When Power Oversubscription Meets Traffic Flood Attack: Re-Thinking Data Center Peak Load Management

Xiaofeng Hou, Mingyu Liang, Chao Li, Wenli Zheng, Quan Chen, Minyi Guo
Department of Computer Science and Engineering, Shanghai Jiao Tong University
{xfhelen,liangmingyu}@sjtu.edu.cn,{lichao,zheng-wl,chen-quan,guo-my}@cs.sjtu.edu.cn

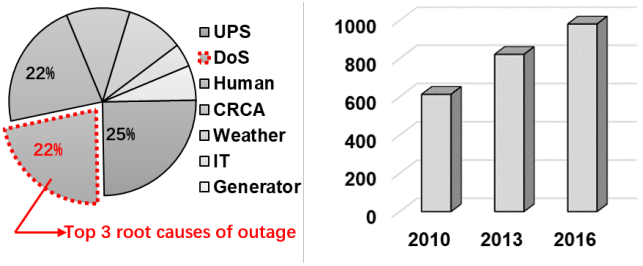
ABSTRACT

The state-of-the-art techniques on data center peak power management are too optimistic; they overestimate their benefits in a potentially insecure operating environment. Especially in data centers that oversubscribe power infrastructure, it is likely that unexpected traffics can violate power budget before an effective network DoS attack is observed. In this work, we take the first to investigate the joint effect of power throttling and traffic flooding. We characterize a special operating region in which DoS attacks can provoke undesirable power peaks without exhibiting network traffic anomalies. In this region, an attacker can trigger power emergency by sending normal traffics throughout the Internet. We term this new type of threat as DOPE (Denial of Power and Energy). We show that existing technologies are insufficient for eliminating DOPE without negative performance effects on legitimate users. To enhance data center resiliency, we propose a request-aware power management framework called Anti-DOPE. The key feature of Anti-DOPE is bridging the gap between network traffic controlling and server power management. Specifically, it pre-processes of incoming requests to isolate malicious power attacks on the network load balancer side and then post-processes of compute node performance to minimize the collateral damage it may cause. Anti-DOPE is orthogonal to prior power management schemes and requires minute system modification. Using Alibaba container trace we show that Anti-DOPE allows 44% shorter average response time. It also improves the 90^{th} percentile tail latency by 68.1% compared to the other power controlling methods.

ACM Reference Format:
Xiaofeng Hou, Mingyu Liang, Chao Li, Wenli Zheng, Quan Chen, Minyi Guo. 2019. When Power Oversubscription Meets Traffic Flood Attack: Re-Thinking Data Center Peak Load Management. In *48th International Conference on Parallel Processing (ICPP 2019), August 5–8, 2019, Kyoto, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3337821.3337856

1 INTRODUCTION

With an explosive growth of various cloud applications, data centers are continuously deploying more servers. Although the load power demand goes up quickly, it is very difficult and expensive to upgrade the power infrastructure in an existing data center facility. As a result, it is more economical for today's data centers to scale out computing resources by aggressively oversubscribing their power system [12, 16, 30, 31, 35]. This proposal is generally based on the assumption that traffic surge and the associated power demand surge does not occur very often (servers rarely reach peak load simultaneously) [31, 35]. In this case, power over-subscription shows great promise in maintaining data center performance scaling trend with attractive cost efficiency.



(a) Root causes of unplanned power outages [3] **(b) Total cost of power outages caused by DoS [3]**

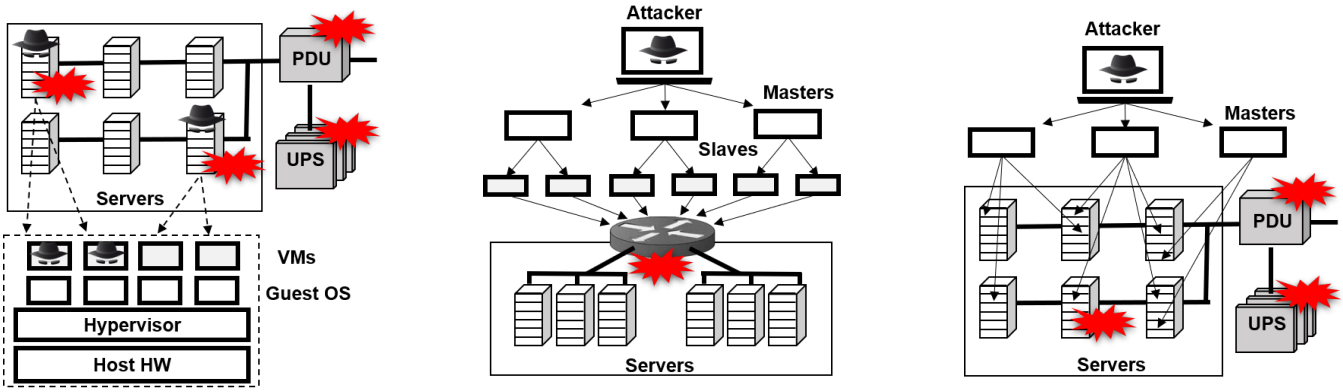
Figure 1: Data centers face a growing amount of attacks that may compromise power provisioning effectiveness.

Unfortunately, real-world data center operating environment can be complicated. It is crucial to study data center peak power management strategies in a highly dynamic (potentially insecure) network environment. Network flood can not only cause denial-of-service (DoS), but also cause unexpected power emergencies, depending on which happens earlier. If a data center scales out its computing and network resources without considering the worst-case traffic scenarios, the benefits of power over-subscription can be significantly compromised. In fact, traffic flood poses an immense threat to cloud service in recent years. According to Verisign and Pnoemon Institute, the frequency of distributed denial-of-service (DDoS) attacks targeted on data centers grows at a rate of 75% in 2016 [47, 50]. It is a remarkable fact that DoS attack has become the top-3 root causes of data center unplanned outages as shown in Figure 1-(a). It has been shown that the impact of power emergency due to denial-of-service (DoS) attack has escalated over the years from 2010 to 2016 (Figure 1-(b)).

In this work we identify a potential availability threat triggered by abnormal network traffics in a power-constrained data center. In general, we consider a sophisticated adversary from the external Internet. The attacker can manipulate power consumption of the targeted application with well-designed traffics. It exploits the vulnerabilities within power management frameworks to jeopardize the cost effectiveness of current data centers which emphasize high utilization. We refer to this type of malicious acts as denial of power

ICPP 2019, August 5–8, 2019, Kyoto, Japan

ACM ISBN 978-1-4503-6295-5/19/08…$15.00
https://doi.org/10.1145/3337821.3337856



(a) VM-based power attack from inside (b) Denial-of-Service attack from outside (c) Power attack from outside

Figure 2: Like DDoS, server power attack could become a grave threat to aggressively provisioned data centers

and energy (DOPE) attack. DOPE is a new class of low-rate but high-power requests targeting unconventional layer of targeted resources (e.g., energy, power, and cooling).

DOPE attack poses a tremendous challenge for managing oversubscribed data centers. On the one side, it is difficult for data centers to track per-request power consumption of web applications due to the pervasiveness and anonymity of the Internet. On the other side, current data centers excessively rely on network load balancer (NLB) and auto-scaling resource allocation to provide built-in defenses against DDoS attacks [45] for gaining the optimal productivity and the maximum uptime. As a result, hostile requests can generate the maximum possible load on their targeted servers without prior detection. In other words, they often harshly utilize servers' hardware components to abuse power resources.

Today, new theoretical attacks can quickly move to practical. It impels us to think about *"How data center should be managed to preserve the substantial benefits of aggressive power provisioning without making it vulnerable to traffic flood?"*

We propose Anti-DOPE, a request-aware power management framework for enhancing the ability of today's power-constrained data centers to defend against DOPE. Our design highlights a two-step power management approach, power-driven forwarding (PDF) on the NLB side and request-driven power management (RPM). In the first step, NLB segregates suspicious power allocation requests and dispatches them on isolated servers based an offline analysis. Afterwards, the second step monitors the instantaneous pressure on the power resources and adjusts the execution of suspicious requests to eliminate power peaks. Overall, this paper makes three main contributions:

- We identify Denial of Power and Energy (DOPE) attack, a new class of threat to aggressively provisioned data centers. By establishing a scaled-down system we discuss how DOPE can leverage low-rate but high-power requests to originate undesirable power peaks.
- We propose a new threat mitigation framework called Anti-DOPE. This technique gracefully supports existing data center power management schemes through a network-aware, two-step defense strategy.
- We thoroughly evaluate the effectiveness of Anti-DOPE through Alibaba container trace simulation. We compare Anti-DOPE with conventional load power throttling mechanisms as well as network traffic tuning techniques.

The rest of this paper is organized as follows. Section 2 introduces background. Section 3 analyzes power oversubscription under network flood. Section 4 depicts the threat model of DOPE. Section 5 proposes the Anti-DOPE framework. Section 6 describes experimental methodology and presents evaluation results. Section 7 discusses related work and Section 8 concludes this paper.

2 BACKGROUND

2.1 Overload Risk in Data Centers

Despite many benefits, oversubscribed data centers have exposed themselves to various power-oriented attacks. As shown in Figure 2-(a), hostile users can compromise the mainstream power management infrastructures and techniques by manipulating VMs inside server racks. According to prior research, attackers can launch power attack [12, 20, 36, 43] through running intensive loads on the controlled VMs or invoking frequent VM mitigation activities. As power budget shrinks, Power Grab [36] can abuse power resources and disrupt operation of its competitors by operating power-hungry VMs. In battery-backed data centers, running task-intensive VMs can drain the precious energy storage and overload server racks without prior detection [12].

With the proliferation of popular on-line, data-intensive services (such as search, social networking, e-Commerce and webmail) hosted in warehouse-scale computers, the past years have witnessed rapidly increasing DoS attack [2]. DoS attackers cripple the targeted online service by sending massive requests through the Internet as shown in Figure 2-(b). These attacks are typically classified into network-layer attacks and application-layer attacks. Network-layer DoS attacks causes the disruption in the legitimate user connectivity (exhaustion of communication protocols and reducing router/switch processing capacity). Application-layer cyber-attacks disrupt the legitimate user services (depletion of the server resources like CPU, memory, disk bandwidth).

Type	Name	Description
Victim	Colla-Filt	Collaborative Filtering is a computing-intensive algorithm used by recommender systems.
	K-means	K-means,a memory-intensive algorithms.
	Word-Count	Word-Count frequently reads text files from the disk in e-commerce domain.
	Text-cont	Text-Context sends requests asking for text.
DoS	http-load	A tool for simulating attacker and generating HTTP traffics.
	AB	ApacheBench can set concurrent requests number.
Normal	AliOS	AliOS imitates accessing Alibaba online service.

Table 1: Evaluated workload for proof-of-concept.

Figure 2-(c) illustrates a trending attack scenario in which users jeopardize power-constrained cloud power infrastructures by sending malicious requests from the external Internet rather than regulating internal VMs. The attack happens when an adversary manipulates a group of recruited agents to send power-consuming requests. Every agent behaves like a normal user at the networking level, but in combination they can constitute an unpleasant power peaks in the victim organization. This attack is a more appealing method to interrupt data center operation because very limited work has been done at a level concerning how the network request pattern may frustrate power management strategies.

3 VULNERABILITY CHARACTERIZATION

We build a scaled-down testing environment for discussing the connections and conflicts between power oversubscription and network vulnerabilities in data centers. It consists of a mini server rack with four leaf node servers. The nameplate power of the server is 100 watts. With the Advanced Configuration and Power Interface (ACPI), we can adjust CPU operating frequencies from 1.2GHz to 2.4 GHz at an interval of 0.1GHz. All the servers are connected to a FAST FSG116 network switcher to ensure our experiment is dissociated from the outer network. We establish an e-Commerce (EC) service based on Spring Boot [6].We implement the main functionalities of EC workloads in accordance with BigDataBench [27]. As shown in Table 1, Colla-Filt (collaborative filtering) is a computing-intensive algorithm used by recommender systems. As two of the basic operations in EC domain, Word-Count reads text files from disk frequently and Text-Context sends requests asking for text. K-means consumes plenty of memory resources to make classification. Correspondingly, DoS attackers can exhaust the victim

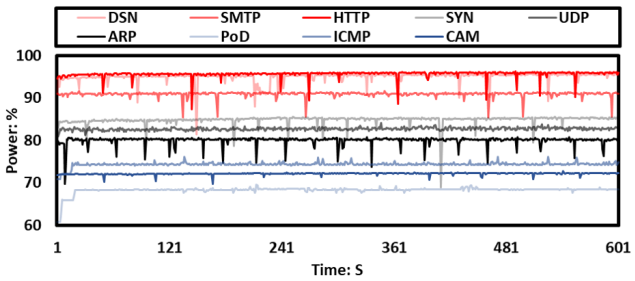
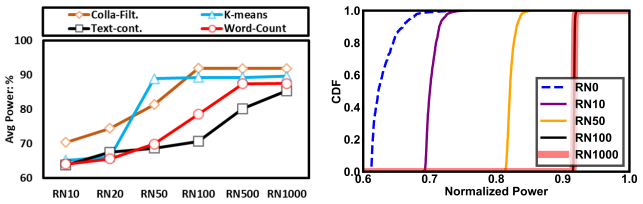
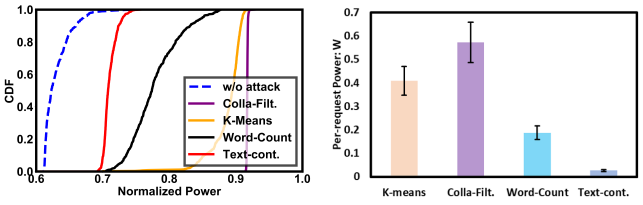


Figure 3: Power profile of typical cyber-attacks



(a) Different traffic volume **(b) Different traffic rate**

Figure 4: The higher traffic rate tends to cause higher power



(a) Power under various traffic **(b) Per-request power intensity**

Figure 5: Power caused by different types of traffic shows that volume-based DoS request has low power intensity

servers' resources with massive queries requesting these services. Besides, they can saturate network connectivity by sending numerous requests. We leverage http-load tool [51] and ApacheBench (AB) [44] to imitate the behavior of the attackers. Based on the cluster data [1] released by Alibaba cooperation, we model the pattern of accessing online EC service to imitate normal users' activities.

3.1 Power Profile of Cyber-attacks

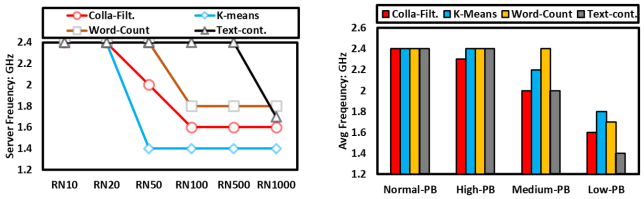
We begin by examining the power usage of various traffic flood. We launch typical network flood [22] targeting different layers with widely used tools [4, 9, 14, 25, 39, 48, 49]. We manipulate the attack force to maximally consume the victim EC service. Meanwhile, we measure the power variation under various cyber-attack scenarios within a 600 seconds observation window.

In Figure 3, the x-axis is time and y-axis represents the power variation. Colored lines demonstrate results of different attacks. Red, black and blue lines respectively represent high, medium and low power usage scenarios.

It is remarkable that application-layer cyber-attacks like HTTP and DNS Flood Attack are more likely to make high power peaks compared to malicious acts in the other layers. While undergoing application-layer DoS attack, EC becomes task-intensive workload, thus it consumes considerable power resources. This observation inspires us to think about this question: *"In today's power-constrained data centers, can application-layer attacks violate the power limits while the network DoS defense system are promoting to gain productivity?"*

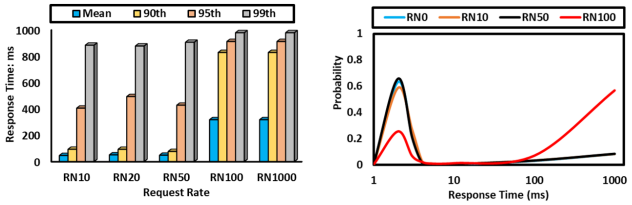
3.2 Power Anomalies Epoch

To answer the above question, we characterize the variation of power usage with different HTTP traffics. We consider two key factors: the traffic rate and the requested service type. We choose



(a) The effect of traffic rate (b) The effect of traffic type

Figure 6: The effect of HTTP DoS attack on power capping



(a) Service time (b) Tail latency distribution

Figure 7: Service quality gets worse with higher traffic rate

HTTP DoS attack because it is one of the most common and representative mechanisms in application-layer attacks. Our design can be easily extended to the other types of the application-layer DoS attacks by simply changing the monitored statistical features.

Figure 4 presents the impact of peak power manipulation for 10 minutes. Figure 4-(a) shows the increase in power usage with larger traffic volume. It is clear that sending more requests per second produces higher power. Particularly for attackers accessing Colla-Filt, K-means and Word-Count service, the attacker elevates the consumed power at a low traffic rate. Figure 4-(b) shows the CDF of power consumption at multiple levels of traffic rates (requests number per second). The x-axis is normalized with respect to the nameplate power of our leaf server node. At the higher network volume, there is lower variance in power usage. The generated power peaks tend to smoothen the fluctuations as the request number (RN) per second gets larger. However, it requires the hostile web users to transmit more queries for generating high peaks. Thus, the likelihood of generating consistent and high-power peaks increases as long as the attacker delivers adequate queries.

Figure 5 compares the CDF of power consumption for querying different web services individually when the traffic rate is 100. There are several interesting observations from these results. First, power usage generated by abnormal users is higher and more stable than the normal. As can be seen in Figure 5-(a), it is highly likely for an abnormal web customer to operate server's power closer to the nameplate, much more than the normal. Colla-Filt's curve is subvertical since it has expended the potential maximum power resource across all servers. Meanwhile, Colla-Filt's CDF is rightest since Colla-Filt represents functions which consistently compute the preference of customers in an E-Commerce application. Figures 5-(b) demonstrates the average power of 4 request types. The query requesting for K-means consumes most power per request while the volume-based traffic seems to consume much less power.

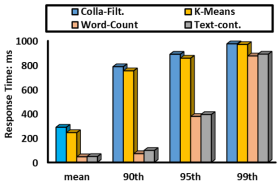


Figure 8: Service time under various traffic types

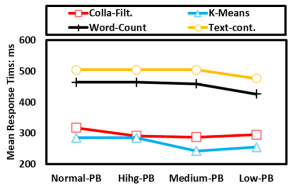


Figure 9: Mean response time under various power budget

All above has demonstrated that it is feasible for a network attack to cause high power consumption in data centers. Regardless of the accessing service types, attackers can trigger power surges with plentiful requests. Particularly, it is easy for some tasks like Colla-Filt, K-means and Word-Count to generate power surges with light traffic rate.

3.3 Impacts on Power-Limited Data Centers

A primary goal of this work is to characterize and analyze power-constrained data centers under network flood. Therefore, we firstly simulate data centers that have different power budget. We configure the normal power budget (Normel-PB) as our baseline (with 100% supplied power). We configure high power budget (High-PB) with 90%, medium power budget (Medium-PB) with 85%, and low power budget with 80% (Low-PB) percent of Normal-PB, respectively. Afterwards, we discuss the relationship between the supplied power and network attacks' properties. We use dynamic voltage and frequency scaling (DVFS) to adjust the server's power when the peaks exceed the budget.

Figure 6 shows the impact of attack rate and request type on server voltage/frequency scaling. In Figure 6-(a), with traffic increasing, it causes larger V/F reduction under scenario of Medium-PB. Colla-Filt based traffic firstly incurs V/F reduction at a low rate because of its highest power intensity. When the traffic rate exceeds certain threshold, the V/F value stays the same and is enough for limiting power within safe line at the cost of performance degradation. When the attack rate is 1000 request per second, Figure 6-(b) indicates that K-means induces more V/F reduction because its power is less sensitive to frequency changes. Summarily, Colla-Filt compromises V/F mechanism at a lower rate but K-means can incur lower system execution speed. It is easy for a cyber attacker to touch the bottom line of power limit and bring frequency reduction. Correspondingly, we care more about the cascading side effects of malignant traffics on the behavior of normal customers. We first investigate the extra service loss under the aggressively power insufficient situation.

Figure 7 indicates that DoS-driven power surges show 7.4X longer mean response time and increase 8.9X 90th percentile of tail latency after the increasing request number exceeds about 100 in an aggressively power-insufficient condition. Namely, with inadequate power budget, larger request number causes power peak and triggers DVFS which in turn aggravates the service quality.

Figure 8 compares the service time of our four observed traffic types. Colla-Filt and K-means arouse more serious degradation of service quality. Aggressive power oversubscription design will

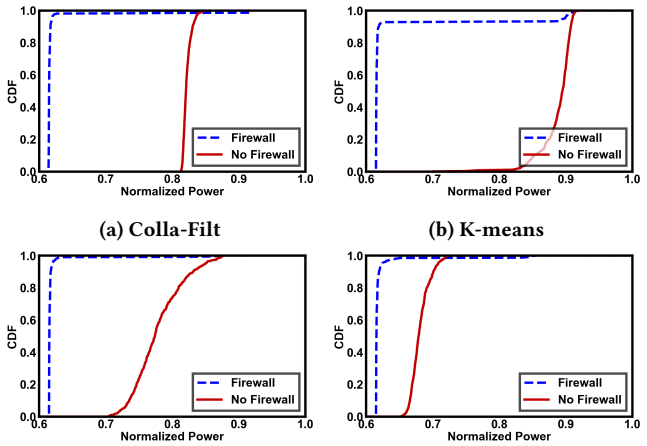


Figure 10: CDF of power usage with and without firewalls for different traffic types

cause severe service availability (Figure 9). The fact of severe decline in service availability is that the online attackers cause power reduction, which in turn compromises the service states. However, it is feasible for a hostile user to manipulate the service power consumption by sending low-rate query flows. The previous figures illustrate that it is possible for traffic types like Colla-Filt, K-mean and Word-Count to controlling the power without being detected by firewall.

3.4 Analysis on Flood Prevention

It is attractive that cyber-attackers can generate unexpected power surges in power-limited data centers. It in turn severely compromises web application's service quality. Nevertheless, the dedicated traffics must elude the perimeter network protection at first. According to a series of annual surveys [2] conducted by Arbor Network company, over 80% data center operators being investigated deploy firewall as security mechanisms defending against network flooding attacks. Therefore, taking firewall representing mainstream network defense systems, we investigate the feasibility of producing high-power peaks when the system is under protection of firewalls.

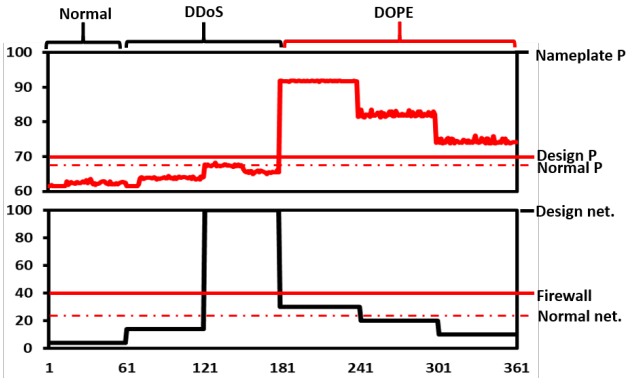


Figure 11: DOPE attack region.

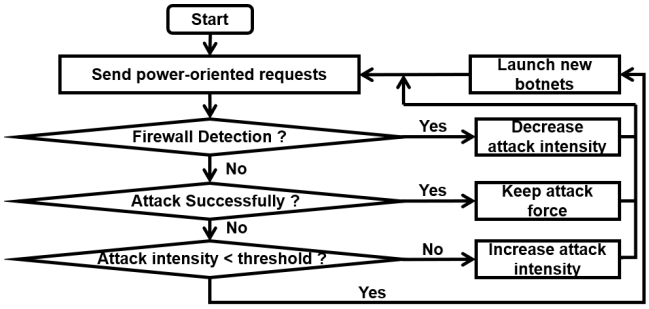


Figure 12: Attack algorithm.

We run DDoS deflate [39] to prevent and mitigate HTTP DoS attack. Deflate is an open-source firewall tool based on the integrated netstat [23] monitor in Linux. We use the default configuration at 150 requests per seconds as the pre-defined firewall rules.

Figure 10 depicts the CDF of power usage for 4 traffic types with and without firewalls. In this situation, the attacker launches 1000 request per second and consumes high power shown by solid lines without firewalls. Nevertheless, it will be caught by firewall shown as dotted lines. It is notable that there are partial high power spikes even with firewalls due to the initiating delay of defense method.

The start time for the firewall to detect the abnormal traffics is different among various traffic types. To some extent, the start time lag is vulnerable as well. It is obvious that the high-volume traffics are easy to be caught by firewall.

4 THREAT MODEL OF DOPE

The above results point out a vulnerable domain where an Internet adversary can induce abnormal power consumption on the target service nodes with selective network traffic types and enough request number. It is feasible according to several characteristics quantified in the previous section. First, the adversary can generate high and stable power peaks with an adequate quantity of task-intensive service requests. Second, normal power management activities such as power capping and battery control are not enough to eliminate the vulnerabilities. They further cause severe degradation in service quality. Third, the adversarial users exhibit no traffic anomalies or unique traffic patterns during overwhelming the target service with high-power queries. They can elude the network defense techniques although network traffic is often monitored for security purposes. In this work we term the above act as DOPE (Denial of Power and Energy). Figure 11 defines the operating region of DOPE. Its request number can be close to the normal while far smaller than the DoS-detecting network capacity. DOPE will violate power infrastructures and managing approaches in an oversubscribed data center.

The DOPE attacker comes from the external Internet. It can be an individual hacker, a botnet master, or an organization for committing cybercrime/cyberwarfare. DOPE only has remote access to the target service application through some public portals, and no other special interfaces. It cannot directly control the internal VMs or server nodes. To implement DOPE, the adversary can first select partial high-power request types through numerous offline

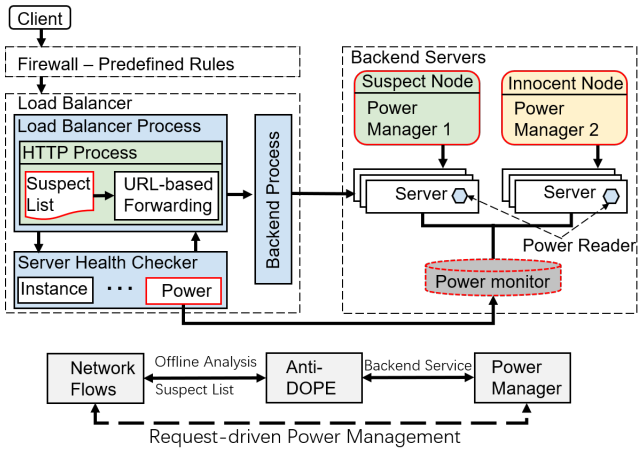


Figure 13: Overview of the Anti-DOPE framework.

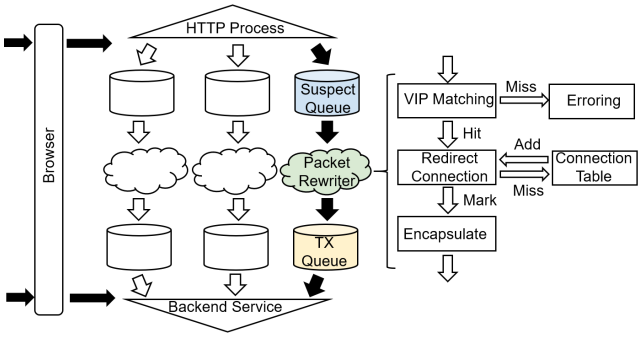


Figure 14: Differentiated power management.

analysis and characterization of power for different service components in current mainstream online workloads. After that, it can launch DOPE attacks with selective traffic types. We propose a simple DOPE attack algorithm as shown in Figure 12. It gradually increases the request number to the bottom limit of the deployed defense systems. During the process, it repeatedly adjusts its request number until an effective DOPE without being detected by network protection approaches.

5 ANTI-DOPE FRAMEWORK

The DOPE attack can bring many issues such as increased energy and carbon footprint, undesirable performance capping, and unexpected downtime. We take several key initial steps to strength the data center's security posture with several new capabilities. We propose a request-aware security optimization framework for existing data center power management schemes. In this section, we first provide a high-level overview of the proposed framework. Afterwards, we discuss the detail on modification of both network NLB and data center power manager sides.

5.1 Anti-DOPE Overview

As shown in Figure 13, Anti-DOPE bridges the gap between network defense systems and power management in today's data centers. Anti-DOPE has two key functioning modules: power-driven forwarding (PDF) and request-aware power managing (RPM). PDF allows splitting the incoming traffic into the suspect flows and innocent flows. Meanwhile, RPM implements differentiated power control strategies based on the power usage information of the data center. The function of PDF and RPM is discussed below:

PDF: PDF is mainly responsible for splitting the risk requests and forwarding requests onto different servers. For isolating high-power requests with low-power ones, it maintains a suspect list inside the HTTP process module for classifying the incoming requests in accordance with their access url. After that, the url-based forwarding module will distribute the classified requests to different backend server nodes through invoking backend process.

RPM: RPM instructs power management of the isolated requests. It keeps a feedback link between server power monitor and server health checker. The server health checker obtains state of power supply infrastructures and the overall power usage of all the servers. When power capping and battery control is required, DPM mechanism starts to de-allocate power budget from nodes under suspicion to make the simultaneous power usage below the budget without big performance degradation.

5.2 Differentiated Power Management

The cornerstone of our framework is that high-power requests are extremely likely generated by DOPE attackers. It is because that the main differences between malicious and normal requests lie in the power demand per request. As mentioned before, most high-power requests are more likely belonging to hostile users. Based on this, differentiated power management (DPM) is the center piece of Anti-DOPE. DPM demands both components in PDF and RPM for working together. The keys of DPM are how to forward risk requests and normal requests onto different servers and how to implement power controlling policies.

Suspect list determines the way of forwarding. Generally, for an online data-intensive (OLDI) application, requests asking for the same service (i.e., url) often require similar computing resources and consume almost equivalent power. Thus, Anti-DOPE establishes suspect list by offline profiling the relationship between power and service types for heterogeneous requests accessing to different service types. Therefore, As shown in Figure 14, for any incoming requests, HTTP process divides them into different queues according to suspect list. Afterwards, the package rewriter will check the validation of request and redirect it to specified servers, called suspect server nodes through modifying and encapsulating the new backend address.

As for power allocating, the goal of DPM is to reclaim the power capacity for legitimate requests. In term of suspected users, DMP makes request execution decisions and regulates the length of throttled requests to meet power budget. As shown in Algorithm 1, DPM determines the throttling patterns of servers running suspect requests at each slot to satisfy currently available power. Once it determines the throttling configuration of each servers, it enforces the allocation with extensive bottom interface. Taking Linux as an example, it can leverage the perf_event interface supported by perf tool to modify the RAPL interfaces provided by Intel processors [7].

Algorithm 1 Differentiated Power Management Algorithm

Require: Power budget: P_{Supply}
1: // At the beginning of each time-slot
2: Obtain current power consumption: P_{demand}, current battery capacity: BA_{init} and predicted power mismatching: $\Delta P = P_{supply} - P_{demand}$
3: Initiate power capping reduction: $P_{reduction} = cap_{max}$, throttling list $TL(m, m) = \langle throttling_1^m \ldots, throttling_n^m \rangle$
4: // battery is insufficient for shaving current peak valley
5: **if** $\Delta P > BA_{init}$ **then**
6: Allocate partial innocent servers to BA based on BA_{init}
7: Update power mismatching: $\Delta P = \Delta P - BA_{init}$
8: // Determine the throttling strategies of the suspected nodes
9: **for** index=1 to n **do**
10: // search the suspect node
11: **for** pm=1 to m **do**
12: // coordinate the different power throttling strategies
13: **if** $P_{reduction} > P_{reduction} - P_{cap}$ **then**
14: // search the optimal throttling
15: Update throttling list $TL(p, q)$, and $p, q \in \langle 1, \ldots, m \rangle$
16: **end if**
17: **end for**
18: Allocate different numbers of servers to BA based on Throttle suspect node at configuration $TL(p, q)$
19: Collect running results at the end of the time-slot
20: **end for**
21: **else**
22: Go to the next time slot
23: **end if**

5.3 Requests Control Model

After dispatching the incoming requests on disparate servers, the manager keep listening to the power monitor. Once detecting a message of power budget violation, batteries discharge at first to seamlessly invoke performance throttling mechanisms. Determining the quantity of throttled request obeys the following principle.

We assume the initial power budget is B_0 for saving cost and energy. The number of incoming requests is Q and they are divided by the classifier into n levels of power usage according to the provided service types. As shown in the former section, each power usage level means a potential saving power range. We assume q_i quests are classified into the i_{th} level. Thus, the whole incoming request flow Q is divided into n parts defined as a vector $<q_0, q_1, ..., q_n>$. Each element in this vector represents the number of requests to be throttled at a certain V/F level. To limit the overall power consumption into the power budget B_0, Anti-DOPE scheduler throttles the execution of partial requests in the vector $<q_0, q_1, ..., q_n>$.

$$\sum_{i=0}^{n} q_i * P_i(f) \leq B_0 \quad (1)$$

In the above formula, $P_i(f)$ is the power consumption of request q_i. $P_i(f)$ always changes with the execution frequency f to meet low-power requirement [46].

Algorithm 1 illustrates the flow of determining the throttling level for each request. Specifically, when a power peak occurs, it first estimates whether backup batteries are adequate for capping this power peak. If the batteries are inadequate, they discharge to support innocent servers and serves as the transformation media for initiating differentiated power throttling(Line 5-7). After that, Anti-DOPE looks up all the suspect requests and find out the optimal throttling strategy (Line 8-16). After determining the throttling level of each requests, the power controller will distribute the total power budget in accordance with the throttling list(Line 17-19).

5.4 Discussion

Rather than focusing on precise detection of malicious requests, the design of Anti-DOPE follow a KISS (keep it simple, stupid) principle. It can effectively thwart the resource starvation attempts of the attacker without distinguishing the malicious and normal ones in comparison with existing proposals.

Anti-DOPE distributes the legitimate request with high demand as suspect one as well. However, it does not compromise the service quality severely due to the high proportion of the malicious. Anti-DOPE outperforms the existing power or network protection approaches by establishing their communication. Conventional power management methods can cover power peaks produced by DOPE, however, it makes the latency issue of normal users worse. Meanwhile, the mainstream network protection mechanisms are incapable of handling with DOPE due to their primary dependency on rate-limiting techniques. It is uncertain for them to decide on the quantity of discarded packets to eliminate power peaks.

6 EVALUATION AND RESULTS

6.1 Evaluation Configurations

We use Alibaba's container trace [1] to imitate normal users' activities accessing EC application. It contains 12 hours long running log of 1.3k machines. We inject the malicious load by recording the running Colla-Filt, K-means and Word-Count service attack in hand. Table 2 summarizes our evaluated four power management schemes for defending against DOPE in the under-provisioned data center environment. We consider two power management baselines: Capping and Shaving. Capping represents the traditional data center designs that only use performance scaling mechanisms to cap power peaks [13]. Shaving uses the state-of-the-art power shaving schemes similar to prior work [18, 31] for better design trade-offs. We also simulate a typical network traffic controlling method, token bucket to manage traffic flood through package offloading. In all, we compare Anti-DOPE with conventional power capping mechanisms and simple network rate limiting strategy to prove its effectiveness.

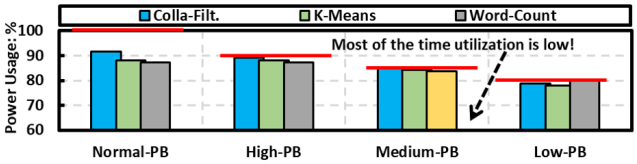
6.2 Effectiveness on Removing DOPE

We first investigate if Anti-DOPE could gracefully manage the power-constrained server cluster without significantly affect normal users. In this experiment, we focus on four power budget scenarios, i.e., Normal-PB, High-PB, Medium-PB and Low-PB (Detailed in Section 3.3). We use a 10-minutes long observation window in the following analysis.

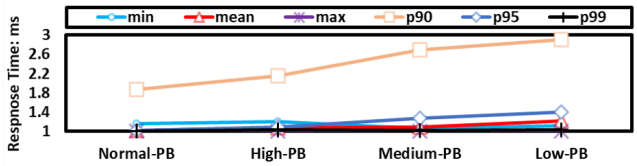
We compare victim server nodes suffering from various DOPE attacks. In this experiment, the original server running EC applications shows relatively low power utilization (indicated by the red line as shown in Figure 15-(a).). Once DOPE starts, we notice that there is a sharp increase in total power consumption.It is obvious that our Anti-DOPE can adjust the power usage to limit the overall

Scheme	Feature	Description
Capping	Performance scaling only	Only uses dynamic voltage and frequency scaling (DVFS) to cap power.
Shaving	UPS based peak shaving	Triggers DVFS only if the UPS used for peak shav -ing runs out of energy.
Token	Power-based token bucket	A modified network traffic controlling algorithm to ensure power limits.
Anti-DoPE	Our proposal	A resilient power capacity management framewo-rk considering per-request.

Table 2: Evaluated Power Management Schemes



(a) Anti-DOPE can constrain power usage into the supplied



(b) Anti-DOPE has slight effect on normal users' service time

Figure 15: Anti-DOPE can effectively allocate power budget with slight performance degradation of the normal users.

power demand within the supply. Remarkably, Colla-Filt has larger reduction with the same V/F regulation.

As shown in Figure 15-(b), Anti-DOPE can guarantee the average response time and tail latency of legitimate users. We choose the situation of good user and Normal-PB as the baseline. Our results show that the mean response time, the 90^{th} and 95^{th} percentile of tail latency is slightly worse than the baseline. This is because the normal traffic is possibly divided into separate servers, and the collection of high-power requests aggravates their execution speed. When the EC application is under attack, our Anti-DOPE still guarantees the service quality as shown in the results. It is notable that the variation of minimum and maximum response time, as well as 99-th percentile tail latency keeps the same since the longest or shortest service time is also affected by other factors such as internet bandwidth.

6.3 Response Time Profile

In this section we further evaluate the impacts of different power management schemes on tail latency and mean response time. They are both key metrics defined in the service-level-agreement (SLA).

Figure 16 illustrates the results of mean response time. We choose Normal-PB as the baseline. For the baseline, all the service response time under different power schemes is below 40 milliseconds and there is no difference among the observed power schemes. In the figure, High-PB, Medium-PB and Low-PB all have lower power budget compared to Normal-PB. We notice that in these low-power-budget

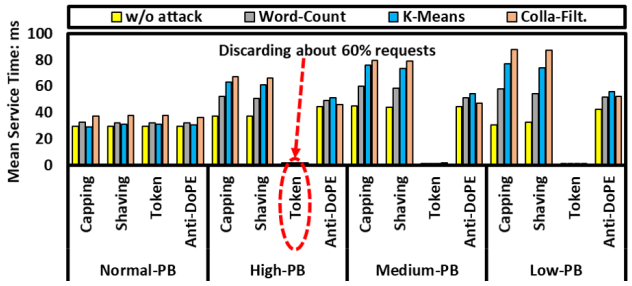


Figure 16: Mean response time while using different power schemes to handle with DOPE.

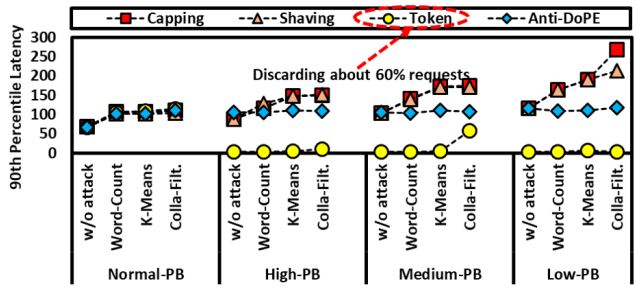


Figure 17: Tail latency while using different power schemes to handle with DOPE.

scenarios (i.e., lower than Normal-PB), all the power management methods increase the mean response time. Nevertheless, our proposal guarantees the minimum mean service time because it mainly deploys V/F adjustment on the attack requests, which is more likely routed onto suspect nodes. It is optimistic that all the methods can ensure a mean service time below 100 milliseconds even in the extremely insufficient power cases. It is interesting that Token has far shorter service time than the others. This is because it abandons more than 60% of the packages to satisfy the power limit.

However, it is more serious when looking at the 90th percentile tail latency. As shown in Figure 17, the tail latency can be up to 236 milliseconds. Compared with normal results with DOPE attack under the nameplate situation, it indicates that DOPE slightly prolongs the tail latency. There is no big difference among different schemes because the power is adequate. Nevertheless, the situation changes under under-provision scenarios. Compared with others, our Anti-DOPE mechanism sustains the service quality of normal users regardless of the supplied power. This is a result of isolating most of the malicious ones. The 90-th percentile latency achieves to 105 milliseconds in a medium attack scenario with capping tools. Besides, it seems that batteries do not function well with such a long-duration power peak when compared Capping with Shaving. Token yields good tail latency since it abandons numerous requests.

6.4 Battery Management Behaviors

Energy storages are more and more critical components in modern data center. Conventionally, batteries are only used as emergency backups which are rarely used. Recently, batteries have been used

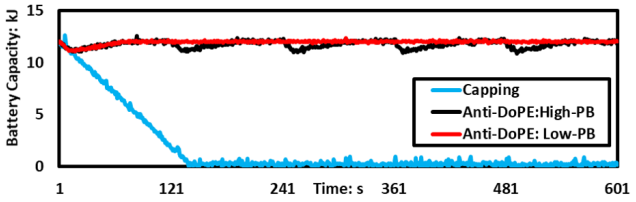


Figure 18: Batteries' behaviors for different power management schemes when facing cyber-attacks

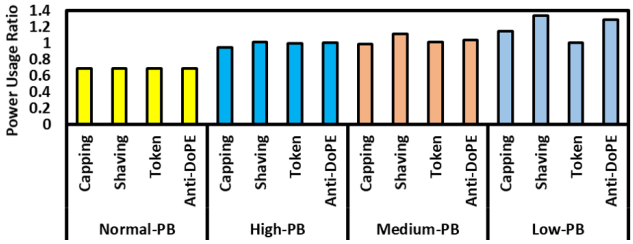


Figure 19: Energy consumption for different power management schemes at different power provision levels

to shave the occasional load power peaks. Therefore, any power-efficient design must ensure that batteries are enough for handling unexpected emergencies. In another word, batteries are used to shave the peak powers without compromising their normal functionality. In this section, we examine how the DOPE attack consumes the batteries under different power management schemes. We simulate a mini battery which can sustain 2 minutes when supporting all the web application nodes. We run the attack under different power provision situations and record the batteries capacity variation when using the batteries differently.

Figure 18 shows the monitored batteries' utilization map of the evaluated server clusters at each timestamp. In the figure, vertical axis represents the remaining capacity of batteries and horizontal axis is the time. We note that servers heavily discharge their associated batteries systems to remove DOPE in conventional data centers leveraging batteries to shave peak power. Since the DOPE generates high and long power peaks, it exhausts the battery as soon as indicated by the blue line. Our proposal mainly uses batteries as the transition medium. The usage of batteries depends on the power budget, booting delay of DVFS, and frequency of attack changes. The dark line illustrates a situation where the attack switches among 3 evaluated DOPE attack types per 2 minutes. Batteries discharge every time when the attack changes. Once the Anti-DOPE finishes reconfiguring the V/F settings, batteries are recharged again immediately. The dark line indicates that batteries discharge once until initiating throttling mechanisms to cap power generated by Colla-Filt based DOPE.

6.5 Energy Utilization and Saving

It is critical for an availability-oriented power management framework to be energy-aware as well. In fact, since Anti-DOPE is orthogonal to existing system, it has no side-effects on workload energy consumption during normal operation. Meanwhile, it does not require additional energy to provide a better performance while dealing with DOPE. Anti-DOPE leverages differentiated power capping (detailed in Section 5) different from other baselines. The appropriate voltage/frequency scaling shortens performance degradation times with a more appropriate energy utilization mode. Thus, Anti-DOPE can minimize the performance loss with as less energy consumption as possible. In other words, Anti-DOPE maximizes the effective usage of power.

We evaluate the energy consumption of the victim server under attack across different scenarios. We compute the total consumed energy with the consideration that it is sourced from batteries together with utility power supply. For comparison, we normalize the power consumption to the supplied utility power energy. As shown in Figure 19, different schemes consume the same energy in the baseline case. When there is a DOPE attack in the power-limited data center, Capping consumes less power since this technique blindly decreases the executing V/F of all the requests. However, the energy saving from capping is often aggressive since it has several side-effects such as degraded service time as shown in Figures 16 and 17. Compared with Shaving, Anti-DOPE uses less energy due to its less dependency on batteries. Summarily, Anti-DOPE guarantees the optimal performance with desired energy utilization.

7 RELATED WORKS

Power and Energy Attack: Vulnerability in server power management framework has been identified recently [13, 21, 42, 43]. Prior works have discussed two representative cyber-attack related power management challenges: energy abuse [13, 42] and power overload [21, 43] An energy abuse-based attack mainly targets the Web application layer, with the intent of merely consuming additional server energy. Differently, we investigate a new type of risk that arises from today's aggressively designed data center power provisioning architectures. On the other aspect, a power overload based attack aims to cause a rare expensive power outage by manipulating a group of virtual machines collocated in the same rack or cluster. In contrast, we consider a more powerful distributed server power/energy drawn and focus on the more frequent collateral damages caused by the attack.

Application-layer DoS/DDoS Flooding Attacks: Current researches on application-layer DoS/DDoS flooding attacks primarily focus on disrupting legitimate user's services by exhausting the internet connectivity [38, 40, 41] and server resources [5, 24, 29]. With asymmetric attacks [5], the bots request web pages that generate higher workload on the server resources (CPU cycles or disk usage) and the server is kept busy in responding to these requests. These attacks overlook power consumption due to aggressive usage of servers' computing resources. Thomas Martin et al. leverage energy-related Denial of Service to drain the battery in mobile computers [33]. It prevents mobile devices entering low power modes by keeping it active, then the battery life can be drastically shortened. Similarly, prior works [19, 21, 28] propose to increase the servers' power consumption by sending low-rate network requests. They exploit the dynamic performance-energy controlling techniques to force the server into high-performance mode, thus, it consumes more energy. Compared with them, DOPE intends to cause unexpected power surges. Ours work mainly characterize

the relationship of internet request properties and generated power anomalies.

Resource Availability Attack: Prior works have investigated the security issue on resource contention. For example, an attack thread can significantly deprive legitimate threads of their resources and cause significant performance degradation [12, 15]. At the chip level, recent work shows that hardware Trojans can be used to block the resources of a network-on-chip system, causing denial of service [32]. At the virtual machine level, a resource-freeing attack (RFA) could assign additional resource to the attacker's VM by modifying a victim VM's workloads [34]. Our work differs from these works in that we look at UPS caused by DoS attack at the data center level by stealth.

Power-Constrained Data Center: Over-provisioning server resources increases cost efficiency due to higher data center utilization. To ensure that the power dissipation stays below the given power budget, aggressive control strategies such as power/performance state tuning [8, 10, 11, 37] and battery-based peak power shaving [12, 17, 18, 26] are employed. Unfortunately, current power management frameworks are largely power budget and job performance driven, having the slightest knowledge of the legitimacy of load power demand. Therefore, a sophisticated attacker can exploit the blindness of these power management schemes to mount a successful power resource-oriented attack.

8 CONCLUSION

Peak power management strategies must be more resilient in aggressively under-subscribed data centers. It allows us to better handle the uncommon power anomalies. In this study, we identify a new type of threat called DOPE. DOPE can overwhelm the power management system of data centers by sending selective network traffics. We propose Anti-DOPE, a resilient request-aware power management framework defending against DOPE. We show that Anti-DOPE can greatly improve the availability of today's aggressively provisioned data center.

ACKNOWLEDGMENTS

We thank all the reviewers for their valuable comments and feedbacks. This work is sponsored by the National Natural Science Foundation of China (No. 61702329 and 61502302). Corresponding author is Chao Li from Shanghai Jiao Tong University.

REFERENCES

[1] Alibaba. 2018. Alibaba Cloud Object Storage Service Trace Data. https://github.com/alibaba/clusterdata.
[2] NetScout Arbor. 2016. *Worldwide Infrastructure Security Report: Volume XII.*
[3] NetScout Arbor. 2017. *Insight into the Global Threat Landscape: NetScout Arbor's 13^{th} Annual Worldwide Infrastructure Security Report.*
[4] BoNeSi. 2019. https://github.com/markus-go/bonesi.
[5] RioRey Company. 2012. RioRey Taxonomy of DDoS Attacks. http://www.riorey.com/x-resources/2012/RioReyTaxonomyDDoSAttacks2012.eps.
[6] Pivotal Software Cooperation. 2018. *Spring Boot.*
[7] Inter Corporation. 2016. IntelÂő 64 and IA-32 Architectures Software Developer's Manual. Volume 3B: System Programming Guide, Part 2.
[8] A. Bhattacharya et al. 2012. The Need for Speed and Stability in Data Center Power Capping. In *IGCC.*
[9] B. Hang et al. 2009. A Novel SYN Cookie Method for TCP Layer DDoS Attack. In *BioMedical Information Engineering.*
[10] C. Hsu et al. 2018. Smooth-Operator: Reducing Power Fragmentation and Improving Power Utilization in Large-scale Datacenters. In *ASPLOS.*
[11] C. Li et al. 2014. Towards Automated Provisioning and Emergency Handling in Renewable Energy Powered Datacenters. In *JCST.*
[12] C. Li et al. 2016. Power Attack Defense: Securing Battery-Backed Data Centers. In *ISCA.*
[13] D. Meisner et al. 2009. PowerNap: Eliminating Server Idle Power. In *ASPLOS.*
[14] D. M. Nessett et al. 1999. Multilayer Firewall System.
[15] D. Woo et al. 2007. Analyzing Performance Vulnerability Due to Re-source Denial of Service Attack on Chip Multiprocessors. In *CMP-MSI.*
[16] D. Wang et al. 2012. Energy Storage in Datacenters: What, Where, and How much?. In *SIGMETRICS.*
[17] D. Wang et al. 2013. Virtualizing power distribution in datacenters. In *ISCA.*
[18] D. Wang et al. 2014. Underprovisioning Backup Power Infrastructure for Datacenters. In *ASPLOS.*
[19] F. Palmieri et al. 2011. Evaluating Network-based DoS Attacks Under the Energy Consumption Perspective: New Security Issues in the Coming Green ICT Area. In *BWCCA.*
[20] F. Palmieri et al. 2014. Adaptive Stealth Energy-Related DoS Attacks against Cloud Data Centers. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.*
[21] F. Palmieri et al. 2014. Energy-oriented Denial of Service Attacks: An Emerging Menace For Large Cloud Infrastructures. In *SC.*
[22] G. Somani et al. 2015. DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions. *CoRR.*
[23] G. Vigna et al. 1998. NetSTAT: A Network-based Intru-sion Detection Approach. In *ACSAC.*
[24] K. Singh et al. 2017. Application Layer HTTP-GET Flood DDoS Attacks: Research Landscape and Challenges. In *Computers & Security.*
[25] K. Yeung et al. 2018. Tools for Attacking Lay-er 2 Network Infrastructure. In *IAENG.*
[26] L. Liu et al. 2015. Leveraging Heterogeneous Power for Improving Datacenter Efficiency and Resiliency. In *CAL.*
[27] L. Wang et al. 2014. Big-databench: A Big Data Benchmark Suite from Internet Services. (2014).
[28] M. Ficco et al. 2017. Introducing Fraudulent Energy Consumption in Cloud Infrastructures: a New Generation of Denial-of-service Attacks. In *IEEE Systems Journal.*
[29] S. Behal et al. 2017. Characterization and Comparison of DDoS Attack Tools and Traffic Generators: A Review. In *IJ Network Security.*
[30] S. Govindan et al. 2011. Benefits and Limitations of Tapping into Stored Energy for Datacenters. In *ISCA.*
[31] S. Govindan et al. 2012. Leveraging Stored Energy for Handling Power Emergencies in Aggressively Provisioned Datacenters. In *ASPLOS.*
[32] T. Boraten et al. 2016. Mitigation of Denial of Service Attack with Hardware Trojans in NoC Architectures. In *IPDPS.*
[33] T. Martin et al. 2004. Denial-of-service Attacks on Battery-powered Mobile Computers. In *PerCom.*
[34] V. Varadarajan et al. 2012. Resource-freeing Attacks: Improve Your cloud Performance (at your neighbor's expense). In *CCS.*
[35] X. Fan et al. 2007. Power provisioning for a warehouse-sized computer. In *ISCA.*
[36] X. Hou et al. 2018. Power Grab in Aggressively Provisioned Data Centers: What is the Risk and What Can Be Done About It. In *ICCD.*
[37] Y. Chen et al. 2005. Managing Server Energy and Operational Costs in Hosting Centers. In *SIGMETRICS.*
[38] Y. Xie et al. 2009. Monitoring the Application-Layer DDoS Attacks for Popular Websites. In *TON.*
[39] Zaf et al. 2018. A Lightweight Bash Shell Script Designed to Block DoS Attacks.
[40] Z. He et al. 2017. Host-Based Dos Attacks and Defense in the Cloud. In *HASP.*
[41] Z. He et al. 2017. Machine Learning Based DDoS Attack Detection from Source Side in Cloud. In *CSCloud.*
[42] Z. Wu et al. 2012. On Energy Security of Server Systems. In *TDSC.*
[43] Z. Xu et al. 2014. Power Attack: An Increasing Threat to Data Centers. In *NDSS.*
[44] The Apache Software Foundation. 2018. ab-Apache HTTP Server Benchmarking Tool. https://httpd.apache.org/docs/2.4/programs/ab.html.
[45] Google. 2016. *Best Practices for DDoS Protection and Mitigation on Google Cloud Platform.*
[46] IBM. 2016. POWER8 Processor Datasheet for the Single-Chip Module.
[47] Ponemon Institute. 2016. *The Cost of Denial of Service Attacks (Data Center Performance Benchmark Series.*
[48] M. Kumar. 2011. DDOSIM Layer 7 DDoS Simulator. https://thehackernews.com/2011/01/ddosim-layer-7-ddos-simulator.html.
[49] Netstress. 2019. https://netstress.org/.
[50] Verisign. 2016. *Verisign Distributed Denial of Service Trends Report.*
[51] In ACME Labs Webmaster. 2016. Multiprocessing http Test Client. https://acme.com/software/http_load/.