Virtualization and Cloud Computing

Table of Contents

Virtualization ................................................................................................................................... 2
Cloud Computing ............................................................................................................................ 3
Cloud Computing Service Models ................................................................................................... 5
Cloud Computing Delivery Models ................................................................................................. 8
Restricting Network Access ........................................................................................................... 11
Notices .......................................................................................................................................... 13
Virtualization

**Virtualization**

Allows you to run multiple systems on a single physical device

If attacker gets access to host system, may be able to compromise all virtual systems

Examples: VMWare, VirtualBox, Xen

**120** Now one of the big things that we run into today is virtualization. Now don't confuse virtualization with cloud computing; although the two mix pretty well.

When we talk about virtualization what we're saying is we want to run multiple operating systems at the same time on a physical piece of hardware.

We do this at the enterprise level to reduce overall cost and increase resiliency of the particular device.
We do this at the end user level primarily for testing; and that's pretty much where it ends.

But since virtualization is happening at lightening speed to do resource consolidation for large enterprises--well I do a little bit more testing on virtualized infrastructures.

There are all sorts of different virtualized infrastructures out there. VMWare is the king right now. VirtualBox and Xen, they're second fiddle to that.

**Cloud Computing**

**Cloud Computing**

Hosting services and data somewhere on the Internet, not locally

Security Concerns

- No physical or logical control over your data
- Don't necessarily know where your data is stored
- Data integration / segregation

**121 Okay so we talked about**
virtualization. Now let's talk about cloud computing.

Cloud computing is utility computing. Just like today when we walk up to the plug on the wall and say: I'd like to plug in my device; I demand electricity right now. When I unplug from it, I'm not paying from it because I'm not paying for it because I'm not using it anymore. So cloud computing allows us to treat computing like it is a public utility.

Also what we do is we don’t have any local infrastructure anymore; all we need is a really strong connection to the internet.

Now the problem is is now what we’re doing is is we're making somebody else in charge of our network here, of all of our services and servers here. Do we have good contracts in place to protect ourselves? Do they have good data integration? Do they have good segregation within that environment there? And how are we paying them for that?

By the way, if you're not paying a really high premium price for this Platform as a Service or Infrastructure as a Service, the likelihood of you getting separation and segmentation-- we'll call it separation at this point-- is pretty low.
Cloud Computing Service Models

Platform as a Service (PaaS) – providers allow applications to be created and ran on their infrastructure

Examples: Google Code, Amazon Web Services

Software as a Service (SaaS) – applications are run remotely over the web, usually on a subscription basis

Example: salesforce.com

Infrastructure as a Service (IaaS) – utilizes virtualization and you pay for what you use

Example: gogrid.com

*122 Let’s talk about the different basic models that are out there. There are three.

A lot of other people will come up with a whole bunch of other 'as a Service' concepts. But if we just look at this from the Cloud Security Alliance standpoint, there are three main ones that we pay attention to. Remember, this is all utility computing.

At the lowest level, under everything, if we don’t want to house servers here and we want to house them somewhere else, Infrastructure as a Service.
Now at that point you have total control over everything. All of the infrastructure whatsoever, it is yours; but also the vendor isn't doing anything more than allowing you to plug in, if you will.

Now Infrastructure as a Service is at the lowest. In the middle is called Platform as a Service. I really like this for small development shops that are happening today.

So inside of your organization, if you're a large enterprise, you may-- the software developers may say: You know, I’ve got a really good idea; but it's going to require that I spin off an implementation of Linux with this particular server and this particular service and these things running and not things running.

And they go to the IT Department and say: Hey can you help us out?

And they go: Yes, we'll see in about three weeks, five weeks, seven weeks; you know, because we-- it's not on our project timeline right now; we really can't do it for you.

Well that's too long for me. I need something now; now, now, now, now, now. Because somebody else has told me to test on this thing.

Well that now, now, now comes in the form of Platform as a Service.

So now what we're doing is if you don't have control of the underlying
infrastructure, we say these are the minimum requirements. This is how much RAM and hard drive and processor and CPU and memory that I need; and now I'm going to place--this specific platform, this operating system and these tools are going to be in place.

The vendor is now managing the underlying infrastructure; and they're also managing all of the configurations. So this particular MySQL version, this particular operating system with these configurations, they're handling that for us. And what that abstraction is going to allow us to do is to not have to worry about setting up all of the services there. And when we're finished we can just tear it right down.

The last one on our list is Software as a Service. We have no control over the infrastructure, no control over the platform. It is a web interface.

All of the web services that you use today, where you have no control underneath of that, that's Software as a Service.

A classic example of this in a complex environment is something like Salesforce.com.
Cloud Computing Delivery Models

**Private** – owned, managed and operated by the organization

**Public** – owned, managed, and operated by the provider for use by the general public – pay-as-you go model or free

- Webmail (gmail, yahoo mail, outlook.com)
- Document collaboration (Google Docs, OneDrive)

**Community** – provisioned to serve a specific group or community – owned, managed, and operated by one or more members or third-party

- A cloud shared among different parts of the government

**Hybrid** – combination of any of the above

- Most common is a combination of Private and Public clouds

**123 Cloud Computing Delivery Models.**
Now this, we take that infrastructure platform or Software as a Service--and how do we purchase it? Do we get somebody to outsource the entire thing for us; and they take care of it but they only take care of it for us? That’s private cloud. Normally we don’t see that that often.

What we normally see is the public cloud where all our data and somebody else’s data, including our competitors’ and our adversaries’ data, is all comingled here. But we have some sort of abstraction that doesn’t allow us to touch each other’s stuff. Hopefully that abstraction and
that security will remain in place. I don't think so.

And so on a public cloud it's utility services. If in theory the plug on the wall that we actually plug into-if in theory that plug on the wall made it so that all of the data that was flowing, all the electricity that was flowing out of there was completely the same, then public cloud would be perfect.

The problem is is it's not electricity; it's files, it's CPU, it's processing of our data, it's transforming of our data. And if somebody else can look at that transformation data, they could gain an advantage over us in the market.

The last one that we look at here is Community. Can you find a group of like-minded people that do what you do or are in the business that you are in?

This usually works with governmental agencies where a state agency will say: Hey I'm one state agency, and there are 30 state agencies or 300 state agencies--depending on how big your state is--and what we want to do is we want to have a cloud that's good for just us. And that's your community cloud.

In theory you could do this for anybody that joins a particular club. But I don't think that it's going to work out too well because the cost of implementing the infrastructure is relatively expensive and you're going
to have to get somebody to support that.

And then finally there’s the Hybrid. And I really like the hybrid idea; because what we can do is we can say: We need a private cloud for this subsection of what we're doing.

Let’s say it's all our services internally for us; and it’s all of our file services, all of our database services. We're going to outsource that to the cloud; but only we can access it on only our equipment.

Right next to that there’s a public cloud where your web, your mail and your DNS are sitting right out there, right next to it, in that public cloud.

So you could do the hybrid where you actually allow those web servers to pull data from your private cloud. It's a complex setup. But it is possible at this point.
Restricting Network Access

Network Access Protection (NAP) – Microsoft Windows scheme that requires computers on the network be in compliance with network patch policies.

Network Admission Control (NAC) – System that isolates devices which are not in compliance with policies.

NAP and NAC require a compliance audit server and a quarantine subnet to place non-compliant hosts in.

Both can (and should) be used with 802.1x and EAP for authentication.

**124 Next what we want to do is we want to talk about that concept of Virtual Private Networking.

And in Virtual Private Networking what we say is: When people are coming in and connecting to our resources, we want to control them.

Well that also happens-- well it happens when we’re dealing with network access, when we’re dealing with our guests that come in and connect to our web services. It happens when a vendor comes in, a contractor comes in and plugs into the wall.
How are we going to restrict their access and protect our environment and segregate these two groups of people? Well usually we'll use some sort of trusted network connection.

Trusted Network Connect, TNC, is the big overall term that encompasses these two terms that you see here: Network Access Protection, NAP, which is a software implementation; and Network Admission Control, or sometimes called Network Access Control-- depends on the vendor that you're dealing with.

In Network Admission Control we're doing it hardware wise. And in that hardware what we do is we isolate devices that are not compliant with the policy, based on the type of hardware that they have.

In Network Access Protection, Microsoft’s implementation-- and this only works with a subset of operating systems that are out there-- what we will do is we will say: Based upon the profile of this device at a software level we will or will not allow it to do certain activities.

Now both should use-- if you're choosing a solution that’s NAP or NAC, you should use 802.1x and Extensible Authentication Protocol as your authentication method. That really speaks to Radius at that point. Those are your two choices.

I don't think personally that the software choice is as good as the hardware choice, just from the
implementations that I've seen. I may be stilted in my opinion.

Notices

© 2015 Carnegie Mellon University

This material is distributed by the Software Engineering Institute (SEI) only to course attendees for their own individual study.

Except for the U.S. government purposes described below, this material SHALL NOT be reproduced or used in any other manner without requesting formal permission from the Software Engineering Institute at permission@sei.cmu.edu.

This material was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The U.S. government's rights to use, modify, reproduce, release, perform, display, or disclose this material are restricted by the Rights in Technical Data-Noncommercial Items clauses (DFAR 252-227.7013 and DFAR 252-227.7013 Alternate I) contained in the above identified contract. Any reproduction of this material or portions thereof marked with this legend must also reproduce the disclaimers contained on this slide.

Although the rights granted by contract do not require course attendance to use this material for U.S. government purposes, the SEI recommends attendance to ensure proper understanding.

THE MATERIAL IS PROVIDED ON AN "AS IS" BASIS, AND CARNEGIE MELLON DISCLAIMS ANY AND ALL WARRANTIES, IMPLIED OR OTHERWISE (INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, RESULTS OBTAINED FROM USE OF THE MATERIAL, MERCHANTABILITY, AND/OR NON-INFRINGEMENT).

CERT® is a registered mark owned by Carnegie Mellon University.