

search site

PRODUCTS

Database
 Middleware
 Developer Tools
 Enterprise Management
 Applications Technology
 Products A-Z

TECHNOLOGIES

BI & Data Warehousing
 Grid
 Java
 Linux
 .NET
 Office
 PHP
 Security
 Service-Oriented Architecture
 Windows Server System
 Technologies A-Z

COMMUNITY

About OTN
 Oracle ACEs
 TechBlast Newsletter
 Oracle Magazine
 Oracle 10g Books
 Education
 User Groups
 Partner White Papers
 All Community Resources

Downloads

Documentation

Forums

Articles

Sample Code

Getting Started

Tutorials

Blogs

Middleware Architecture Series



Week 1 of 9: Security

Middleware Architecture Series

Spotlighting Innovative Use of Middleware Technology

Security: No Longer An Afterthought

At Celeritas Technologies LLC, keeping high-performance applications secure is a process, not a goal

Dan Norris, a senior consultant with Celeritas Technologies, knows how to make multi-tiered applications secure.

"Attacks against the application tier are on the rise," explains Norris. "The proof is the dramatic increase in buffer overflow-type attacks and buffer overflow root compromises—both are at the application level. There is not usually anything that the operating system can do to help, and certainly firewalls and virus scanners can't address these kinds of attacks."



Dan Norris,
 Celeritas
 Technologies

Norris points out that oftentimes, these attacks result in a companies' sensitive data being posted on the Internet. "There have been some notorious attacks making headlines, and companies are now doing more than simply installing firewalls and virus scanners. Securing the application tier has become much more important."

Recently, a manufacturing firm hired Celeritas to architect a secure customer-facing portal. Considering the customer was the U.S. federal government, security was a top priority.

Security Versus Usability

Securing an application today often means choosing between usability and level of protection—the more levels of protection put into place meant more hoops users must jump through, reducing the application's usability.

"It's always a balancing act," explains Norris. "For really secure setups, we typically install an application and then tighten down the security to the point where the application stops functioning, and then we back off just slightly. That can take a lot of time and be costly, so you have to help the client decide what level of security they truly need." Determining that need must balance the cost of a break-in against end-user accessibility. Adds Norris, "For our client, the cost of a compromise was very high, but they also wanted the application to be easy to use."

Norris explains the client wanted to use an already deployed public key infrastructure (PKI) credential to authenticate to Oracle Application Server and into the new portal application. This way, users could continue to use their same credential file with the new application while maintaining a single sign-on (SSO) experience.

Says Norris, "The most important factor is that every user has a two-factor authentication. Users must provide their PKI certificate file, and then they must also enter a pass phrase."

This two-factor approach keeps things simple for the user while still providing strong front door security since attackers likely won't be able to obtain both pieces of information at the same time. Attackers may be able to intercept the PKI certificate but won't have the pass-phrase, or vice-versa, attackers could hack the pass-phrase but will lack the needed PKI certificate. It's like having the lock but not the key, or the key but not the lock.

"Deploying PKI sign-on with single sign-on is a lot of work," admits Norris. "But in this case, it was worth it because the end-users had an easy single sign-on with a good amount of protection."

In addition to Oracle Application Server Portal fitting nicely into the PKI/SSO design, Norris likes how easy Oracle Application Server makes turning services on and off. "Most people are used to doing this at the operating system," he explains. "You want the OS to have as few services running as possible with the least amount of privileges. With Oracle, you can do this at the application tier as well, turning on just those services you need. Because Oracle Application Server works well with PKI, more services can easily be turned on after a user is authenticated, making the application secure without reducing its overall functionality."

First Steps are Often the Little Things

Once the authentication system is determined, and the operating system and application services are reduced to the bare minimum, it's time to start building the security around the application itself. Norris recommends starting with the obvious.

"My first step is always the physical security of the servers and network components," he explains. "Are they locked up and secure? Can anyone get to the server and yank out a network cable? Are the back-up tapes in a secure place so they can't be walked off with? These seem obvious, but many times we've seen really secure code running on hardware that had no physical security, especially in smaller companies. You couldn't ascertain and authorize who was physically touching the parts of the system."

Next, Norris ensures logon, audit, and user accounts all fall within a secure methodology. He explains, "For system administrators who need to logon and make changes and manage configurations, we only allow encrypted terminal sessions on the UNIX boxes. We don't allow anybody to logon to the servers directly as the group owner account, rather we force everybody to logon as their own normal user account, and then use an audit utility to see who logged on when. If there's a vulnerability in one of those accounts, we would have an audit trail to know how an attacker made it in and through whose account they came in on." In addition, on each node Norris installed intrusion detection software (IDS) that sends out alerts on any abnormalities in the access mechanisms to the servers. He also installed a number of log-watching utilities to ensure the normal UNIX system logs are monitored with any key data highlighted for easy retrieval.

"In the long run, you can

"If you miss the little things, you've left yourself open to an attack"

manage a single place for all of your SSL certificates no matter how many actual middle-tier servers you scale up"

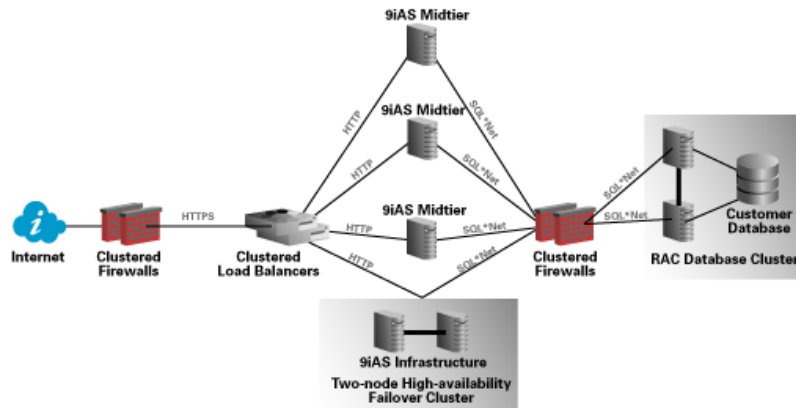
Norris then goes about masking actual system information. "As a security architect, I want to make sure that attackers know as little as possible about our environment. If they know we're running Apache of a certain version, it's not hard for them to go and find what vulnerabilities that particular version has and exploit them."

While Norris admits masking system information doesn't guarantee security, he argues it makes attacking a system much harder. Says Norris, "Masking often discourages an attacker enough that they'll go find someplace that's easier to attack and stop wasting their time with us."

Norris warns, "It's these little things that can be so important. You can have the best application design in the world, but if you miss the little things, you've left yourself open to an attack."

Securing the Internet Tier

The first layer a user hits has to be just as secure as any of the layers behind it. For Norris' client, this meant a single point of entry for all Internet interactions. "The only thing that users will see is a cluster of very restrictive front-end firewalls," he explains. (See architecture below.)



The only traffic that firewall cluster will permit is HTTPS, and only those ports are available to actual authenticated customers. He adds, "This means that if someone wants to attack us, they've only got one little point to do it."

After the firewall comes a load balancer layer. Norris configures the load balancers in such a way to only allow certain URL requests to be passed through, and they serve as the termination point for the HTTPS protocol that will terminate SSL connections on the hardware.

Norris also uses the load balancers themselves as part of his system-masking technique. He explains, "With our load balancers, attackers are going to have a tough time finding out whether there is one server in the system, or twenty servers. It's another way to frustrate the bad guys."

After the load balancer layer are the actual Web servers. "In our case, we used middle-tier servers for Oracle Application Server," explains Norris. "Even though the only way you can get to those is by jumping through a couple of secure layers, we still shut down any unnecessary service, making sure that only the bare minimum services are running on those servers. If anyone did happen to find a way to attack us, they wouldn't be able to find much once they got in."

Securing the Application Tier While Maintaining Performance

After the Internet-fronting tier is the application server layer and its interactions with the backend application and database. Here, Norris had to architect sufficient security but also maintain high availability and performance.

"It's one of the reasons we like using Oracle Application Server clusters," explains Norris. "Besides scalability and its Web cache failover features, Oracle Application Server lets us turn on and off the services we need. We want activate services in such a way to keep the system secure, but this also saves CPU and memory overhead." Norris also used load balancers in such a way to keep a lot of secure socket layer (SSL) management off the application tier. SSL encryption and decryption can hog system resources, so handling these functions was key to maintaining high performance.

"To be good at security design, you have to convey security-related risks in a non-threatening manner"

Explains Norris, "There was nothing on this portal site that passed in the clear across the Internet. Because of this encryption/decryption workload, we decided to use the SSL termination in our load balancer cluster, and we could manage our SSL certificates there as well."

The dataflow is fairly simple. The load balancers have a software proxy on them that receives HTTPS packets. Proxies are different from a virtual IP address in that a proxy will have to do some protocol translation. In this case, in conjunction with SSL termination hardware, the proxy translates the protocol from HTTPS to HTTP by ripping off the encryption and encapsulation around the packets. The load balancer then sends the HTTP request to the Web server; in this case, to the middle tier servers running Oracle Application Server.

Norris adds the performance gain achieved is difficult to quantify, but points out that doing SSL encryption/decryption operations on the server side in software is expensive compared to the cost of doing it in a piece of the dedicated hardware.

"The typical numbers are that in software a typical host may be able to handle 20 to 30 SSL connections per second," says Norris. "Using SSL acceleration hardware, the load balancer cluster can terminate somewhere in the range of 1,000 or more connections per second. So it's a dramatic performance booster."

Norris also likes the ease of management. "In the long run, you can manage a single place for all of your SSL certificates no matter how many actual middle-tier servers you scale up."

Norris continues his thorough approach to security all the way to the backend. Behind the application server cluster is another pair of firewalls that will accept the SQL*Net traffic from each of the application server nodes and allow it to pass through to the appropriate database node.

This second firewall layer establishes a virtual "demilitarized zone" ahead of the portal repository and database. This way, in the unlikely event that the application tier is compromised, the damage to the actual data would be limited thanks to the second layer of firewalls. "Putting in a DMZ using firewalls ensures one last level of defense to ensure attackers can't get to the database tier," says Norris.

Finally, the backend itself is kept highly available and scalable using real application clusters (RAC) that house the actual portal repository and necessary customer data in a shared-disk environment.

Adds Norris, "The RAC servers on the backend are admittedly somewhat small, but being able to dynamically add another node to our RAC environment gave us a little bit less concern about being able to address scalability in the long run."

Advice for Security Architects

When it comes to security, consultants need to know the different options available to their customers. Good security starts on day one of the design, and carries through to good management and system administration.

"To be good at security design, you have to convey security-related risks in a non-threatening manner," says Norris. "As architects, we regularly talk to clients about high availability and scalability, and we have to explain security in much the same manner-as simply decisions that must be made during the design and build process. If you try to address security towards the end of a project, when deadlines loom and chaos reigns, it likely will be full of holes. Start with security in mind, and help your customers understand their options."

Dan Norris(norris@Celeritas.com) is a senior consultant for Celeritas Technologies, LLC in Kansas City. In his 6+ years of consulting experience, he has helped many companies design their infrastructure to be highly available and more secure. He has filled many roles including a system architect, Unix system administrator, Oracle database administrator, application server administrator, and security architect.

Dan has been a technical presenter at international user group conferences, regional seminars, and Oracle user groups throughout the Midwest. He was selected as a beta tester for the Oracle9i Certified Master DBA practicum and is also a Oracle9iAS Oracle Certified Associate. Additionally, he holds Unix administration certifications from Sun and HP.

Please rate this document:

Excellent Good Average Below Average Poor

[Rate and View Results](#)

Give us your feedback on middleware topic: What did you think of this session? [Send](#) feedback to: architects_us@oracle.com

[\[Back to Middleware Architecture Series Home Page\]](#)

For more information call
>> 1-800-633-0994 <<

 [E-mail this page](#)

 [Printer View](#)