a compatible environment on Apache is straightforward, with little or no change required in the application. This transition allows organizations to eliminate the issues surrounding IIS security problems, since Apache replaces IIS as the execution environment. Specific solutions mentioned were:

- Nearly all third-party application servers and applications have Apache-compatible plug-ins available today, making the transition from IIS to Apache straightforward
- Using the ChiliSoft CASP product, current ASP applications can be moved from IIS to Apache with little or no change to either the application or the developer's environment.
- Applications written using the ISAPI for IIS can be moved to Apache through the use of the ISAPI-compatible interface provided with Apache
- Applications written for CGI-based environments can most probably be moved to Apache with little or no change, and often experience performance enhancements because of the use of Apache modules to execute the programs.

APPLICATION DEVELOPMENT ALTERNATIVES

While Apache provides compatible application environments for nearly all IIS applications today, many organizations may wish to consider other alternatives that are not based on Microsoft specific technologies. In particular, the use of application environments other than ASP provide organizations with significantly more flexibility in the choice of both hardware platforms and operating systems, and may provide a more reliable application as well.

Objectives

The objectives that organizations will have in considering other application platforms may include:

- Cross-platform support across hardware platforms and operating systems
- Higher reliability of the application
- Well-supported environments from third party vendors and the availability of trained developers
- A well-planned transition from current ASP environments for new and current applications.

ASP Transition

While ASP remains extremely popular as a development and execution environment, many organizations have experienced issues related to the reliability of the underlying execution environment as well as the well-documented security issues. At the same time, for many organizations developer training and the effort required to re-implement existing applications may seem to outweigh the known disadvantages of ASPs.

Regardless of the end execution environment(s) selected, organizations must begin by developing a solid plan for transitioning off of ASP technologies. While the specifics of this plan will vary by organizations, at a minimum it should include a prioritized list of applications to be moved, and an incremental developer-training program to begin the re-training process. Applications may be prioritized in several ways, including security vulnerabilities, exposure to public Internet access, and dependence on Windows specific technologies.

Java-based Applications

One of the most attractive alternatives for many organizations is the use of Java-based environments for the development and execution of Web-based applications. Java has the most desired attributes for Web applications environments, including strong cross platform support, a significant third party application community, widely available developer expertise, and a strong security record. There are several alternatives available for creating Java applications today, with the most common being based on the Java 2 Standard Edition (J2SE) and Java 2 Enterprise Edition (J2EE) specifications from Sun Microsystems.

J2SE Web applications use a set of facilities that allow the creation of dynamic Web pages via Java: Java servlets and Java Server Pages (JSPs). Both of these approaches allow Java code to be executed that can perform tasks such as page navigation, personalization, and access to external data sources. J2SE based applications have capabilities that closely parallel ASPs, with the added advantage of being based on an industry-standard cross platform specification and providing superior scalability based on distributed application principles.

The Apache Software Foundation (ASF) has sponsored a large-scale project called Tomcat that has resulted in a fully production-ready, highly scalable J2SE execution environment. Tomcat is the reference implementation for both JSPs and Java servlets, and is included in the Covalent Apache FastStart product. For applications that do not have to implement complex business logic or transaction semantics, Tomcat provides an excellent alternative to ASPs.

J2EE includes a wide range of additional capabilities, most significantly Enterprise Java Beans (EJBs). EJBs allow complex business rules and access to external systems to be encapsulated within containers; these containers can be connected into a variety of different applications. J2EE environments generally include an application server that is responsible for user session management, application assembly, and high reliability aspects such as clustering, load-balancing, and failover of application components. These application servers are most appropriate for complex, enterprise level applications that must implement complex business transactions.

Issues Resolved

For organizations that have decided to transition away from ASPs, Java-based application environments present a strong alternative based on their cross-platform capabilities and strong industry support. Most organizations will be best served by a well-planned transition, focusing on the most critical applications first. Java-based applications resolve the following issues:

- Wide choice of operating systems and hardware platforms for application execution
- Exceptional third-party support for applications and developers
- Migration path from simpler applications based on J2SE environments such as Tomcat to complex enterprise applications based on the J2EE standard.
- High-performance, high-reliability distributed application architectures for the most demand Web applications

COVALENT TECHNOLOGIES RECOMMENDATIONS

Covalent Technologies, Inc, the leading supplier of Apache based Web technologies, is well-positioned to provide products and services to organizations concerned about IIS security issues and considering a transition away from IIS based applications. While each organization will have unique needs and priorities, Covalent recommends the following overall process:

- Assess your current environment, with emphasis on particularly vulnerable applications such as those exposed to the public Internet
- Provide immediate protection to the most critical applications through the use of gateway products such as Covalent's ASG
- Transition selected applications from IIS, either by using products such as Chilisoft CASP or by porting the application to different execution environments.
- Consider implementing a long-term plan to remove current dependencies on Microsoft specific technologies, moving to industry-standard platforms such as J2SE and J2EE.

Covalent's Professional Services Organization (PSO) can work with you to determine the best immediate and long-term action plan. Depending on your needs, Covalent's PSO can assess your current environment, assist with installation and configuration of Covalent's ASG, and help plan and execute an application transition plan.

Web applications become more critical to every organization every day. Executing those applications in a secure, reliable, and flexible environment will provide significant benefits for every organization; Apache provides precisely that environment, and Covalent adds the products required for enterprise environments.

whitepaper

Covalent
technologies inc.



Covalent Technologies, Inc.
303 Second Street, Suite 375 South
San Francisco, CA 94107
415/856-4200
www.covalent.net

Sales 415/856-4245 or 800/444-1935