This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2415794, IEEE Transactions on Cloud Computing

Cloud-Trust - a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds

Dan Gonzales, *Member, IEEE*, Jeremy Kaplan, Evan Saltzman, Zev Winkelman, Dulani Woods

Abstract— The vulnerability of Cloud Computing Systems (CCSs) to Advanced Persistent Threats (APTs) is a significant concern to government and industry. We present a cloud architecture reference model that incorporates a wide range of security controls and best practices, and a cloud security assessment model – Cloud-Trust – that estimates high level security metrics to quantify the degree of confidentiality and integrity offered by a CCS or cloud service provider (CSP). Cloud-Trust is used to assess the security level of four multi-tenant IaaS cloud architectures equipped with alternative cloud security controls and to show the probability of CCS penetration (high value data compromise) is high if a minimal set of security controls are implemented. CCS penetration probability drops substantially if a cloud defense in depth security architecture is adopted that protects virtual machine (VM) images at rest, strengthens CSP and cloud tenant system administrator access controls, and which employs other network security controls to minimize cloud network surveillance and discovery of live VMs.

Index Terms— Cloud computing, cyber security, advanced persistent threats, security metrics, virtual machine (VM) isolation

1 Introduction

The flexibility and scalability of CCSs can offer significant benefits to government and private industry [1][2]. However, it can be difficult to transition legacy software to the cloud [3]. Concerns have also been raised as to whether cloud users can trust CSPs to protect cloud tenant data and whether CCSs can prevent the unauthorized disclosure of sensitive or private information. The literature is rife with studies of CCS security vulnerabilities that can be exploited by APTs [4] [5][6][7].

Virtualization, the basis for most CCSs, enables CSPs to start, stop, move, and restart computing workloads on demand. VMs run on computing hardware that may be shared by cloud tenants. This enables flexibility and elasticity, but introduces security concerns. The security status of a CCS depends on many factors, including security applications running on the system, the hypervisor (HV) and associated protection measures, the design patterns used to isolate the control plane from cloud tenants, the level of protection provided by the CSP to cloud tenant user data and VM images, as well as other factors.

These concerns raise questions. Can the overall security status of a CCS or a CSP offering be assessed using a framework that addresses the unique vulnerabilities of CCSs and can such assessments be applied to alternative CCS architectures and CSP offerings in an unbiased way? The federal government has issued security controls that CSPs must implement to obtain FEDRAMP CCS security certification [8] that are based on National Institute of Standards and Technology (NIST) cloud security guidelines [1]. However, these do not provide high-level decision-makers with an overall assessment of CCS security status or the degree of confidentiality and integrity offered by specific cloud architectures [9].

The main contributions of this paper are to develop a CCS reference architecture and a cloud security assessment model – Cloud-Trust – that provides quantitative high level security assessments of IaaS CCSs and CSPs. Cloud-Trust can assess the relative level of security offered by alternative CSPs or cloud architectures. Cloud tenants can use it to make decisions on which CSP security options or cloud security features to implement. We illustrate the use of Cloud-Trust by applying it to the case where the cloud tenant is a U.S. government agency and examine how well four alternative CCS architectures protect U.S. government data.

Cloud-Trust is based on CCS unique attack paths that cover the essential elements of an IaaS cloud architecture. It is based on a Bayesian network model of the CCS, the class of APT attack paths spanning the CCS attack space, and the APT attack steps required to implement each attack path. It provides two key high-level security metrics to summarize CCS security status quantitatively:

- Probability an APT can access high value data
- Probability the APT is detected by cloud tenant or CCS security monitoring systems

The first security metric estimates whether high value data (designated as "Gold" data in this paper) is likely to be compromised or erased from the CCS. The second metric assesses whether the CSP provides cloud tenants sufficient CCS network monitoring, file access, and situation awareness data to detect intrusions into a tenant's cloud network, and whether the tenant's security and monitoring systems contribute to the intrusion detection.

This paper is organized as follows. Section 2 discusses trust zones. Section 3 presents a cloud reference model and cloud security control features. Section 4 describes CCS unique attack paths and vulnerabilities that can be exploited by APTs. Section 5 describes Cloud-Trust. The final section provides Cloud-Trust results for four alter-

- Manuscript received 4 April 2014, revised October 2014. Published 2015.

- *Daniel Gonzales, Jeremy Kaplan, Evan Saltzman, Zev Winkelman, and Dulani Woods are with the RAND Corporation, Arlington VA, 22202 (e-mail: [gonzales, saltzman, jkaplan, zwinkelm, dwoods]@rand.org).*
- *Evan Saltzman is with the The Wharton School, University of Pennsylvania, Philadelphia, PA 19104 (e-mail: esalt@wharton.upenn.edu).*

.

native cloud architectures, and describes how CloudTrust can be used to assess the security capabilities of alternative CSP offerings.

2 Physical And Virtual Trust Zones

We define a trust zone (TZ) as a combination of network segmentation and identity and access management (IAM) controls. These define physical, logical, or virtual boundaries around network resources. Cloud TZs can be implemented using physical devices, virtually using virtual firewall and switching applications, or using both physical and virtual appliances.

IAM systems use usernames, passwords, and access control lists (ACLs), and may use Active Directory Domain Controllers [10], Federated Trusts [11], and multifactor authentication mechanisms using time limited codes or X.509 certificates. IAM servers can also use hardware information to make access decisions. For example, devices without a pre-validated MAC address can be prevented from joining a network. Routers using ACLs and IP address white listing can prevent an unauthorized device from accessing network resources. These are examples of hardware based TZ enforcement.

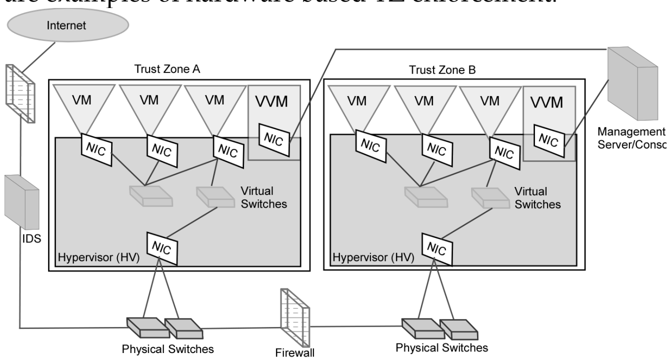


Fig. 1. CCS Network Segmentation Scheme

An example of a more complex network CCS segmentation scheme is shown in Fig. 1. It uses defense in depth approach to restrict network connectivity to VMs running in a CCS. Both real and virtual Network Interface Cards (NICs) are used to isolate network segments. The network segmentation approach is based on the virtual networking capabilities offered by VMware in their ESX HV [12]. It enables a hybrid strategy that uses both virtual network and physical firewall barriers to protect information in TZs A and B shown in the Fig. 1. Amazon Web Services (AWS) offers a similar capability called Virtual Private Cloud (VPC) [13]. An APT attack with the goal of exfiltrating data at rest on a resource in TZ B in Fig. 1 must first circumvent the network segmentation and establish network access to the target resource. By staging the attack from a trusted IP address (whitelisted by the firewall(s) protecting that zone), the attacker may gain network connectivity to the target. Assuming the data at rest is encrypted and brute force decryption is not feasible, the attacker must also gain access to the credentials and keys required to decipher the data. This access is typically governed by policies and accounts on the domain controller. Access is granted for legitimate requests from users that have been authenticated and who are authorized. Successfully spoofing these requests, or otherwise gaining access to the keys after access has been granted to a legitimate user, would provide the attacker with the ability to decrypt the data.

Compromising data from TZ B in Fig. 1 while it is in flight presents different challenges. Data in flight may transit other segments of the network with lower barriers to access for the attacker. For example, if a server in TZ A retrieves data from TZ B, the data is now in this less protected zone, and may be diverted or copied and transmitted over the Internet. If the data is in flight using a protocol that does not guarantee end-to-end encryption such as SOAP, and instead uses point-to-point transport level encryption such as REST over HTTPS, the data will be decrypted at various points in transit, possibly in memory, before it reaches the application layer at the destination endpoint. On the other hand, relying on capturing data in flight makes it much more difficult to compromise the entire dataset.

The security of TZ implementations depend on correctly configuring domain controllers, firewalls, routers, and switches that are used in segmenting and restricting access to portions of the cloud network and on “locking down” secure communications between users and domain controllers to prevent SOAP interface or signature wrapping attacks [14]. Misconfiguration of IAM servers, domain controllers and other network devices can introduce vulnerabilities in the cloud network and let attackers enter restricted TZs. Careful configuration management is a key factor that must be taken into account in assessing cloud security status. To ensure such vulnerabilities are not inadvertently created in a CCS well trained system administrators (sys-admins) are needed to set up, maintain, and correctly patch this infrastructure.

3 CCS Reference Model And Architectures

This work is limited to one cloud deployment model, infrastructure as a service (IaaS) clouds. The layers of the software stack below the Guest OS are under the control of the IaaS CSP: the virtual machine manager (VMM), HV, computing and storage hardware, and the CCS network. Only the guest OS that forms the foundation for VMs is assumed under the control of cloud tenants. IaaS cloud tenants provide their own applications and data. The Guest OS may be specified by the CSP policy, or control of the guest OS configuration may be shared between the CSP and cloud tenant. Because of the shared control of the IaaS cloud software stack the security profile and status of the CCS depends on both CSP and tenants.

The CCS reference model is shown in Fig. 2. CSP management and security servers are segregated from cloud tenant VMs by subnets, firewalls, domain controllers, and internet access points. Tenant VMs are networked using a software defined network (SDN) shared by all cloud tenants. A CSP domain controller controls access to virtual TZs used by cloud tenants. TZ gold, which contains more valuable Agency data, is housed within Agency TZ A. This provides multiple access control boundaries to prevent external cloud users, for example from the tenant B TZ, from accessing data in the Gold TZ.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2415794, IEEE Transactions on Cloud Computing

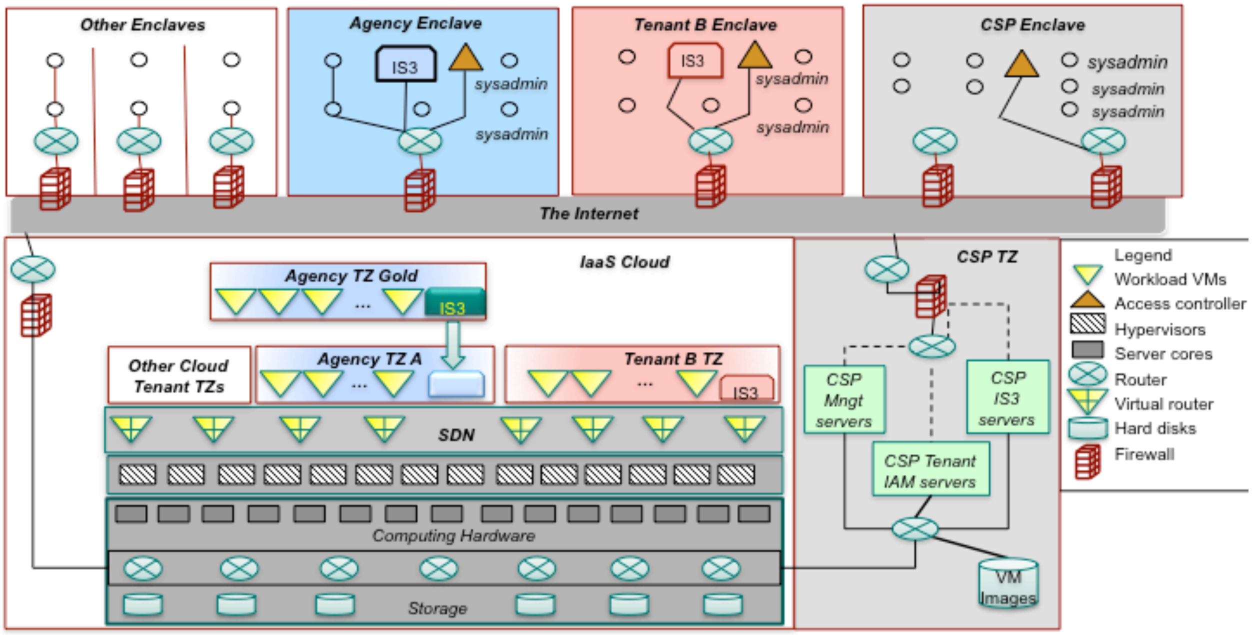


Fig. 2: CCS Reference Model

The CSP TZ is segregated from tenant TZs and contains cloud management servers, SDN controller servers, CSP tenant IAM servers, and CSP Information System Security System (IS3) servers. CSP sys-admins communicate with CSP management systems through a separate firewall and Internet port to isolate CSP communications traffic. It is a best practice to isolate CSP management and monitoring systems from cloud tenant VMs, as illustrated in Fig. 2 [15]. Our cloud reference model is based on this best practice and design tenets developed by the Defense Information Systems Agency (DISA) for securing enterprise networks [16].

Early information systems were designed largely to manage computing resources, apportion costs, and improve performance. As cyber threats grew, enterprise network security capabilities grew in an attempt to keep pace with the threat. Modern firewalls block IP ports and protocols and inspect packets. They also include host-based Intrusion Detection Systems (IDSs), keystroke logging, reverse web proxy servers, DMZs, IAM servers, security incident event managers (SIEMs), and other more exotic detection and protection systems. Network performance monitoring tools, such as Netflow, and log file analyzers are used to identify suspect data flows or configuration changes, and automated software distribution systems rapidly patch OS installations and applications. Cyber security systems have been adapted so they perform similar functions in CCSs, although virtualization presents new challenges to both the attacker and defender.

We call the cloud systems that detect and prevent the actions of malware and bad actors the Information System Security System (IS3). IS3 systems can generate lots of data and have high false alarm rates. Well-trained sys-admin personnel are needed to monitor and manage IS3 servers. A cloud IS3 includes IDSs, host based security systems, fire -walls, IAM servers, reverse proxy web servers, syslog servers, and SIEM servers (all capable of functioning effectively in a virtual environment). The SIEM aggregates event data produced by security devices, network infrastructures, systems and applications. Event data is combined with contextual information about users, assets, threats and vulnerabilities. The data is normalized, so events, data and contextual information from disparate sources can be correlated and analyzed for specific purposes, such as network security event monitoring, user activity monitoring and compliance reporting. Fig. 2 shows the location of IS3 servers used by the CSP, the Agency, and other tenants. We assume tenants provide their own IS3s to monitor and manage their TZs.

System protection and risk reduction involve numerous actions not performed directly on the CCS. These include physical protection measures, vetting employees, security awareness training, maintaining a vulnerability management data base, and participating in national vulnerability organizations and fora (e.g., SANS). We do not include employee training or vetting activities in Cloud-Trust, but note they are important for securing CCSs and CSPs.

A wide range of options exist for configuring, segmenting, and applying security controls to a CCS. Many types of security systems can be added. It is the beyond the scope of this paper to enumerate all possible cloud security controls. We focus on a few new promising CCS specific security capabilities. An important security attribute is how CSP sys-admins manage the CCS. We assume management is performed off-site. As described above we assume CSP sys-admins control CSP management servers using a dedicated Internet portal. CSP sys-admin traffic is accepted by the CSP control port firewall and routed to CSP management servers only if the traffic originates from an approved list of IP addresses. CSP management applications are isolated by hosting them on dedicated servers in their own CCS subnet. However, they cannot be completely isolated from tenant VMs, as they must monitor tenant VMs. Fig. 2 shows routers connecting tenant and CSP management subnets. These subnets can be isolated in hardware by using separate NICs for public and control plane (i.e., management) networking.

TABLE 1
CCS Architecture Security Controls

	VM Images At Rest	VM Migration	CSP Sysadmin IAM	Data Center physical security	Hypervisor, BIOS, CPU	VM Isolation	Tenant IAM	App. White-listing
Cloud Arch 1	Not encrypted	Unencrypted memory pages and packets	Single factor	All CSP employees have access	HV, BIOS not signed CPU without TPM	No network, CPU isolation	Single factor	No
Cloud Arch 2	Not encrypted	Unencrypted memory pages and packets	2 factor – time limited token code	CSP employee access limited & controlled + USB server ports disabled	HV, BIOS not signed CPU without TPM	No network, CPU isolation	Single factor	No
Cloud Arch 3	Not encrypted	Unencrypted memory pages and packets	2 factor – time limited token code	CSP employee access limited & controlled+ USB server ports disabled	HV, BIOS not signed CPU without TPM	No network, CPU isolation	2 factor – time limited token	No
Cloud Arch 4	Encrypted at rest + file access monitoring	Encrypted memory pages and packets	2 factor – time limited token code	CSP employee access limited & controlled+ USB server ports disabled	Signed HV, signed BIOS CPU with TPM	Virtual PANs, temporal CPU isolation	2 factor – time limited token	Yes

Table 1 shows four cloud architectures based on the reference model with progressively more security controls. More robust security controls are shaded. All four architectures use an SDN for tenant VM networking.

The first has a minimal set of security controls. The second incorporates additional data center physical access, CSP sys-admin authentication, and server hardware port controls. In the first architecture any CSP employee can enter the cloud data center. In the second, CSP sysadmins are not permitted in the data center. Employees authorized to enter the data center carry electronic access control cards and their movements are tracked in the data center. CSP sys-admins must use two factor authentication to login to CSP management servers, and they must sign in as named local users and not as root. Some cloud management products now offer such capabilities [17], which make it easier to identify unauthorized processes running with high privilege levels.

The third architecture includes the security controls of the second one and applies these security controls to all Agency sys-admin and regular users. Agency cloud users must login to Agency VMs using time sensitive two factor authentication methods.

The fourth architecture includes additional cloud infrastructure hardening measures. VM images are encrypted in storage. VM image store directories are monitored for access attempts, image changes, and TZs are isolated using more robust measures.

The HV and (Basic Input/Output System (BIOS) used in the CCS present potential additional points of vulnerability. HVs contain source code also found in an OS and may have large code bases, which means they may contain significant vulnerabilities. New technologies have have been developed to protect HVs and BIOS and to detect unauthorized HV or BIOS tampering. NIST has developed guidance for hardening BIOS [18]. Server vendors and microprocessor manufacturers now provide capabilities to verify CPU authenticity, the unaltered state of key chips on the motherboard, and which can securely measure and store BIOS and software boot time information. These make use of the Trusted Platform module (TPM) [19]. TPM has been integrated with the boot time measurement and remote attestation capabilities of Intel and other microprocessors [20]. There are many options to consider in this area. A particular CSP may implement a commercial HV that utilizes all of the security capabilities offered by TPM. Or the CSP may choose to not implement any of the security options available for a particular HV, microprocessor, or server. Or the CSP may use a custom designed HV, with its own unique security features. In this case the complete set of HV and server security features may not be public information. For the sake of illustration we consider only two options in this area. In cloud architectures 1 to 3 we assume a non-signed HV and servers and CPUs without TPM are used. In these cloud architectures, the integrity of the BIOS and HV cannot be verified during boot up.

Cloud architecture 4 is more secure. All servers in the CCS are assumed to use trusted BIOS (signed BIOS), TPM, and CPUs capable of making secure boot time measurements, such as Intel Trusted Execution Technology equipped CPUs [20]. Some HVs, such as VMware's vSphere 5.1 and later, are available in modular form, with each module containing a PKI signature that can be independently used to verify boot time and the unmodified state of the software code base during boot up. In the future, protected memory CPUs may have TPM capabilities built into the microprocessor, and may be used to verify the unmodified state of the HV code base dynamically at periodic stages at runtime [21] [22].

Table 1 shows other security attributes of the CCS architectures we consider. One is the degree of isolation of tenant TZs. An important aspect of this isolation is whether cloud users in other TZs can surveil the public portions or private tenant subnets in the cloud beyond their own subnet or TZ. If tenant VM names and IP addresses are readily available within the cloud, a cloud user from outside the Agency may be able to use standard network surveillance tools to identify the names and IP addresses of VMs used by Agency users (as may be possible in the Amazon Web Services cloud if certain security controls are not implemented by the tenant [5]).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2415794, IEEE Transactions on Cloud Computing

These capabilities have significant security implications, as shown by Ristenpart [5]. A wide range of network configurations is relevant to this security dimension. To make the analysis tractable we consider only two options in this area. The first is when cloud tenants have a wide range of surveillance capabilities at their disposal. The second is when the cloud tenant is not able to conduct surveillance operations across tenant TZs. The second option is adopted only in the fourth cloud architecture. It can be implemented in a variety of ways. One is by using dedicated hardware, perimeter firewalls, and IDSs to protect the Agency TZs, as is the best practice for securing enterprise networks [16]. However, this reduces cloud flexibility and elasticity. New cloud security technologies, like virtual firewalls and virtual networking using encrypted packets provide similar capabilities in a fully virtualized environment. AWS offers a EC2 service called Virtual Private Cloud (VPC) with these capabilities [13]. VMware offers networking and security capabilities in a product called VxLan [23] to support the isolation of VMs and VM TZs, and has recently received a patent on such capabilities [24].

VMs in storage and migration also require protection [25] [26]. If the VM image is altered in storage and compromised by malware, an adversary may gain control of the VM even if it is spun up in a highly protected TZ. If an adversary can gain access to a VM when it is stored in memory during migration or to the VM packet stream when the VM is moved, the adversary can obtain crypto keys or other credentials that provide access to sensitive data and applications in the agency TZ [27]. VM images can be encrypted to prevent VM image inspection and compromise. Live VMs could also be protected during migration by encrypting them in memory and during their movement by encrypting VM IP packets. These security controls are part of the fourth cloud architecture, as indicated in Table 1.

4 CCS Node Classes

The abstracted view of an IaaS CCS is shown in Fig. 3. It is the starting point for Cloud-Trust, and is based on the types of nodes in a CCS. These are labeled node classes, because many individual nodes of each type or class will be present in the CCS.

To simplify the analysis we assume all nodes in each node class are identical in terms of their security properties (before any malware is introduced we assume they are identically configured and that if there are system or node configuration errors these are common across all nodes in a node class). Therefore, it is not essential to distinguish between individual elements in each node class, and we can define a Bayesian network model in which the nodes of the network are CCS node classes, and not individual system components of the CCS. This Bayesian network model forms the basis of Cloud-Trust.

The columns in Fig. 3 indicate the TZs node classes belong to. The types of nodes classes are indicated in the first column. Node classes reflect the segregation of CSP and tenant network paths. The CCS architecture shown in Fig. 3 also has the feature that VM traffic within a TZ can be confined in that zone and segregated if all intra-TZ message traffic is routed by the V routers. This functionality is consistent with SDN or virtual networking capabilities provided by leading HV vendors and CSPs.

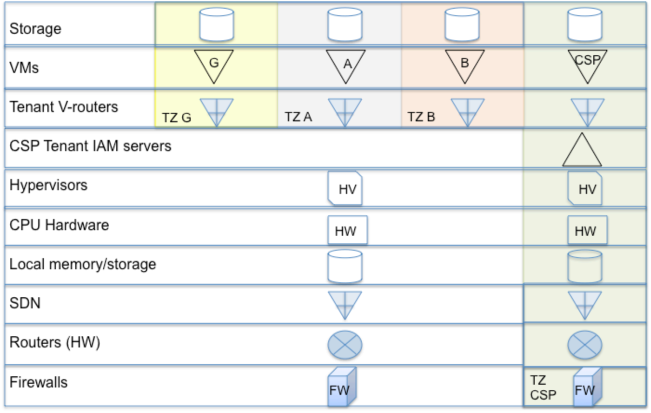


Fig. 3: CCS Node Classes

The attacker's objective is assumed to be the data store in TZ Gold in the upper left hand corner of Fig. 3. The APT will have to traverse the network of node class objects from bottom to top to gain such access if the attack starts from outside the cloud.

Using such node class diagrams, a cyber attack against an IaaS cloud can be represented by a directed graph of edges and nodes. The types node classes included in the node class diagram depend on the specifics of the cloud architecture examined. To find the set of edges that represent technically feasible cyber attacks we investigate specific CCS vulnerabilities identified in the literature. These are used to develop a set of attack paths that span the set of all feasible paths through the CCS infrastructure to the APT target.

5 CCS Attack Paths

CCS attacks can be divided into outsider or insider attacks. Outsiders can gain access to the cloud using three attack paths. The first exploits weaknesses in cloud access control mechanisms. Such weaknesses may exist in firewalls or IAM servers used by the CSP or cloud tenants. The second starts by stealing valid credentials of a cloud user at some location outside the cloud (for example from a host inside a government agency). The third outsider attack path starts with the attacker using valid credentials and prior legitimate access to the cloud.

Insider attack paths start inside the cloud when the attacker already exploits credentials for at least one cloud TZ, for example the CSP TZ. The ingress attack paths we consider are shown in Table 2.

The attack paths are defined in two variants. The first we call a "Stuxnet" variant where the APT requires little or no command and control (C2) by the external human attacker. In this case the APT has the surveillance information it needs to conduct all stages of the attack, or capabilities needed to independently do surveillance. The second attack variant is one where the APT has much less capability and information about the CCS environment. In this case we assume it must communicate with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2415794, IEEE Transactions on Cloud Computing

an external control authority and be updated with new capabilities during the attack.

Table 2

Cloud Specific Attack Ingress Paths

	Attack Name	Key Node Classes Exploited
1	VM side channel attack	Physical machine
2	SDN virtual router	Hypervisor, virtual router
3	VM attack through the hypervisor	Hypervisor
4	Live VM attack	VM
5	Corrupting VM images 1	VM image and VMs
6	Disk injection to Live VM	VM
7	VM migration attack 1	Local storage
8	VM migration attack 2	V-router
9	VMM control compromise 1	VMM, V-router
10	VMM control compromise 2	VMM, hypervisor
11	CSP sys admin + physical access	Physical machine, VM
12	Corrupting VM images 2	VMs and VM images
13	Undetected configuration modification	CSP firewalls and IAM servers
14	Nested Virtualization	Hypervisor

The target data that the APT attempts to access in all these attacks is located in a cloud TZ controlled by a government Agency –TZ Gold (G). We assume Agency users with TZ G access are also able to log into VMs in the Agency's TZ A. We do not assume that Agency network traffic is not restricted between A and G TZs. We also assume that Agency VMs operating in the same TZ run on the same physical machines and HVs.

VM CPU Timing Side Channel Attack

This attack is based on VM vulnerabilities identified by Ristenpart, et. al. [5]. It is representative of a class of attacks that take advantage of VM co-residency, which arises when VMs of two or more users share the same hardware. If the attacker's VM is co-resident with the target VM it may be able to glean information from the target VM by observing the hardware's behavior.

First the APT obtains access to the cloud and conducts surveillance. If the target is in a public cloud the only barrier to entry is a valid credit card to establish an account. The attacker instantiates VMs as needed to collect information on servers and VMs. To surveill the cloud the attacker will run legitimate code or malware.

We define VMs as being co-resident when they operate within the same physical machine and same HV. A variety of techniques can be used to detect and establish co-residency [5].

If the tenant is not guaranteed exclusive use of hardware, the instantiation of a VM that is co-resident with a target VM is generally governed by chance. However, it may still be possible using techniques described by Ristenpart [5] called "Cloud Cartography." Other co-residency checks use network trace routes. Since the first network "hop" from a VM is its HV, if that HV is configured to report itself when a trace route is conducted, co-residency can be detected using IP addresses. In a similar way, "distance" can be determined by ping packet round trip times. The lower the round trip time, the more likely the VMs are co-resident [5].

If the VM operates an external facing service such as a website, still other load analysis techniques may be feasible estimate co-residency [5].

Once co-residency is achieved, the attacker uses a prime-trigger-probe technique to monitor activity on the shared CPU's cache. The attacker's goal is to obtain an agency user's password, which may be done by analyzing an agency user's inter-keystroke timings [5] [29].

Once credentials have obtained, they are used to directly logon to the target's VMs. Once inside the agency network, additional surveillance may be conducted to identify and gain access to the targeted Gold data.

Software Defined Networking Attack

This attack exploits potential vulnerabilities in SDNs [30]. Virtual switches are special purpose VMs that may be co-resident with guest VMs on the same HV. Other configurations are possible including one where the virtual switch logic and code are integrated with the HV.

First, the APT gains access to VMs in a cloud TZ (e.g., TZ B) that are logical and network "peers" to the target VMs. This can be done through legitimate means if the only barrier to obtaining a CSP account is payment.

With legitimate or stolen credentials, the APT gains regular user access to a TZ B VM. The APT installs malware on the TZ B VM, which enables the APT to control the HV (exploiting a HV vulnerability).[1] Once the HV has been compromised, the APT is able collect information from the host machine's RAM such as additional credentials, network architecture, and decryption keys to compromise additional VMs and physical machines as necessary. We assume that credentials obtained for one VM in a TZ can be used on other VMs in the same TZ.

The APT obtains credentials to logon to a VM on the machine hosting the VM with access to Gold data. The APT compromises the HV on this machine. This time, the APT uses malware to modify the behavior of the virtual switch. This could include changes to the code in the virtual switch, the routing table, or both, so network packets destined for or emanating from the target (gold TZ) VM are copied and directed to a VM under APT control. Encryption of network traffic within the agency's virtual enclave could deter such an attack.

The APT obtains the targeted information over time by filtering the inbound and outbound network traffic to the target VM. The CSP design pattern that makes this attack possible is putting SDN based VMs co-resident in the same physical machines with cloud tenant VMs.

VM Attack Through the HV

This attack starts in much the same way as the SDN attack above. The APT obtains valid government user credentials (through spearfishing, surveillance, or use of malware) that can be used to access a VM operating in the agency's TZ in the cloud. A related attack path exists in the public cloud. Then the attacker obtains a public cloud account and initiates VMs in TZ B with the objective of compromising the HV and obtaining co-residency with a target agency VM running in TZ A or TZ G.

HV compromise proceeds as in the SDN attack. The APT installs malware that exploits a vulnerability in the HV that enables privilege level escalation [31] [32]. Once the HV is compromised, the APT collects data from the host machine's RAM such as additional credentials, network architecture, and decryption keys to compromise additional VMs and physical machines as necessary. This data is used to locate target VMs and to obtain co-residency.

Once the attacker has successfully executed the above steps and becomes co-resident with the target VM, the APT can extract relevant data from the memory of an operating VM in the Gold TZ and can gain access to Gold data.

Live VM Attack

In this attack we assume each VM has at least one "local" active administrator account. For this is a local username-password account the VM doesn't seek network validation of the logon. The hashed user name and password - that is targeted by the APT.

We also assume all VMs in agency TZ have the same local sys-admin logon accounts. The success of this attack and variations on it are dependent primarily on the agency's configuration of its VMs.

First, the attacker illicitly obtains agency credentials from outside the cloud to gain regular user access to a VM in the Agency's TZ A. With these credentials, the APT gains regular user access to an agency VM in TZ A. Depending on the tenant's security configuration, the APT may need to work around additional hurdles such as access via a restricted range of IP addresses (i.e., IP white listing restrictions). We assume for this attack that these additional security controls are not in place.

The APT installs malware to extract the file containing the hashed local sys-admin password (such malware would require some form of malicious privilege escalation). The APT moves the hashed password file to a location under its control where it decrypts the password file. Using this local sys-admin password, the APT logs into additional TZ A VMs and installs a key logger to collect additional credentials. The APT repeats this combination of local sys-admin password and key logger exploits until eventually, the APT obtains credentials sufficient to gain access to a VM running in the Gold TZ that has access to gold data.

Corrupting VM Images 1

In this attack VM images are compromised and used to gain access to agency Gold data [33]. We assume agency reference VM images are stored in the cloud. The success of this attack is dependent on the VM image storage controls used by the CSP and agency. For this attack to be effective VM images images would not be encrypted, no file access monitoring used, and only single factor authentication would be available to tenants.

First, the attacker uses valid (insider) or stolen (outsider) credentials to access the agency's image store in CCS. These credentials are assumed to grant the intruder access to TZ A and to a shared storage directory accessible by the CSP and agency users with TZ A access credentials. Agency VM images are stored in this shared storage directory. The outside attacker uses stolen credentials to access and copy one or more agency VM images. The insider would be a CSP sys admin or an Agency sys admin who has access to the shared VM image store.

The attacker modifies the VM image to include malware that monitors data accessed by VM. The attacker uses the same agency credentials to insert the modified VM image into the image store. Agency personnel use the infected VM image to instantiate new VMs. The malware on infected images remains dormant for a period of time to avoid triggering startup timing alarms.

Malware on the VM monitors the data accessed by the VM user. When the Gold data is accessed, the APT uses additional exploits on the target VM to access Gold data. This may involve caching credentials to allow the APT to directly access the VM or to deposit Gold data in local storage using previously stolen credentials.

Disk Injection to Live VM

The attacker attempts to gain access to agency Gold data by placing malicious code in the local attached storage of the targeted VM [34]. Necessary pre-conditions for this attack are that the VM in TZ Gold is operating on a physical machine that hosts VMs in other TZs, and the attacker can conduct network surveillance inside the cloud. The APT can then attempt co-residency with the target.

Using similar surveillance, pivoting, and compromise steps associated with earlier attacks, the APT gains access to a VM that is co-resident with a target VM operated by the Agency in its gold TZ.

In this attack, after the APT logons to a VM co-resident with the target VM, the APT exploits a vulnerability in the HV to compromise it.

Using the compromised HV, the APT writes malicious code to the local storage of the target VM. The HV also makes a minor change to the native "root/admin" level job scheduling system of the OS that ensures the malicious code will be called. When the privileged (root/admin level) job is called on the target VM, the malicious code is loaded and run. The malware then beacons its readiness to take further action by communicating to the APT's human controller. The APT is directed by the attacker to access Agency Gold data.

This attack becomes much more difficult if the local storage of the VM is encrypted. In such case, both the encryption regime and the HV must be compromised in order to complete the attack.

VM Migration Attack

VMs are migrated or moved frequently in clouds to prevent the overheating of servers and to optimally allocate workloads to available physical machines and resources. During workload migration VM memory pages including the OS are copied and moved to a new location. This attack takes advantage of the exposure of a VM during VM migration operations in the cloud [27].

Through spearfishing and surveillance, the APT obtains user credentials for a VM operating in cloud tenant B TZ.

Using a VM in TZ B the APT monitors network traffic (without additional compromises, this presumes that the VM is receiving and can control the behavior of its virtual or physical NIC to put it into promiscuous mode so that it can capture packets not addressed to it). APT uses

VMs in TZ B to capture and filter network traffic.

When a live VM transfer is detected, the attacker's VM stores the associated packets. Useful information is extracted from the captured VM (certificates, credentials, file access information) and is used to compromise additional VMs in other TZs, until the attacker compromises a VM in TZ Gold. At this point the attacker gains access to Agency Gold data.

CSP personnel with physical access

CSP personnel with physical access to the CSP datacenter can breach security controls by direct access to physical machines. However, the large number of machines complicates the task. Precision requires the attacker first identify the hardware hosting the target data.

The CSP insider cannot make physical contact with every machine in the datacenter, but he has several methods to locate the machine hosting the agency TZ G VMs. The first is to enumerate all of the Agency's live VMs. CSP management servers hold such data. A CSP sys-admin will have access to this data.

The list of agency VMs might be very large. The attacker must reduce the list so he can visit each machine. Information that can help narrow the list is configuration data. This includes security group and other tenant created configurations that reveal the topology of the tenants resources in the cloud, specific services that can be assumed based on specific ports that are open, identity and access management data that shows which tenant users have CSP accounts and what they are allowed to do with specific resources, tenant naming conventions for their machine images or instances, and relative memory, disk, CPU, and I/O sizing of various instances. These steps are likely to identify, for example, large machines, which host web server, file share, or database processes, and those that have limited access.

Once the CSP insider knows which machines to visit, he must map these to the datacenter layout. Datacenters that are segmented or compartmentalized, or keep access to physical and logical maps separate, will complicate the task. In contrast, datacenters that have a single point of entry will make this attack easier.

A CSP insider can inject malware using a USB drive. USB ports on physical machines are a well-known vector for inserting malware and ex-filtrating data. These attacks can be instrumented to work in an environment where the user is unprivileged and possibly unaware of the payload.

To compromise the target the attacker must induce it to load the malware. Physical machines with protections against 'autorun' execution may require additional manipulation via a KVM management interface. Such access may require access tokens that would identify the insider, but this depends on how such tokens, including local machine administrator account passwords, are managed. Given the familiarity a CSP insider is likely to have with the software, hardware, and virtualization stack, they may have the option of employing minimally invasive, and therefore hard to detect malware such as an in memory rootkit. Such malware will not be directly detected by disk or network based scans.

The malware injected via physical access provides a breach point for the attacker. The breach itself can provide network access and elevated privileges on the HV hosting the target VM. By beaconing to known attacker control nodes, the attacker can establish a link to control the execution of the rest of the attack.

VM Manager (VMM) Control Compromise

CSP sys-admins use VMM capabilities to migrate live VMs, to allow for hardware servicing without execution interruption, or to debug faults using core dumps and memory page snapshots. An attacker can repurpose VMM utilities to compromise agency data.

VMMs can be a privileged VM that runs on top of the HV. The VMM has access to ring 0 privileges and can see other VM's memory and configuration values. If the attacker gains direct access to the VMM, or is able to corrupt the VMM control channel, they would gain a great deal of maneuverability within cloud infrastructure.

Compromising the VMM or VMM control provides the attacker with a path to Agency Gold data by making keys and other sensitive data in an Agency VM visible to the attacker [35]. It also provides the attacker with a mechanism to move resources across TZs, for example by moving memory dumps, machine state, or entire VM instances from one physical machine to another. For example, the attacker could use the VMM to take a memory snapshot of an Agency VM in TZ Gold. The attacker could then proceed to access sensitive access tokens or data.

This attack has three key steps: identifying the HV hosting the target, gaining access to the VMM control channel, and executing a snapshot or memory dump request from the VMM.

One path for this attack begins from a compromised node or insider in the CSP enclave. This person, or node would allow the attacker to identify the HV that contained the target VM. Identifying the HV implies identification of the VMM. Other outsider attack paths also exist.

Once the target VMM is identified, the attacker acquires the ability to send it valid requests. If the physical machine has a separate network interface card (NIC) installed to isolate command channel traffic to the VMM, the attacker may need to compromise a CSP enclave node with access to that network. Or a CSP insider with access to a CSP management enclave C2 node can do this if the CSP insider has sys admin privileges.

If VMM traffic transits the same NIC as all other traffic to and from the HV, an outside attacker may be able to gain access to and control the VMM from C2 nodes outside of the CSP management enclave. In order for this attack path to be successful the attacker will have to compromise one or more nodes in the cloud network that are in the network path between the target and the CSP management servers in the CSP TZ.

If management function requests are run using a protocol that does not require authentication, network access to the control channel gained in the previous step might be the only requirement for successfully dumping the memory of the target VM. Otherwise, a credential may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2415794, IEEE Transactions on Cloud Computing

need to be compromised. If this is the case, the insider or compromised node in the CSP enclave could be used to surveil the host and network for valid credentials. The attacker establishes a destination for the memory dump, presumably outside the Agency's TZ.

The attacker sends a message to the VMM managing the target VM instructing it to dump its memory into a location in TZ B. The attacker examines the memory dump and identifies needed credentials to access agency Gold data, or finds the Gold data directly in memory.

Corrupting agency VM images 2

VM images and instances are vulnerable to attacks from the time they are created, during their transfer to the CSP, in storage in the cloud, while running, and when they are migrated within the cloud. Any unauthorized access or manipulation of VM image file can undermine trust in it. Infecting images can be more damaging than 'stealing' them because instances based on the infected image will continue to process sensitive data and may expose it to further exploitation. Integrity checks that are both stringent and regularly performed can provide some assurance regarding the health of the image, but such checks can be defeated.

Modifying a startup script in a VM image provides a simple example of how an attacker can gain control of a VM. Adding just a few commands can open a backdoor (i.e., exploit a vulnerability in the OS) or send a beacon signal to the attacker's command and control infrastructure. The attacker can then connect to the VM and continue to the next phases in the attack. Because the attack is inserted into the startup routine, it will run every time a VM instance based on infected image is spun up. VM instances are also susceptible to manipulation.

Implementing this attack requires three steps: gaining access to the CCS, modifying images or instances while avoiding integrity check violations, and creating a beacon that indicates successful compromise of a live VM.

Attacks that exploit vulnerabilities of networked devices may also apply to CSP management servers. An APT could use spearfishing, surveillance, and installation of malware to gain access to a physical machine and sysadmin account credentials in the CSP enclave providing them with a command and control (C2) node and network access to cloud management servers in the cloud that host dormant VM images and instances.

From the C2 node, the CSP enclave can be surveilled for the target: physical machines hosting the agency VM images and instances can be infected. Once found, the C2 node can be used to access the target and establish the ability to read and write from the VM images and instances. At this point a second APT package can be injected into the agency VM image or instance. The C2 node can also be used to access and manipulate or defeat the aforementioned integrity checks by replacing hash values, or causing hash collisions.

When an authorized agency user requests the VM image or instance be spun up, integrity checks are circumvented and the modified startup scripts are run activating backdoor and beaconing malware. The activation of the beacon and backdoor can be scheduled to run at a later time or made to run so quickly such that the deviation from a baseline startup time may not be noticeable.

The attacker can use the backdoor on the infected VM to install additional malware if required. The attacker waits for a compromised VM to be started by an authorized user with TZ Gold credentials. When this occurs the attacker has network access and elevated privileges on the target VM to access Agency Gold data.

This attack can be defeated by encrypting Agency VM images stored in the CCS and ensuring cryptographic keys are controlled by the Agency and not the CSP. There are a number of ways of implementing a secure key managment system for CSPs. One approach has been developed by Tysowski, et. al. [46].

Undetected configuration modification

Restricting traffic to a single whitelisted IP address associate with an agency enclave is a common baseline security control-best practice for limiting access to resources provisioned in the cloud. This, in theory limits access to the cloud resources to traffic emanating from the agency enclave, but it does not extend all agency enclave protections and monitoring to agency resources in the cloud (for example, an IDS may be absent in the cloud, and the CSP SIEM may not receive data from firewalls protecting Agency TZs). Therefore, the agency has less situational awareness regarding activity on its cloud based resources than it does within its enclave. This may allow the attacker to obtain access to sensitive data in the cloud.

Typically the CSP will provide an implementation of this control to agency users. For example: defining security groups for particular VMs. Permissions to create, modify and remove these configurations are granted to agency users with CSP accounts according to agency provisioning using the CSP's IAM schema. For example: agency user A is allowed to create VMs and set security groups, agency user B is allowed to start and stop instances but cannot create them or modify their configurations. In order to modify the security group configuration and whitelist IP address corresponding to its C2 nodes, the attacker need only gain access to the CSP credentials for agency user A.

Whitelisting an additional IP address for an APT C2 node outside the agency's enclave allows the attacker to user the credentials it has acquired from inside the agency's enclave to access the agencies resources in the cloud without any of the agencies monitoring tools detecting the access. If the CSP does not notify the agency that a configuration change has been made, and the attacker restores the original configuration after the access is complete, the agency may never learn of the access.

This attack has three steps: obtaining credentials for agency CSP resource configuration modifications and for agency A and G TZ access; modification of agency CSP resource configurations to white list the IP address of the attacker's C2 node; and access of agency Gold data from the newly whitelisted enclave.

The attack begins by compromising a node within the agency's enclave via spear phishing or other methods.

This node allows the attacker to perform surveillance that subsequently yields a valid credential being used by agency users to access a resource on an agency VM in the gold TZ, as well as a valid credential for an agency user with the authority to modify configuration of agency CSP resources.

Once the attacker has an agency user's credential for modifying configurations it may be able to use this credential from nearly anywhere because this access may not confined to whitelisted IP addresses. Once it logs into the CSP management interface and adds an additional IP address to the agency's Gold VM whitelist this task is complete. An additional step to cover its tracks after the attack is complete might involve reverting the configuration to its original setting.

Armed with the VM access credential obtained in the first step, and a network path from its C2 node outside the agency's enclave, the attacker can proceed with essentially unmonitored access to the agency's VM using legitimate credentials.

Nested virtualization

A nested virtualization attack [36] uses an additional unauthorized HV to access sensitive data and credentials. The additional HV could be inserted either between the normal HV and the physical hardware, or between a guest OS and the normal HV. In the former case, the additional HV will provide an attack surface that spans all of the VMs on the original HV. In the later case, the additional HV could be confined to a specific guest OS.

The target for the attack is a VM running in TZ G or is a VM image with stored TZ G credentials that is at rest. Finding the VM image at rest, or finding the physical machine that the target VM is or will be spun up would be accomplished by surveillance of Agency VM operations. Either target is likely to begin with the attacker gaining access to the CSP management enclave in order to perform sufficient surveillance. An insider working for the CSP can do the surveillance.

Attacking the VM image and inserting the unauthorized HV provides the advantage that the operation can be performed before the image is loaded into the CSP infrastructure (while it is still in the agency enclave).

Targeting the image at rest, the attacker would 'wrap' it with an additional HV (which would boot up first). Targeting the physical machine would require that the attacker either be able to reboot the machine and cause it to load the attacker's HV first, and then load the CSP's HV, or implement a 'blue pill' rerouting of a live HV without rebooting [36].

Once an attacker has successfully nested a HV at either layer, one of the main advantages, in addition to gaining access to memory and other sensitive resources, is that the rest of the stack would function 'normally'. The guest VMs continue to run on virtualized infrastructure, and the original HV thinks it is running on CSP hardware.

Once the attacker has succeeded in injecting a HV that it controls, it has gained a stealthy point of access to sensitive VM data and credentials. However, unless the attack is completely autonomous, it may require additional surveillance and C2 activities. The HV may therefore have to beacon to another node to complete the attack.

Nested virtualization attacks exploit the fact that both the intended hosts and guests might not have mechanisms available to verify the other parties. The guests are supposed to run on a virtualized platform and may not be able to detect that they are not running directly on a CSP sanctioned HV. Similarly, both the CSP HV and the CSP hardware provide interfaces that do not discriminate between consumers of their resources. Absent specific restrictions, an additional attacker controlled HV could be a consumer that is as accepted as a guest OS, or CSP controlled HV.

6 Bayesian Network Model

We apply Bayesian network statistics to the attack paths described above. Attack paths have been used to understand the vulnerability status of information systems [37]. They have also been used to develop probabilistic measures of enterprise network security [38] [39]. We extend this approach to CCSs by constructing an acyclic directed graph using the attack paths defined above [40]. We apply these attack paths to the CCS node classes defined in Fig. 3. The resulting directed graph is shown in Fig. 5.

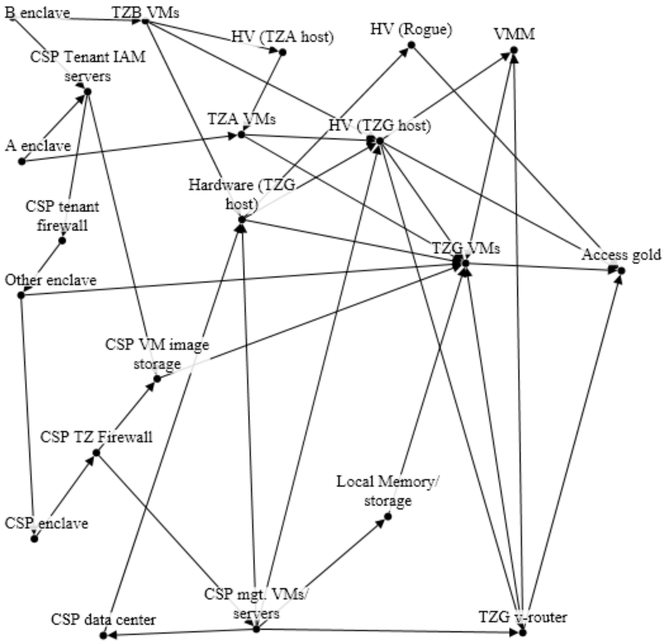


Fig. 5: IaaS CCS Infiltration Bayesian Sub-Network

Cloud-Trust relies on conditional probabilities that represent the probability that a vulnerability in an individual CCS component can be exploited by an APT, if other CCS components have already been compromised.

These conditional probabilities correspond to the directed edges shown in Fig. 5. This approach enables us to factor in the contributions that specific CCS security features can have in reducing the vulnerabilities of nodes in the CCS and which then can contribute to a reduction in the overall security profile of an IaaS cloud. Our model of CCS architectures includes the security features and controls the CSP provides, what the CSP permits the customer or cloud tenant to provide, and what the cloud tenant actually provides.

The complete security model consists of two Bayesian sub-networks: an infiltration sub-network and an exfiltration sub-network. Only the Cloud-Trust infiltration sub-network is shown in Fig. 5. The infiltration sub-network characterizes the probability that an APT will be able to access the gold data, while the exfiltration network characterizes the likelihood that the APT can exfiltrate the accessed gold data. We assume the two sub-networks are independent, i.e., the infiltration strategy is independent of the exfiltration strategy employed by the attacker (In a subsequent paper we will examine the relationships of infiltration and exfiltration strategies and will extend Cloud-Trust to exfiltration networks).

We denote the infiltration Bayesian network $B^{IN} = (G^{IN}, \Theta^{IN})$, where G^{IN} is a directed acyclic graph with nodes that are random variables and links that are direct dependencies between those random variables. Let $V^{IN} = \{A_i\}_{i=1}^{n}$ be the sequence of random variables representing the nodes of V^{IN} such that binary random variable A_i denotes whether node $i \in \{1, \dots, n\}$ has been accessed by an APT to infiltrate the gold data. Furthermore, defining $a_{ij} = \Pr(A_i | A_j)$ as the probability that node i is accessed by the APT given that node j is accessed by the APT, we can define the link set as $L^{IN} = \{(i,j) : a_{ij} > 0, i, j \in \{1, \dots, n\}, i \neq j\}$. Observe that the link set L^{IN} gives the nodes between which an APT can traverse with positive probability. Since G^{IN} is a directed graph, we must have that $a_{ij} \neq a_{ji}$. A consequence of the acyclic property is that either $a_{ij} = 0$ or $a_{ji} = 0$. The component Θ^{IN} represents the set of quantitative parameters in the network; for each parameter $\theta_i \in \Theta^{IN}$for $i = 1, \dots, n$, we have that $\theta_i = \Pr(A_i | \pi_i)$, where $\pi_i = \{A_j : a_{ij} > 0\}$. In other words, the set π_i is the set of parents of node i or the set of nodes from which an APT can access node i with positive probability. Note that we will make the Markov assumption that the random variable $A_i$depends only on its parents π_i, i.e., the access of a node i depends only on which node j it was accessed by the APT and not the history of how the APT accessed node j.

While the above exposition characterizes the (unopposed) attack carried out by the APT, the SIEM has an opportunity to detect the APT's attack. Hence, we define the binary random variable D_i^{IN} indicating whether the SIEM detects an APT access of node class $i = 1, \dots, n$ indicating whether the SIEM detects an APT exfiltrating data through node class $i = 1, \dots, n$. If we let A denote the event that the gold data has been accessed by the APT and undetected by the SIEM, we can calculate the probability of undetected access as

$$\Pr(A) = 1 - \prod_{i=1}^{n} [1 - \Pr(A | A_i) \times \Pr(A_i) \times [1 - \Pr(D_i^{IN})]]$$

where the probability that node class i has been accessed is given by

$$\Pr(A_i) = 1 - \prod_{j=1, i \neq j}^{n} [1 - \Pr(A_i | A_j) \Pr(A_j)]$$
$$= 1 - \prod_{j=1, i \neq j}^{n} [1 - a_{ij} \times \Pr(A_j)]$$

for $i = 1, \dots, n$. We assume that $\Pr(A_1) = 1$, i.e., the APT accessing the enclave node class is taken as given. Hence, we have a system of n equations and n unknowns, i.e., $\Pr(A_2), \dots, \Pr(A_n), \Pr(A)$. Since the Bayesian network is acyclic, solving this system is algebraically simple using substitution.

Our model can also estimate the probability that the SIEM and associated IS3 systems will detect an attack that would infiltrate the gold data. Let D^{IN} be a binary random variable indicating whether the SIEM detect an attack that would infiltrate the gold data. Then

$$\Pr(D^{IN}) = \frac{\Pr(A | no\ detection) - \Pr(A)}{\Pr(A)}$$

where $\Pr(A | no\ detection)$ is the probability that the APT can infiltrate the gold data without any detection (i.e., $\Pr(D_i^{IN}) = 0$ for $i = 1, \dots, n$. The above equation shows what percentage of attacks that would otherwise be successful in infiltrating the gold data can be detected.

Although mathematically simple, our Bayesian network approach imposes constraints on how APT attacks can be represented. We have assumed the Markov property when defining the conditional probabilities. In some attacks it may be necessary to return to previously compromised nodes to proceed with the attack. In our mathematical formalism this type of circular path is forbidden. We account for such a possibility and for the possibility that an attacker may have to traverse the ingress path more than once, by including a probability of APT "beacon success" at nodes where additional attack software code or APT commands are needed. This eliminates cycles in the infiltration attack graph.

In general, it is possible to expand the node set to allow for attack path histories to influence the probability of node access; however, the tractability and insight that could be gleaned from the model output might be hampered. Using a similar approach, we could also remove the assumption that the infiltration and exfiltration processes are independent, but it's not clear that such complexity would add value to the model.

7 Illustrative Cloud-Trust Results

To apply the model conditional probabilities are needed for all network edges – the probability that given the APT has access to the starting node of the edge, the APT will be able to exploit a vulnerability, conduct surveillance and identify, or obtain co-residency with the target node at the end of the edge. Over 50 probability scores are needed to fully characterize the Cloud-Trust infiltration Bayesian network for a typical cloud architecture. For the illustrative cloud architecture assessments presented in this paper over 400 probability score inputs were estimated using a variety of methods. Below we describe how some of these estimates were made.

The scope of Cloud-Trust does not include the security measures used in external network enclaves. So we assume that an APT can gain access to relevant external network enclaves and to cloud credentials stored there.

For some attack paths the attacker obtains initial cloud TZ credentials by legitimate means. This is the initial step of the VM side channel attack. In this case the attacker has a legitimate public cloud account that enables him to instantiate a VM in the public cloud in TZ B. The second step in this attack is to move from a VM in TZ B to be co-resident with the target VM in TZ Gold. As described earlier in the attack narrative there are various mechanisms that can be used in public clouds to conduct surveillance and to move a VM into a preferential location in the cloud so the attacker becomes co-resident with the target. We assess the success of these methods for specific cloud architectures using two probability scores p_s and p_cr. The value of these probabilities estimates, shown in Table 3, are derived from the literature that applies to different public cloud offerings [5].[14][13].

We do not estimate the probability that a specific HV will have exploitable vulnerabilities. Instead, we consider a generic HV. HVs are large code bases that resemble OS kernels, so we assume the probability is high that an unsigned HV contains vulnerabilities. On the other hand, if the HV is signed and trusted boot time measurements are available from the manufacturer we reduce this probability significantly as indicated in the table (in other words we assume the HV still has vulnerabilities, but during a reboot they will be detected and a "pristine" version of the HV can be re-installed from a Gold Disk.

Table 3

Selected Cloud-Trust Probability Scores

	Attack Step	Cloud Arch 1	Cloud Arch 2	Cloud Arch 3	Cloud Arch 4
1	TZ B Access from Tenant B enclave	1	1	1	1
2a	Establish co-residency – surveillance (p_s)	1	1	1	.1
2b	Establish co-residency – VM movement (p_cr)	0.5	0.5	0.5	.01
3	Obtain credentials (varies by attack)	0.1	0.1	0.01	..01
4	Exploitable hypervisor vulnerability	0.9	0.9	0.9	.01

The discussion above illustrates the types of probabilities used in our approach: one, probabilities which represent the likelihood that a particular type of activity can be accomplished in the cloud (that is whether cloud security controls are present or absent which would constrain or eliminate the activity); two, probabilities that reflect the likelihood the attacker can gain access to one or more cloud resources (e.g., whether IAM controls are in place to restrict attacker access); and three, the probability that a specific type of cloud component contains a vulnerability or property which can be exploited by the attacker to maneuver to or gain access to another cloud component. In many cases such probabilities cannot be determined precisely by analytical means. For example, all vulnerabilities that are present in a HV may not be known. In cases were there is significant uncertainty in a vulnerability value, we assign one of five values to the conditional probability: very high (set equal to 1), high (set equal to .9), medium (set equal to one half), low (set equal to .1), and very low (.01). We estimate probabilities of APT detection for each node class in the cloud architecture using a similar approach. There is also uncertainty regarding specific conditional detection probabilities. In these cases we also estimate the probabilities of detection on a five level scale. Based on reports available in the open press on the extent and longevity of APT attacks we do not ascribe high detection probabilities to most edges in the Bayesian network [41][28][42].

An alternative means to determine conditional attack probabilities is to use the common vulnerability scoring system (CVSS) [43]. CVSS scores associated with specific CCS components could be used to estimate these conditional probabilities. Such an approach would resemble that suggested by earlier authors [38].

Table 4

Cloud-Trust Assessment Results

Cloud Architecture	Infiltration Probability with APT Detection	Infiltration Probability without APT Detection	Detection Probability (APT accesses gold data)
1	0.89	0.99	0.1
2	0.84	0.98	0.14
3	0.25	0.46	0.46
4	0.004	0.007	0.5

Illustrative Cloud-Trust results are shown in Table 4 for the cloud architectures defined in Table 1. The cloud architectures with more capable security controls are estimated to have lower probability of successful APT infiltration. Not surprisingly, if the APT is detected prior to gold data access, the probability of infiltration is reduced. This effect is most pronounced in cloud architectures 3 and 4, which have more robust security controls. However, one can see that even with robust security controls, the estimated probability of APT or threat detection are less than or equal to 1/2 in all cases. The estimated cumulative APT detection probability is ~ 1/2 in architectures 3 and 4 because the individual APT detection probabilities for individual CCS components are estimated to be small (with the exception of file access monitoring of the Agency Gold data store in TZ G) and because file access monitoring may not provide an effective APT detection capability if the APT accesses TZ Gold using valid stolen credentials. Cloud-Trust accounts for this possibility in the overall assessments scores given above.

Cloud architecture 4 has the lowest probability of APT infiltration. This architecture makes extensive use of encryption to protect VM images at rest and live VMs during migration. It also uses a signed HV, and robust sys-admin access control methods to verify the identity of both CSP sys-admins and Agency cloud users. These security controls make it more difficult for the APT to steal valid credentials and obtain a long lasting presence in key CCS components, such as the HV. If the HV or BIOS are modified by the APT, there is a good chance the HV modification will be detected (especially if the APT is a complex and large code base), even if the APT itself can not be detected. The compromised HV or BIOS can then be deleted, and a "pristine" version of the HV can be re-installed from a Gold Disk.

7 Conclusion

We have demonstrated how Cloud-Trust can be used to assess the security status of IaaS CCSs and IaaS CSP service offerings, and how it is used to compute probabilities of APT infiltration (high value data access) and probabilities of APT detection. These quantify two key security metrics: IaaS CCS confidentiality and integrity. Cloud-Trust also produces quantitative assessments of the value and contribution of specific CCS security controls (including several optional security controls now offered by leading commercial CSPs), and can be used to conduct sensitivity analyses of the incremental value of adding specific security controls to an IaaS CCS, when there is uncertainty regarding the value of a specific security control (which may be optional and increase the cost of CSP services).

Potential next steps

The scope of initial version of Cloud-Trust is limited to IaaS CCSs and CSPs. It also does not include all possible insider attacks. Potential next steps should be to extend Cloud-Trust to include the full range of insider attacks, and to Platform as a Service (PaaS) and Software as a Service (SaaS) CSPs.

It would also be useful to develop a full set of data exfiltration APT attack steps that span the component space of CCSs and CSPs. It would also be to explore how CVSS data could be used to estimate APT attack probabilities. A robust sensitivity analysis methodology could also be developed to see which nodes and edges Cloud-Trust results are most sensitive to.

Acknowledgment

This work was supported by the Institute of information and infrastructure protection (I3P), the Department of Homeland Security (DHS) National Cyber Security Division, and the RAND Corporation, and was performed under DHS contract #2006-CS-001-000001.

The authors thank Inette Furey of DHS and Martin Wybourne of Darmouth College for sponsoring this research and Richard George (Johns Hopkins University Applied Physics Laboratory), Dr Shari Pfleeger (Dartmouth University), and Kartik Gopalan (Binghamton University (SUNY)) for useful conversations.

References

[1] W. Jansen and T. Grance, "Guidelines on security and privacy in public cloud computing," *NIST Spec. Publ.*, pp. 800–144, 2011.
[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing." NIST, 2011.
[3] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 142–157, 2013.
[4] L. Vaquero, L. Rodero-Merino, and D. Morán, "Locking the sky: a survey on IaaS cloud security," *Computing*, vol. 91, no. 1, pp. 93–118, Jan. 2011.
[5] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 199–212.
[6] A. Sood and R. Enbody, "Targeted Cyber Attacks-A Superset of Advanced Persistent Threats," *IEEE Security and Privacy.*, Vol. 11, no. 1, Jan.-Feb., 2013.
[7] B. Krekel, "Capability of the People's Republic of China to conduct cyber warfare and computer network exploitation," U.S.-China Economic and Security REview Commission, Northrop Grumman Corp., DTIC Document, 2009.
[8] *FedRAMP Security Controls*, Federal Chief Information Officer's Council, [Online]. Available: http://cloud.cio.gov/document/fedramp-security-controls. [Accessed: 29-Oct-2014].
[9] S. Zevin, *Standards for security categorization of federal information and information systems*. DIANE Publishing, 2009.
[10] M. Walla, "Kerberos Explained," May, 2000. [Online]. Available: http://technet.microsoft.com/en-us/library/bb742516.aspx. [Accessed: 12-Jan-2014].
[11] Microsoft, "Federation trusts," Aug 22, 2005. [Online]. Available: http://technet.microsoft.com/en-us/library/cc738707(v=ws.10).aspx. [Accessed: 12-Jan-2014].
[12] VMWare, "Network Segmentation in Virtualized Environments," BP-059-INF-02-01, 2009 [Online]. Available: http://www.vmware.com/files/pdf/network_segmentation.pdf [Accessed: 29 Oct 2014].
[13] Amazon Web Services, "AWS | Amazon Virtual Private Cloud (VPC) – Secure Private Cloud VPN." [Online]. Available: http://aws.amazon.com/vpc/. [Accessed: 12-Jan-2014].
[14] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "All your clouds are belong to us: security analysis of cloud management interfaces," in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, 2011, pp. 3–14.
[15] V. J. Winkler, *Securing the Cloud: Cloud computer Security techniques and tactics*. Elsevier, 2011.
[16] *Network Infrastructure Technology Overview*, Version 8, Release 3, Defense Information Systems Agency, August 27, 2010.
[17] G. Keeling, R. Bhattacharjee, and Y. Patil, "Beyond the Hypervisor: Three Key Areas to Consider When Securing Your Cloud Infrastructure Platform," presented at the VMWorld 2012, San Francisco, August, 2012 [Online]. Available: http://www.vmworld.com/docs/DOC-6257 [Accessed: 29-Oct-2014]
[18] A. Regenscheid, "BIOS Protection Guidelines for Servers (Draft)," 800-147B, Jul. 2012 [Online]. Available: http://csrc.nist.gov/publications/drafts/800-147b/draft-sp800-147b_july2012.pdf [Accessed: 29-Oct-2014].
[19] Trusted Computing Group, "Trusted Platform Module (TPM) Summary." [Online]. Available: http://www.trustedcomputinggroup.org/resources/trusted_platform_module_tpm_summary. [Accessed: 12-Jan-2014].
[20] S. Chalal, et. al., "Evolution of Integrity Checking with Intel® Trusted Execution Technology: an Intel IT Perspective." Intel, Aug. 2010 [Online]. Available: http://www.intel.com/content/dam/doc/white-paper/intel-it-security-trusted-execution-technology-paper.pdf. [Accessed: 29-Oct-14].
[21] M. Hoekstra, "Intel® SGX for Dummies (Intel® SGX Design Objectives) | Intel® Developer Zone," 26-Sep-2013. [Online]. Available: http://software.intel.com/en-us/blogs/2013/09/26/protecting-application-secrets-with-intel-sgx. [Accessed: 13-Jan-2014].
[22] F. McKeen, et. al., "Software Guard Extensions (Intel® SGX), Instructions and Programming Model," *Intel Softw. Solut. Group Tech Rep*, Jun, 2013.
[23] T. Huber, "VXLAN – What it is, Components that Make it Work, and Benefits | VMware SMB Blog - VMware Blogs," September 17, 2013. [Online]. Available: http://blogs.vmware.com/smb/2013/09/vxlan-what-it-is-components-that-make-it-work-and-benefits.html. [Accessed: 13-Jan-2014].
[24] W. Lambeth, et. al., "Private allocated networks over shared communications infrastructure," U. S. Patent: 8619771 [Online].
[25] D. Perez-Botero, "A Brief Tutorial on Live Virtual Machine Migration From a Security Perspective," *Princeton Univ., Princeton, USA*. 2011. [Online]. Available: http://www.cs.princeton.edu/~diegop/data/580_midterm_project.pdf [Accessed: 29-Oct-2014]

[26] M. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," *ACM SIGOPS Oper. Syst. Rev.*, vol. Volume 43, no. Issue 3, Jul. 2009.

[27] M. I. Gofman, et. al., "SPARC: A Security and Privacy Aware Virtual Machine Checkpointing Mechanism," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, 2011, pp. 115–124.

[28] Mandiant, "M-Trends 2010: the Advanced Persistent Threat." Mandiant, 2010.

[29] D. X. Song, D. Wagner, and X. Tian, "Timing Analysis of Keystrokes and Timing Attacks on SSH," in *USENIX Security Symposium*, 2001, vol. 2001.

[30] J. Fortes, "Cloud Computing Security: What Changes with Software-Defined Networking?," presented at the ARO Workshop on Cloud Security, Mar 11, 2013.

[31] M. J. Schwartz, "New Virtualization Vulnerability Allows Escape To Hypervisor Attacks - InformationWeek," Jun. 13, 2012. [Online]. Available: http://www.informationweek.com/security/risk-management/new-virtualization-vulnerability-allows-escape-to-hypervisor-attacks/d/d-id/1104823 [Accessed: 13-Jan-2014].

[32] E. Ray and E. Schultz, "Virtualization security," in *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, 2009, p. 42.

[33] T. Katsuki, "Crisis for Windows Sneaks onto Virtual Machines," Aug. 20, 2012. [Online]. Available: http://www.symantec.com/connect/blogs/crisis-windows-sneaks-virtual-machines. [Accessed: 13-Jan-2014].

[34] F. Breedijk, "Blackhat talk: Cloudburst – VMWare guest to host escapes by Kostya Kirtchinsky," Jul. 31, 2009. [Online]. Available: http://www.cupfighter.net/index.php/2009/07/blackhat-cloudburst-vmware-guest-to-host-escape/. [Accessed: 13-Jan-2014].

[35] Homeland Security News Wire, "Japanese pharmaceutical crippled by insider cyberattack," *Homeland Security News Wire*, Aug. 22, 2011. [Online]. Available: http://www.homelandsecuritynewswire.com/japanese-pharmaceutical-crippled-insider-cyberattack. [Accessed: 13-Jan-2014].

[36] J. Rutkowska, and A. Tereshkin, "Bluepilling the Xen Hypervisor," presented at the 2008 Blackhat Conference, Las Vegas, NV, Aug. 2008.

[37] O. Sheyner et. al., "Automated Generation And Analysis Of Attack Graphs," *Secur. Priv. 2002 Proc. 2002 IEEE Symp. On*, pp. 273, 284.

[38] A. Singhal and X. Ou, *Security Risk Analysis Of Enterprise Networks Using Probabilistic Attack Graphs*, National Institute of Science and Technology Interagency Report 7788, Gaithersburg, MD, August 2011.

[39] M. Frigault, et. al., "Measuring network security using dynamic bayesian network," in *Proceedings of the 4th ACM workshop on Quality of protection*, 2008, pp. 23–30.

[40] I. Ben-Gal, "Bayesian networks," *Encycl. Stat. Qual. Reliab.* 2007.

[41] C. Glyer and R. Kazanciyan, "The 'Hikit' Rootkit: Advanced and Persistent Attack Techniques (Part 2),"[Online]. Available: https://www.mandiant.com/blog/hikit-rootkit-advanced-persistent-attack-techniques-part-2/ [Accessed: 04-Feb-2013].

[42] D. Alperovitch, *Revealed: Operation Shady RAT,* McAfee, 2011 [Online]. Available: http://noramintel.com/wp-content/uploads/2011/08/McAfee-wp-operation-shady-rat.pdf. [Accessed: 29-Oct-2014].

[43] P. Mell, K. Scarfone and S. Romanosky, "CVSS v2 Complete Documentation," Jun. 2007, [Online]. Available: http://www.first.org/cvss/cvss-guide.html. [Accessed: 14-Jan-2014].

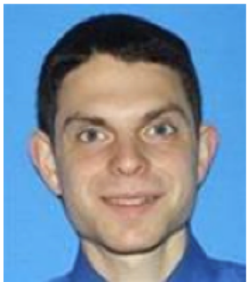
[44] H. Thimbleby, S. Anderson, and P. Cairns, "A Framework for Modelling Trojans and Computer Virus Infection," *Comput. J.*, vol. 41, no. 7, pp. 444–458, Jan. 1998.

[45] K. Ingols et. al., "Modeling modern network attacks and countermeasures using attack graphs," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, 2009, pp. 117–126.

[46] P. K. Tysowski and M. A. Hasan, "Hybrid Attribute- and Re-Encryption-Based Key Management for Secure and Scalable Mobile Applications in Clouds," IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 172–186, 2013.

Daniel Gonzales received a Ph.D. in Theoretical Physics from MIT in 1985, and a B.S. in Physics from Stanford University in 1979. His research includes command, control (C2) and communications and intelligence systems, information assurance (IA), and cloud computing. He has examined performance, testing, and strategy issues for enterprise computing systems, radios, biometrics systems, cloud computing, and electronic warfare (EW) systems for the U.S. Army, U.S. Navy, DHS, and the Office of the Secretary of Defense.

Dr. Jeremy M. Kaplan received a B.A. in physics from Columbia College in 1967 and a Ph.D. in physics from Columbia University in 1976. He is a private consultant and RAND adjunct staff member. He researches information system security. At the Defense Information Systems Agency (DISA) he led development of architecture, systems engineering, test, and communications planning for DoD nuclear C2. He received the Presidential Rank Award of Meritorious Executive in the Senior Executive Service. He has also created leading-edge concepts for the dynamic modeling of networks.



Evan Saltzman received a B.S. degree in mathematics and economics from the College of William and Mary and his M.S. in operations research at Georgia Tech. Evan was at RAND from 2010 to 2014. He has examined supply chain and workflow management systems, and was an adjunct faculty member in the School of Management at George Mason University. Prior to joining RAND, Evan worked as an ORISE Fellow at the Centers for Disease Control and Prevention (CDC).

Zev Winkelman received a B.S. in computer engineering from the University of Michigan, a M.S. in Criminal Justice from the John Jay College of Criminal Justice, and a Ph.d. in public policy from U.S. Berkeley. He is on the faculty of the Pardee RAND Graduate School and researches big data analytics, social media, and cyber security. He has more than 15 years of experience in computer engineering and software development. He has implemented systems that allow analysts to fuse, analyze and visualize datasets across multiple domains.

Dulani Woods received an MS in Agricultural Economics from Purdue University and BS degrees in Mechanical Engineering and Marine Engineering/Naval Architecture from the US Coast Guard Academy. He is a data analyst and modeler at RAND. His research has included assessing design tradeoffs for a fleet of Offshore Patrol Cutters for the U.S. Coast Guard. He served as a Coast Guard Officer from 1995 to 2004.