

ASP.NET Web application security in VDS hosting

Asked 15 years, 11 months ago Modified 9 years, 8 months ago

Viewed 394 times



1



We are designing asp.net web application in wcsf. Web application will be deployed to Windows Server 2003 shared [VDS](#) hosting. Web site will be used for b2b, monthly service fee and credit card transactions used in web application so it must be secure site. I want to consider what must be done before deployment and i need an answer to a few questions:

- 1) How can i copy protect my site & my code. Is code signing enough? What should i do for reflector protection? Obfuscation is enough?
- 2) What about Windows Server 2003 [VDS](#) hosting pros & cons?
- 3) MS SQL Server 2005 Express is suitable for that kind of business use? Limitation of mssql express (4GB storage and 1Gb ram) cause bottlenecks in website?
- 4) Can I transfer database from express edition to mssql 2005 workgroup edition in the future without problem?

asp.net

sql-server

hosting

vps

Share

Improve this question

Follow

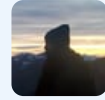
edited Apr 17, 2015 at 10:42



Stefan Steiger

81.9k ● 69 ● 399 ● 454

asked Jan 17, 2009 at 13:28



Erhan

1,166 ● 8 ● 11

It's VPS not VDS please edit... – [Mischa Kroon](#) Jan 17, 2009 at 14:18

No...he does mean Virtual Dedicated server as far as I can see. – [Kev](#) Jan 17, 2009 at 14:38

Yes i mean virtual dedicated server. – [Erhan](#) Jan 17, 2009 at 19:52

2 Answers

Sorted by:

Highest score (default)



1



1. Obfuscation would be sufficient. Code signing doesn't provide any protection from prying eyes. Also most respectable hosters aren't interested in your intellectual property and data. They're in the hosting business at the end of the day.
2. I work for a hosting company and we have a fast growing number of customers hosted on our virtualisation platform. One of the advantages for both the customer and ourselves is that if you need more CPU or memory, we can add them almost immediately because this is just a config setting on

the VM. Ultimately it all depends on how your supplier has built his virtual environment. You really need to gather more information about the capabilities and limits of their environment, how many servers per node they typically run etc, and decide if it's adequate for you.

3. SQL 2005 express will perform as well as full fat SQL 2005. The limitations are in functionality (e.g. no olap, that kinda thing) and as you correctly point out, the maximum amount of memory it can utilise and max size of DB. Performance will degrade however if you outgrow the maximum amount of memory available.
4. SQL 2005 Express databases can be backed up and restored to any SQL 2005/2008 product without any problems.

Without knowing the volume of traffic that will hit your application it would be hard to predict whether a virtual server is the best choice of environment. The same applies to SQL 2005 express. If the app is database intensive you may find yourself under increased memory pressure - the more memory SQL server has the more it can cache. I think you need to do some of your own capacity planning and decide for yourself what is the optimal solution for now and later down the line.

Share [Improve this answer](#)

[edited Nov 24, 2013 at 7:48](#)

[Follow](#)

answered Jan 17, 2009 at 14:08



Kev

120k ● 53 ● 305 ● 391

Kev - don't take this the wrong way but are you concerned that Erhan took your reply as the whole answer? You gave very useful information but it may leave him just focused on protecting his code. If he doesn't protect the site and data, he's gonna get hammered by hackers. – [Mark Brittingham](#)

Jan 18, 2009 at 3:19

Um...I guess technically I answered all his points, but your answer also added value by making extremely valid additional points. I guess this is the problem with SO's one answer only approach. I'd at least hoped he'd have given u an upvote, the that's there is mine. – [Kev](#) Jan 18, 2009 at 16:26



Erhan -

1



Your primary security objective isn't to protect your source code, *it is to protect your customers' data*. To that end, you have to make sure that A) your communication is secure (SSL) b) your code implements appropriate security (isn't subject to SQL injection attacks) and c) your database is secure.



For (A) - communication security - you'll need to get an SSL Cert from someone like [Thawte](#) (that is who we use and I like them). When you install the Cert on your server, you'll want to ensure that the web site only accepts secure (SSL) connections - at least for all pages with

sensitive data. For example, all forms with credit card data must be submitted via SSL.

For (B) - code security - this is worthy of an entire [book](#)! You'll must **not**, for example, construct SQL calls by just appending forms/request data collected from the web page (this leaves you open to [SQL Injection](#) attacks). **All** SQL arguments must be parameterized. etc. etc. No one can explain security in an SO post - you have a *lot* to learn.

For (C) - SQL Server security - this too is a complex topic but there are a few key points. Based on your description, I'm assuming that the SQL Server is running on the same server as your web app. This means that you do *NOT* want it accepting *ANY* connections from the network. Don't be tempted to leave this open so that you can access it easily from the SQL Management Studio on your desktop...this is just horrible, stupid security practice. You can access it via Remote Desktop after signing on to the remote server. Also, disable the 'sa' account - just do it right now. People finding you on port 1433 (if you don't disable all network access) will be flinging brute force password guesses against the sa account every second of every minute, etc. for years. We get about 17,000 a day on a non-web site database that we maintain for a few customers. There is more to learn here but if you take those two steps, you're pretty well covered. Just to be sure, though, I recommend that you also run the MS security surface analysis tool and follow its

recommendations, where appropriate, to reduce the visibility of your SQL Server.

As far as virtual machine hosting, it *should* be fine. However, dedicated hosting only runs about \$199 a month at [MaximumASP](#) (who I use, recommend and think are among the best in the business!). I would recommend that you consider using your own dedicated server just so that you can easily scale your business if it proves successful.

With respect to your database, SQL Server Express will work for this and, yes, you can upgrade to the Workgroup Edition in the future. The database limits won't cause a bottleneck for your web site unless you actually hit those limits. You'll need to figure out whether that is likely. That being said, I really would *not* recommend trying to run a professional web site on the free version of the MS database - especially one holding financial data. With any success whatsoever, you *are* likely to bang into those limits. Many web hosting companies (and MaximumASP in particular) will let you add a professional version of SQL Server to the price of your server for a minimal price (e.g. \$25 a month or something similar).

Finally, you appear to be a bit confused about Web Applications versus Windows apps. In a web app, your users will not have access to your source code/DLLs (IIS blocks access to a variety of files including dlls and .config files). They aren't downloading and installing an application so you will not need sign your code and you

have no need for obfuscation. "Reflector protection" doesn't even make sense to me in this context.

If you are worried about your *hosting company* seeing your code, you should find a different hosting company. That being said, I just can't imagine that any reputable hosting company really cares about seeing your source code.

Good luck!

Share Improve this answer

edited Jan 17, 2009 at 14:40

Follow

answered Jan 17, 2009 at 14:20



Mark Brittingham

28.9k ● 12 ● 82 ● 111

Nice post but I disagree with you about 'Virtual' dedicated servers. we're hosting some pretty heavy duty virtual servers on our platform. They're not a 'hobby' environment and are entirely suitable for high traffic applications. – [Kev](#) Jan 17, 2009 at 14:37

Thanks Kev - I've amended my post. I went with a Virtuozzo virtual server for awhile for a specific customer and, while it worked pretty well overall, it gave us more hiccups than any of our dedicated servers. Of course, that is *old* technology by now! – [Mark Brittingham](#) Jan 17, 2009 at 14:39

If it's done properly it works like a dream. We have 64GB mem/16 cores per node, the back end store is EMC Clariion SAN/RAID with SAS disks etc etc. We run all of our shared platform on it now (1200 sites per server - 15 VM's split across 2 nodes) and it rips. – [Kev](#) Jan 17, 2009 at 14:45

Well, that is *big* iron. I guess I hadn't thought of how much firepower you could bring to a single server.

– [Mark Brittingham](#) Jan 18, 2009 at 3:20

Not only that, tricks like being able to migrate a VM from one node to another without the customer noticing are also very cool. I added a comment to your observations on my answer, no hard feelings taken. – [Kev](#) Jan 18, 2009 at 16:31
