Cyber Defenses for Physical Attacks and Insider Threats in Cloud Computing

Jakub Szefer
Yale University
jakub.szefer@yale.edu

Pramod Jamkhedkar
Princeton University
pjamkhed@princeton.edu

Diego Perez-Botero
Princeton University
diegop@cs.princeton.edu

Ruby B. Lee
Princeton University
rblee@princeton.edu

ABSTRACT

In cloud computing, most of the computations and data in the data center do not belong to the cloud provider. This leaves owners of applications and data concerned about cyber and physical attacks which may compromise the confidentiality, integrity or availability of their applications or data. While much work has looked at protection from software (cyber) threats, very few have looked at physical attacks and physical security in data centers. In this work, we present a novel set of cyber defense strategies for physical attacks in data centers. We capitalize on the fact that physical attackers are constrained by the physical layout and other features of a data center which provide a time delay before an attacker can reach a server to launch a physical attack, even by an insider. We describe how a number of cyber defense strategies can be activated when an attack is detected, some of which can even take effect before the actual attack occurs. The defense strategies provide improved security and are more cost-effective than always-on protections in the light of the fact that on average physical attacks will not happen often – but can be very damaging when they do occur.

Categories and Subject Descriptors

K.6 [**Managements of Computing and Information Systems**]: Security and Protection; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*

Keywords

Data Center Security; Physical Attacks; Insider Threats; Cloud Computing; Migration; Cloning; Virtual Machines

1. INTRODUCTION

Physical attacks on computers are less frequent and harder to launch, compared to software attacks. However, a physical attack may give an attacker absolute control over a secure sever, resulting in potentially greater damage. For this reason, most institutions that deal with highly sensitive data, including military and financial companies, go to great lengths to protect their servers and data centers from physical attacks. For example, to get into the building, one needs to scan one's hand, pass a guard, then go through a hallway where one is trapped if found suspicious. Also, some servers are locked in cages and separated from other systems [20]. Many physical attacks are, however, carried out by insiders who have authorized physical access to secure servers, making them difficult to prevent. Physical attacks on information technology (IT) infrastructures have been identified as one of the most overlooked aspects of IT security [12], and there has been no, or limited, cyber-physical defenses proposed. Furthermore, as cloud computing becomes prevalent, data centers will become prime targets for attackers.

Problem Overview: Defense mechanisms against physical attacks have typically followed two approaches: 1) physically controlled access and surveillance (entirely in physical space) and 2) design of secure servers with pervasive encryption and hashed storage (entirely in cyber space). On the one hand, physical access control mechanisms [11] may prevent unauthorized physical access to a certain degree, but are not effective against insider attacks. On the other hand, pervasive encryption, or continuous encryption of all memory (and swap disk space), is done by secure processor proposals, e.g. [22, 15, 6, 3], in which everything is automatically encrypted in DRAM. While the performance overhead may be acceptable, such architectures are not implemented in existing commodity processors – and hence are not available in current servers inside data centers.

Malicious Insiders: A particular concern is that a malicious cloud provider employee, who has "legitimate" physical access to the data center, can trivially launch physical attacks. This is precisely the scenario where our cyber-physical security countermeasures come into play. Today, any person, authorized or not, can launch physical attacks. In our proposed system, thanks to the use of the physical sensors, any sensitive data will be moved or scrubbed before an authorized employee or outsider has a chance to launch a physical attack. Note that we add the missing, physical, piece to the plethora of current, cyber-only, defenses. Also, our solution applies equally well to external attackers, insiders and maintenance personnel. Hence, our proposed solution is different from the existing ones, which are primarily aimed at keeping out the intruders, or are cyber-only mechanisms which focus on software attacks and not physical attacks.

ASIA CCS'14, June 4–6, 2014, Kyoto, Japan.

http://dx.doi.org/10.1145/2590296.2590310.

Our primary contributions are:

- A new cyber-physical defense strategy, where cyber defenses are used to protect against physical attacks (Section 2), including a new defense based on virtual machine (VM) cloning;
- Qualitative and quantitative analysis of four cyber defense strategies for physical attacks (Section 3); and
- A formalization of concepts and ideas needed to reason about physical attacks in data centers or other distributed networked systems (Section 4).

Key Advantages: Our cyber-physical defense framework is cost-effective. The cyber defenses against potential physical attacks are triggered only when the possibility of such an attack is detected by the surveillance and physical access control mechanisms. This precludes the need for pervasive encryption and hashing. It also offers a realistic solution against insider threats.

Paper Organization: Section 2 provides an overview of our cyber-physical defense architecture. Section 3 provides an analysis of the four defense strategies. Section 4 discusses physical attacks and equations needed to reason about them. Related work is described in Section 5 and we conclude in Section 6.

2. CYBER-PHYSICAL DEFENSES

Our work is based on a cloud computing paradigm where computation is performed inside VMs. Our goal is to protect code and data inside the VMs from physical attackers.

2.1 Threat Model and the Attacks

We aim to protect against physical attacks on compute, storage and networking equipment in a data center setting. This work is orthogonal to, and complements, work on defenses of these entities from cyber attacks. There is already a plethora of security research on software defenses, e.g. [4].

The main protected entities in our cyber-physical defense framework are the VMs running on the physical servers in a data center, which contain applications and data belonging to cloud customers. By extracting the contents of the VMs, an attacker can gain valuable information, including proprietary or sensitive data or programs belonging to cloud customers.

The human threat we are worried about is a physical attacker who has or gains physical access to a data center and the servers where the VMs are running. A physical attack can be carried out by different types of individuals including outsiders, maintenance individuals, insiders, etc. We assume that once an attacker has physical access to a machine, at that point in time the attack can be considered successful. Physical attacks include cold boot style attacks for extracting information from memory chips of the server [10], or stealing hard drives to extract data from the local disk. Attacks can also include turning off the power, or allowing the machine to overheat by turning off the cooling, to cause loss of availability to the VMs running on this server.

2.2 Attack Detection and Defense Timeline

Our defense framework against physical attacks is based on the time difference between potential attack detection and the actual attack, as shown in Figure 1, which we have first proposed in [24]. We now, however, consider three timestamps in our framework: time of detection (t_{detect}) denoting the time at which a physical attack is detected, and the time of attack (t_{attack}) denoting the time at which the attacker has physical contact with the equipment. We also consider $t_{pre-empt}$, which denotes a pre-attack time when some defenses can be pro-actively launched. The physical sensors mentioned previously are used to trigger a warning at t_{detect}. Scheduled events, such as maintenance, could be used to trigger pre-emptive actions at $t_{pre-empt}$.

Figure 1 shows different security mechanisms and their relationship with these three time stamps. Mechanisms shown in case A are preventive mechanisms that operate wholly in the physical space, and their effectiveness ends after an attack occurs. The goal of these measures is to delay t_{attack}, possibly forever (i.e., to shift t_{attack} as far right as possible). While these defenses may prevent outsider access, they are ineffective against insider attacks. Mechanisms shown in case B are detection mechanisms that operate in physical space, and their defensive purpose ends after an attack has been detected; although they can be utilized for evidence gathering as the attack proceeds, and for forensics afterwards. Pervasive encryption mechanisms shown in case C assume that an attacker can attack without any warning ($t_{detect} = t_{attack}$), and data is kept encrypted all the time. Such mechanisms, even though effective, either incur significant overhead, or require new hardware.

In our defense framework, we capitalize on the fact that effective detection mechanisms (case B) coupled with effective protection mechanisms (case A), both operating in the physical space, will induce a significant delay between t_{detect} and t_{attack}. The physical detection mechanisms (case B) would produce a warning at t_{detect} that can trigger appropriate cyber defense mechanisms, based on the expected time available for realtime response (T_r). We have identified four primary types of cyber defenses, which fall under cases D and E, that could be used against physical attacks to protect VMs. Our defenses consist of three reactive defenses (case D) and one proactive defense (case E).

1. *Migrate:* Applications and data are moved away from the physical servers being attacked so that when the attacker gains access to these servers, the data is no longer there.
2. *Encrypt:* Applications and data are encrypted within the servers, so that when the attacker gains access to the servers, the attacker cannot make use of the data since it is in encrypted form.
3. *Delete:* Applications and data are deleted from the servers so that when the attacker gains access to the servers, the data is no longer there.

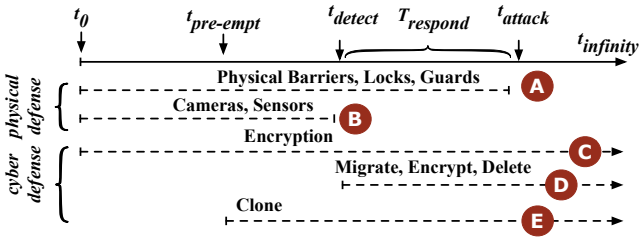

Figure 1: Defense strategy timeline. Our new proposed defense strategies are highlighted with Ⓓ and Ⓔ.

4. *Clone:* A VM with applications and data is cloned in anticipation of a potential attack; when an attack is detected we only need to erase the memory and any local storage on the victim server as a clone is already active and running in a different location.

Each defense strategy has its own limitations in terms of the time taken to carry out the response, practical feasibility, computation cost, and the security protections it offers; we present evaluation of these four in Section 3.

2.3 Cyber-Physical Defense Architecture

The Cyber-Physical Defense architecture consists of three main components: physical sensors, security management infrastructure, and compute equipment (e.g. servers).

2.3.1 Sensors and Physical Security Monitors

Data centers routinely include motion sensors, cameras, electronic locks on doors, etc., [19] and others can easily be added. These sensors provide an infrastructure that can signal an alert about an impending physical attack. Such a warning system provides various degrees of response times for defense mechanisms to be triggered.

We propose a physical security monitor which collects all the data from the attached sensors, formats it and converts it to the standard API (application programming interface) calls to the management infrastructure.

2.3.2 Security Management Infrastructure

The security manager integrates all the data from the various sensors. In order to calculate minimum response times based on attacker paths and hurdles, the data center floor plan, location of access doors, and location of servers can be automatically extracted from CAD (computer aided design) files describing the data center. The possible paths to a given server and the hurdles (e.g. locks on cages) can be pre-calculated, based on these inputs, to optimize reaction times to threats. The management software also calculates defense times for individual VMs based on bandwidth availability, server availability, safe destination servers, VM sizes, etc. The defense trigger mechanism also takes into consideration scheduled events (such as cleaning and maintenance) and mobility of data center personnel to trigger appropriate defenses. The defense strategies are translated into actual VM management commands so that the compute equipment can carry out the needed defensive countermeasures.

2.3.3 Compute Equipment

The cloud providers use management software such as OpenStack [17] to manage the actual compute servers. Figure 2 shows a possible realization of our cyber-physical defense system using the OpenStack cloud computing framework. The sensor and physical monitor infrastructure provides inputs via a modified Nova API. The inputs are then passed, via the Queue, to the management infrastructure which includes the Threat Evaluation and Defense Selection logic. This obtains floor plans and sensor locations from a CAD database. The defense triggers are passed via the message Queue to a modified Nova Compute server, which then carries out the defense strategies. Nova database is used by the OpenStack components to store (and retrieve) information about the different physical servers, status of the running VMs, etc.

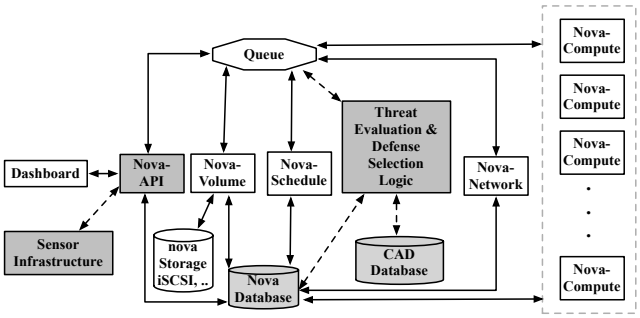

Figure 2: Our cyber-physical security architecture overlays onto the OpenStack cloud architecture. Modules shown in gray are OpenStack parts that would need to be added or modified.

3. FEASIBILITY OF CYBER DEFENSES

We now describe the defense strategies in more detail, and also evaluate the feasibility of our defenses with respect to the available attack response times. We follow the instance type definitions for Amazon's EC2 instances [1] (see Table 9 in the Appendix).

3.1 Migrate Strategy Analysis

When employing *migration* as a defense strategy, confidentiality, integrity and availability are all preserved after the attack; even if the attacker destroys or steals the equipment. Use of live VM migration allows for reduced downtime, and erasure of the source host's memory ensures that attacker does not gain information from the server he or she captures. Thus, our proposed defense is actually realized by complementing migration with the erasure or deletion of relevant RAM or local disk sectors[1], as described in Table 1.

Table 1: Migration based defense strategies.

Strategy	Details
Migrate + Erase RAM	The VM is migrated and the VM's main memory contents on the local machine need to be scrubbed.
Migrate + Erase RAM & Disk[1]	Same as "Migrate + Erase RAM", but if any of the VM's main memory has been swapped to local disk by the hypervisor, that disk space needs to be erased as well.
Migrate + Encrypt RAM	The VM is migrated, but the local main memory that was allocated to the VM is encrypted (local copy of encryption key is scrubbed).
Migrate + Encrypt RAM & Disk[1]	Same as "Migrate + Encrypt RAM", but any local disk contents (e.g. swapped memory of the VM) are encrypted (local copy of encryption key is scrubbed).

3.1.1 Migrate Strategy – Feasibility Analysis

The time taken to enact live migration as a defense depends on the type of applications which are running inside the VMs. We base our evaluation on the 7 types of workloads that most commonly arise in data centers [13], which are listed in Table 2.

To evaluate the migration strategy, our test bed is comprised of two hosts with identical hardware and software

[1]The local disk need not be erased or encrypted if always-on encrypted storage is used. The encryption key, however, still needs to be scrubbed from the system's main memory.

Table 2: Data Center Workloads

Workload	Benchmark
Mail Server	mstone [16] as remote SMTP client; smtp-sink [21] as SMTP server inside VM
App Server	Faban Benchmarking Framework [7] as remote client; Glassfish Server [9] with sample Java EE application inside VM
File Server	Dbench [5] inside VM
Web Server	Faban Benchmarking Framework [7] as remote client; Apache HTTP Server [2] inside VM
DB Server	Sysbench [23] inside VM
Stream Server	VideoLAN [26] inside VM; Wireshark [28] capturing stream packets remotely
Idle Server	No workload

configurations. Each host comes with dual quad-core Intel Nehalem CPUs (1.6GHz), on top of which a KVM hypervisor is running. The network connecting both hosts supports 1Gbps speeds. Meanwhile, the VMs being migrated come with 1 GB of dedicated RAM and 1 CPU core. For each workload, we ran 2-minute long benchmarks and performed migrations at five different migration points within that 2-minute interval; those results were then averaged. This solves the problem of variations due to the migration point [13].

The results are shown in Table 3. From the data, it can be seen that total migration time for a 1 GB VM (and thus the time needed to deploy the defense) is on the order of 10 seconds. The *liveness* of this strategy is clear: downtimes experienced by users are negligible (< 1 second) for all but one server workload. A key fact to bear in mind is how critical available bandwidth is to live migration. For instance, it took us 19.4 seconds to migrate an Idle Server when using a 100 Mbps link, but only 2.6 seconds with a 1 Gbps link. The overhead is the difference between execution time when migrating and when not migrating, divided by the time when not migrating.

Table 3: Migration Performance and Resource Usage with Data Center Workloads with 1Gbps link.

Server Type	Total Time (s)	Down -time (ms)	Data Sent (MB)	Over -head (%)	Avg. BW (Mbps)
Mail	8.9	300	800.39	15.71	702.86
App	7.5	4050	728.21	6.25	766.52
File	7.2	250	697.87	12.71	764.77
Web	7.2	250	645.33	10.08	707.08
DB	7.4	450	717.52	1.39	765.22
Stream	8.0	650	716.21	3.03	705.31
Idle	2.6	200	212.34	0.00	629.11

3.2 Cloning Strategy Analysis

When employing VM cloning as a pre-emptive defense strategy, confidentiality, integrity and availability are preserved. Cloning strategies are listed in Table 4. Cloning relies on proactively cloning the VM and running multiple clones of the VM; if any clone is in danger of attack, that clone is simply terminated while other clones keep running.

3.2.1 Cloning Strategy – Feasibility Analysis

The cost of cloning comes from the duplicate resources used as the VMs run, e.g. DRAM memory allocated to each clone. This can be dealt with by installing more resources.

Table 4: Cloning based defense strategies.

Strategy	Details
Clone and erase RAM	Scrub RAM containing VM's memory pages.
Clone and erase RAM & Disk	Scrub RAM and any disk containing VM's memory pages.

Also, to be able to use cloning as a defense strategy, there needs to be sufficient time to proactively clone the VMs. Cloning re-uses many of the facilities of migration and has almost identical total time. Details of the cloning strategy are omitted for space reasons, they are available in [18].

The time required to clone a 2GB VM ranges between 4 and 15 seconds if the full bandwidth of a 1Gbps link is available between the host machines. As with live migration, the bandwidth and current processor load will influence the cloning time. However, since cloning is proactive, the cloning time can be selected when there is available bandwidth and the servers are not under heavy load.

3.3 Encrypt Strategy Analysis

When employing encryption as a reactive defense strategy, confidentiality is preserved after the attack. Availability, however, is not preserved if the attacker destroys or steals the equipment and there is no redundant backup copies stored elsewhere. The defense is realized through encryption of relevant RAM or local disk sectors, as described in Table 5.

Table 5: Encryption based defense strategies.

Strategy	Details
Encrypt RAM	Encrypt all of VM's memory pages in RAM.
Encrypt RAM & Disk	Encrypt all of VM's memory pages in RAM and any memory that has been swapped to local disk.
Move to Encrypted Disk & Erase RAM	If swap space is already using encrypted storage, move all memory pages to swap space, and erase RAM contents.

3.3.1 Encrypt Strategy – Feasibility Analysis

Encryption will be affected by both by the algorithm used as well as the hardware available. We benchmark symmetric key encryption using the popular TrueCrypt [25] . Based on the speeds and the VM's RAM and local disk sizes (from Table 9) the times for encryption of RAM are obtained and shown in Table 6. On the test servers (see Section 3.1.1 on migration) the AES algorithm performed at 450 MB/s and Twofish at 409 MB/s. We also tested the encryption on a different system with Intel Core i7 processor with dedicated AES instructions. The AES algorithm benefited greatly from the new Intel AES-specific hardware instructions and achieved 2253 MB/s All measurements were done when the CPU was not utilized for any other purpose.

Table 6: Encryption of VMs' RAM (a) without and (b) with dedicated AES encryption instructions.

	Encrypt RAM	micro	m1.small	m1.medium	m1.large
(a)	AES	1.38 s	3.89 s	8.56 s	34.16 s
	Twofish	1.52 s	4.28 s	9.41 s	37.58 s
(b)	AES	0.27 s	0.77 s	1.71 s	6.82 s
	Twofish	1.52 s	4.28 s	9.41 s	37.58 s

3.4 Delete Strategy Analysis

When employing deletion as a defense strategy, confidentiality is preserved, but integrity and availability are not, as the applications or data are not available anymore after the deletion. The defense is realized through deletion of relevant RAM or local disk sectors, as described in Table 7.

Table 7: Delete based defense strategies.

Strategy	Details
Erase RAM	Scrub RAM containing VM's memory pages.
Erase RAM & Disk	Scrub RAM and any disk containing VM's memory pages.

3.4.1 Delete Strategy – Feasibility Analysis

RAM and disk erasure times depend on the write speeds of the memories and disks. Data can also be overwritten with a random pattern. Our results in Table 8 are for overwriting RAM and disks with zeros.

On the test servers (see Section 3.1.1) sequential writes to main memory can sustain 6481 GB/s. For disk, we use the 70 MB/s throughput (assuming the erase is just writing 0s at the full speed of the disks we have tested). Table 8 summarizes the times needed to perform the erase strategy for the different types of VMs.

Table 8: Erasing RAM and local disk of VMs.

Operation	micro	m1.small	m1.medium	m1.large
Erasing RAM	0.10 s	0.27 s	0.59 s	2.37 s
Erasing local disk	8.9 s	0.14 s	0.14 s	0.14 s

Note that *micro* instances have some local swap disk, while other instance types only have configuration files that count as data on local disk (they use remote persistent storage).

3.5 Comparing the Defense Strategies

We have presented details of the four defense strategies: migrate, clone, encrypt and delete. In our evaluation, the migration strategy takes longer (about 9s), compared to encrypting data or erasing the VM, but provides the most security, and availability is maintained. The encryption strategy is faster with less than 7 seconds even for the largest VMs, if AES-specific hardware instructions are available. The exception is when there is a lot of VM memory that has been swapped to disk and that disk has to be encrypted on the fly. Use of encrypted swap space, however, alleviates this problem. The fastest strategy, but also offering only confidentiality protection, is the delete strategy. All data is lost, but the attacker does not get to it. Alternatively, cloning is a very good strategy if there are enough resources to create clones, pre-emptively, of the VMs. The defense then is quite fast (as fast as the erasure strategy) because, if the attacker threatens one clone, the clone can simply be erased. Moreover, cloning can be configured to use less network resources and take more time.

4. REASONING ABOUT PHYSICAL ATTACK AND RESPONSE TIMES

The key to accurately determining the response time for a given VM is to calculate how "close" is the nearest potential threat from the server on which the VM is running. This calculation must be carried out taking into consideration the physical locations of different individuals in a data center, their type, and how easily each of those individuals can reach the server. We denote the current location of different individuals in a data center as the threat context. For a given threat context, the estimation of time to respond is not absolute, but relative to the type of security required by the VM. For instance, one VM may consider the presence of a maintenance person in the vicinity as a threat, whereas another VM may not consider it as a threat according to which security level was selected for the VM. Therefore, two VMs operating on the same server, under the same threat context, may perceive the same threat with different urgency and hence have different response times. The calculation of the estimated response time available for a given VM is carried out in the following steps:

1. Obtain the locations of all the potential attackers (i.e. the threat context) and the location of the host server.
2. For each attacker, calculate all paths from the attacker's position to the server.
3. Calculate the response time for each attacker based on the shortest path (in terms of travel time and time to clear hurdles) for that attacker.
4. Calculate the available response time as the smallest of all the response times for each attacker.

Step 1 is calculated from inputs provided by the sensors and physical access control mechanisms providing locations of all individuals in the data center. Step 2 is calculated based on the result of Step 1 and the pre-calculated physical layout of the data center. We define Step 3 as follows. For a given VM vm running on Server s, and an attacker a, let $P(a, s)$ represent the set of all the paths from the position of a to the position of server s. The response time for VM vm from threat a while being hosted on server s is:

$$T_r(vm, s, a) = min_{p \in P(a,s)}\{T_w(a, s, p) + T_h(a, vm, p)\} \quad (1)$$

where $T_w(a, s, p)$ is the amount of time required by the attacker a to cover the distance between himself and the victim server s on a given path p and $T_h(a, vm, p)$ is the amount of time required by the *attacker* to clear all the hurdles between himself and the victim server s on path p. The value for $T_w(a, s, p)$ can be calculated by the length of path p divided by the average running time for a human. In order to calculate $T_h(a, vm, p)$ we need to know what type of hurdles are present on path p, and how much time attacker a would take to clear those hurdles. Whether or not an individual (intruder, maintenance personnel, etc.) is an attacker or not is determined by the $vm's$ security policy.

Step 4 calculates the nearest threat as follows. Let A be the set of all the potential attackers detected in the data center. The available response time for a given VM vm hosted on server s is the minimum of all the response times calculated for each potential attacker, calculated as follows:

$$T_r(vm, s) = min_{a \in A}\{T_r(vm, s, a)\} \quad (2)$$

The type of defensive action chosen must fit this T_r.

5. RELATED WORK

Numerous physical measures are used in data centers to ensure the security of the data center infrastructure from physical attacks [11]. Barriers, alarms, entry control, contraband detection, CCTV (closed-circuit television) surveillance, and other means have been used for protecting data

centers [8]. Many physical security measures mainly focus on environmental, not human, factors such as fire security or the failure of supporting utilities (e.g., power) [27].

Preventing outsiders from gaining access to data centers typically includes measures like security locks, biometric authentication, isolation of secure areas, sign-in books, two-factor authentication, etc. These mechanisms are often complemented with mechanisms for detecting physical intrusions such as cameras, sensors, surveillance guards, etc. These physical defense measures provide time constraints on any potential attackers. Each measure slows down the attackers, although it does not necessarily stop them.

An example of a "moving target defense" [14] strategy is one where VMs are pro-actively moved from one server to another. Past work, however, has not looked at using physical triggers as a means of influencing moving target defenses.

6. CONCLUSION

We presented details of a cyber-physical security framework for cloud computing data centers that combines the security mechanisms in cyber and physical spaces, and exploits the power of virtualization to provide dynamic security against physical attackers. The framework protects against *human* attackers who can use physical access (illegitimate or legitimate as in the case of insider attacks) to extract applications or data from the compute infrastructure inside the data centers. We secure applications and data through the use of cyber defenses based on the strategies of migration, encryption, deletion and cloning. While the first three are reactive defenses triggered by the detection of a potential human attacker, the last one, cloning, is a preemptive strategy that can be used for the most security-sensitive applications. We leverage physical intrusion detection systems to warn of an impending physical attack to trigger the first three defenses: migration, encryption and deletion. When there are enough resources for pre-emptive cloning, it can provide faster response when an attack actually happens. We hope our work will simulate more research into the use of cyber defenses for protecting against physical attacks.

7. REFERENCES

[1] Amazon EC2 Instance Types. http://aws.amazon.com/ec2/instance-types/.

[2] The Apache HTTP Server Project. http://httpd.apache.org/.

[3] D. Champagne and R. B. Lee. Scalable architectural support for trusted software. In *Proceedings of the International Symposium on High Performance Computer Architecture*, HPCA, pages 1 –12, January 2010.

[4] W. Dawoud, I. Takouna, and C. Meinel. Infrastructure as a Service Security: Challenges and Solutions. In *Proceedings of the International Conference on Informatics and Systems*, INFOS, March 2010.

[5] Dbench filesystem benchmark. http://dbench.samba.org/.

[6] J. S. Dwoskin and R. B. Lee. Hardware-rooted trust for secure key management and transient trust. In *Proceedings of the ACM Conference on Computer and Communications Security*, CCS, pages 389–400, October 2007.

[7] Faban Harness and Benchmark Framework. http://java.net/projects/faban/.

[8] L. J. Fennelly. *Effective Physical Security*. Butterworth-Heinemann, 3rd edition, 2003.

[9] GlassFish - Open Source Application Server. http://glassfish.java.net/.

[10] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Communications of ACM*, 52(5):91–98, May 2009.

[11] S. Heare. Data Center Physical Security Checklist. Technical report, SANS Institute, December 2001. http://www.sans.org/.

[12] K. J. Higgins. The 10 Most Overlooked Aspects of Security, November 2006. http://www.darkreading.com/security/application-security/208808177/the-10-most-overlooked-aspects-of-security.html.

[13] D. Huang, D. Ye, Q. He, J. Chen, and K. Ye. Virt-lm: a benchmark for live migration of virtual machine. In *Proceedings of the International Conference on Performance Engineering*, ICPE, pages 307–316, March 2011.

[14] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, editors. *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*. Springer, 2011.

[15] R. B. Lee, P. Kwan, J. P. McGregor, J. Dwoskin, and Z. Wang. Architecture for protecting critical secrets in microprocessors. In *Proceedings of the International Symposium on Computer Architecture*, ISCA, pages 2–13, June 2005.

[16] mstone multi-protocol testing system. http://sourceforge.net/projects/mstone/.

[17] OpenStack. *OpenStack Compute: An Overview*.

[18] D. Perez-Botero. *Pwnetizer: Improving Availability in Cloud Computing Through Fast Cloning and I/O Randomization*. Master's Thesis, Princeton University, Princeton, NJ, 2013.

[19] S. D. Scalet. 19 Ways to Build Physical Security into a Data Center. http://www.csoonline.com/.

[20] J. S. Schultz. Should you trust mint.com? From New York Times http://bucks.blogs.nytimes.com/2010/07/06/should-you-trust-mint-com/?_r=0.

[21] smtp-sink(1) - Linux man page. http://linux.die.net/man/1/smtp-sink.

[22] G. E. Suh, D. Clarke, B. Gassend, M. van Dijk, and S. Devadas. Aegis: architecture for tamper-evident and tamper-resistant processing. In *Proceedings of the Annual International Conference on Supercomputing*, ICS, pages 160–171, June 2003.

[23] SysBench: a system performance benchmark. http://sysbench.sourceforge.net/.

[24] J. Szefer, P. Jamkhedkar, Y.-Y. Chen, and R. B. Lee. Physical Attack Protection with Human-Secure Virtualization in Data Centers. In *Workshop on Open Resilient human-aware Cyber-physical Systems*, WORCS, June 2012.

[25] TrueCrypt - Free Open-Source On-The-Fly Disk Encryption Software. http://www.truecrypt.org/.

[26] VLC media player. http://www.videolan.org.

[27] M. E. Whitman and H. J. Mattord. *Principles of Information Security*. Cengage Learning, 2011.

[28] Wireshark: the world's foremost network protocol analyzer. http://www.wireshark.org/.

APPENDIX

Table 9: Local server resources used by different Amazon EC2 VM instance types.

Resource	micro	m1.small	m1.medium	m1.large
VM's RAM	613MB	1741MB	3840MB	15370MB
VM config	$\leq$ 10MB	$\leq$ 10MB	$\leq$ 10MB	$\leq$ 10MB
Total RAM	623MB	1751MB	3850MB	15380MB
Local Swap for VM	$\leq$ 613MB	n/a	n/a	n/a
VM config	$\leq$ 10MB	$\leq$ 10MB	$\leq$ 10MB	$\leq$ 10MB
Total Disk	623MB	10MB	10MB	10MB