Cloud Computing and Confidential Data:
An Analysis of Security Implications and Mitigation Strategies

Stuart Hutchings
ICPSR, University of Michigan
Ann Arbor, MI 48104, USA

Felicia LeClere
NORC, University of Chicago
Chicago, IL 60602, USA

and

Bryan Beecher
ICPSR, University of Michigan
Ann Arbor, MI 48104, USA

ABSTRACT

The Inter-University Consortium for Political and Social Research (ICPSR), with funding from the National Institutes of Health, assessed the feasibility of using public cloud computing infrastructure to enable secure and controlled access to confidential research data. Our first step was to assess usability and functionality, and our second step, which we report on here, was to assess whether the computing environment could be managed securely. To ensure an objective assessment of our security risk profile we contracted the services of an independent ethical hacker (or "white hat") to evaluate the effectiveness of our strategy. While there is more work to do, as security risk mitigation is a continual effort, we feel confident cloud computing offers a grand opportunity to more effectively leverage IT resources. This affords education and research institutions an opportunity to expand their vision how their data is utilized without increasing risks to their intellectual and fiduciary responsibilities. In this presentation, we discuss the findings of the hacker's evaluation of our approach to managing security in the cloud.

Keywords: Cloud Computing, Security, Confidential Data, Ethical Hacker

1. INTRODUCTION

This report provides analysis from a project at the Inter-University Consortium for Political and Social Research [1] funded by the National Institutes of Health, Office of the Director [2] through the Challenge Grant Program [3]. The primary goal of the grant was to test whether confidential data, that is data distributed under license or contract, could be effectively and safely disseminated via the computing cloud. Historically, data licenses and contracts put the burden of securing data files on the user. This often involves elaborate data security plans that may involve purchasing new technology or securing existing networks and machinery. This grant tested whether dynamically configured temporary computing environments in a cloud would provide users a secure environment in which to analyze confidential data.

What is cloud computing?

Cloud computing [4] is defined as the delivery of elastic computing resources (e.g., networks, servers, storage, applications, and services) as a metered service rather than a specific product.

There are three general service segments or models, known as "Software as a Service" (SaaS), "Platform as a Service" (PaaS), and "Infrastructure as a Service" (IaaS). Below are examples of each:

- SaaS – Yahoo Mail, Google Docs
- PaaS – Google Apps, Force.com
- IaaS – Amazon Web Services (AWS), RackSpace

Each approach offers benefits of flexibility, scalability and 'pay as you go' costing. And each has risks, some relating directly to security.

ICPSR Experience with Cloud

ICPSR has been utilizing the capabilities of several public cloud providers since 2009, generally for fail over and replication of select functions such as DNS, our search service, and encrypted storage of copies of non-confidential archived data. Based on this experience we concluded early in the project that an optimal cloud-based computing environment for our use case would be very similar to traditional solutions, and that we would be able to leverage existing expertise, skills, tools and management infrastructure. The question became would the threat model also be similar in nature or would cloud introduce unique additional risk vectors.

Scope of this analysis

This project was not intended to create a 'perfect' security solution; rather we sought to define the risk specific to utilizing a computing cloud. Independent of this project, existing ICPSR archives have been assessed as requiring our IT systems to meet the FISMA [5] Impact Rating of Low, presumably in part due to practices in place to anonymize the data. We, however, did not limit our vision and sought to understand what was possible with current technology. Thus we assumed from the outset that what we developed may not be impenetrable and that this would most likely be an iterative exercise, common to any IT security management effort. We understood there would mostly likely be inherent vulnerabilities of some of the components we would use, such as Microsoft Windows, but that these vulnerabilities were a concern independent of a computing cloud infrastructure as we would need to address them as well with traditional alternatives. Also out of scope of this paper is the web based management system we developed for submitting, reviewing and approving requests for access to confidential data archives.

2. OUR ASSESSMENT OF RISK

There are many articles and papers on the subject of the benefits and risks associated with cloud computing or specifically with public cloud providers [6],[7],[8]. Among these are issues such as loss of governance, risk to compliance (regulatory, licensing), co-location of compute resources and storage, etc. A general discussion and understanding of these is necessary when planning to utilize the services of any provider. For the purpose of this analysis, we focused on the technical requirements with the intent, should the project prove feasible, a production implementation plan would review the compliance requirements, among other legal and policy considerations. It is worth noting that a common theme we have observed in the discussion of public cloud is the perception of loss of control, which we theorize is at least in part due to a lack of physical access and or proximity to the computing resources. Our experience has taught us that any network attached computer, be it Internet, Intranet, or locally bound, is potentially at risk and that therefore appropriate precautions are necessary.

We elected to utilize Amazon's Web Services (AWS) [9] as our public cloud provider. We have been utilizing their offerings since 2009. Selecting AWS addressed several of the potential concerns we identified with public cloud providers. Their security, in particular management of their data centers and compliance with various Federal and international standards, [10][11][12] very likely exceeds anything individual educational or research organizations could or do provide. Additionally they offer a service which we felt would help address the risk associated with co-location, as will be explained later in the paper.

We sought to provide a similar experience for our users to what we understood them to be accustomed to today; that being a Microsoft Windows desktop in their office on which they access and analyze the confidential data. Whether our solution utilizes Citrix [13], VMware [14], or Microsoft's Remote Desktop Service [15] or Hyper-V [16], in each scenario fundamentally we are describing delivery of a Microsoft Windows desktop experience and the security risks and limitations associated with that operating system. In general, the delivery mechanism does not compensate or eliminate those concerns, so it is our belief that whether this was a local implementation or one hosted in AWS, the security ramifications of Windows remain generally the same. Therefore this required the security and management components associated with Windows, such as Identity Management, Configuration Management, Patch and Vulnerability Mitigation and so forth.

3. OUR PROPOSED SOLUTION

We elected to create a template computing instance using Microsoft Windows 2008 R2 server running the Remote Desktop Service (RDS) as a host, formerly known as Terminal Server, residing in the AWS Elastic Computing Cloud (EC2). RDS utilizes a Remote Desktop Protocol (RDP) connection (via port 3389) between a client and the server to provide access to applications or in our case a full desktop. It is essentially what is known as a "thin client" solution. Installed upon each server would be the applications requested by the researcher necessary for their analysis work and a copy of the requested data, they would connect remotely from their local desktop computer. Now RDP is not a very secure protocol [17] so to mitigate that concern, we would create a Windows 2008 R2 server running Remote Desktop Service in the Gateway role. This provides the ability to tunnel RDP through an HTTPS (port 443) connection and so the RDS Gateway could be set up as the only entry point to the RDS Hosts, managing access via policies and firewall rules as to what resources individual researchers could access while minimizing the risks associated with RDP. With this model, we create an instance of the RDS Host server from the master template, add any additional software requested, a copy of the data requested, and configure rights of the identified users.

AWS EC2 VPC

Unfortunately this approach, while addressing our desire to provide a comparably user experience to current practice, still exposed us to not only our identified risks but also the management challenges of configuring individual accounts on individual servers and creating associated access policies on the RDS Gateway. Operational this is not wise and prone to human error, a common security risk. Therefore we elected to utilize the AWS Virtual Private Cloud (VPC)[18] offering. This is effectively a cloud within a cloud. It allows for creation of a dedicated secure connection using an IPsec tunnel between a existing legacy infrastructure, such as ours, and an isolated segment of AWS compute resources. This permitted us to create a small virtual private network (or networks) within AWS strictly for our use. The benefits include the ability to prohibit communication between computing resources within AWS, restrict access to storage we've allocated to only addresses within our network, and control via security groups and firewall rules the extent to which this VPN can access (and be accessed by) our Intranet based resources. While it is now technically feasible to both permit direct Internet connection through AWS to this VPN and through our legacy Intranet, this would greatly reduce the benefit to us of using this approach and therefore we did not opt to utilize that particular capability.

By effectively extending our Intranet to this isolated segment of Amazon's infrastructure, our campus network and security services were positioned to proactively monitor the VPN for unusual activity. We elected to place the RDS Gateway on physical hardware in our local data center and restrict access to the VPC and our Intranet to strictly the services and protocols necessary for managing the RDS Hosts we would instantiate as needed and to permit the RDP communications between the RDS Gateway and the Hosts initiated by the researchers remotely via the Internet. No direct Internet access is permitted in our configuration to the VPC. Overall this greatly reduced our exposure to co-location concerns and theoretical attacks, and it provided us the means to leverage our on-premise resources, such as:

Identity Management utilizing our University's sponsored account system [19] for provisioning and managing user accounts, rights, password quality, and self-serve support for password changes and resets.

Configuration Management by joining these servers to our production Active Directory domain, thus providing us the ability to manage the configuration via Group Policies. This permits us to ensure the servers capabilities, user accounts, and accessible functions are limited to just what is necessary to a) support the RDS Host role and b) provide a reasonable user experience. It also allowed for adjustments as needed or as new threats are identified, and is configured to actively monitor (and correct) for alterations to these defined settings.

Patch Management through our campus hosted implementation of Tivoli Endpoint Manager for Patch Management [20] to distribute and implement Microsoft security patches and 3rd party software updates. This permits us to ensure the servers are current on patching to minimize our exposure to known vulnerabilities.

End of Life

Once the research request is complete, it is equally important we take adequate precautions with removing the data from the computing cloud’s storage. To that end, using an automated process we return the data via an encrypted connection to our on premise storage. After analyses to ensure any results do not contain portions of the original confidential data, the results would be provided directly to the researcher. The storage allocated with the server instance in EC2 would be scrubbed and overwritten several times before being de-provisioned. At which point the server would be deleted.

Miscellaneous

Intentionally, we elected to not include virus protection, anti-malware, or data encryption software with the RDS Host master image. Our concerns were mostly related to the performance impact and the belief that our approach of maintaining a current master image and no direct Internet access would significantly minimize the risks these technologies would protect against. We also anticipated the cost to performance could undermine the acceptance rate of this solution by the researchers. However we were prepared to alter that decision should the ethical hacker determine that some or all of these components were necessary to address vulnerabilities he/she would expose. Figure 1 provides a visual representation of our model.

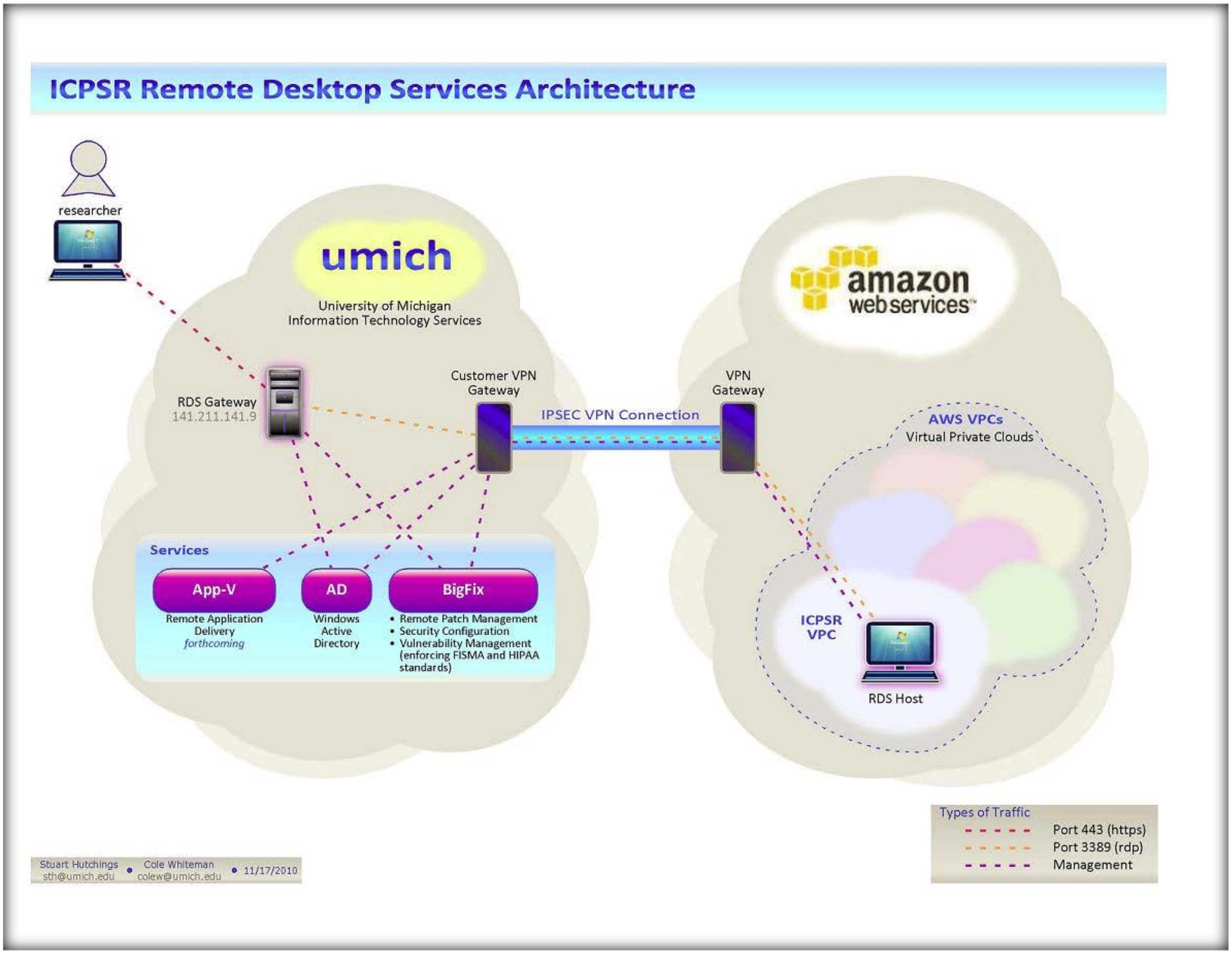

Figure 1

4. ETHICAL HACKER ANALYSIS

ICPSR contracted the services of an independent ethical hacker, also known as a “white hat”, to assess our model. The contractor is currently a Ph. D student at Johns Hopkins University performing applied cryptography research and has several years of professional experience working as a security and privacy evaluator. We presented him with two tasks: 1) Analyze and assess the threat vectors of this model; and 2) Perform penetration testing based on these threats to evaluate what security vulnerabilities it possessed and recommend modifications to remediate. We did not provide him with our assessment of the threat vectors, as we did not want to limit nor influence his own analysis.

He was given a general overview of our approach, an account that would be typical of a researcher and instantiated an RDS Host server (with an example set of non-confidential data) for analysis and penetration testing.

Results

During the fall of 2011 he performed this work and submitted his final report to us in December 2011. Below is a listing of threat areas he identified, the results of his testing, recommendations for enhancements and how we responded. In total he identified five threat vectors. For the purposes of this paper, we have summarized his analysis of the threats, their description and the details of the vulnerabilities identified but we have included all concerns raised.

Limitations

The penetration testing analyzed the overall security of the system with respect to an active attacker given access to a cloud-provisioned virtual machine. Such an attacker had limited privileges on the virtual machine and the threat model assumed that the attacker would be able to perform attacks from within the virtual machine and also that the attacker would be able to perform attacks on the connection between his or her computer and the virtual machine. The threat model did not take into account the possibility of an attacker using his or her host machine to attack other computers on the University of Michigan network. Such an attack would be outside the scope of this analysis as an attacker with direct access to the University of Michigan network may propagate all sorts of attacks against the ICPSR infrastructure as well as against any other departments within the university.

Attack Vector – Access Control Policies

Users should have read/write access to their home directory and its sub directories but no access to any outside directories. In particular, users should be explicitly prohibited from accessing system directories or other users’ home directories. Users should be prohibited from accessing any programs that have not been explicitly configured for their use. Finally, users should be denied access to any Turing complete programming languages that may be installed on the virtual machine.

Findings identified several ways an attacker could compromise the ACL system. It should be noted that most of these were actually due to bugs in the Windows ACL system itself, and thus cannot be sufficiently mitigated. This included: the ability to browser the local file system, though no ability to read other users directories [high risk]; access PowerShell [high risk]; execute shell commands via batch scripts [moderate risk]; good password strength requirements but insufficient length, which could be cracked given sufficient time [low risk]; and while all standard means of copying off data were unavailable he was able to demonstrate it was plausible using DNS as a covert channel, however he deemed that also of low risk.

Recommendations included disabling the ability to browse the file system; uninstall PowerShell; disable execution of batch scripts; and seek a solution for the DNS covert challenge, and eliminate external DNS queries.

Our Response was to disable browsing of the file system, modify rights to access PowerShell as it cannot be removed, disable execution of batch scripts via user access controls, and are actively investigating how best to address the DNS concern.

Attack Vector – Network Security

Network security has two major parts: protocol-based security in which the security of the RDP protocol itself is analyzed and virtual host based security in which the security of each network-facing service on a given VM is analyzed. In the case of the former he looked at packet sniffing, SSL, man-in-the-middle, and session stealing attacks. In the case of the latter he enumerated and analyzed running services on a VM and also on the RDP gateway.

There are several types of attacks that are commonly perpetrated against network protocols. Packet sniffing attacks are passive attacks in which a malicious user observes all traffic traveling between a host and a server. In egregious cases these attacks can leak sensitive data such as usernames and passwords. In situations where such sensitive information is transmitted, good protocols typically use SSL in order to secure their connections using cryptography so that a passive attacker cannot see the data being transmitted. SSL works by having a server transmit a signed SSL certificate containing its public key as well as a list of supported key negotiation protocols. If a client decides that it trusts this public key, the client then sends its public key as well as a list of key negotiation protocols that it supports. The server will pick the most secure key negotiation scheme that both the client and server support. Recently, an attack on SSL has been released called BEAST. BEAST exploits an improperly implemented cryptographic scheme in the SSL protocol and it allows messages encrypted as part of an SSL session to be decrypted by an active attacker. SSL is also vulnerable to a type of attack called man-in-the-middle attacks where an active attacker attempts to sit in the middle of an SSL connection, pretending to be the server to whom the user wants to communicate. If a user accepts the attackers pubic key as the server’s, the user will believe that he or she is communicating over a secure channel where none actually exists. The attacker will initiate an SSL connection with the user and a separate connection with the server, which will believe it, is legitimately talking to the user that initiated the connection. Session stealing attacks occur when a user attempts to re-open a closed session between a client and a server, with itself in place of the client. Such an attack may occur when a client has uncleanly closed a connection to the server but where the server has not yet timed out the connection. In this case, an attacker can often reuse the old session key and re-open the connection to the server.

Findings indicate he was able to perpetrate attacks of varying severity against individual virtual hosts and against the Remote Desktop Services Gateway. One specific area of concern was the use of TLSv1, which is utilized for RDP connection for security, as it is considered broken at least in the sense that previous attacks have been developed that can compromise it. However, those attacks do require actions on the part of the user which would be very difficult to perform. Through one of his attacks he was able to compromise the default Administrator account of the RDS Host.

Recommendations include configuring RDP to warn about invalid certificates and not connect by default.[High]. Configure the firewall to block TCP connect scans [Medium]. Disable services of the Gateway that are not absolutely needed for the intended functionality, which he understood would take time to assess. Investigate whether IIS 7, required by the Gateway can be upgraded [High]. He also suggested we consider removing TLSv1 and establishing an alternative using TLSv1.1 or 1.2, though he admitted this may not be an option. [Medium]

Our Response is that we would modify the configuration of RDP to warn on invalid certificates. As to the rest, we are assessing the impact of the proposed changes. If they would impact the operations of our solution, we will seek alternative mitigation strategies. As to the attack on the Administrator account, we will implement a change that will generate a new unique complex password on a daily basis using an internally known common key. Should we experience an event where an attacker is able to launch a brute force attack on this account, the window time he/she would require to complete the exercise may not be sufficient and the information gained quickly rendered useless.

Attack Vector – Virus, Malware, and Trojans

There are several categories of computer viruses, worms and trojans that VMs must be robust against. Attacks of this nature are typically designed to leverage human behavior to initiate and propagate the attack, and known weaknesses of versions of services of the operating system being attacked. In addition to concerns about viruses propagating the network, there is the additional concern of attack sites that can be used to perform various nefarious actions using a victim's web browser.

Findings indicated he was unable to find attack vectors for this category of attacks due to the security model in place on the virtual host. While a virus or malware is of low risk, he felt the harm that could be caused should it occur could be catastrophic and therefore anti-viral/malware offering should be added to our environment as a precaution. Additionally, he was able to identify an incorrectly configured setting in our disabled web browser that could permit to configure the browser to trust a compromised site and via scripts execute an attack using handcrafted HTML files.

Recommendations included installing an antivirus solution [Low risk] and correcting the Internet Explorer script execution setting [Medium risk] and then prevent changes to the browser options [Medium risk].

Our Response is that we will pursue the changes he recommends though we have not determined how we will configure the anti-virus software, as we would like to minimize the impact to the user experience.

Attack Vector – Denial of Service Attacks

Denial of service (DOS) are malicious attacks propagated with the intention of making a network resource unavailable for use for an extended period of time. They cause lost productivity for users and headaches for administrators. Unfortunately, they are also rather difficult to protect against. There are three classes of DOS attacks that I looked at: packet flooding attacks, distributed packet flooding attacks, and account lockout attacks.

Findings showed our server configuration was extremely vulnerable to packet flooding DOS attacks. In the course of this test, he identified that we had not configured a lockout of the users account should too many attempts of a bad user name/password combination occur. Ironically if we elect to create a rule to lock account on too many attempts, it would in turn open us up to a different approach for performing a DOS attack.

Recommendations included configuring our firewall to detect packet flooding and to block IP addresses of perpetrators [High]. He also suggested implementing our account lockout by IP addresses for too many failed login attempts rather than user accounts (this can be done using a state full Firewall) [Low].

Our Response was to implement the suggested firewall configuration modifications and to implement account lockout rules in the manner he proposed.

Attack Vector – Virtualized Hardware Security

Perhaps the most catastrophic on a virtual machine is one in which an attacker is able to escape the VM and attack the underlying VM host. Such attacks are extremely dangerous because once an attacker has escaped a VM jail he or she can then attack any other VMs on the machine and can steal all sorts of secrets and credentials. The good news is attacks that allow an attacker to compromise a VM host from within a VM are relatively few and far between. When they are found, however, the damage that they can cause is lasting and serious.

In addition to VM breakout attacks, another security concern is the idea that an attacker may compromise the VM server itself and then have access to all of the virtual disks stored on it. In such a case an attacker can use standard forensics tools to capture important data from the hard drives while being able to completely ignore any ACL systems in place. This would allow an attacker to capture password hashes for all user accounts and to access any confidential data stored in user accounts.

In theory, attackers may also conceivably be able to connect virtualized hardware to a VM in order to compromise the security. In practice however, I found no way to perpetrate such an attack using a cloud environment, so I think this attack vector is unlikely.

Finally, there is also the threat that an attacker will analyze virtualized memory dumps in order to recover encryption keys and passwords that have been stored in memory. While these attacks not exactly commonplace, I have personally carried them out on memory dumps with much success by using tools introduced in a paper on Cold Boot Attacks from the 2008 USENIX conference proceedings.

Findings while he was able to perform some basic foot printing within the virtual server, such as determining the versions of products Amazon was running for the cloud hypervisor, he could not find critical information presumably useful in performing an attack, such as the IP address of the underlying server from within the VM and could not perform any attacks to break out of the VM either. In order to attack this system from the outside, an attacker would need to perpetrate an attack on the Amazon EC2 infrastructure itself which he felt was an exceedingly bad idea if the attacker's goals include not getting caught. Since he wasn't able to break out of the VM he also wasn't able to access the underlying virtual disk store or virtualized dumps of VM memory from when the machine was suspended. Additionally his research determined there are, at least currently, no known attacks that are useful for breaking out of the hypervisor software utilized by Amazon. This does not mean that such attacks will never happen and, in fact, such attacks have been reported in the past. Anticipating and reacting to such attacks is therefore rather difficult. The best way to protect against such attacks is to ensure that all VMs are patched with the latest version of the virtualization tools. In addition, having such stringent access controls as are existent in the EC2 environment as-is will make exploiting any such security holes a monumental task and will be a serious barrier for even the most determined attacker.

Recommendations included keeping the hypervisor tools updated [high risk] and to consider VM-level encryption as a precaution [Medium].

Our Response we agreed with his observations and are considering the risk/reward of adopting encryption, it would reduce our exposure should we be compromised but at a high cost to performance and the user experience.

5. CONCLUSIONS

The use of an independent ethical hacker to assess our security efforts was critical to the success of this project. In general, we were not surprised by the observations made by the ethical hacker. Even the rather basic errors he discovered in configuration which impacted security, although admittedly a tad embarrassing, demonstrated that delivery of this system still depends on the same diligence and approach to security required with managing our own infrastructure and resources. The issues he identified were almost entirely challenges we would face and issues we would have had to protect against whether this was locally hosted, using our on premise physical infrastructure, or remotely hosted at a public cloud provider. Admittedly more effort will be needed to ensure each use case's regulatory and compliance concerns are or can be addressed. And while we believe we satisfied the concerns presented by co-location with other clients of a public cloud provider, it is reasonable to assume further efforts made me needed if a higher level of isolation is demanded for specific confidential data. However, our results affirmed our belief that institutions such as our own can responsibly utilize cloud and public cloud providers.

Looking ahead, we see opportunities for enhancing the security of our cloud hosted assets through implementation of proactive monitoring software, two-factor authentication, utilizing additional functionality available in our existing management products that can, for example, audit and remediate security settings based on Federal defined standards. Further, we see value in evaluating technologies that can assess the devices of researchers attempting to access the confidential data to ensure them meet a defined specification, for example that their operating system is fully patched. And we believe the white hat exercise should be repeated in the coming months with the intent of utilizing a different hacker, with a different perspective and skill set.

We encourage educational institutions to assess the value proposition of the computing cloud.

6. REFERENCES

[1] **ICPSR**
http://www.icpsr.umich.edu/icpsrweb/ICPSR

[2] **NIH, Office of Director**
http://www.nih.gov/icd/od/

[3] **Challenge Grant**
http://grants.nih.gov/grants/funding/challenge_award/

[4] Peter Mell, Timothy Grance, "**The NIST Definition of Cloud Computing"** NIST Special Publication 800-145. September 2011.
http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[5] **Federal Information Security Management Act (2002)**
http://csrc.nist.gov/groups/SMA/fisma/index.html

[6] "**Cloud Computing Security Risk Assessment**", European Network and Information Security Agency, November 2009.
http://www.enisa.europa.eu/act/rm/files/deliverab les/cloud-computing-risk-assessment

[7] "**Top Threats to Cloud Computing**", Cloud Security Alliance. March 2010. https://cloudsecurityalliance.org/research/top-threats/

[8] S Subashini, V. Kavitha. "**A Survey on Security Issues in Service Delivery Models of Cloud Computing"** Journal of Network and Computer Applications. July 2010.
http://www.sciencedirect.com/science/article/pii/S1084804510001281

[9] **AWS**
http://aws.amazon.com/

[10] **AWS Federal Government** http://aws.amazon.com/federal/

[11] **AWS Security**
http://aws.amazon.com/security/

[12] **AWS SLA**
http://aws.amazon.com/ec2-sla/

[13] **Citrix**
http://www.citrix.com/lang/English/home.asp

[14] **VMWare**
http://www.vmware.com/

[15] **Microsoft Remote Desktop Services** http://www.microsoft.com/en-us/server-cloud/windows- server/remote-desktop-services.aspx

[16] **Microsoft Hyper-V**
http://www.microsoft.com/en-us/server-cloud/hyper-v-server/

[17] Rob VandenBrink, **"Cyber Security Awareness Month – Day 9 – Port 3389/TCP (RDP)"**, Internet Storm Center, October 2009.
https://isc.sans.edu/diary.html?storyid=7303

[18] **AWS VC2**
http://aws.amazon.com/vpc/

[19] **MCommunity** http://www.itcs.umich.edu/mcommunity/forms/

[20] **Tivoli Endpoint Manager**
http://www-01.ibm.com/software/tivoli/solutions/endpoint/index.html