Fred Cohen & Associates - Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

The Virtualization Solution

Virtualization is increasingly being touted and used as the solution to the many ills we have in securing operating systems. But for many, it's just a case of "back to the future" with a twist. The twist could be good – or could be bad – for security – depending on how we use it. And if history has anything to teach us, it's that what can go wrong will go wrong – in the worst possible way. But there is always hope, if we can just figure out how to couch it.

Operating systems will save us

Underlying the basic notions of operating system protection is the notion of separation. The operating system provides separation of files, directories, processes, users, and devices from each other, even though the hardware allows them to interact arbitrarily. In its role as mediator, the operating system prevents user processes from subverting the operating system and each other by limiting where in they can read and write, domination of input, output, storage, and processing resources, and by intermediating the hardware mechanism so that only the operating system has direct access. If we implement the operating system correctly, it can provide provable guarantees over the processes, users, files, and devices, and we can have a secure operating system environment that removes the need to have every program protect itself.

Or at least that's how I recall the theory as taught to me some 30+ years ago.

Of course it turned out that implementing the operating system so it was all secure was too hard because the size of the operating system, libraries, drivers, and other related components was too big to prove security properties. And in practice, both obvious problems (like sending bad information to operating system calls and having them use the bad information to do bad things) and less obvious problems (like race conditions aver access to files based on pathnames resulting in setting things altered after being tested) exploited the intractability of getting the operating system "right", resulting in the eternal flow of faults that produced security failures.

Security kernels will save us

The solution? Security kernels! Of course, since we can't build a big secure operating system, we can build a small secure security kernel that makes all of the key security-related decisions, and then allow for the more complex processing associated with the full featured operating system to do all of the other work.

Or at least that's how I recall the theory as taught to me some 25+ years ago.

Of course it turned out that implementing the security kernel so it was secure was too hard because its size grew too large, and of course the security models we used didn't match the realities of the world. Even if you can't attack the security kernel, you can attack the higher level OS mechanisms, the libraries, the user level processes, and users themselves, and they will ultimately bend to your will as an attacker.

Perimeter controls will save us

The solution? We put controls at perimeters. After all, if you can't get past the perimeter, you can't get in, and everybody inside the perimeter is authorized anyway. Sure, there are covert channels, but those are too obscure to even bother with, and we can do theoretical analysis of them to limit the bandwidth to the deviations allowed in usage. How big can that be anyway?

Or at least that's how I recall the theory as taught to me some 20+ years ago.

Of course it turned out that the implementation of firewalls, by its nature, had to allow some things in and out or there was no reason to have it, and if you can get any bits in and out, you can get any other bits in and out. So as we limited the ports on IP protocols, all of the traffic was diverted through a few ports, and everything was then built up on those ports – Web, DNS, perhaps a few more. And of course, we saw tunnels through the firewalls into user processes that avoided any operating system or library level restrictions, running programs that the administrators didn't control, if necessary. The whole Internet became a giant covert channel.

Intrusion detection will save us

The solution? Intrusion detection and response! We will implement automated systems to tell the good from the bad! Never mind that we already know it's impossible to do perfectly, we will do it imperfectly, but so well that no real person will ever figure out how to get around it. After all, how hard can it be?!?

Or at least that's how I recall the theory as taught to me some 15+ years ago.

Of course it turned out that intrusion detection and response still isn't good enough to stop anybody that is willing to try the detection and response system out on their new attacks, and of course there are at least 50 ways to defeat your intrusion detection and response systems, and the overhead keeps getting higher, and...

Encryption will save us

The solution? Encryption will save us! We can use public key infrastructure combined with integrity controls and digital signatures, certificates, and all that other encryption stuff to secure all of the communications so it doesn't matter who you talk to, it will be private, and you will only be able to run the programs that are certified, so that will be safe. After all, we trust our vendors, their vendors, their vendors, and so forth!

Or at least that's how I recall the theory as told to me some 10+ years ago.

Of course it turned out that the implementation of encryption was flawed, but even if it wasn't, everybody wanted to talk to anybody any time from anywhere, and you can't resist the flood called the users. And of course when we encrypt, we can no longer detect intrusions, so we had to break the encryption at the firewalls to detect intrusions and respond to them, to audit, etc. and then re-encrypt for the rest of the trip. But we threw hardware at it, and it only increased costs by 10% or so. And we waste so much of our computer time on the other stuff, that the 10% is hardly even noticed. But in the end it didn't help much because even if you encrypt and sign everything, if anything is defeated almost everything else eventually gets reached. And of course we also started to build embedded languages so that we could load code that the IDS couldn't see and the firewall couldn't stop and that supported the encryption, and that ran in the authorized Web browsers, and that communicated without bothering the operating system, and these were the higher level languages embedded in our Web browsers, twitter clients, and so forth.

Virtualization will save us

The solution? Virtualization will save us! We will separate everything into its own virtual computer and run it from there. Then we will control the interactions of these virtual computers using a small security kernel and they can run whatever operating systems, software, and everything else they want! Not only that, we can balance loads so that as one computer gets too heavily loaded, we can move the virtual computers to other machines and that will prevent denial of services and other outages!

Or at least that's how I recall the theory as told to me a few years ago.

Of course... it will turn out that all of the same problems we had with operating systems, security kernels, perimeter controls, intrusion and anomaly detection and response, encryption, and everything else I have mentioned will still be problems in the virtualized environment, except that we won't have the same control over where things happen, and we will lose the ability to keep track of things in even more ways.

The truth: none of these things will save us – and I'm not sure we need to be saved

Put simply:

- The problem is that we don't really want security as much as other things.

Neither operating systems, security kernels, perimeter controls, intrusion and anomaly detection and response, encryption, virtualization, nor dark of night will change this.

But having said this, we have a responsibility to help move the World in the right direction, to the best of our understanding of what that direction is. Simply sitting back and criticizing is not enough. Rather, we must ask and answer:

- How will we use the opportunity that virtualization brings us to improve the situation?

This is where the security community can make real progress, and do so fairly quickly.

Ten things to know about virtualization

The first thing to note about any security solution that will survive serious scrutiny is that it has to be presented honestly. Hyperbole and exaggeration will not produce better security any more for virtualization than it has for anything else in the past. Whatever solutions we have will have their limits and we need to make them clear going in and along the way. So here are ten things to know about virtualization and tell everyone else when they tell you how it will solve the World's problems.

1. Virtualization is not new. At least 30 years ago, mainframes ran multiple operating systems in virtual machines. And at least 40 years ago, when a card deck was submitted and there was more than one mainframe available with the proper capabilities, the card deck might be run on any available machine, depending on the operator's inclination and available resources and priorities.

2. Virtualization is a form of separation. It separates different virtual things from other virtual things in space, time, and/or in other ways. Don't expect it to solve behavioral problems, content control problems, or encryption or coding problems.
3. Virtualization is only as good as the quality of the mechanisms that implement it. If we cannot build a secure kernel that meets our security needs, and we can't build a secure operating system that meets our security needs, what would ever make anyone imagine that we could build a secure virtual environment that meets our security needs? And if we can do those other things, why do we need to do virtualization to meet any security need?
4. To be useful, virtual systems will need to communicate. That means that we will need all of the same controls that we need when non-virtual systems communicate. That means firewalls, IDSs, process separation, and the list goes on and on.
5. If we cannot securely manage one computer at one location running one kernel and one operating system with one set of software, applications, and users, what makes anyone believe we can run a multitude of computers at a multitude of locations with a multitude of kernels, operating systems, IDSs, and so forth? Isn't the definition of insanity doing the same thing over and over again and expecting different results?

Now that we know some of the things we cannot expect from virtualization, how about some of the things we can expect?

1. Virtualization is being used to provide separation at levels where separation is not otherwise being commonly applied. For example, in Web browsers, separation is being used between sessions in different “tabs” through a virtualization approach. Perhaps this sort of separation will reduce the impact of breakdowns in protection within one virtual environment on other environments within the same larger environment. Of course there are many assumptions inherent in this approach, but if it works because there is a willingness to put more quality into this separation, it may be a net positive.
2. Virtualization is being used to try to disaggregate risks. In particular, distribution of processing resources over location with the ability to reassign resources brings both the potential for abuse and the potential for increased reliability, depending on how it is managed. Similarly, the reduction of activities performed by a given virtual machine means that breakdowns in internal protection can only spread throughout that virtual machine without violating separation mechanisms between machines or using inter-machine communications to spread further. Of course there is a side effect of increased risk aggregation in the hardware platforms which are handling more things than when they were assigned to physical machines, and increased risks associated with the movement of operating environments and content from place to place, and so forth, but that's the nature of the tradeoff. Virtualization aggregates risk in one way to disaggregate it in another way.
3. Virtualization has the potential, but not yet the promise, of distributing decisions across infrastructure to reduce the impact of successful attacks on select elements of the infrastructure, because of the use of multiple paths and redundant mechanisms to assure integrity of results and processes producing them. Of course this makes the

mechanisms less efficient as a tradeoff for increased integrity, but that is the case at all levels of mechanisms. Reliable composites from less reliable components. This has not been thoroughly thought out for malicious circumstances of course, but there is some serious potential.

4. For things that are not extremely important from a privacy or integrity perspective, are commodity services that are available from any number of more or less equivalent sources, and provide little benefit to the malicious attacker or likely harm to the individual user or enterprise, like personal emails, non-private calendar entries, access to search engines, short messaging service for fun, and so forth, virtualization may have little risk impact and far greater operational efficiency than “owned” infrastructures and systems. In this sense virtual computing is like toilet paper. We depend on it heavily for our needs, but we aren't worried that one or another supplier will have a major failure. There are plenty of other vendors, and unless we have to keep changing vendors quite often or sustain substantial outages, minor problems can be tolerated.
5. We can also reasonably expect that the hyperbole for virtualization will exceed the reality in terms of security, while the utility of virtualization will exceed the hyperbole in ways we do not yet specifically anticipate. Creative people will find creative things to do and make it increasingly worth our while to use their virtual resources. And as a side effect, more and more people will use them for more and more things, regardless of the security implications.

Security cannot simply say "no" - it won't work

This is, of course, thematic in the security space. It's easy to stand back and identify the problems likely to result from virtualization, which will largely not be “properly” secured to the specific requirements of any particular person or organization. To some extent, this has already been done, both here and elsewhere.

And yet, essentially every reader of this article uses virtualization on a daily basis and trusts it to one level or another, often to excess. Whether it is getting weather information from a non-primary source on the Internet, using a search engine that is freely available over the Web, making online purchases for commodity items, or simply using VoIP telephone services; there is a compelling argument for using these services. They work – until they don't.

Simply saying "no" to virtualization is foolishness, among other reasons, because it won't work. The user base of the World is moving away from the desktop computing and storage model to the network-based model. And as the resources of virtualized environments are improved, larger and more capable enterprises with the knowledge, skills, and capacities to protect themselves against large scale negative consequences to their users and themselves, the providers will act to improve protection at distributed and yet centralized facilities that they operate. They will start to bear the burden, but as part of the deal, they will also take more of the pie.

They key thing to consider for the future, is how much of the pie can you give them and what you really need to be sure of, to what extent, and at what cost. In other words, risk management, as it has always been.

Consider this future

- Essentially all of your content is in the virtualized environment – encrypted by means you trust and control – and duplicated across carriers and providers – so that from your inexpensive and easy to secure computer, which you can buy a new one of from a local store for $200 at any time, you can access anything of yours, without possessing any of it other than the images that appear on your screen when you need them.
- The cost to you is some level of inconvenience at having to access things over networks, a loss of availability when enough networks are down, some level of periodic fees for maintaining the required services, and the periodic update costs for your inexpensive hardware used to access the content as and from where you like it.
- You have access, on demand, to vast amounts of computing power, shared data sets used by millions of others around the globe, and protected by the investment of those same millions of others. You can use it for whatever purposes you can identify, and you pay only for what you use when you use it, plus some overhead based on total utilization. As part of a global group of users, you act to keep the prices low and service levels high, by buying from the open market, or by buying from a small number of shared cooperatives. While the maximum total usage is limited, like any other utility, there are different use periods with different pricing, and there is a limit to total capacity, but you can always buy more by paying more to more of the providers.
- Your providers are highly competitive, and there are no major tie-ins, so that when one increases prices, reduces services, or becomes intolerable to your needs, you can instantly and transparently use them less and others more. But at the same time, there are global standards that assure interoperability, so you can move from one to another at any time. Providers who don't offer this lose business, while those who do, compete over price, performance, reliability, security, and other parts of the market.
- You don't have to worry about running complex IT infrastructure and building lots of customized applications, but as a side effect, you don't get everything quite your way. You get access to lots of outstanding and well thought out and designed applications and underlying mechanisms that others have thought through to convince you to keep using their services, and you can build your own services and offer them to others through the virtual global computing environment.

Of course the future is virtually here today. Except for the standards, interoperability, low cost access devices, workable global encryption scheme, automated distribution of resources across providers, global competitive market with enough maturity to count on but not so much that it stagnates, complete set of useful and compatible services suited to everyone from a microbusiness to the largest enterprises, lack of major tie-ins, and a few other things or course.

Is it early to jump in whole heartedly? It sure is. But on the other hand, if the user community can demand and get all or even most of these things, it becomes an increasingly compelling vision for the IT world of the future. Just think of how compelling it is for governments around the world, your competitors, and your customers. The question is not whether or not you can say 'Yes' to this future; it's “Where and when can you say 'Yes' to which of these futures?”