Power Attack: An Increasing Threat to Data Centers

Zhang Xu Haining Wang
Department of Computer Science
College of William and Mary
Email: {zxu, hnw}@cs.wm.edu

Zichen Xu Xiaorui Wang
Department of Electrical and Computer Engineering
Ohio State Univeristy
Email: {xuz, xwang}@ece.osu.edu

Abstract— **Entering the era of cloud computing, data centers are scaling in a fast pace. However, as the increasing number of servers being deployed in data centers, the data center power distribution systems have already approached peak capacities. Since the upgrades of the power systems are extremely expensive, power oversubscription has become a trend in modern data centers as a cost-effective way to handle power provisioning. Under benign workload of data centers, power oversubscription works well as servers rarely peak simultaneously. However, power oversubscription makes data centers vulnerable to malicious workload that can generate power spikes on multiple servers at the same time, which may cause branch circuit breakers to trip and lead to undesired power outages. In this paper, we introduce a new security concept called power attack and exploit the attack vectors in platform as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS) cloud environments, respectively. To demonstrate the feasibility of launching a power attack, we conduct series of hardware experiments and data-center-level simulations. Moreover, we give a detailed analysis on how different power management methods can affect a power attack and how to mitigate such an attack. Our experimental results and analysis show that power attacks will pose a serious threat to modern data centers and should be taken into account while deploying new high-density servers and power management techniques.**

I. Introduction

With the ever-increasing demand of cloud services, data centers have experienced significant growth in their scale. The number of servers in data centers has surged from 24 million in 2008 to over 35 million in 2012 [5]. Correspondingly, the power consumption of data centers has increased by 56 percent from 2005 to 2010 [22], with an even faster speed in recent years. Thus, the rapid server deployment in data centers has caused their power distribution and cooling systems to approach peak capacity [14]. However, it is very expensive to upgrade the power infrastructures of data centers and the related cost is commonly in hundreds of millions of dollars.

To support more servers with the existing power infrastructures, power oversubscription has become a trend in data centers [12], [27], [17]. The key feature of oversubscription is to place more servers on the power infrastructure of a data center than it can support if all the servers would not reach their maximum power consumption at the same time. Since servers rarely peak simultaneously with normal workloads, oversubscription allows many more servers to be hosted than traditional provisioning that relies on the server nameplate power ratings, without the need of upgrading the power infrastructure. However, power oversubscription makes it a possibility that the power consumption of servers might exceed power capacity, resulting in an increasing risk of power outages.

From the security perspective, this hidden risk induced by power oversubscription leaves data centers vulnerable to malicious workloads that can generate power spikes on multiple servers at the same time. We define the creation of such a malicious workload as a *power attack*. Without obtaining a privileged access, an attacker can launch a power attack as a regular user. The simultaneously occurred power peaks could produce the overloading of electrical circuits and then trigger the trip of circuit breakers (CBs) at the rack level or even a higher level of power facilities, leading to undesired power outages. The ultimate goal of a power attack is to fail the victim's power facility and cause an interruption or termination of the computing services running on the blackout servers. The damage of a power attack is twofold: both cloud service providers and the owners of other computing services running on the blackout servers suffer from service interruptions and financial losses.

In this paper, we systematically investigate the feasibility of launching power attacks in three main-stream cloud service business models: platform as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS), respectively. In the case of PaaS, we choose high performance computing (HPC) as one of its typical workloads, and conduct a set of experiments based on HPC benchmarks. We observe that an attacker can generate power spikes by adjusting workloads but those system-utilization-based load balancing mechanisms can hardly detect such an attack. In the case of IaaS, we introduce a new concept called parasite attacks that leverage controlled virtual machines (VMs) to significantly increase the power consumption of the host physical machine. Moreover, we demonstrate that VM migration can trigger high power spikes by conducting a set of experiments. If the VM migration routine can be inferred by attackers, the power spikes generated during migration can be exploited to help trip the CBs. In the case of SaaS, we use web services as its typical workload and conduct a set of experiments to demonstrate that specially crafted web requests can trigger power spikes and consequently

NDSS '14, 23-26 February 2014, San Diego, CA, USA
Copyright 2014 Internet Society, ISBN 1-891562-35-5
http://dx.doi.org/doi-info-to-be-provided-later

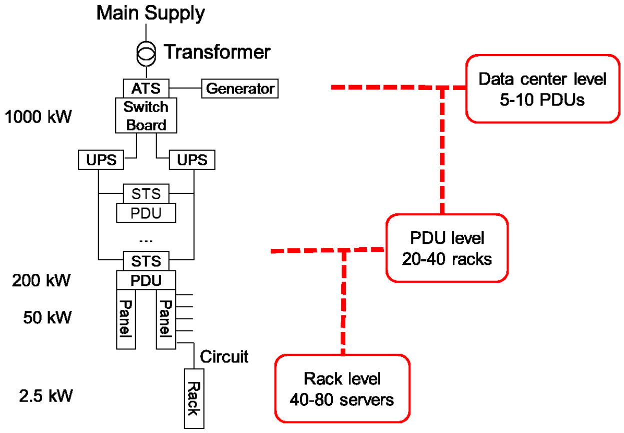

Fig. 1. A typical data center power distribution from [12]

trip the CBs.

Based on our rack-level experimental results, we further conduct a series of data-center-level simulations by using traces and configurations of the Google's data center at Lenoir, North Carolina, USA. The simulation results show that by injecting malicious workload, an attacker can generate power spikes in a data center scale, which pose a serious threat to the availability and reliability of data centers. While the focus of this work is on the attacking side, we also present different approaches to mitigate the power attacks in an effective manner.

The remainder of the paper is structured as follows. Section II introduces the background of power infrastructures in a data center. Section III presents our threat model of power attacks. Sections IV, V, and VI present how to launch a power attack in the PaaS environments, IaaS environments, and SaaS environments, respectively. Section VII shows the data center level simulation results. Section VIII provides a detailed discussion on how some new power management techniques will affect power attacks. Section IX presents the defense against power attacks. Section X surveys related work, and finally Section XI draws the conclusion.

II. Background

In this section, we introduce a typical power infrastructure employed in most data centers. We then discuss the practice of power oversubscription in data centers for cost reduction and its implications on the power security of data centers.

Today's data centers commonly have a three-tier power distribution infrastructure to support hosted computer servers [6], though the exact architecture may vary for different sites. Figure 1 shows a simplified illustration of the three-tier hierarchy in a typical data center. High-voltage power (60-400 kV) from the utility grid is scaled to medium voltage (typically 10-20 kV) through an outside transformer and then fed to an Automatic Transfer Switch (ATS). The ATS connects to both the utility power grid and on-site power (e.g., diesel) generators. From the ATS, the primary switchgear of the data center scales the voltage down to 400-600 V, which is supplied to Uninterruptible Power Supplies (UPS) via multiple independent routes for fault tolerance. To protect the power infrastructure against electrical faults, the switchgear is normally equipped with a circuit breaker (CB) that would trip if the total power consumption of the data center exceeds its rated capacity. Each UPS supplies a series of Power Distribution Units (PDUs), which are rated on the order of 75-200 kW each. A PDU breaks up the incoming power feed into multiple branch circuits and has a breaker panel where circuit breakers protect individual circuits from ground short or power overload. The PDUs further transform the voltage to support a group of server racks. It is important to note that many components in a data center power system have limited capacities. For example, a PDU can generally handle 75-225 kW of load and a rack-level branch circuit typically has a capacity of 6 kW [6]. Violating such capacities may cause circuit breakers to trip, leading to the shutdown of the servers connected to a branch circuit or even the entire data center. A typical 1 MW data center may house ten or more PDUs. Each PDU can support approximately 20 to 60 racks while each rack can include about 10 to 80 computer servers [36].

As mentioned before, many data centers keep deploying new high-density servers (e.g., blade servers) to support their rapidly growing business. As a result, their power distribution systems have already approached the peak capacity. In order to minimize the high capital expenses of upgrading their power infrastructures, data centers recently started to adopt power oversubscription as an important methodology to fully utilize their existing power infrastructures [6]. For example, Google, HP, and IBM researchers have proposed various ways to implement power oversubscription in data centers [12], [27], [17]. Google recently conducts analysis on three kinds of workload traces they collected from real data centers: search, webmail, and MapReduce [12]. Their study shows that the peak power is as high as 96% of the rated capacity at the rack level, but much lower (72%) at the data center level, because the power consumption of different racks rarely peak simultaneously. Therefore, they conclude that there is a substantial oversubscription opportunity, which would allow 38% more servers to be safely hosted in their existing data center, without spending a huge amount of money to either upgrade the power infrastructure or build new data centers.

It is important to note that traditional data centers commonly adopt a conservative provisioning methodology to host servers based on their nameplate power rating and thus have very small probability for power overloading. However, today's data centers increasingly rely on power oversubscription to avoid or defer the costly power infrastructure upgrades, which significantly increases the opportunity of having undesired power capacity violations.

As we can see, a strong assumption made for power oversubscription is that the power consumption of most racks or PDUs in the data center never peak at the same time, which has been demonstrated to be valid with normal data center workloads in numerous studies (e.g., [6], [12], [15]). Unfortunately, an unsafe implementation of power oversubscription could lead to a serious vulnerability for data centers, e.g., a malicious attacker may manipulate many servers to have their power peak simultaneously, which can then lead to the violation of some rated power capacities in a data center. As a result, the overloading of electrical circuits could trigger branch circuit breakers to trip, leading to undesired outages and then the disruption of important services. To prevent undesired power overload, the power consumption of each rack enclosure, each Power Distribution Unit (PDU), and the entire data center must be carefully provisioned and then properly controlled at runtime, in order to stay below the desired power limits at each level [36]. With the pervasion of outsourcing cloud services such as infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS), the workload of the data center will be impacted severely by the cloud service users. When all the users of cloud services are benign, the workload of a data center will follow the normal pattern and power oversubscription can be applied safely. However, an adversary can gain access to data center resources and make changes to workload easily. By deliberately adjusting the workload of the servers within a rack, an adversary can make all servers reach power peaks simultaneously and the circuit breaker might be tripped. Moreover, a more advanced attacker may even generate power spikes in servers within same PDU or even UPS to cause higher level utility failure.

In this paper, we will demonstrate how an adversary can generate power spikes in three main-stream cloud business models, PaaS, IaaS and SaaS, respectively. Our observations and experiments will prove that current power management strategies in a data center will face the serious threat of a power attack.

III. Threat model

In this section, we describe the threat model of power attacks. In particular, we present the reasonable assumptions we made for the study of power attacks.

The target of a power attack can be a rack, a PDU, or even the entire data center, and we assume that the victim has the following configuration features. (1) The target is running certain cloud services that are available to public. The target can run IaaS, PaaS and SaaS, and any users including an attacker can subscribe the services. (2) The target data center deploys power oversubscription as its power management solutions. (3) The target data center monitors and manages power consumption at the rack or PDU level. In a large data center, it is very difficult to monitor power consumption of all servers in a very fine-grained manner. And the accurate power sampling for thousands or tens of thousands servers will induce high overhead [24]. Therefore, power monitoring is at the rack or PDU level, instead of the server level. (4) The target data center performs certain routines such as virtual machine migration and deploys basic load balancing systems.

The adversary who launches a power attack could be an individual hacker, a botnet master, a competing cloud service provider, or an organization for committing cybercrime/cyberwarfare. However, we assume that the attacker is always from outside. The resources and capabilities of the attacker has are detailed as follows. (1) The attacker communicates with the target via the public service interface provided by the cloud. The attacker accesses the target as a regular user, and no additional privilege is available to the attacker. (2) The attacker has sufficient resources to launch a large scale attack. The attacker has the capability of subscribing the target's service with a large number of user accounts and generating a large amount of malicious workload/requests to the target. (3) The attacker can infer internal information of the target by exploiting certain probing techniques. Some network probing tools have been leveraged to infer the physical topology of a data center [29], revealing the connection between the IP address of a machine and its physical location, as well as verifying if two VMs reside in the same physical machine. Moreover, for easy management, normally data center administrators place the servers that provide the computing services for the same enterprise or group of users in the same rack. Also, the IP addresses of the physical machines that connect to the same rack share the same network ID and hence are close to each other. Therefore, we assume that the attacker is able to approximately locate the target machines that are within the same rack or PDU[1]. On the other hand, to successfully launch a power attack at the rack level, it is not required to pinpoint all these servers belong to the same sack. As long as attackers know one target and its IP address, they could simply launch a mini brute-force attack by injecting malicious workloads to a range of IP addresses, which cover the target and most of the other machines in the same rack.

The process of launching a power attack is also the process of consuming the services provided by the target, and the attacker must pay for the computing services. Thus, there is a cost related to launch a power attack. However, the damage caused by a power attack could be catastrophic. Once a CB is tripped, all servers connected will be blacked out and all services running will be interrupted. Such damage is much more severe than that caused by traditional attacks such as DoS attacks. Therefore, for those attackers who commit a cybercrime or cyberwarfare, we assume that they have a strong motivation and rich resources to launch a power attack.

In different cloud environments, the attacker has different control over the target's computing resources and services. For instance, in IaaS, the attacker can obtain the full control over owned virtual machines. But in SaaS, the attacker can only access the target by issuing network requests. Therefore, the key technical challenge of launching a power attack is how to construct effective attack vectors in different cloud environments, i.e., how to generate more power consumption of the target in different circumstances. In the following three sections, we detail the attack vectors in three main-stream cloud business models, PaaS, IaaS, and SaaS, respectively.

IV. Power Attack in PaaS

In this section, we investigate the attack vectors in PaaS environments and design corresponding experiments to evaluate the power attacks. Based on the experimental results, we further conduct damage assessment and analysis.

A. PaaS and attack vectors

Platform as a service (PaaS) provides computing platform to users. The service vendor will manage the hard devices, OS, and middleware, but users can customize the applications running on the platform. With PaaS, application developers are

[1] An example in Amazon EC2 is shown in Appendix.

TABLE I. CONFIGURATION OF SERVERS USED IN EXPERIMENTS.

	Server A	Server B	Server C	Server D	Server E	Server F	Server G
CPU	2*Xeon W3540 Dual Core	2*Xeon 5520 Quad Core	2*Xeon 5130 Dual Core	2*Intel E4600	2*Intel E4600	2*Xeon Dual Core	2*Xeon Dual Core
Memory	2*1GB DDR2	6*1GB DDR3	4*1GB DDR2	2*1GB DDR2	2*1GB DDR2	2*512MB DDR, 6*1GB DDR	16*1GB DDR2
Hard Disk	1*7200RPM SATA	6*7200RPM SATA	4*7200RPM SATA	2*7200RPM SATA	2*7200RPM SATA	2*7200RPM SATA	2*7200RPM SATA
Host OS	Ubuntu 12.10/3.5.0-17 kernel	Ubuntu 12.10/3.5.0-17 kernel	Ubuntu 12.10/3.5.0-17 kernel	Ubuntu 12.10/3.5.0-17 kernel	Ubuntu 12.10/3.5.0-17 kernel	Ubuntu 12.10/3.5.0-17 kernel	Ubuntu 12.10/3.5.0-17 kernel
VMM	Xen 4.1.2	Xen 4.1.2	Xen 4.1.2	Xen 4.1.2	Xen 4.1.2	Xen 4.1.2	Xen 4.1.2

able to save considerable cost and complexity of running the underlying computing platforms.

In a PaaS environment, a user can run various applications flexibly on the platform, i.e., the user can get full control over the workload pattern. This feature of PaaS provides an attacker the opportunity to launch a power attack. The attacker can subscribe the platform from the service provider and run specially crafted workload. The malicious workload can lead to a significant rise of power consumption.

Some PaaS providers [1] deploy load balancing mechanisms to prevent a workload burst. A load balancing system normally monitors the system utilization of the servers and dynamically schedules workload. However, load balancing is not equal to power balancing. It is very hard to accurately model the power consumption of a machine with respect to system utilization. With these characteristics of PaaS, we present a potential attack vector as follows: an attacker can subscribe multiple servers that are within the same rack (connected by same breaker) from providers and run specially crafted applications/workload on them.

The attack workload should be designed to significantly increase the power consumption of victim servers in two phases. First, a heavy workload is generated to exercise system utilization to a high level. This will directly increase the power consumption of victim servers. The system utilization should reach a certain cap under such a high workload, e.g. the system utilization may reach the cap of load balancer or CPU utilization reaches the 100% cap. In the second phase, after reaching system utilization cap, the workload is no longer increased. Instead, the patterns or configurations of the workload will be adjusted. Since different workload configurations yield different power consumptions, by adjusting workload patterns or configurations, the power consumption of victim servers will be further increased to an even higher level without increasing system utilization. In this way, the target CB could be overloaded.

B. Attack evaluation

To evaluate the feasibility and effect of power attack in PaaS, we conduct experiments in a testbed that simulates the PaaS environment. The configuration of all the servers used in our experiments can be found in Table I. While high performance computing (HPC) has become a pervasive service nowadays, the demand of tremendous resources and parallel computing makes HPC a suitable candidate of PaaS. Thus, we use HPC as the workload of PaaS in our study.

1) Single Server: First, we conduct single server experiments to figure out how different workloads may affect the power consumption and system utilization of a server. The testbed of this round of experiments is Server A in Table I. SPECCPU2006 is used as the benchmark in our experiments. SPECCPU2006 is a CPU intensive benchmark that is widely used as HPC benchmark. There are different benchmarks in SPECCPU that perform different computations. Since these benchmarks will yield different workload patterns, we can regard them as different HPC applications running in PaaS environments.

Figures 2 and 3 illustrate power consumption and memory utilization of different benchmarks in SPECCPU2006. These benchmarks are carefully selected so that they consume similar amount of memory. Since our testbed has four cores, we run four copies of SPEC benchmarks in parallel to fully exercise all cores. Since all benchmarks can exercise the CPU to reach the same utilization, we do not show CPU utilization in the figure. As Figure 2 shows, different workloads yield very different power consumptions. Figure 3 illustrates the memory consumptions of these benchmarks. Note that SPECCPU involves negligible disk and network activities, so their impact on power consumption is insignificant and can be ignored.

As Figure 3 shows, while all these benchmarks induce the same CPU utilization, benchmark 465 consumes the least memory. However, benchmark 465 consumes more power than many other benchmarks. For instance, while benchmark 462 consumes around 150 W power, benchmark 465 has power spikes up to 175 W, which is over 15% more than that of benchmark 462. We can also see that the memory consumption of benchmark 462 and that of benchmark 456 are very close. The average memory utilization for 462 is 24%, while the average memory utilization for 456 is around 25%. With the same CPU utilization, similar memory usage, and negligible I/O activity, we can regard that benchmark 462 and benchmark 456 consume very similar amount of system resources. However, from Figure 2 we can see that benchmark 456 consumes over 20% more power than benchmark 465. Our observations indicate that system utilization, i.e, resource consumption, cannot accurately determine power consumption.

We also run another HPC benchmark, High Performance Linpack (HPL) on the testbed. HPL is a benchmark to calculate random matrix production. It has multiple parameters to configure, which will affect the performance of the benchmark. In our experiments, we take the following root parameters

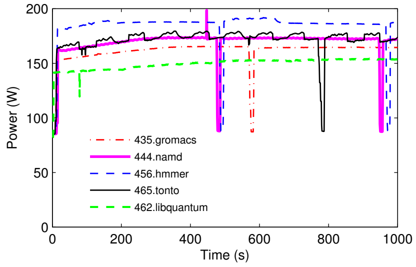

Fig. 2. Power consumption of the server with different SPECCPU workloads.

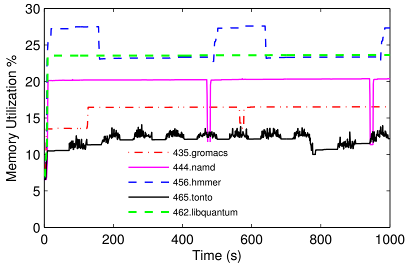

Fig. 3. Memory consumption of the server with different SPECCPU workloads.

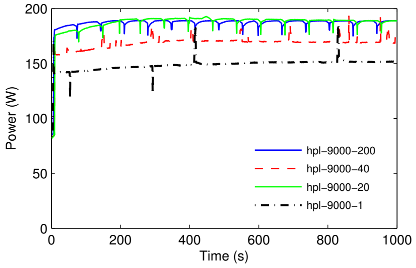

Fig. 4. Power consumption of the server running HPL benchmark with different configurations.

into consideration: the processor grid, i.e., the number of processors, the problem size N which is the size of input matrix, and the block size NB which determines how HPL solves the matrix production problem. Since our testbed is a 4-core machine, we fix the number of processors to be 4. To make the input size consistent, i.e., make HPL consume the same amount of memory, we fix the parameter N as 9000. We adjust the value of NB among 200, 40, 20, and 1. As the CPU utilization and memory utilization remain the same in these experiments, we do not present their results here.

Figure 4 illustrates the power consumption of our testbed while running HPL with different configurations. The adjusted parameter, NB, will determine the way HPL solves the problem. From the figure we can see, with a different value of NB, the power consumption of our testbed differs significantly. When the block size is set to 1, the testbed only consumes less than 150 W power. By contrast, the testbed has power consumption near 190 W when the block size is set to 20. Such results indicate that even for the same application, different parameters or configurations can yield considerably different power consumption.

2) Rack-level Cluster: To verify if an attacker can generate significant power rise by adjusting the workload beyond a single server, we further conduct experiments in a rack-level cluster. We setup a 4-server rack with Server D, Server E, Server F, and Server G in Table I. These four servers are connected to the same switch and circuit breaker, resembling a rack in a real world data center.

We run SPECCPU2006 on all servers in the rack. The overall power consumption of the entire rack is recorded. These benchmarks are configured to exercise CPU to full utilization (reaching the cap) and the memory usage percentage of each server on these benchmarks are the same as those in our single-server experiments. Therefore, we only present the results of power consumption at the rack level, which are illustrated in Figure 5.

As Figure 5 shows, the rack level results concur with our single server results. While the CPU utilizations in all cases reach the cap, some benchmarks generate more power consumption than others. Such an observation indicates that even after the system utilization reaches a cap, an attacker has the potential to increase the power consumption of target by adjusting the workload, e.g., the attacker can change the workload from benchmark 462 to benchmark 456 to further increase power consumption.

Then we design two malicious traces to launch a power attack against a rack in PaaS environments. The first trace is based on SPECCPU2006. At the beginning, the workload behaves as a normal workload that generates moderate system utilization and power consumption. We use benchmark 462 with light workload configuration to represent such a moderate workload. Next, the attacker can change the workload to exercise the system utilization to a certain level to significantly increase power consumption. We use benchmark 462 with heavy workload to represent the malicious workload during this phase. Finally, after system utilization reaches the cap, the attacker tunes the workload to further increase power consumption. Here we use benchmark 456 to represent the malicious workload of this phase.

The second malicious trace is based on High Performance Linpack (HPL). While SPECCPU2006 can be used as running independent workloads on different machines, we use HPL to simulate the scenario where multiple servers are coordinated to run the same task in PaaS. Each of the four servers works as a node in the working cluster and they communicate with each other via OpenMPI. HPL will distribute the workload to each node for high performance computation. In this round of experiments, we configure HPL with different problem sizes (N) and different block sizes (NB). The power attack based on HPL is mounted as follows. Fist, the input size of the workload is set to be moderate, we use HPL with N set to 1000 and NB set to 5 to represent the moderate workload. Next, the attacker can enlarge the input size to increase the system utilization. In our experiments, we increase the block size to 4000 during this phase and more CPU cores are exercised. At last, the attacker can change the workload pattern to further increase power consumption. In our case, we modify NB from 5 to 100.

The evaluation results of the malicious workload are illustrated in Figure 6. It is evident that both malicious traces can generate a significant rise in the overall power consumption of the rack. As Figure 6 shows, the power attack can trigger over 30% increase in power consumption.

C. Damage assessment and analysis

Our experimental results above validate that in PaaS environments, an attacker can generate abnormal high power consumption by adjusting workload running on target machines.

The damage caused by such a power attack is at two levels. A relatively light damage can be overheating the IT equipments

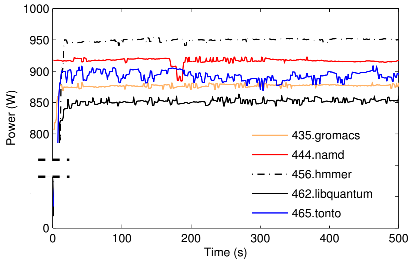

Fig. 5. Power consumption of the rack while running different benchmarks from SPECCPU.

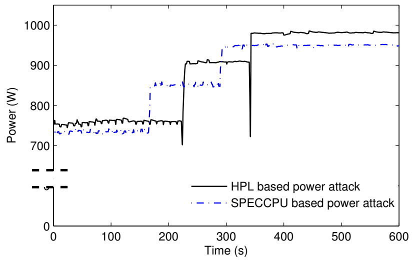

Fig. 6. Power consumption of the rack under power attack.

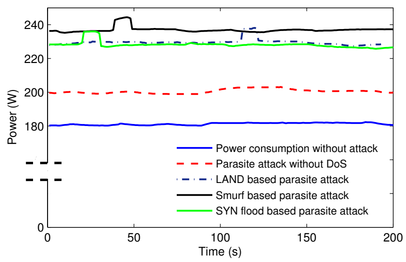

Fig. 7. Power consumption of the server under parasite attack.

and degrading the performance of the victim servers. During our experiments, when launching a power attack against the rack with our malicious traces, the CPU cores of server E were overheated, resulting in system failure. In a rack where power is aggressively oversubscribed, a power attack can lead to more serious consequence: the trip of circuit breaker (CB). The 4-machine rack used in our experiments is located in a server room with the total number of 16 servers. The entire server room can be regarded as a PaaS rack, where all servers are connected by the same CB. Users can run different applications on the servers in the room and only the four servers in our small rack are under our control. When we conducted experiments that run the SPECCPU 456 benchmarks, the CB of the server room was tripped. This accident indicates that power attacks can be a serious threat in real world.

However, our experimental results do not stand for the most powerful power attack in real world. First, the HPC benchmarks we used are only CPU-intensive. The memory and I/O devices are not fully exercised in our experiments, leaving space for further increase of power consumption. In real world, an attacker can include memory and I/O intensive workloads to further increase power consumption. Second, the servers used in our experiments have poor power proportionality. These servers consume over 60% of peak power when being idle. Such poor power proportionality will significantly reduce the effects of power attack because there is not much room for the increase of power consumption. In real world, a power attack against data centers with more advanced servers should be able to produce more significant impacts than our experimental results.

V. Power Attack in IaaS

This section describes the potential attack vectors in the IaaS environment and presents the experimental results and analysis of evaluation on two attack vectors: parasite attack and VM migration.

A. *IaaS and attack vectors*

Known as infrastructure as a service, IaaS is a cloud-service model in which the provider offers physical or virtualized infrastructure along with other resources to users. Amazon's Elastic Cloud (EC2) is a popular IaaS service. In the EC2 environment, a user can instantiate virtual machines (VMs) via the interface or API provided by EC2. The booted VMs are under full control of the user. In other words, while the service vendor manages the hypervisor and physical devices, the user can determine the OS, middleware, application and data running on the VMs. IaaS provides a cost-effective way for enterprises to modernize and expand their IT capabilities without spending capital resources on infrastructure. However, IaaS-based data centers are also exposed to the threat of power attack.

First of all, the IaaS business model allows an attacker to have more control over the target. The attacker can instantiate many VMs with minor cost and run any kind of workloads on the VMs.

Second, IaaS divulges a considerable amount of internal data center information to the attacker. For the convenience in management, an IaaS data center often uses some well-known topology and networking configuration strategies [29]. Thus, the attacker can infer the internal structure of the data center and locate the target inside the data center via network probing.

Third, the widely used virtualization techniques in IaaS expose performance vulnerability to malicious attackers. In particular, the additional layer introduced by virtualization makes many system activities such as I/O operations more costly. The induced high overhead can be exploited by attackers to generate power spikes.

Based on these vulnerabilities of IaaS, we propose two attack vectors to launch a power attack in IaaS environments.

The first attack vector is parasite attack that leverages controlled VMs to attack the host physical machine from inside, resembling a parasite consuming its host from inside. On one hand, the controlled VMs can directly run intensive workloads to increase the power consumption of the host. On the other hand, the controlled VMs can exploit the vulnerability of virtualization to further abuse more resources and power of the host system. For instance, DoS attacks towards a parasite VM can consume considerable resources of the hypervisor [30], potentially increasing the power consumption of the host. Using these two attack strategies together, a parasite attack can significantly increase the power consumption of a target system.

The second vector is VM migration that is a routine operation in the cloud. Certain VMs require live migration to perform maintenance and update. VM migration is a high power consuming operation. If an attacker can understand how VM migration is performed in an IaaS data center, VM migration can be exploited to help launch a power attack.

Knowing that a number of VMs are being migrated to a rack, the attacker can launch a power attack like a parasite attack against the rack at the same time period. Since VM migration itself can cause high power spikes, it will greatly aggravate the power attack and cause the trip of CB.

B. Evaluation of parasite attack

The complete process of a parasite attack is as follows. First the attacker keeps instantiating VMs and infers their physical locations with the strategies mentioned above. In this way, the attacker can finally place many VMs on the physical machines within a target rack. Then, the attacker can run intensive workloads on the controlled VMs to increase the power consumption of the host systems. During this phase, the parasites fully consume the resources that are allocated to them by the hosts. Finally, the attacker can launch some special attacks, e.g., DoS attacks towards the parasite VMs. Since the parasite VMs are under full control of the attacker, an attack towards parasites can ensure a success. Due to the performance penalty of virtualization, such an attack can trigger unexpected system activities at the hypervisor level, leading to resource abuse of the host system. As a result, the power consumption of the entire host system can be further increased to a higher level.

To evaluate the feasibility of a parasite attack, we build up a virtualized system with multiple VMs and launch attacks against one of the VMs. The host machine is Server B in Table I. We run 4 virtual machines over the host, including the "parasite VM" controlled by the attacker. These VMs are installed with Ubuntu 12.10 and they are configured with 512 MB memory and 4 vcpus. The open-source tool *hping3* is used to launch DoS attacks. Three different types of DoS attacks are launched: TCP SYN flood, Smurf, and LAND attacks. The power consumption of the host machine is recorded.

At the beginning, all these 4 VMs are running certain workloads so that their system utilizations remain around 25%, which is normal in real world. Under this scenario, the host consumes around 180 W of power. In the next step, we run intensive workload on the VM controlled by the attacker. With the parasite VM being fully exercised, the power consumption of the host is increased to around 200 W. Then, we launch DoS attacks against the parasite VM. Under DoS-based parasite attacks, the power consumption of the host is increased to above 230 W, with power spikes that can reach 245 W. The experimental results are shown in Figure 7, and they clearly demonstrate that parasite attacks can increase the power consumption of the host by over 30%.

C. Exploiting VM migration

Since users have more control over VMs and more internal information is available in IaaS, VM migration can be exploited to help launch a power attack in IaaS. To measure the power consumption spikes generated during a VM migration, we conduct three rounds of experiments. First, a basic VM migration is conducted to verify that for a server involved in the migration, it will experience a power rise. In the second round, the scenario where VMs are migrated within a rack is emulated to show the impact of intra-rack VM migration on the overall power consumption of a rack. Finally, we emulate the scenario where multiple VMs are simultaneously migrated from other racks to a target rack to demonstrate the threat caused by inter-rack VM migration.

In the first round of experiments, Server B and Server C in Table I are used as our testbed. The VMs running on the servers are initialized with 512MB memory, 8G image size, 4 vcpus and default credits (512). We set server B as the monitored server whose power consumption will be recorded. We first set server B as the destination server, migrating 1 idle VM from server C to server B. Then we set server B as the source server, migrating 1 idle VM from it to server C.

Figure 8 illustrates the power consumption of the monitored server. The figure demonstrates that during the migration, as either the source or the destination, the server will experience a rise of power consumption. The cause of a short period of power spike is the initialization and operation of VM migration. At the source side, the memory contents need to be duplicated; additional computation is required to prepare the transition; and networking devices are also exercised to transfer VM information. At the destination side, additional resources are allocated to the new incoming VM, increasing the server's power consumption.

In the second round, we emulate the intra-rack VM migration as following. We connect Server A, Server B, and Server C to the same circuit breaker, making up an IaaS rack. We boot 8 VMs in each of the 3 servers with the SPECCPU benchmarks running on them. In this round of migration, we migrate 4 VMs from server A to server B, 4 VMs from server B to server C, and 4 VMs from server C to server A.

The power consumption of the entire rack during the migration is illustrated in Figure 9. We can see that there are several crests of power consumption. This is due to the different configurations of the servers in our testbed. Although all the migrations are started at the same time, different configurations lead to different migration time and different power consumption. These results indicate that when multiple VMs are migrated together as in our experiments, the rack will experience some unexpected power spikes. In our experiments, the rack has already been working in a high power consumption state. During the migration, the power consumption of the rack further rises from 560 W to 640 W. The power spikes over 600 W last for over 15 seconds. Suppose the CB of the rack has a rated power capacity of 600 W, as the power consumption of the rack is below 560 W both before and after migration. If the migration strategies do not take the migration power spikes into a serious account [8], [32], the power spikes will trip the CB of the rack, resulting in disastrous server shutdowns.

In the third round, we emulate the inter-rack VM migration. In real world data centers, it is common that a number of virtual machines are migrated simultaneously, probably towards the same rack. For example, periodic live VM migration has been commonly adopted as an effective way to perform server consolidation for higher resource utilizations in data centers [33], [34]. In our experiments, we set server B in a separate rack and set servers A and C in the other racks. Therefore, the power consumption of server B is recorded as the power

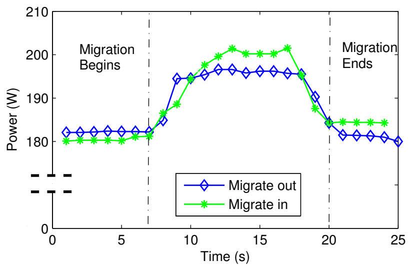

Fig. 8. Power consumption of an involved server during VM migration.

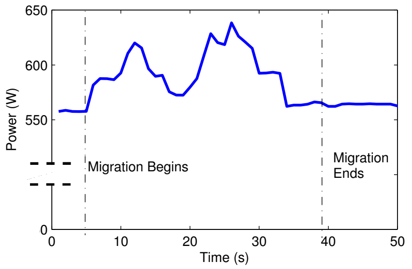

Fig. 9. Power consumption of the rack during intra-rack VM migration.

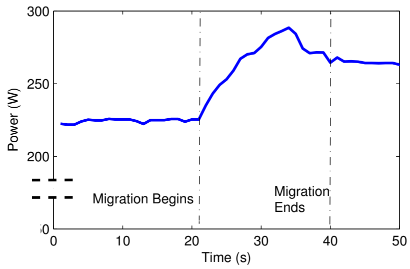

Fig. 10. Power consumption of the target rack during inter-rack VM migration

consumption of the target rack. We migrate 2 VMs from server A to server B and another 2 VMs from server C to server B in parallel. Server B originally runs 4 VMs with the SPECCPU workload and all the VMs that are migrated to server B also run SPECCPU on them.

Figure 10 illustrates the power consumption of the target rack during the migration. At the beginning, the target rack has power consumption around 225 W. When the migration begins, the power consumption of the target begins to rise rapidly. Within a short period, the target can reach a power peak over 280 W. After the migration ends, the power consumption of the target rack reduces to around 260 W. Such results indicate that if there are multiple VMs migrated to one rack simultaneously, the target rack will experience significant power spikes and such power spikes can be exploited by an attacker to trip the CB of the target rack.

D. Damage assessment and discussion

Our experimental results verify the feasibility of launching a power attack in IaaS. With one parasite VM residing in the host, a parasite attack can increase the power consumption of a virtualized system by more than 30%. Such an attack is as powerful as the attack in PaaS. In real world, the parasite attack effect can be higher than what we achieved in the experiments since the attacker is able to place more than one parasite VMs on a target. More parasite VMs imply that more controlled VMs on the host run in full utilization, generating more power consumption. In addition, more parasite VMs can also make DoS attacks more powerful. As Figure 7 shows, Smurf attack incurs more power consumption than TCP SYN flood and LAND attacks. The reason is that TCP SYN flood and LAND attacks can only affect one victim VM, but Smurf attack broadcasts packets to a range of IP addresses and makes VMs communicate with each other, which can affect multiple VMs. Thus, if the attacker is able to launch DoS attacks to multiple parasite VMs, the impact of a power attack upon the host will be more significant.

In addition to parasite attacks, we also demonstrate that VM migration can be exploited to help launch a power attack. Although attackers can hardly directly manipulate the VM migration routine in a data center, they can infer how and when VM migration is conducted and launch a power attack against the rack that is conducting VM migration. As our experimental results demonstrate, VM migration can increase the power consumption of a rack by over 30% even when the rack is already imposed with heavy workload. Therefore, launching a power attack against a rack conducting VM migration can amplify the damage caused the power attack and trip the CB more easily. VM migration can also be leveraged to help mount a power attack in PaaS and SaaS environments, where virtualization and VM migration are used.

VI. Power Attack in SaaS

With SaaS being the most popular cloud service model, we exploit the attack vectors in SaaS scenarios and conduct a set of experiments to verify their feasibility.

A. SaaS and attack vectors

Software as a service (SaaS) delivers the application managed by third-party vendors to cloud clients, and users can access the applications via client-side interfaces. The most typical SaaS service is web service. Compared with PaaS and IaaS, the users of SaaS have much less control over the infrastructure. The service vendor manages the underlying hardware, middleware, OS and applications, which are transparent to users. A user can access the application only via the interfaces provided. Therefore, standing at the perspective of power attacker, SaaS provides very limited control over the target. The attacker can only access the target via certain APIs or interfaces (usually web browser). However, as pointed out by many previous works, certain specially crafted web service requests will consume more system resources, therefore resulting in the potential of a power attack.

In a typical web service, HTML pages are dynamically generated when receiving requests. Some contents of the requested web page need to be constructed on the fly or fetched from database. During this process, two levels of caching, object cache and in-memory cache, are used to help to optimize the performance. Normally many cache misses can produce considerable negative impact on the system performance and lead to the increase of power consumption. Thus, an attacker will attempt to generate requests that trigger a large number of cache misses to launch a power attack.

Moreover, different computation will induce different power consumption for a system. For instance, floating point operations may consume more power consumption than integer operations. In modern processors such as x86 processors, Arithmetic Logic Unit (ALU) performs integer operations while the Floating-Point Unit (FPU) takes the responsibility of executing floating point operations. FPU is more power hungry

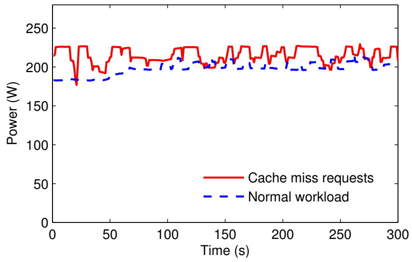

Fig. 11. Power consumption of the server under normal workload and cache miss workload

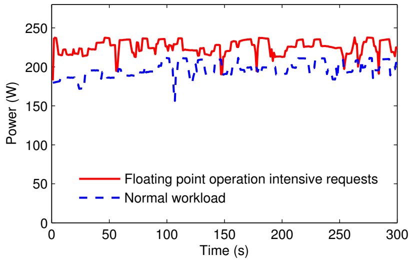

Fig. 12. Power consumption of the server under floating point operation intensive workload

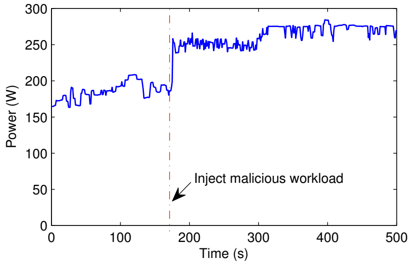

Fig. 13. Power consumption of the server under power attack.

than ALU, indicating that floating point operations are more power expensive than integer operations. Meanwhile, different arithmetic computation will consume different amount of resources as well, e.g., division operation is more costly than add and multiplication operations. Such power consumption discrepancy in computation provides another attack vector. An attacker can launch a power attack by sending requests that involve a large number of expensive floating point operations.

B. *Attack evaluation*

To evaluate a power attack in SaaS environments, we set up a testbed to deploy web services and conduct a series of experiments to generate power spikes. The server used in the evaluation is Server B in Table I.

The RUBiS benchmark is used in our evaluation. RUBiS is a web benchmark that emulates online-shopping web services. RUBiS provides various features of classic online shopping websites such as browsing goods, selling items, bidding on items, registration and viewing user information. Meanwhile, RUBiS also provides client emulators that behave as real world users. The "transition table" defined by RUBiS describes the behaviors of the emulated clients. By modifying the transition table, the client emulator can generate different request patterns.

We deploy RUBiS as a 3-tier web service, in which Apache 2, Tomcat7, and MySQL are used as the web server, the application server, the database, respectively. We populate the database with 100,000 unique items and 50,000 different users. To Make RUBiS more suitable for our experiments, we modify the source code of RUBiS to include some additional functionalities. For instance, we make RUBiS capable of performing "discount" operations, i.e., a user can have coupons to get discount, reducing the buy-out prices of items by certain percentage. The users can purchase multiple items or a certain number of one item, and RUBiS will calculate the overall price. We also modify the client emulator to make it more flexible and capable of generating specially crafted requests.

In our experiments, we first explore the requests that will trigger cache misses. RUBiS provides a "default" transition table that defines the normal traffic, so we use it to represent the normal workload of RUBiS. During the experiments, 4,000 clients are emulated. To generate the malicious cache-miss traffic, we modify the transition table so that the clients will continuously browse items in a totally random manner. In this way, both of the object caches and in-memory caches will be flushed frequently, resulting in considerable cache misses. Figure 11 shows the comparison of power consumption between normal traffic and cache-miss traffic. The results demonstrate that cache-miss traffic can generate significantly more (over 15%) power consumption than normal traffic.

We also conduct experiments to verify that floating point operations generate more power consumption than integer operations. First, we populate the database to set the prices of all items to be integer numbers. Then we emulate 4,000 clients to browse and purchase items with the access pattern provided by the default transition table. Such requests represent the normal workload of the web service. After that, we update the prices of all items to be floating point numbers. While the requests still follow the access pattern provided by the default transition table, the clients are crafted to purchase multiple items and use coupons while checking out. Such malicious requests cause the server to perform a considerable amount of floating point operations. The experimental results are illustrated in Figure 12. Compared with the normal workload, it is evident that those malicious requests generating a large number of floating point operations can force the web server to experience a significant rise of power consumption.

Finally, we combine the two attack vectors mentioned above to generate a malicious trace and then launch a more powerful attack. Again, we use the trace of 4,000 emulated clients generated by the default transition table to represent a normal workload of the server. To launch the attack, we craft the malicious requests to trigger both cache misses and expensive floating point operations. While running the normal workload, we launch the power attack by injecting crafted requests from 2,000 malicious clients. These malicious clients perform browse-and-purchase operations. Each of the malicious client first browses random goods with floating point prices, then purchases a random number of the browsed items. Meanwhile, the client uses coupons to get discount on the items bought. In this way, the server has to perform numerous add, multiplication and division floating point operations. Moreover, since the clients are browsing and purchasing items in a random fashion, a large number of cache misses are triggered. Figure 13 illustrates the power consumption of the victim server under the power attack, which is mounted at 180s. The results clearly demonstrate that our power attack can induce a significant rise in power consumption of servers in SaaS environments.

TABLE II. SIMULATION ENVIRONMENT SETUP.

Parameter	Value
# of Servers	139,200
# of racks	approximate 700
# of PDU	approximate 20
# of CBs	approximate 30 (per PDU + per DC)
Capacity of PDU-level CB	150kW
Capacity of DC-level CB	1MW
CPU Per Server	dual-core 2.0GHz Xeon
DRAM Per Server	16GB
Disk Per Server	2TB
Est. Peak Power per Server	240Watt

C. Damage assessment and discussion

Our experimental results verify that specially crafted web requests can generate significantly more power consumption of servers in SaaS than normal requests. In our SaaS experiments, the power attack can increase the power consumption of a victim server by 30 to 40 percent, which is even more significant than those in PaaS and IaaS environments. Therefore, the damage caused by a power attack in SaaS can be as great as in PaaS and IaaS. In general, the attack impact upon SaaS mainly depends on three factors, the per-request power consumption, the malicious request rate, and the attack duration. To make an attack powerful and stealthy, seeking an attack vector with high per-request power consumption is the key.

Besides the attack vectors mentioned above, certain web applications expose particular vulnerabilities that can be exploited by attackers. For instance, an attacker can launch algorithmic complexity attacks [10] against web applications that involve with many hash table operations. Algorithmic complexity attacks can make hash table operations always suffer from the worst case complexity, therefore consuming much more resources. For web applications deployed with large databases, requests that compete on database locks can also generate significantly more resource consumption [26].

VII. DATACENTER LEVEL SIMULATION

While we have shown the feasibility of mounting a power attack at the server and rack levels, such attacks could be spawned to the data center level, which may lead to more severe and disastrous consequences. In this section, we study the impact of power attacks at the data center level (DC-level) including the large size PDU-level based on simulations. We first introduce the setup of simulations and then present the simulation results and analysis.

A. Simulation Setup

1) Platform: Based on our server-level and rack-level experimental results, we build the simulation models and configure the data center parameters following the description of the Google data center in Lenoir, North Carolina, USA [11]. We assume there exists a simple workload management scheme in the data center, which can distribute all workloads to each PDU evenly. All simulation parameters and their values are shown in Table II.

In the simulation, we build the similar power infrastructure as in Figure 1. Based on our threat model, a power attacker can obtain the knowledge of racks and servers in the same PDU to launch attacks to each single PDU inside the power infrastructure. We use a boolean checker as the CB in the simulation implementation. Once a PDU-level CB is tripped, servers connected to the corresponding branch circuit shut down consequently. As a result, we can observe that the monitored victim PDU power drops down to zero. All services dispatched to servers in this PDU will be redistributed to other servers with stable power supply from different PDUs. A similar CB is implemented for the whole data center. From the public information of the Google data center, we find that the nameplate capacity of the DC-level CB is smaller than the sum of all capacity of PDU-level CBs. The reason is due mainly to the unplanned capacity increase of the Google DC (adding more servers and PDUs) without updating the whole power infrastructure. Thus, there exists a possibility that the DC-level CB could be tripped without failing any PDU-level CBs, which is confirmed by our simulation results shown in Figure 17.

2) Workloads: Two workload traces are used in our simulation, named as “Original” and “Attack” representing normal and attack activities, respectively. The original trace consists of the daily workloads of the Google data center [11]. The attack trace includes the workloads similar to HPC workloads, which can increase the power consumption of a target by up to 30%[2] in a short time. As a sample of the daily workload, we show a three-day workload trace from Google in Figure 14. Note that, in Figure 14, we define three regions as peak, medium, and valley, with respect to the workload dynamics. The peak region is above the top 10-percentile of the workload, e.g., the workload density of 30,000 queries per second (qps) is the top 4-percentile of the workload. By contrast, the valley region is below the 90-percentile of the workload. The rest in between is the medium region. The three regions represent the three typical running states of a data center, busy, normal, and idle. We design simulations that launch the power attacks in these three regions, respectively, to examine the power attack impact at the DC level under different running states of the data center.

B. Simulation results

Prior to the simulation of the DC level power attacks, we first simulate the power attacks at the PDU level to demonstrate the impact of power attacks on a victim PDU and how PDU level attacks can affect the power consumption of the entire data center.

When an attacker acquires the information of those servers located in the same PDU, the attacker can launch a power attack against this PDU to trigger a power outage. We simulate this scenario where the power attacker targets several particular PDUs. The capacity of the PDU-level CB is shown in Table II.

First, we show a snapshot of the power attack on a large size PDU (including 40 racks, 650 machines) in Figure 15. The attack begins at time 12min and lasts for 22 minutes till the PDU-level CB is tripped. The whole PDU fails and all the servers powered by this PDU are shut down completely. Thus,

[2] The number is based on our experimental results in Sections IV,V, andVI.

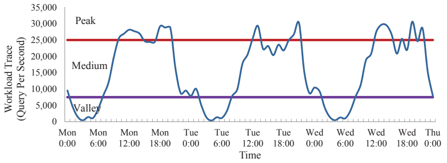

Fig. 14. The workload trace of a three-day period in May 2011 from the Google cluster in [11].

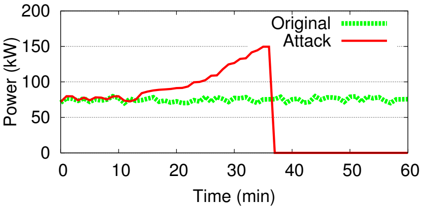

Fig. 15. The snapshot of power attack on one PDU. This is the first attack shown in the scale of data center as in Fig. 16.

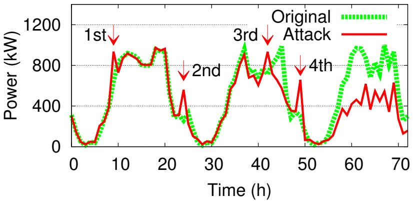

Fig. 16. The snapshot of continuous power attacks on multiple PDUs in the scale of data center. There are multiple attacks launched at time between 9-10(1st), 24-25(2nd), 43-44(3rd) and 49-50(4th).

compared with the power behavior of the original workloads, the power consumption of the PDU drops immediately after the success of the power attack. In our simulation, we have performed four similar power attack attempts to four different PDUs sequentially at the scale of a data center. The results are shown in Figure 16, in which each attack is represented as an arrow and the first arrow is the attack shown in Figure 15.

All attacks successfully trip the targeted PDU-level CBs. However, the power curve of the entire data center recovers shortly after the first three attacks. This is because when one PDU fails, the workload manager in DC restarts and evenly redistributes those workloads from the servers lost power to other servers with stable power supply. Thus, the impact of PDU-level power attacks is reduced at a certain degree at the cost of power load increase in other PDUs. However, after the fourth attack in 49-50 hour, the computing capacity of whole data center (i.e., the number of available servers) is greatly diminished due to the power outages. The remainder of available servers cannot support the significant increase of original workload density starting at 58th hour, regardless of the workload management. Thus, the load balancer of the data center starts to reject service requests and the rejection ends till 69th hour. During this period of time (i.e., from 58th to 69th hours), only about 53% of service requests are processed and the rest are rejected. As a result, the DC-level power consumption is just half of the original amount, which is clearly shown in the area of the oscillating power curve in Figure 16. The entire data center finally resumes to normal between 69th and 70th hours with the significant decrease of workload supply.

Next, we target at tripping the DC-level CB. As aforementioned, it is possible that the DC-level CB is tripped without the trip of any PDU-level CBs. Here we illustrate three power attacks that target at the DC-level CB in three workload regions, respectively, in Figure 17. When DC is processing workloads in the peak region, it is defenseless to the 30% power increase, as shown in Figure 17(a). Although the impact of the power attack is mitigated by the load balancer at some degree, the margin to the power threshold of the DC-level CB is very small. As a result, the whole data center quickly fails under the power attack. For the power attack in the medium region, in Figure 17(b), we observe the similar results. Especially, when the original workload increases uphill and still in the medium region, the power attack successfully shuts down the entire data center. However, unlike the power attack in the peak region, there exist possibilities that power attacks could fail in this region as the total power consumption (i.e., Original+Attack) is smaller than the capacity of the DC-level CB. Attackers could either increase the size of attack workloads (at a risk of being discovered by the data center administrators) or find the right time to launch such an attack again. For the power attack in the valley region, due to the same reason as the failed attack attempts in the medium region, we have not succeeded in triggering a power outage in the DC with the same malicious workloads. Although not all power attempts lead to power outages of the target, our simulation shows two observations: (1) there is a noticeable possibility of a power attack success at the data center level, especially at peak times, which leads to disastrous consequences. (2) The damage of power attacks could be weakened by pre-defined DC management policies to some extent. Next, we discuss those results in details.

C. Damage assessment and discussion

Our simulation results further demonstrate the potential threat of power attacks. For example, the PDU-level simulation shows that a power attack can trip the CB at the PDU level. The data center-level simulation demonstrates that a power attack could potentially shut down the entire data center. Table III lists some statistics of all the simulated power attacks. The attack against the entire data center succeeds when the workload is in the peak and medium regions, but fails in the valley region. Moreover, the power attack in the medium region takes several attempts and lasts longer than that in the peak region, due to the workload management policies commonly employed in a data center, such as load balancing. It is important to note that such management policies are not originally designed to defend against power attacks, though they could slightly weaken the impacts of a power attack only to a limited degree. On the other hand, from an attacker's perspective, our results suggest that a power attack is more effective when the data center workload is in the peak region. Since the workload traces of many data centers are accessible to the public and usually follow a well-known diurnal pattern, it is not difficult for the attacker to figure out when it is the best time to launch an effective power attack.

Comparing the number of accessible servers and the attack duration (Table III) at both the DC and PDU levels, it is obvious that a power attack requires less resources at a lower (i.e., the PDU) level. To attack a rack or all racks in a PDU, the attacker only needs to access a moderate number of servers. However, the attacker would need to have the knowledge that

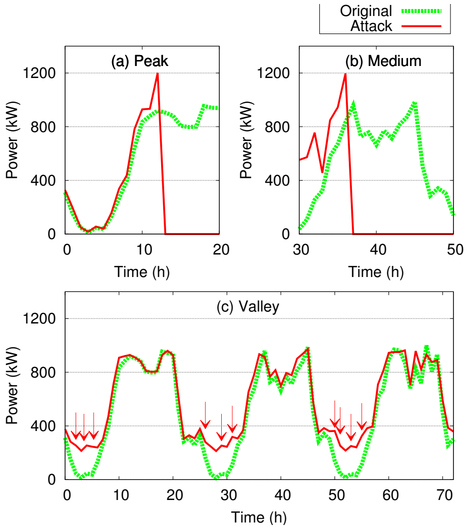

Fig. 17. Power attack launched at the DC level in three regions, (a) peak, (b) medium, and (c) valley. Each red arrow in (c) represents one power attack attempt.

those servers are located within the target rack/PDU. On the contrary, to launch a power attack against the entire data center, the attacker does not need to know such location information. In our simulation at the DC level, we assume that all the servers are accessible to the attacker. However, in the real world, such an assumption may not be true. For example, some servers in a data center may be disconnected from the Internet or have some strong security protection, so the attacker cannot gain the access to them. However, as discussed in previous sections, a 30% power increase is not the greatest amount of increase a power attack can generate. Therefore, even if a power attacker can only access just a portion of servers within a PDU or a data center, it is still possible that the overall power consumption would be increased by about 30%, leading to the disastrous server shutdowns.

VIII. Impact of New Power Management Solutions

In this section, we discuss the impacts of some new power management strategies on power attack. Although these strategies have not yet been widely deployed, it is highly likely that they could be employed in future data centers. Some of the strategies may mitigate the threat of power attack while the others may increase the risk.

A. *Power capping*

Power capping is a solution that can limit the maximum power consumption of a target unit within a user-specified power cap in a data center. For example, server-level power capping [18] leverages feedback control theory to limit the power consumption of a server. Similarly, the power consumption of multiple servers in a rack or PDU can also be capped [27], [28], [35]. For an entire data center, a hierarchical power control solution called SHIP [36] has been proposed to provide power capping hierarchically at three different levels: rack, PDU, and the whole data center. For all those power capping strategies, the power consumption of the target is monitored periodically in real time and dynamically controlled to ensure that it stays below the specified power cap. For instance, Dynamic Voltage and Frequency Scaling (DVFS) is commonly used to lower the CPU frequencies (and voltage) of selected servers when the current power consumption is higher than the cap. In the meantime, within the cap, power capping tries to run the servers at their highest possible frequencies for optimizing system performance. Power capping can also allow a data center operator to host more servers (i.e., power oversubscription), without upgrading the power infrastructure, by having a power cap that is just slightly lower than the rated capacity of the corresponding CB.

TABLE III. Attack Statistics.

Scenario	Infected Machines	Attack Duration (min)
DC_Peak	139200	27
DC_Medium	139200	182
DC_Valley	139200	N/A [3]
PDU_Peak	231	13
PDU_Medium	445	16
PDU_Valley	698	17

Power capping can definitely help to defend against power attack, because power attack is to generate power spikes while power capping is to shave power spikes. However, in practice, there are three major challenges that prevent power capping from becoming an effective defense solution: reactive manner, the selection of control period, and long settling time. First, power capping works in a reactive manner because its periodic power sampling determines that it can only respond to any power budget violation. Any power spikes occur between two consecutive power sampling points (i.e., within a control period) cannot be detected by power capping. Since the control period can be as long as several minutes at the data center level [36], a power attacker can easily launch an attack successfully before power capping can even detect it. Second, in power capping, the selection of control period is a trade-off between system responsiveness and computation/communication overheads. A power capping controller needs to periodically collect the power and performance information from all the controlled servers through the data center network, make centralized and computational-intensive capping decisions, and then send the decisions back to the servers to change their hardware DVFS levels for power capping. A control period has to be long enough for all those steps to finish. Therefore, the control period can be longer than 2 minutes for the SHIP hierarchical controller [36]. However, 2 minutes is already long enough for a CB to trip even when it has only a 25% power overload [13]. As shown in our hardware experiments, a 30% or higher power rise can be easily generated by an attacker through various ways such as parasite attack. Finally, a power capping controller normally cannot immediately drag the power consumption lower than the CB capacity within one control period, even if it detects a power attack. Most controllers need a settling time of at least six or more control periods [36], which means a total time interval of 12 minutes (with a control period of 2 minutes), for power to return after a power spike. Clearly, a power attacker can launch multiple attacks within such a long interval.

In summary, although power capping can mitigate power attack to some extent, it cannot completely prevent power

[3] No attack at valley is successful.

attack due to the three reasons discussed above. More importantly, power capping is mainly designed to allow more aggressive power oversubscription in data centers [12], which can actually lead to a greater risk of power attack.

B. Server consolidation and energy proportionality

Servers are well known to consume too much power even when they are idling. Some recent studies show that current servers still draw about 60% of their peak power at idle [23]. This fact is far away from the ideal case where a server's energy consumption can be *proportional* to its workload, which is called *energy proportionality*. An energy-proportional server would consume little energy at idle and its energy consumption would increase proportionally to its workload intensity. Since the average server utilization in typical data centers is only 20-30% [23], energy-proportional servers would lead to a significant amount of energy savings. While today's servers are still not yet energy-proportional by themselves, a recently proposed power management strategy, called server consolidation, can help make a data center more energy-proportional by dynamically migrating and consolidating the workloads onto a small number of servers and shutting down other servers for energy savings. For example, some recent studies [33], [8], [32], [34] have proposed VM placement solutions that rely on live VM migration for server consolidation.

While these server consolidation solutions can indeed reduce the overall energy consumption of a data center, they may actually also increase the risk of having power attacks. The key idea of server consolidation is to consolidate workload, so that only a smaller number of servers are used with high utilization, which generally comes with high power consumption for each server. In addition, most server consolidation solutions try to put consolidated workload on servers in the same rack or connected to the same PDU for easier management. This strategy could also lead to less cooling costs, because only those Computer Room Air Conditioning (CRAC) units that are near this rack/PDU needs to be running, while other CRACs near those shutdown racks/PDUs can be turned off as well for energy savings [4]. Therefore, server consolidation clearly would increase the power consumption of the rack or PDU that is selected to run the consolidated workload. This thus would push them further to the edge of having a power outage, providing an attacker a better opportunity to launch power attacks.

Whereas future server hardware will certainly become more energy proportional to their workloads, energy proportionality may also provide more opportunities for a power attacker. The key reason is that energy proportionality can allow more aggressive power oversubscription, which in turn increases the likelihood of having power outages. For instance, for a today's server that is not energy proportional, suppose its peak power is 200 W and it consumes 80% of the peak power, i.e., 160 W, when it works at a 20% utilization. Therefore, for a rack equipped with a CB that has a 2000 W of rated power limit, the rack is likely to host 12 such servers with power oversubscription based on the power values at 20% utilization. Now, let us suppose that we have energy-proportional servers that consume only 40 W (20% of peak) of power at a 20% utilization. With aggressive power oversubscription, now the rack can host up to 50 servers. In such a case, an attacker can more easily increase the power consumption of the rack to about 4000 W, simply by increasing the server utilization to only 40%, resulting in significant overload and immediate trip of the CB and thus the shutdown of the rack.

IX. Mitigation Methods

The difficulty of defending against a power attack roots in three aspects. First, although power oversubscription is the major vulnerability exposed to power attacks, it is also one of the key techniques to reduce the operational cost of a data center. As data centers continue to scale up in a fast speed and it is extremely expensive to upgrade data center power infrastructures, power oversubscription has become the trend and will be more aggressive to accommodate more servers in a data center. Second, it is challenging to monitor power consumption of each server accurately in a large scale data center. Since deploying power meters for every server in a data center is too costly [9], [24], current power management solutions tend to approximate the power consumption of each server via utilization-based modeling. However, our work demonstrates that system utilization cannot precisely reflect power consumption. Without accurate and timely measurement of power consumption of servers, it will be difficult to detect and prevent power attacks. Third, with the pervasion and easy access of cloud services, an attacker can consume the computing resources of a data center like a normal user. Although the intention of attackers is very different from that of normal users, it is very difficulty to distinguish attackers from normal users and deny their service requests at the beginning.

In spite of these difficulties, there exist feasible approaches to mitigating the consequence of a power attack. Tracking down the power consumption of individual incoming requests and taking corresponding reaction can be a promising way to defend against a power attack at the server level. Shen et al. [31] built models estimating the power consumption of requests throughout their execution life in a very fine-grained fashion. Such an approach can effectively throttle high-consumption request rate and thus suppress power spikes, which will mitigate power attacks to some extent. It also has minor impact on the service performance.

At the cluster and data center levels, we propose a new load balancing strategy, called *power balancing*, that uses the estimated power consumption as an important factor (along with CPU utilization or throughput) to distribute incoming service requests. Different from traditional load balancing algorithms that are based on system utilization and amount of workload, power balancing captures service requests that consume a large amount of power and evenly distributes them to servers connected to different branch circuits in a data center. As a result, the chance of tripping a branch circuit breaker is minimized. We leave the detailed design, implementation, and evaluation of the power balancing mechanism as our future work.

The deployment of per-server UPS is an alternative way to defend against power attacks. When each server contains a mini-UPS, a short period of power outage will not bring down the server. Besides, per-server UPS is also promising to improve energy efficiency [6]. However, replacing data-center-level UPS with tens of thousands of mini-UPSes is not an easy task. Different UPS deployment mechanisms will bring in great impact on data centers, and hence it will take time to have per-server UPSes be widely deployed in data centers.

X. Related Work

While we are the first to propose the concept of power attack, there are numerous research works studying power management in different computing environments.

A number of studies focus on improving power management in data centers. Some works seek to save energy by adjusting workload distribution algorithms in a data center [39], [20], while some other works aim at reducing power consumption of individual servers [7], [37]. However, even with these solutions, data centers are still under high power provisioning pressure and rely on power oversubscription to handle this pressure.

A research conducted by Fu et al. [13] demonstrates how much power consumption and how long such consumption lasts will trip a CB in a data center. Their study shows that the time to trip a CB has functional relationship with the amount of power that exceeds the rated power of the CB. The more power consumption exceeds the rated power of CB, the less time it takes to trip the CB. Their study provides the theoretical support for more aggressive power oversubscription.

A study on power consumption of high performance computation benchmarks is conducted by Kamil et al. [16]. They analyzed the power consumption patterns of different HPC benchmarks including NAS [2], STREAM [3], and High Performance Linpack (HPL). Their work supports our argument that different computation workloads will lead to considerably different power consumption patterns. Their study also demonstrates that HPL is the benchmark whose power consumption is the closest to that of real world computation-intensive scientific workload.

Although many server consolidation solutions only take resource consumption into account [19], [8], there exist previous works studying the power and energy savings brought by server consolidation [25], [32]. While these studies focus on the power consumption before and after VM migration, we demonstrate that the additional power consumption during migration can be exploited by a malicious attacker. The work of Liu et al. [21] models the power consumption during VM migration. They demonstrated that different VM migration mechanisms and configurations will lead to different migration power consumptions. Their work implies that an attacker can impose additional workload to the to-be-migrated VMs to increase power consumption during migration.

Some of the previous studies on web services demonstrate that there are different ways to increase resource/power consumption with specially crafted web requests. The work of Wu et al. [38] observes that cache misses generate more power consumption at a web server. A research conducted by Crosby et al. [10] introduces the computational attack against web servers. By sending requests with certain data sequence, an attacker can force some data structure operations to suffer the worst case algorithm complexity, therefore costing extra computing resources and thus resulting in more power consumption.

Wu et al. [38] introduced a concept called energy attack. While energy attack also attempts to increase power consumption of a target, it is a different concept from power attack. The goal of energy attack is to enlarge the operational cost of a victim (usually a web service provider) by increasing overall energy consumption, but our power attack can trip CBs in a data center, which can lead to more disastrous damage. Normally an energy attack increases the power consumption of victim moderately for a long period, which has high demand of stealthiness. In contrast, a power attack needs to generate significant power spikes in a relatively short period.

XI. Conclusion

In this paper, we investigate the vulnerability of power oversubscription in data centers and introduce the concept of power attack. We explore different attack vectors in PaaS, IaaS and SaaS environments, respectively. In PaaS, we demonstrate that an attacker can manipulate running workloads to significantly increase power consumption. In IaaS, we propose the concept of parasite attack and further show that VM migration can be exploited for helping to mount a power attack. In SaaS, we craft high power consumption requests that can trigger cache misses and intensive floating point operations to launch a power attack. Our experimental results show that a power attack can easily increase power consumption of a target by over 30% in different environments and our power attack trips the CB of our server room. We further conduct a data center level simulation based on real world traces. The simulation results indicate that a power attack can bring down a PDU or even an entire data center. Moreover, we discuss the impact of various power management schemes upon power security of data centers and propose effective defenses to mitigate power attacks.

As the future work, on one hand, we will further explore more efficient and stealthy power attack vectors in different data center environments; on the other hand, we will systematically study defense techniques, develop prototypes, and conduct experiments to evaluate their effectiveness against power attacks in real scenarios.

XII. Acknowledgement

We would like to thank the anonymous reviewers for their insightful feedback. This work was partially supported by ONR grant N00014-13-1-0088 and NSF grant CNS-1143607.

References

[1] "China national grid," http://en.wikipedia.org/wiki/CNGrid.

[2] "Nas parallel benchmarks," http://www.nas.nasa.gov/Resources/Software/npb.html,2007.

[3] "stream," http://www.cs.virginia.edu/stream/.

[4] F. Ahmad and T. N. Vijaykumar, "Joint optimization of idle and cooling power in data centers while maintaining response time," in *Proceedings of ASPLOS*, 2010.

[5] S. Bapat, "The future of data centers," http://citris-uc.org/files/Bapatallocated/The_Future_Of_Data_Centers-1.pdf.

[6] L. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures on Computer Architecture*, vol. 4, no. 1, pp. 1–108, 2009.

[7] M. Chen, X. Wang, and X. Li, "Coordinating processor and main memory for efficient server power control," in *Proceedings of the ACM 2011 international conference on Supercomputing*, pp. 130–140.

[8] H. W. Choi, H. Kwak, A. Sohn, and K. Chung, "Autonomous learning for efficient resource utilization of dynamic vm migration," in *Proceedings of ACM ICS '08*, pp. 185–194.

[9] G. Contreras and M. Martonosi, "Power prediction for intel xscale processors using performance monitoring unit events," in *Proceedings of ACM ISLPED '05*, pp. 221–226.

[10] S. A. Crosby and D. S. Wallach, "Denial of service via algorithmic complexity attacks," in *Proceedings of the 12th USENIX Security Symposium, 2003*, pp. 29–44.

[11] J. Dean, "Designs, lessons and advice from building large distributed systems," http://goo.gl/nc9K.

[12] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of ACM ISCA' 07*, pp. 13–23.

[13] X. Fu, X. Wang, and C. Lefurgy, "How much power oversubscription is safe and allowed in data centers," in *Proceedings of ACM ICAC '11*, pp. 21–30.

[14] S. Gorman, "Power supply still a vexation for the nsa," *The Baltimore Sun*, 2007.

[15] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2008.

[16] S. Kamil, J. Shalf, and E. Strohmaier, "Performance and energy modeling for live migration of virtual machines," http://crd.lbl.gov/assets/pubs_presos/CDS/ATG/powereffreportxt4.pdf.

[17] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," in *Proceedings of the IEEE ICAC' 07*.

[18] ——, "Power capping: a prelude to power shifting," *Cluster Computing*, vol. 11, no. 2, pp. 183–195, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1007/s10586-007-0045-4

[19] X. Li, Q. He, J. Chen, K. Ye, and T. Yin, "Informed live migration strategies of virtual machines for cluster load balancing," in *Proceedings of the 8th IFIP international conference on Network and parallel computing, 2011*, pp. 111–122.

[20] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *Proceedings of IEEE INFOCOM'11*, pp. 1098–1106.

[21] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of ACM the 20th international symposium on High performance distributed computing 2011*, pp. 171–182.

[22] J. Markoff, "Data centers' power use less than was expected," http://www.nytimes.com/2011/08/01/technology/data-centers-using-less-power-than-forecast-report-says.html.

[23] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: eliminating server idle power," in *ACM Sigplan Notices*, vol. 44, no. 3, 2009, pp. 205–216.

[24] D. Meisner and T. F. Wenisch, "Peak power modeling for data center servers with switched-mode power supplies," in *Proceedings of ACM/IEEE ISLPED '10*, pp. 319–324.

[25] ——, "Dreamweaver: architectural support for deep sleep," *SIGARCH Comput. Archit. News*, vol. 40, no. 1, pp. 313–324, Mar. 2012. [Online]. Available: http://doi.acm.org/10.1145/2189750.2151009

[26] N. Mi, G. Casale, L. Cherkasova, and E. Smirni, "Burstiness in multi-tier applications: Symptoms, causes, and new models," in *Springer Middleware 2008*, pp. 265–286.

[27] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," in *Proceedings of ACM ASPLOS 2008*.

[28] P. Ranganathan, P. Leech, D. Irwin, and J. S. Chase, "Ensemble-level power management for dense blade servers." in *Proceedings of IEEE ISCA'06*.

[29] 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 ACM CCS'09*, pp. 199–212.

[30] R. Shea and J. Liu, "Understanding the impact of denial of service attacks on virtual machines," in *Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service*.

[31] K. Shen, A. Shriraman, S. Dwarkadas, and X. Zhang, "Power and energy containers for multicore servers," in *Proceedings of ACM SIGMETRICS'12*, pp. 403–404.

[32] H. Shirayanagi, "Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks," in *Proceedings of the 2012 IEEE Symposium on Computers and Communications*, pp. 460–467.

[33] A. Verma, P. Ahuja, and A. Neogi, "pmapper: Power and migration cost aware application placement in virtualized systems," *Middleware 2008*, pp. 243–264.

[34] A. Verma, G. Dasgupta, T. Kumar, N. Pradipta, and D. R. Kothari, "Server workload analysis for power minimization using consolidation," in *Proceedings of the 2009 conference on USENIX Annual technical conference*.

[35] X. Wang and M. Chen, "Cluster-level feedback power control for performance optimization," in *Proceedings of the 14th IEEE International Symposium on High-Performance Computer Architecture*, 2008.

[36] X. Wang, M. Chen, C. Lefurgy, and T. W. Keller, "Ship: Scalable hierarchical power control for large-scale data centers," in *Proceedings of IEEE PACT'09*, pp. 91–100.

[37] Y. Wang, X. Wang, M. Chen, and X. Zhu, "Power-efficient response time guarantees for virtualized enterprise servers," in *Proceedings of IEEE RTSS'08*, pp. 303–312.

[38] Z. Wu, M. Xie, and H. Wang, "Energy attack on server systems," in *Proceedings of USENIX WOOT'11*.

[39] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proceedings of IEEE INFOCOM'12*, pp. 1431–1439.

Appendix

In Amazon EC2, the physical location of a VM can be associated with the its type, "zone", and IP address. An attacker can customize the VMs by specifying in which zone the VM will be instantiated and what is the instance type so that the VM will be located in a certain physical area. After the VM is booted, the attacker can further infer the "location" of the booted VMs as well as their host physical machines via IP address and packet round time information [29]. Since the VM will change its IP address, i.e., its physical location every time it is rebooted, the attacker can place a VM to a target machine or rack by keeping rebooting the VM until it reaches the desired location. In this way, the attacker can deploy many VMs on those physical machines that are within the same rack.