Firewall Deployment for Multitier Applications

By Lenny Zeltser

This article examines considerations for deploying firewalls as part of a network perimeter around Internet-facing servers. The discussion focuses on situations that may warrant strict separation of network resources into dedicated subnets, and explains how to enforce access restrictions using firewalls in a way that matches business and technical requirements of multitier applications. The article introduces several network architectures that use a single firewall as well as firewalls deployed one behind another in series, and addresses the strengths and weaknesses of each approach.

Partitioned network architectures can be used to protect multitier applications accessible over the Web. Following the trend of designing applications in an expandable and scalable manner, these applications are often created by using modules that run on different servers and that typically form three distinct groups: presentation, middleware, and data tiers. Let's begin by examining how the architecture of such applications may influence the design the network's security perimeter.

Multitier Applications

By segmenting a Web-based application into several logical tiers, software architects isolate core functional areas into groupings that can be designed, developed, and maintained somewhat independently of each other. The following tiers are present in some way in most Web-facing applications of moderate complexity:

- *Presentation* components are usually adjacent to the Internet and are the only modules directly accessed by end users. Such publicly accessible services are often implemented using Web, DNS, and mail servers. Software running on these servers, operating as part of a unified system, presents the application to users and handles interactions between users and back-end components. Programmable logic of the application at this tier is implemented using mechanisms such as CGI scripts, servlets, JavaServer Pages (JSPs), and Active Server Pages (ASPs) that are used to generate the application's user interface. In addition to interacting with human users, presentation systems may communicate with other hosts across the Web, frequently through the use of protocols such as SOAP, ebXML, and WSDL.
- *Middleware* components execute business logic of the application in response to requests issued by presentation servers on the user's behalf; they are not directly accessed by end users. Such middleware components are usually implemented using application servers such as BEA WebLogic, IBM WebSphere, and iPlanet Application Server, as well as via custom daemon-style programs. Application servers provide an environment within which middleware components can operate. They are based on frameworks such as Enterprise Java Beans (EJBs), CORBA, and .NET. Other servers at the middleware tier provide auxiliary services that collaborate with the application server and may host application-level authentication and authorization mechanisms such as those implemented by Netegrity SiteMinder and Entrust GetAccess.
- *Data* components are typically hosted by database and directory servers that run software such as Oracle Database, and iPlanet Directory Server. They may also be implemented using custom programs and flat files. These are typically the most confidential resources of the application because they maintain sensitive data such as customer records, account information, and monetary transaction statements.

This article discusses applications that define three distinct tiers, but these concepts are expandable to four and more tiers as well. The logical view of a Web-based multitier application is presented in Figure 1. Sample workflow of an application built according to this architecture can be described as follows:

1. An Internet user issues a request via a Web browser to the Web server.
2. The Web server preprocesses the request and relays it to the application server.
3. The application server obtains necessary information from the database, processes the request and responds to the Web server. The Web server, in turn, formats and displays the response to the user.

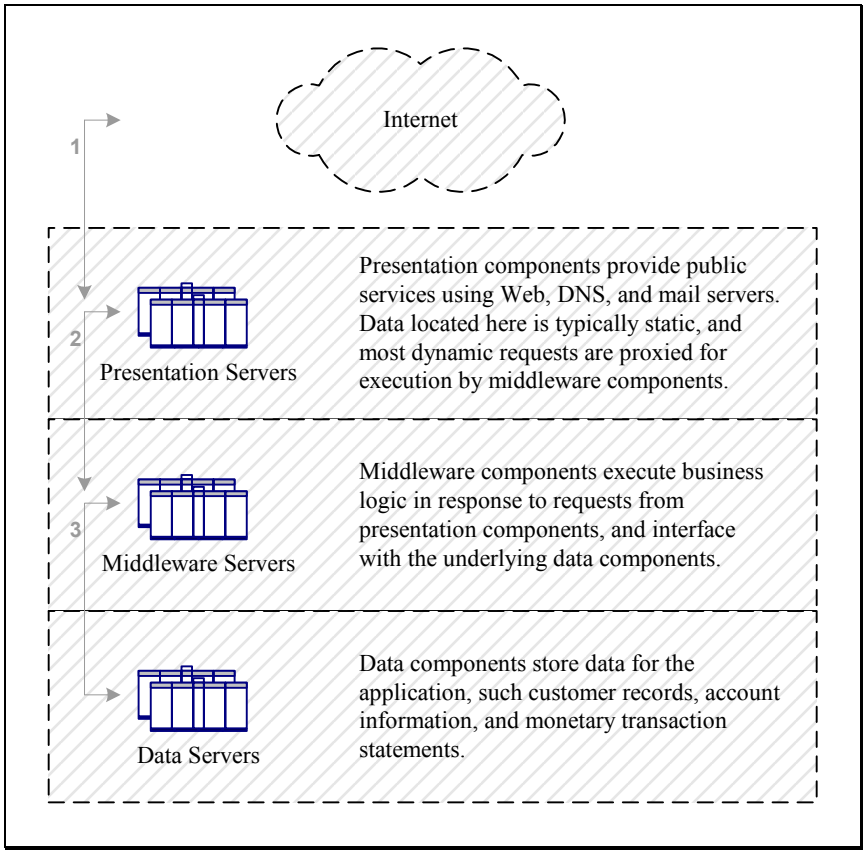

Figure 1 – Multitier Application Architecture

Multitier architecture of the application allows you to design the network in a way that mirrors the grouping of the application's components so that you can segment resources based on their exposure sensitivity and the likelihood that they can be compromised. Let's proceed by examining how to host a multitier application on one subnet behind a single firewall.

A Single Firewall and One Subnet

The idea of resource separation is based on the understanding that network resources differ in the extent of acceptable risk. Risk, when used in this context, is comprised of two factors: the likelihood that a resource will be compromised and the sensitivity of the resource itself. For instance, a well-configured Web server that hosts CGI scripts is more likely to be compromised than one serving only static HTML pages. Deploying a firewall in front of the server tends to decrease the likelihood that it will be compromised, thus decreasing its overall risk exposure.

Similarly, a database server that stores login credentials may warrant additional layers of protection because it is generally more sensitive than the Web server that generates the application's front-end.

Figure 2 illustrates a common network architecture that uses a single firewall to protect components of a multitier application. Here, all servers are hosted on a single subnet, located directly behind the firewall. The firewall, in conjunction with the border router, is responsible for thwarting network-based attacks coming from the Internet. Servers are hardened and are watched over using network and host-based intrusion detection systems. This helps protect the application against attacks that use protocols that are not blocked at the network's border, or that do not come from the Internet. Such defense-in-depth elements are common to all solid designs and are not explicitly shown on the diagram.

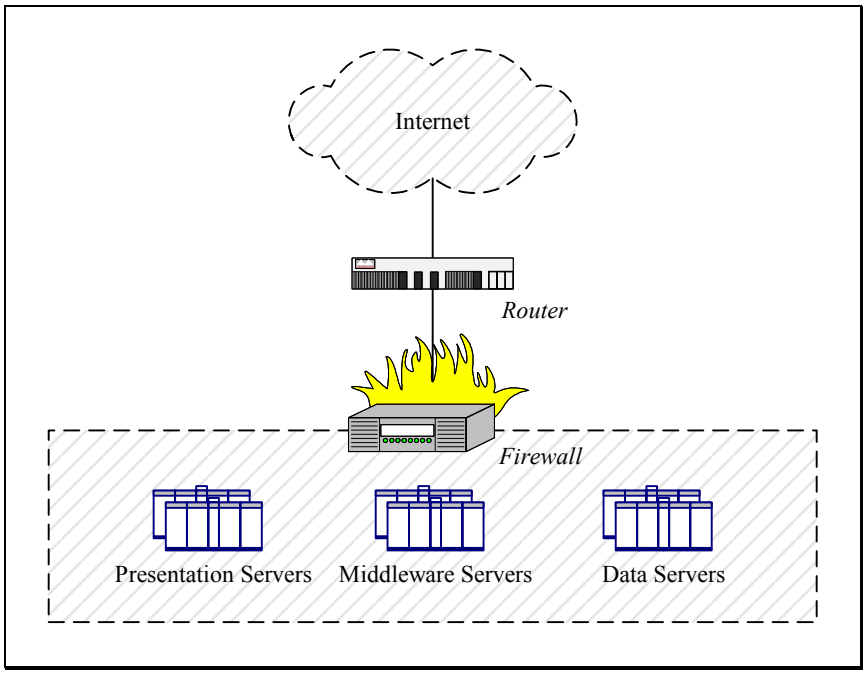

Figure 2 – A Single Firewall and One Subnet

In this design, all servers are hosted on the same subnet, and are warranted equal protection by the firewall that separates them from the Internet. Keep in mind that even though production servers are on the same subnet, in this design they are already separated from the corporate network used by the company's employees for internal operations. This, in itself, may be sufficient to achieve the desired extent of resource separation.

Using a single firewall to protect servers is relatively inexpensive and simple to manage, which is a significant advantage of this solution. You may elect to use this architecture if the expense of deploying and maintaining multiple firewalls is not warranted by the risk of hosting servers on the same subnet. For example, having a single subnet may be justified when further separating the servers does not substantially mitigate a significant risk.

Consider a scenario in which the database server listens on only a single port, used by the middleware server to retrieve data. In this case, placing a firewall between the servers improves security only marginally because this port still needs to remain open. Segmenting the internal network into separate subnets will, of course, help to protect against other types of attacks, but you might need to address other, more significant weaknesses before you decide to deploy additional firewalls or subnets. For environments in which additional separation is appropriate,

let's consider a somewhat more elaborate network architecture that uses a single firewall but that segments the network into multiple subnets.

A Single Firewall and Multiple Subnets

One way to segment the network without introducing another firewall is to use multiple interfaces on the firewall to create several subnets. A design based on this principle is presented in Figure 3, in which the firewall splits the network into three subnets, each dedicated to hosting a particular tier of the application. The firewall is multi-homed, which allows administrators to assign a different security policy to each interface.

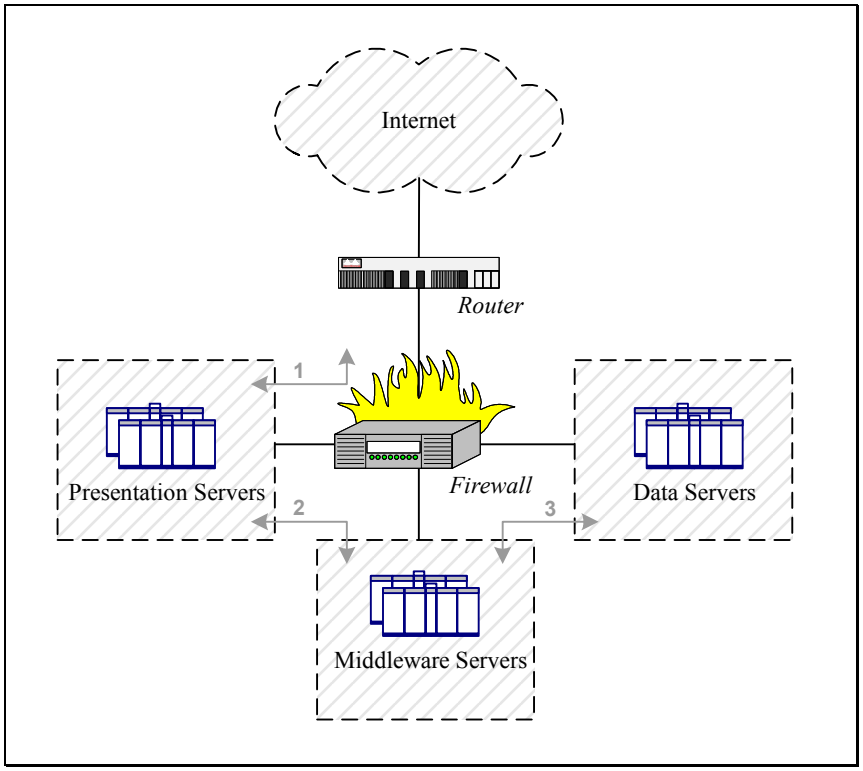

Figure 3 – A Single Firewall and Multiple Subnets

In this configuration, Internet users can directly access only presentation servers, which have access only to middleware servers, which can access only data servers. Juxtapose this design against the architecture of a multitiered application in Figure 1, and you will see that it closely mirrors the roles and requirements of the application's components, and allows a fine-grained control over access to servers in each tier. At the same time, the firewall's rulebase used to implement access restrictions in this scenario is more complicated than one in which all servers lived on the same subnet. This increases the likelihood that the firewall will be misconfigured, introducing its own risks into this design.

Hosting each tier of an application on a dedicated subnet is a powerful technique because it allows network designers to configure the network in a way that closely matches the application's security requirements, albeit at an added cost of maintaining a more complex firewall rulebase and managing servers located on different subnets. This approach, evident in several designs presented in this article, mimics the design of a large ship split into multiple watertight compartments to resist flooding: If one of the sections is compromised, other areas retain a high chance of maintaining their integrity.

Using a single firewall to segment the network is one of the most affordable ways of separating application tiers, but it is not without limitations. A single logical firewall, even if redundant in hardware, presents a single point of failure for the design, especially when it enforces security policy for subnets that host servers of different risk levels. If the firewall is compromised or misconfigured, an intruder could obtain access to all subnets, including the most sensitive segment that hosts data servers. Moreover, the firewall may become a performance bottleneck because it needs to examine traffic passing between all subnets. Let's take a look at an alternative design that uses multiple firewalls to eliminate some of these deficiencies.

Deploying Firewalls in Series

To eliminate the reliance on a single firewall, while still retaining fine-grained control over inter-subnet communications, you can use multiple firewalls to guard subnet boundaries. One such design is presented in Figure 4, in which three firewalls are deployed in series, one behind another. As the sensitivity level of hosted resources increases, so does the number of firewalls located between the Internet and the potential target. In this configuration, tiers of the application are "sandwiched" between firewalls, and a dedicated firewall moderates communications between adjacent subnets according to the application's architecture and the security policy.

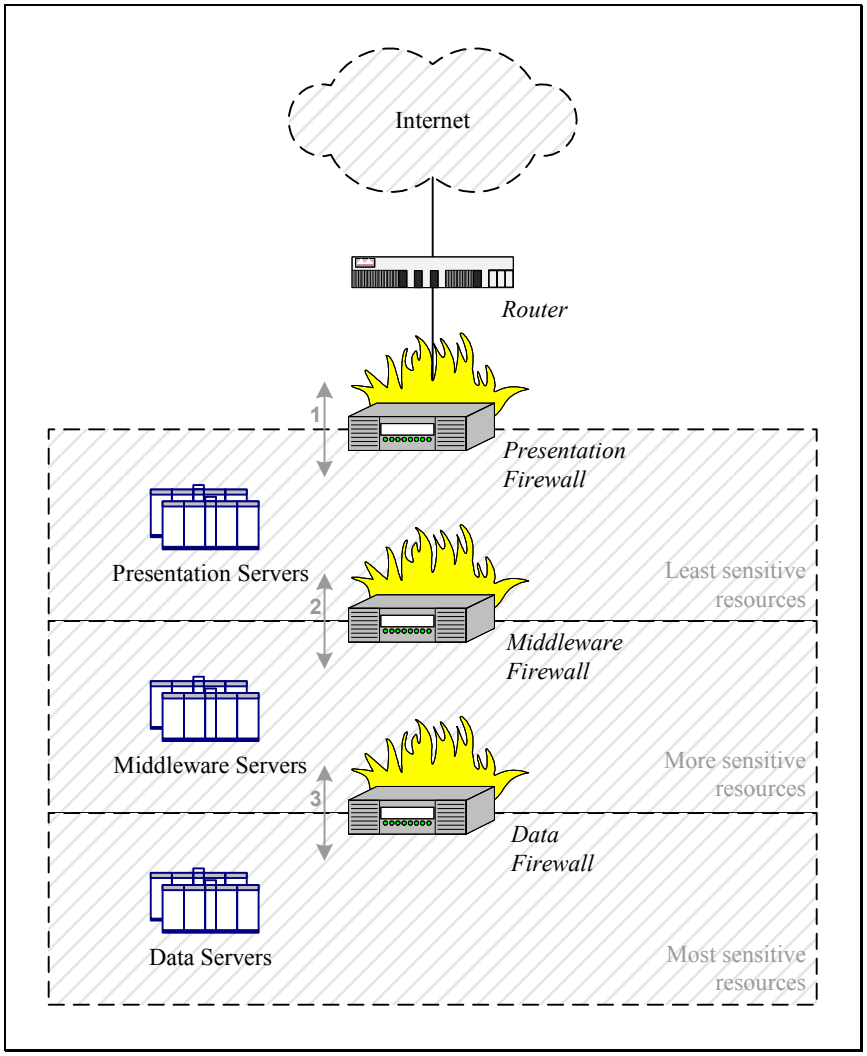

Figure 4 – Firewalls in Series and Three Subnets

When deploying firewalls in series, each firewall's hardware components may be scaled up or down independently of other devices, depending on the nature of network traffic passing through the device. You can also deploy different firewall devices according to requirements of a specific subnet boundary. For example, you may consider implementing the presentation firewall as a reverse proxy to enforce strict control over traffic targeting presentation servers. This works especially well if presentation servers are accessed via HTTP, since HTTP proxying techniques are well understood and have the ability to tightly restrict operations performed by the application's end users. Reliability of proxy firewalls tends to come at the expense of performance, and you may decide to implement middleware or data firewalls using faster filtering devices such as stateful firewalls or even static packet filters.

Such flexibility is difficult to achieve when a single firewall is responsible for regulating traffic that crosses subnet boundaries. However, added security comes at the expense of manageability, which tends to be a significant cost factor when deploying and maintaining multiple firewalls. Middle ground can be achieved between single-firewall and multifirewall designs, so let's look at hybrid architectures that attempt to reach a compromise between cost and security.

Hybrid Multifirewall Architectures

One of the ways to retain the flexibility of dedicating a subnet to each application's tier while decreasing the number of firewalls is to deploy only two firewalls. Figure 5 presents one such design. Middleware and data firewalls have been collapsed into a single device, but the application's tiers still reside on dedicated subnets.

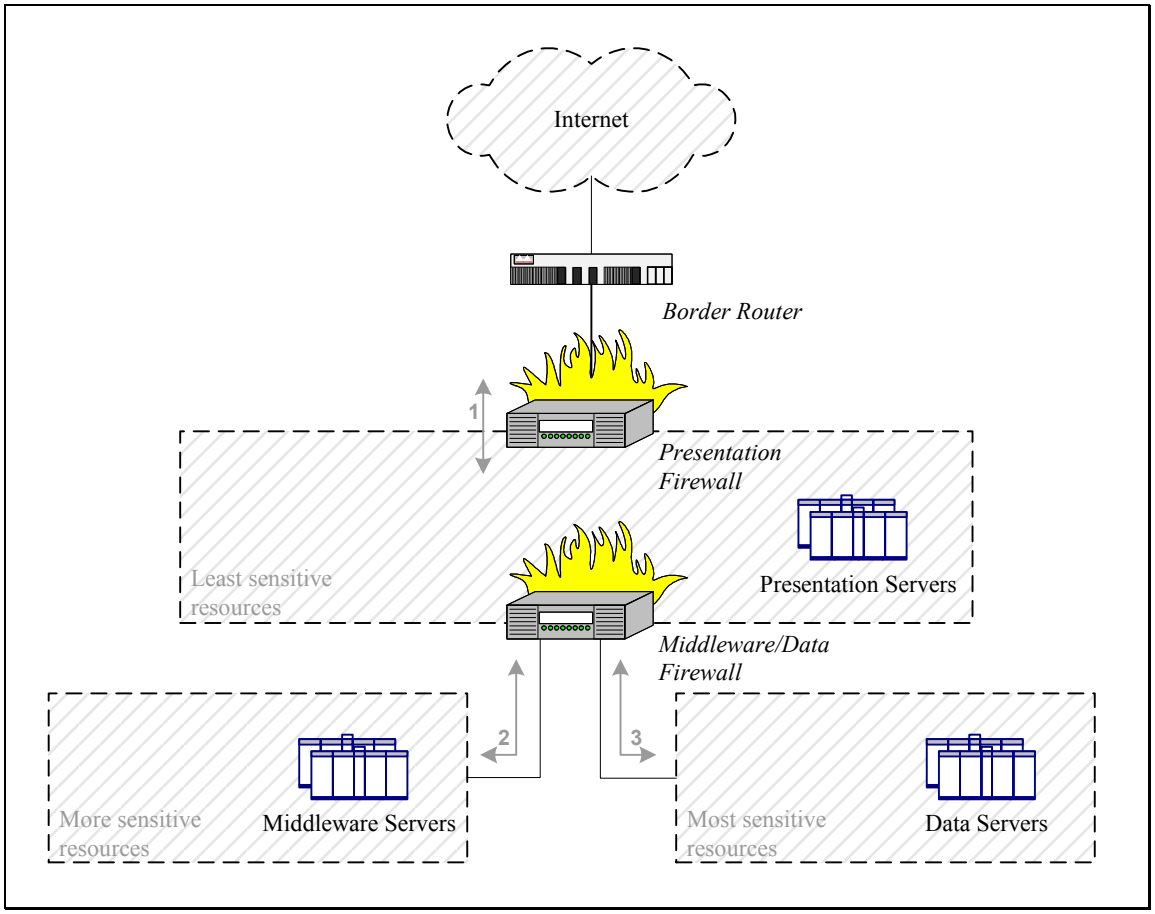

Figure 5 – Hybrid Multifirewall Architecture

As in all designs that we examined, firewalls control what communications can take place across subnet boundaries. Yet, unlike in the single-firewall solution presented earlier, this architecture separates middleware and data servers from the Internet using two firewalls deployed in series. We still have the option of using one type of a firewall – say, a reverse proxy – in front of presentation servers, and deploying another type of device as the middleware/data firewall.

If you prefer to focus the design on isolating data servers from other tiers, you could share the firewall between presentation and middleware subnets instead. This configuration is illustrated in Figure 6. You still have the opportunity to use one firewall technology in the front, and another type of device behind it, but now presentation and middleware tiers are assumed to be more similar in security and performance requirements, and rely on the same firewall for protection.

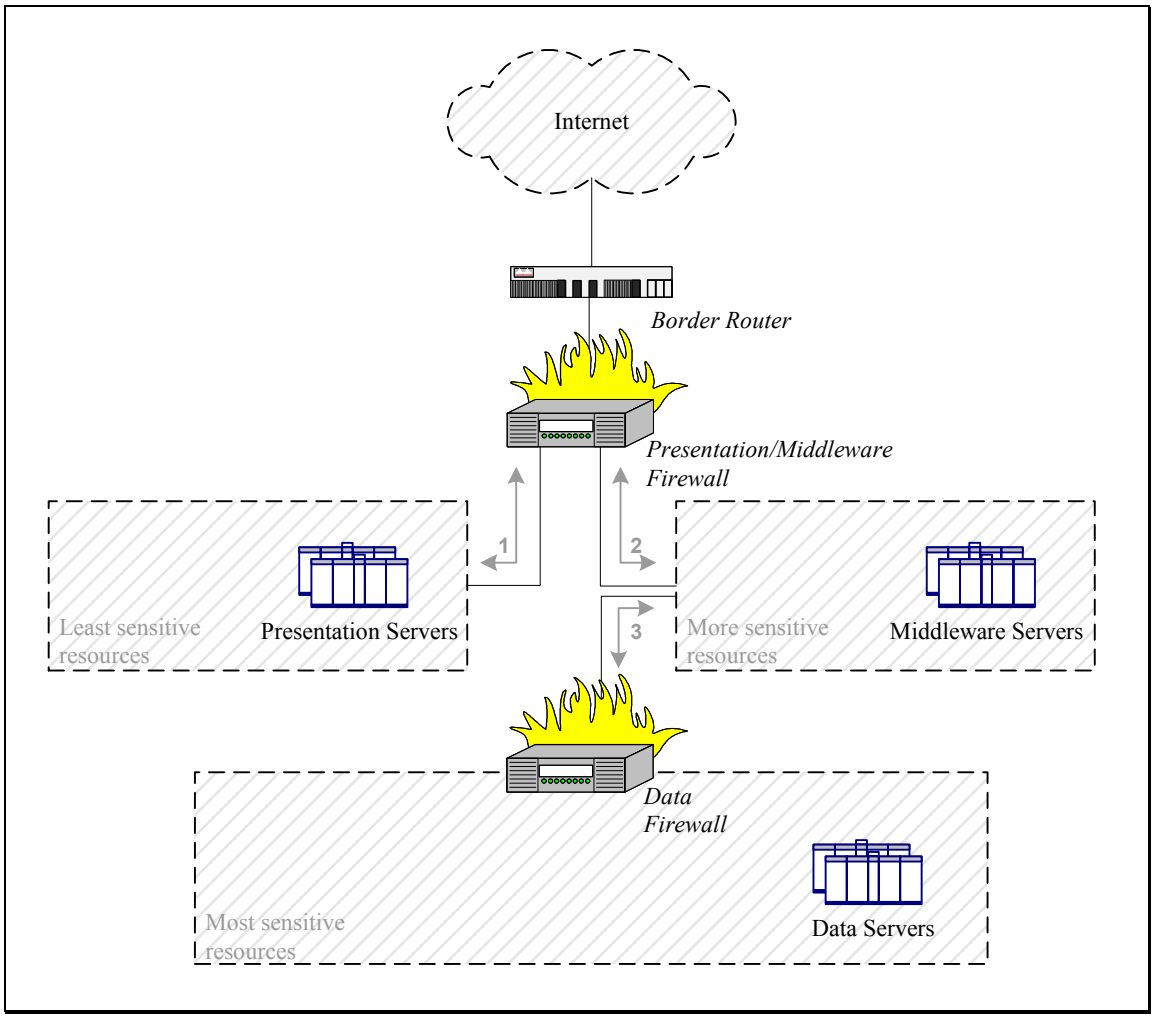

Figure 6 – Alternate Hybrid Multifirewall Architecture

You can further mix and match these configurations by combining some of the application's tiers into a single subnet, if that best suits your budget and requirements. One such configuration is shown in Figure 7. Here middleware and data servers are on a subnet separate from presentation servers, in another attempt to group resources according to their sensitivity levels and the likelihood of a compromise. In this design, the presentation subnet most closely resembles a classic DMZ, with middleware and data servers separated from presentation servers by a dedicated firewall. This design is most appropriate for applications when the distinction between data and middleware tiers is not as clear cut as in a traditional multitier architecture.

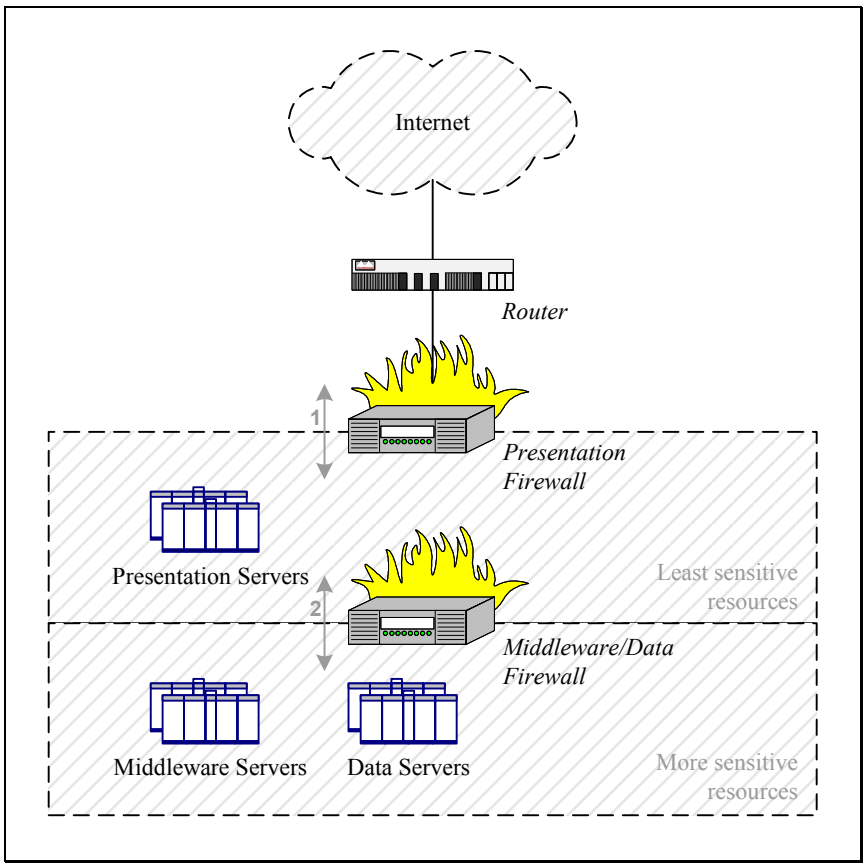

Figure 7 - Firewalls in Series and Two Subnets

Conclusion

In the course of this article, we have examined several Internet-centric firewall designs in an attempt to meet security and performance requirements of multitier applications. In all scenarios, servers hosting application components were separated from the company's corporate network used to conduct internal business, as an initial step to segregate resources with different security requirements. To tightly control interactions between the application's tiers, we looked at hosting tiers of the application on dedicated subnets. By deploying firewalls in series we were able to significantly increase the difficulty of obtaining unauthorized access to sensitive resources from the Internet. At the same time, each firewall layer increased the design's complexity, contributing to the cost of deploying and maintaining the infrastructure, and increasing the likelihood that it will be misconfigured.

The network design appropriate for your environment depends on the nature of your application and the risks that you are trying to mitigate by setting up a security perimeter around your servers. As we discussed, relying on a single firewall or combining application tiers into a single subnet often decreases the amount of control that you have over how application components are accessed. However, beware of jumping to a design that incorporates three firewalls in series, without first considering less expensive alternatives. In this article, we only touched upon some of the many ways of deploying firewalls with respect to each other, and we did not examine the relationship between firewalls and other perimeter defense devices. When designing your network, consider how other components of its perimeter, such as intrusion detection systems, routers, and VPNs, may impact security of the infrastructure, and select a design that matches your application's architecture and your company's business needs.