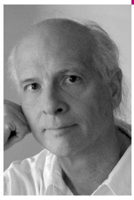
RIK FARROW

musings



rik@usenix.org

VIRTUALIZATION IS ALL THE RAGE these days. We now have multiple alternatives to both server and desktop virtualization software, and virtualization is fast becoming the new "green." As I watched the rush to virtualize unfold, I wondered who had considered the security implications of virtualizing servers?

As with many other shiny new technologies, people don't want to poke too deeply into the works because they might not like what they see. It is human nature to focus on the good side of things and to ignore the messy parts for as long as possible.

Perhaps you are one of those who believe that virtualization makes running servers more secure. Whether you are or not, I invite you to replicate an experiment I ran that helps to resolve the security issues.

You can relax, because the experimental setup is trivial. I so love running thought experiments for this reason, even if the outcomes can vary depending on the hardware or software used to run the experiment. But enough talk. Let's get this experiment running!

The Experiment

First, we need a control. For our control, we will use a modest rack of 15 servers. Each server runs a single application, as we learned a long time ago that running a single service per hardware host supports ease of management, patching, fault isolation, and security.

For our test case, we will take these same 15 servers and virtualize them. This isn't a radical idea; it's merely the rage these days, as we get to utilize our systems much more fully. When running servers on platforms that we outfitted with excess resources because we really didn't know how much we needed and overprovisioning is always the safe bet, we had been wasting resources and energy. The virtualized servers run nicely on a single, if built-out, server, and we can migrate any of them to another virtualization server if they need more resources.

HYPOTHESIS

Okay, now for the hypothesis: Has moving our 15 servers into a single virtualization host made the collection more or less secure? Recall that we simply moved the existing servers over. We didn't patch software, replace buggy software, or move to more secure scripting languages or database services. We just installed the same software within guest domains. I believe the answer here is obvious, but I will spell it out just in case: No, the systems are not more secure than they were. How could they be, as we have not changed anything about the services they were running, and the supporting software?

If they are not more secure, are they less secure? Consider that we have added a new software layer, the hypervisor, along with its supporting software. As with any other software, the virtualization software itself has bugs, including security vulnerabilities. You can visit VMware's security advisory page [1] to get a feeling for this, or create a search using the words "vulnerability" and the name of your favorite virtualization software.

Adding virtualization software increases the attack surface. The attack surface represents what portions of a system are vulnerable to a potential attack; it includes everything from PHP scripts, the Web server, and the system libraries to the underlying OS. To this stack we have added both the hypervisor and its supporting software. In the case of VMware, the bare metal hypervisor, ESX, is 32 megabytes of software. We don't really know how many thousands of lines of code go into making up compiled code with a disk footprint of 32 MB, but surely this took tens of thousands, more likely hundreds of thousands of lines of code. We know there are bugs in the hypervisor code, as some have been patched already. I believe that adding a hypervisor must increase the attack surface beyond where we were before we combined our 15 servers on a single server.

The same will also be true if we decide to use Xen or some other virtualization software as our hypervisor. We have added software, and since software has bugs, the attack surface increases.

ATTACK SURFACE

But let's examine our experimental setup more carefully, On the one hand, we have our legacy rack of independent servers; on the other, we have the virtualized servers, all running on the same hardware. We located our servers on separate hosts partially as a means of increasing their security, and we gave that physical isolation up when we virtualized them. Looking at recent vulnerabilities in virtualization software, we can see that bugs in the hypervisor can give a local attacker root or local system access to the entire system. Thus, we have given up the protection we once had in isolated systems by going virtual. An exploit in one virtualized server can provide unfettered access to all servers, as they are hosted on the same hardware.

There is nothing magic about virtualization. It is merely another OS technology, newly developed outside the world of IBM mainframes, that suffers from the vulnerabilities inherent in any software. And, as with any software, the more features that get added, the greater the potential attack service. Dan Bernstein's DNS software is secure largely because it is so featureless. You cannot bind both an authoritative DNS server and a caching server to the same IP address with djbdns, and the related simplicity reduces the attack surface.

MIGRATION

Did you get the same results running the thought experiment on your hardware that I got on mine? I suspect so, unless adding positive numbers together produces a zero or negative result using your hardware/software stack. But let's not stop there, as there are file systems to consider.

I actually first approached the issue of virtualization security from the other angle: how having virtualization can make services more secure. I imagined an organization that only runs virtualized desktops and pondered how this would impact patch management. If these desktops get rebooted daily, then patching a single image overnight means that all desktops get the same patched image when they reboot the next day. Fantastic!

But that picture presumes that all users run the same software and is overly optimistic. At best, we have simplified our update procedures to patching just a handful of desktop images and having assurance that they will be the only versions used. And we have created a network rush hour by booting all those virtualized desktops, using networked file servers to do this.

I also wondered about the ability to patch servers by installing the patches to their disk images. If the disk images are not in use, this is simple to do. When the images are being used by a guest, the issue is similar to patching any mounted and in-use file system. Binaries and libraries that are currently loaded cannot be overwritten, but there are tricks, such as renaming the binary, that can be used. I will have more to say about disk images later.

One of the cool features promised by virtualization is the ability to migrate guest operating systems. Suppose a virtualization host doesn't have the resources to support all of the guests we have installed there? We can simply "migrate" that system, even without shutting it down! Although this sounds really cool, consider that we are migrating an entire system over the network. In my darkest thoughts, I have installed a network sniffer and seen not only the entire guest but also the contents of kernel memory, including any cached credentials, as the system gets migrated. I suspect that encryption of guests as they are being migrated is on the drawing boards of virtualization providers, but that is the least of the issue.

One of the cool features of Xen and VMware is that they do use disk images. You can download these images from the Internet or build them yourself. If you need to load-balance a Web service by adding a new server, you just point the guest at the image you prepared earlier and fire it up. Let's ignore for the moment the notion that you may have created the disk image months earlier and not patched it since, as you need to spin it up now. And what about the disk image itself?

Guest disk images have the marvelous property that they can be mounted and manipulated just like any other file system. But there is a large difference here, in that when you mount a disk image, the access controls that were present under the guest host no longer apply. If you can mount the disk image, you are root (or an administrator) and now have total access. This really is no different from having root access to a file server that contains sensitive data or one that is used for network booting of systems. But it does mean that all these same problems exist in the virtualized world.

When I was learning about Xen, I made a mistake in editing the /etc/fstab file that prevented a guest from booting (a change in the name of the swap device). I could have started over and rebuilt the Xen guest, but that would have taken me many hours and could result in unfixing things I had already fixed, or the introduction of new mistakes. Instead, I figured out how to use the losetup command and loop devices to mount the image and edited /etc/fstab. I've done this with VMware images as well [2].

This useful ability to mount disk images implies that it can be used by attackers as well. Access to the root domain, where guest images may be stored, means access to everything that appears within those images as well. The hypervisor also has access to the memory granted to guest images, so there really are no secrets that are not available to the hypervisor. Running guest images is akin to running software within a debugger and all that that implies (see Chow et al. [3] for an example).

Lineup

I really don't want to convince you that running virtualized servers is not a good idea. I think it is a wave of the future, appearing now, and it is pretty unstoppable. What I do want to do is suggest that you don't "drink the Kool-Aid" that hypes that idea that virtualization is more secure than isolated servers. Virtualization is not more secure, and it cannot currently be more secure. Perhaps someday we will have hardware that includes real support for isolating guests, but that day has not arrived yet (and appears to be uncomfortably far in the future, beyond the five-year horizon). You can and should use virtualization and you must be aware of the added vulnerabilities in doing so.

In that vein, Wenjin Hu, Todd Deshane, and Jeanna Matthews, who are among the authors of *Running Xen,* offer us a great explanation and comparison of the virtualization possibilities available in Solaris 10. Not only do they compare these, but they also define the different types of virtualization possible in a way that will help you understand similar technologies under Linux or other operating systems. You can also find a book review of *Running Xen* in this issue.

Next up, Edward Walker takes a look at cloud computing clusters. Walker wondered how Amazon's EC2 cloud computing would compare to a dedicated research cluster in terms of performance, and his benchmarks may surprise you.

Alva Couch then considers how the laws of thermodynamics apply to sysadmin. Couch writes about transforming problems through the use of virtualization and explains the tradeoffs involved in so doing.

Robert Solomon presents a case study in setting up Asterisk. Solomon had set up simpler Asterisk VoIP systems before, but this installation replaces an aging proprietary one for a medium-sized office with very specific requirements. He explains the hardware as well as the Asterisk tweaks necessary to perform an all-page and to unlock the front office door.

Brad Knowles explains how best to populate your network with your own NTP servers. NTP will not work well if you configure NTP servers as you would other servers. Getting the most efficient setup from the perspective of network traffic and server load is an interesting challenge, as is choosing the right hardware. In this issue Knowles also gives us a review of *The Book of IMAP*.

Sandeep Sahore shares his cfsize program. Sahore wondered why there weren't UNIX applications for decreasing the size of files without first splitting them or truncating them to zero length, and cfsize is his answer to these problems.

Jason Dusek examines the problems with concurrency. Dusek became intrigued by the mistake of conflating parallelism with concurrency, and he digs deeply into why concurrency is both a difficult and a currently critical problem.

David Blank-Edelman explains some of the tools you can use in Perl for handling MIME attachments, offering some concrete examples. Pete Galvin continues his thread, started in the August issue, about system analysis. In this issue, Galvin focuses on Solaris-specific tools that help in analyzing problems. Dave Josephsen then encourages us to create Event Brokers for Nagios, providing us with a great example of his own. We have Nick Stoughton explaining the role the USENIX Association (that is, you) has in certain standards bodies. Doing this work requires funding, most of it just to cover travel expenses, and we need to understand this role and decide whether the organization should continue to support it. I certainly think we should. Nick's discussion is followed by more pages than ever of book reviews.

Finally, we have conference reports. The 2008 USENIX Annual Technical Conference is covered in great detail, followed by reports on Hot Topics in Autonomic Computing and on Storage and File Systems Benchmarking.

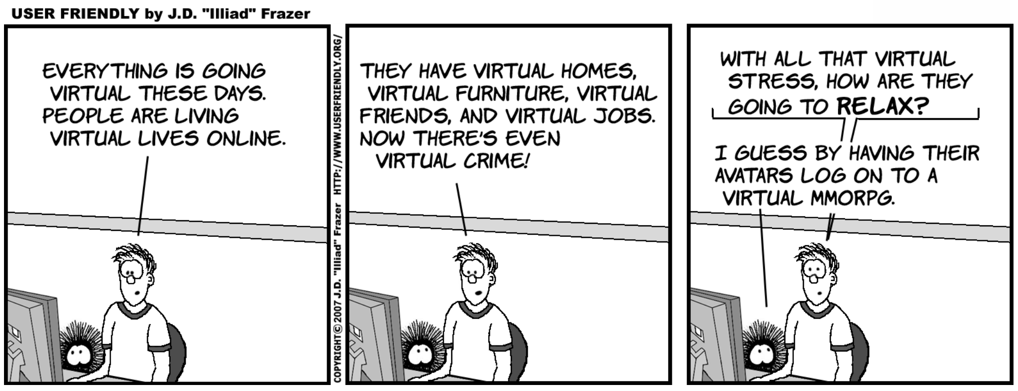
You may have noticed that a lot of this issue is devoted to virtualization. Virtualization is hot, useful, and important, yet, as I suggested above, it comes with its own share of security problems. Most of these are not new. All that I ask is that you remain aware that adding another abstraction layer to an already deep software stack won't make security problems vanish. Instead, simple arithmetic suggests that these problems can only increase.

REFERENCES

[1] VMware security advisories: http://www.vmware.com/security/advisories/.

[2] Mounting VMware disk images under Linux: http://legroom.net/2007/08/05/how-mount-vmware-disk-images-under-linux; http://www.cromoteca.com/en/blog/mountflatvmwarediskimagesunderlinux/index.html.

[3] Jim Chow, Tal Garfinkel, and Peter M. Chen, "Decoupling Dynamic Program Analysis from Execution in Virtual Environments," *Proceedings of the 2008 USENIX Annual Technical Conference*, pp. 1–14: http://www.usenix.org/events/usenix08/tech/chow.html.


USER FRIENDLY by J.D. "Illiad" Frazer
EVERYTHING IS GOING VIRTUAL THESE DAYS. PEOPLE ARE LIVING VIRTUAL LIVES ONLINE.
COPYRIGHT © 2007 J.D. "Illiad" Frazer HTTP://WWW.USERFRIENDLY.ORG/
THEY HAVE VIRTUAL HOMES, VIRTUAL FURNITURE, VIRTUAL FRIENDS, AND VIRTUAL JOBS. NOW THERE'S EVEN VIRTUAL CRIME!
WITH ALL THAT VIRTUAL STRESS, HOW ARE THEY GOING TO RELAX?
I GUESS BY HAVING THEIR AVATARS LOG ON TO A VIRTUAL MMORPG.